DIGITAL ASSISTANT FOR WORKSPACE APPS

BY

SEE LING XUAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Digital Assistant for Workspace Apps

Academic Session: June 2022

I SEE LING XUAN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
No 25, Lot 48557, Jalan Tiga,
Taman Yuk Kwan, 31000
Batu Gajah, Perak.

Ooi Joo On
Supervisor's name

Date: 8 September 2022

Date: 9/9/2022

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 8 September 2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***See Ling Xuan*** (ID No: ***19ACB01373***) has completed this final year project entitled "*Digital Assistant for Workspace Apps*" under the supervision of Ts Ooi Joo On (Supervisor) from the Department of Computer And Communication Technology, Faculty of Information And Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

SEE LING XUAN
(*Student Name*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**DIGITAL ASSISTANT FOR WORKSPACE APPS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : Xuǎn

Name : See Ling Xuan

Date : 8 September 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Ts Ooi Joo On who has given me this bright opportunity to engage in this project of Digital Assistant for Workspace Apps and has given me plenty of recommendations helped me in completing this report. It is my first step to establish a career in digital assistant and workspace apps fields. A million thanks to you.

Finally, I must say thanks to my parents and my friends for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Workspace app, or a digital workspace is a technology framework used to deliver, manage, and centrally controls the components of the IT assets of an organization including apps, data, and endpoints. Microsoft Teams, act as an instance of digital workspace is a platform for business collaboration developed by Microsoft to offer functions of workspace chat and videoconferencing, file storage, and application integration to increase an organization's work efficiency. The problem is that way too many functions to be configured or used in the Microsoft Teams consider time waste to achieve a task with many steps. Besides, the problems include unfamiliar users may apply unsuitable functions in Teams to complete their task and the tedious process to complete a similar task which require repetitive steps. This project proposes a digital assistant to be used in Microsoft Teams for improving efficiency to complete an organization's task using Teams workspace, by simplifying and giving advisory on steps required to complete a task and automating repetitive tasks. AI and algorithms will be implemented in this project for the digital assistant can have features of understanding towards human's commands and give suitable output for those commands, to further predict and suggest the future command or possible step may need to be done by the users. Moreover, the digital assistant should allow users to automate tasks by creating and running workflows and using templates. The proposed system will be achieved by applying machine learning to train the digital assistant model for it can study and execute every Teams' function or the function combinations and allow user customization on its steps to complete certain task. The output of this project will be an AI-based digital assistant designed for simplifying and automating tasks in Microsoft Teams.

Index Terms – Digital workspace, workspace app, Microsoft Teams, digital assistant, AI.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AGI	Artificial general intelligence
AI	Artificial Intelligence
AR	Augmented Reality
ASI	Artificial Super Intelligence
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
IPA	Intelligent Personal Assistant
JS	JavaScript
ML	Machine Learning
MS	Microsoft
NLP	Natural Language Processing
RPA	Robotic Process Automation
VR	Virtual Reality
VS	Visual Studio
XP	Extreme Programming

Chapter 1

Introduction

In the first part of this chapter, the background, technical knowledges, motivation, problem statements, impact, significance, and contribution of this project will be presented. Besides, in the second part of this chapter, the project scope, and objectives to develop this project will be discussed.

1.1 Background Knowledge

The Covid-19 crisis had rapidly accelerated the trends of digital transformation and emergence of digital workspace apps at breakneck speed. Due to the pandemic, remote working had become one of the most important trends in workspace app for almost all enterprises and organizations across the world forced to have their employees work remotely at home. In result, the shift to digital workspace from physical workplace is a revolutionary change in every organization to remain the business efficiency and productivity [2]. A digital workspace, or workspace app is an integrated technology framework designed to deliver and operate applications, data, and endpoints. The main objective of a workspace app is to ensure the productivity of an organization business. Besides, it includes sub features such as individual and group task management, brainstorming, file-sharing and storage, and others. The digital workspace solution helps to provide better employee experience by increasing the workstyle flexibility of them to move the business forward and to empower the organization to stay competitive, productive, and collaborative.

A digital workspace, meant to be virtual equivalent to the physical workplace should be a connected and collaborative work environment for boundaryless access from any location and device. The benefit is that employees can work together seamlessly and stay in constant communication to increase their engagement and can receive clear task requirements and alerts via a workspace app. For now, virtual collaboration is a key trend that has increased in significance over the years which to replace almost every in-person office interaction to the collaboration in virtual workspace [2]. Via virtualization technologies, working beyond physical boundaries will become the norm for team members can collab on the same project in real-time via virtual conferences and social forums. This shift allows employees to communicate and collaborate in many new and effective ways. Due to the pandemic, the digital workspace trends empower employees to be more productive and comfortable while working via conducting the businesses online and to bridge the gap between a physical and virtual experience. A success workspace app and solution should provide a unified, secure, and efficient experience for the employees to maintain their productivity. For example, some modern workspace apps nowadays including MS Teams, Google Workspace, Facebook Workplace, and others.

Besides, upon the digital transformation, AI is used to complete multiple various functions automatically from software perspective. The State of the Digital Workplace report from CMSWire discovered that AI and machine learning have the most effects in their organization to automate simple and repeatable tasks (24%) and help employees to quickly search for a content, people, and app they need to complete a specific task [1]. In a workspace app, AI and machine learning are important to provide guidance to users and automate some tasks for the employees can spend more efficient time tending to the task they would need to complete. The automation process of a workflow indicates the planning and auto-execution of tasks according to standard workflow rules created by people to route their tasks, data, or files between their organization members according to the pre-defined business rules to trigger a task automatically. To address the challenges brought by remote working, automation of repetitive and routine tasks, personalization, and prioritization of step have become several important ways for a digital workspace to help people work smarter and faster. For example, an AI assistant in digital workspace can expand data science to give users the ability to analyse on a complex data without a data science background. In result, to empower them from making a more informed decision quicker. To achieve it, an integration platform within a workspace app is important for users can extract and analyse data from various sources. The most important function of an AI assistant in workspace apps is to enable automatic tasks completion especially for the repetitive and routine operational tasks. The trend of using algorithms to automate some routine activities has rapidly accelerated by the pandemic and expected to grow in the future [2].

Microsoft Teams is chosen to be the focused workspace app to integrate the AI and machine learning based digital assistant developed in this project. Microsoft Teams acts as a collaboration platform and digital workspace app can help most of the organization business to go through an unavoidable digital transformation [20]. For its performance, MS Teams had achieved a spectacular growth from 20 million daily active users in November 2019 to 115 million in year 2020 [20]. This had turned MS Teams into the de facto standard for workspace apps and the vendors are focusing on extending Teams on areas which have not yet been fully developed now. The trend is clearly state that Teams has high extensibility to integrate with most of the business apps to make the digital workspace become more powerful and convenient to use. In conclusion, a digital AI assistant should be developed in MS Teams to increase its productivity and business efficiency for users to use MS Teams as a workspace platform.

1.2 Study About AI and RPA

In this section, the definition and meaning of AI will be simply discussed for better understanding to the IT knowledges will be applied in this project. In year 1950s, the fathers of the field, Minsky and McCarthy stated that AI is the simulation of human intelligence performed by machines such as computer systems [19]. However, before the definition, the concept of AI first denoted by Alan Turing's seminar work by asking a question of can machines think [4]. For example, the applications of AI such as NLP and speech recognition. The foundation of AI requires specialized hardware and software to build the algorithms for machine learning in an AI [17]. Machine learning is simply one component of AI and is where a computer system learns how to execute a task rather than being hard coded how to do so [19]. For a legacy AI program to work, it needs to be trained from huge amount of data and analyse the data for correlations and patterns to make future prediction. The rule, or called algorithm is the core of an AI to provide computer device with the step-by-step instructions for it to complete a specific task. During the training process, AI can study to convert the data received and analysed into actionable information, and continually fine-tune algorithms through trial and error to generate the most accurate and desired outputs. In general, AI can be developed using popular programming languages such as Python, R and Java.

Neural network is the key to machine learning inspired by the human brain. Neural networks are trained datasets that teach an AI on what it should spit out when presented with specific information during training [19]. Neural networks in AI are the interconnection between algorithms that feed data into each other and vary the weight attributed to the data when it passes between the layers [19]. Deep learning is a subset of machine learning which comprised of a huge neural network. Besides, deep learning expands the neural networks into sprawling networks with many sizeable layers of processing units those are trained using large amounts of information [19]. Different from machine learning, deep learning eliminates the manual human intervention required to process data [4]. AI should be able to undergo processes of learning, reasoning, problem solving, perception, and using languages to simulate human intelligence and to achieve complex machine intelligence such as speech recognition and computer vision [8]. As consider the AI capability, it will be the main technology to be applied in this project implementation to achieve the smart assistant function on workspace apps.

Besides from the AI technology, the RPA process is also studied in this chapter to investigate its contributions, strengths, and possibilities to be applied in the project. RPA is an automation technology for a software can emulate humans' operation to interact with other software or systems [24]. The RPA process allows a software robot to understand the context in a certain software's view, and able to perform the same operations as human can perform such as trigger certain event on the page via hitting a keystroke or inputting certain parameters into the system's UI. In other words, through the RPA process, a developed bot can replace human's workload on some tedious repetitive jobs especially in the work peak hour to increase productivity. This software technology was developed for auto executing rules-based business processes via automation technology but still in progressing where its strength is not strong enough to fully mimic human's digital actions [25]. However, the advantages of RPA still involved it accelerates the digital transformation process, improve productivity and reduce employee's workload. In contrast, some of the weak points of RPA including the unstable performance due to not yet fully developed RPA software available in this market that lower the reliability of a RPA bot, and time wasted for the immature RPA bot to identify suitable element on the software or system's screen and perform correct actions on it. Sometimes, a bot might need to spend more time than human's operation to perform a step due to the high intricacy of the step that is not fulfill the objective of RPA to increase productivity. Moreover, as both RPA and digital transformation are still in implementation today, RPA is considered as a technology with potential to growth for fully replacing most of the human's repetitive workloads to reduce manual errors and boost human's productivity in every industry area that involve digital operations. Due to its potential on developing the digital transformation, RPA and AI will be investigated and experimented to combine both advantages and implement the project for developing an RPA bot with AI capability that can increase user's experience and productivity when using the workspace app of Microsoft Teams as decided as the scope of this project.

1.3 Motivation

Current workspace apps including MS Teams are still in the development process of digital transformation and may not be efficient enough to fully shift the physical work mode to remote work mode due to some issues. For example, the tasks overload for an organization business to withstand in the competition from modern world, especially when the digital transformation immediately surged due to the Covid-19 pandemic. It had added stresses onto both employees and workspace apps to ensure the productivity of remote working as same as working physically. Besides, the limited availability of IT tools such as software and apps to complete a task also become a challenge in workspace apps to reduce the time that employees need to switch between apps. Furthermore, the increasing functions to be used in a digital workspace also force employees to learn and use many diverse categories of functions and tools which is time wasting and reduce in productivity. Moreover, most of the workspace apps now are less efficient due to the immature implementation of business automation process to complete tedious and repetitive tasks. Therefore, a smart all-in-one workspace or a smart assistant to assist users in a workspace app is one of the most important trends for digital transformation nowadays.

In this project, MS Teams is chosen to be the focused workspace app to increase users' conveniency and productivity to complete their tasks. MS Teams had achieved 145 million daily active users in April 2021 which the engagement is through the roof [10]. Besides, the integration between MS Teams and Microsoft 365 apps reached more than 30 billion collaboration minutes in a day in the MS Teams platform [10]. This skyrocketing performance had showed the trend of MS Teams and indicated that most of the organizations will prefer to select MS Teams for implementing their remote or hybrid working model due to its productivity and efficiency. However, MS Teams still has the expandability and spaces for development in its functions and features, to make it can align with the rapid speed of digitalization to maintain its good result and performance.

In conclusion, this project is motivated to be conducted to increase the efficiency and productivity of MS Teams as the norm of workspace apps, and to solve the problems faced by the workers who use MS Teams to perform their business tasks. The problems of current MS Teams will be discussed in the next section.

1.4 Problem Statement

In this section, three problem statements will be discussed to indicate the problems and weaknesses of MS Teams for now.

1. Way too many functions to be configured and used

As a persistent chat-based collaboration platform, MS Teams consists of a lot of various functions and features to aid business communication such as document sharing, online conferences, meeting scheduling, shift scheduling and others. The abundance functions in MS Teams had made it become a powerful workspace app and can be used to satisfy the requirements of almost all enterprises and organizations under different case and requirement. However, the plenty of functions can also cause problem of **many steps and configurations required before user can achieve a specific task**. This will be a strong weakness to a workspace app which lower the employees' conveniency and productivity to use MS Teams for collaboration and to complete their tasks. For example, user may need to complete a task in MS Teams by using its functions combination such as to combine the Microsoft 365 integration function and the screen sharing function to share files and resources to the project team members or the online meeting participants. Besides from the increased number of steps to perform a task, the feature of MS Teams to allow tiny configuration of user also lower its productivity such as users can make a lot of configurations to create and manage a team channel. Therefore, the way too many functions and features to be configured and used in MS Teams can increase the task complexity and steps for users to complete a specific task. In result, it will cause **time waste** due to the increase of time needed by a user to configure and use a functions combination in MS Teams.

2. Tedious process

The last problem statement is the **tedious process needed by a user to perform repetitive tasks or routines tasks which require similar steps**. This problem will cause non-productive time to be wasted and lower the productivity of an organization business. There are some general routine tasks in the daily business processes of most of the organization such as email responding to common inquiries and the repetitive follow-up steps for similar task in an organization's business flow. In result, the repetitive steps for similar task to be performed periodically are time, resources and manpower wasting especially for the tasks can be automated. In MS Teams, the tasks automation can be helped by one automate tool, which is the Power Automate App, but its automation features are still not complete and powerful yet to maximize an organization's productivity. The weakness is that the business automations can be performed by Power Automate in MS Teams are still not fully covered the MS Teams functions and lacking enough user customization on the task automation process. Therefore, the tedious process will be a problem to cause inconveniency of users to perform similar tasks by applying almost the same steps.

3. Unsuitable functions used by users

The development of MS Teams until year 2021 had brought more than 56 common features and integrated with a lot of productivity tools such as SharePoint and Word [10]. This MS Teams trend can be an advantage to provide many useful functions those users can use to complete their tasks productively but to unfamiliar user, it will cause **extra time needed for them to complete a specific task**. For example, a new user to MS Teams will be hard to learn all functions in MS Teams in short time period and they may confuse with some similar functions and cause them to apply unsuitable functions to perform a task. Besides, the performed task has **risk to produce an inaccurate result** due to the inappropriate ways and steps to perform that particular task by users. To solve this problem, an **organization** will need to train their new employees on the ways to use MS Teams via lesson or guidance which is **time and resources costed**. Moreover, even though an employee had been attended the lesson to learn how to use MS Teams, the plenty of functions will still be possible to confuse the users and cause them to use unsuitable function, or unsuitable steps to complete their tasks. In result, time will be wasted for a user to recomplete an inaccurate task and search for the suitable function and steps to perform a specific task, and both time and resources will be wasted for an organization to prepare training programs for new employees.

1.5 Impact, Significance and Contribution

This project is to benefit and contribute to the target employees who using MS Teams as their workspace app in their organization. The digital assistant proposed can **help the workers to complete more tasks by reducing the time spend to complete a specific task**. It is important for users to complete their tasks with simpler and more direct command in MS Teams and to spend less steps and time for them to complete each task productively. Besides, AI is important to be developed in a workspace app to assist employees' works such as to provide guidance steps for users to complete their task and use the MS Teams' functions step by step. Moreover, the proposed system which is able to predict and suggest the next possible command may inputted by the users also important to help users in decision making such as to decide which function or tool to use for the next step. In result, it can save users' plenty of time needed to learn all functions in MS Teams and avoid them to be confused by the similar functions in Teams. This feature will be friendly to unfamiliar users to MS Teams for even new users can also complete their tasks correctly and obtain desired outputs from MS Teams.

From the **organizations and enterprises' view**, having a smart assistant in workspace app can also save their **costs and resources to build training programs** on how to use MS Teams properly for their employees. Furthermore, the assistant can offer better working experiences and high flexibility to employees to work in anywhere by using any of their own digital devices but still can maintain the productivity and collaborative experiences as same as they are working in the physical workplace. For example, the manager or a project leader can easily manage their employees via the team channel function in MS Teams to conduct any project with the assistant's helps.

Another importance of this project is **to automate repetitive tasks** in MS Teams to reduce time waste for employees to perform repetitive steps for routine or similar tasks. In the recent years, the switching to virtualization technology not only helps an organization to change its work mode into online mode, but also helps to reduce the expenses and manpower for businesses in an organization. The task automation had become a very important trend for now and the future to maximize the outputs from minimum effort. Based on this trend, business automation had become an important area to be developed to help employees from totally be rescued from the repetitive tasks. Therefore, this project aims to achieve task automation in MS Teams such as to build the business flow chain and allow users to customize their automation process of certain tasks by using macro features of the digital assistant. In result, it contributes to not only to remove the steps to perform repetitive task, but also leading the MS Teams to a smarter workspace app which will benefit more than 145 million of its users and help the organization to have more competitive advantage in competition of the modern world.

Lastly, this project is significant to implement an AI assistant with NLP ability to allow employees to interact with it via normal languages to perform tasks in MS Teams. This is important to **provide a simpler way for users to execute a Teams function for completing tasks** and deal with the outputs they have generated in MS Teams. This feature will undeniably to be a major step to forward the smart workspace app into the next stage in the digital transformation process. Simply, this project is just a small foundation for one day in the future human can use flexible natural languages to easily receive information and complete a task in smart workspace apps such as MS Teams under a fully low-code environment.

1.6 Project Scopes

Laureys had predict that AI and automation will claim their space in the workspace apps start from year 2021 [20]. Currently, MS Teams provides AI-driven components including live captions, meeting transcripts, automatic translations, and superimposing the presenter's video feed on top of the content. However, it is just the first state of democratization of workspace intelligence in the digital transformation. Therefore, the scopes of this project include implement a digital assistant into MS Teams workspace app and propose algorithms for the smart assistant can assist users when using MS Teams functions and to automate workflows. The target users are all users who are using MS Teams for their collaboration within organization and to increase their business productivity.

Basically, this project will be divided into three major phases:

Data Collection

All related data will be collected in this phase. For example, the information of all functions can be executed by users in MS Teams and the way users perform those functions. By referring to the objective one to simplify users' steps to accomplish MS Teams functions, the digital assistant developed should be able to automate those too many functions in MS Teams by only user commands to build an entirely low-code environment, and to save users' time. Besides from the steps to conduct each function in MS Teams, the connection between Teams functions will also be analysed for some functions which need to be executed by the combination of multiple small functions. For instance, users would need to create a new team before they can add a new channel into their created team. Also, the data of required parameters of different use case and scenario to perform specific task in MS Teams will also be collected for the sake of ensuring users had inputted necessary number and types of parameters while performing certain MS Teams function via the digital assistant. The example can be showed is that one of the MS Teams functions to create new team required necessary parameter of at least team name, and the optional parameters of team description and team privacy before the bot can automate the process. To summarize, first of all, the way to execute a specific function in MS Teams and the parameters needed for every single function are necessary to be collected in the first step of this phase, for the digital assistant can automate MS Teams function via user command only to simplify steps required and to guide users from performing the correct Teams function.

In the second part of this phase, other requirements of this project should be collected including UI requirements and other functions or commands those can be executed by the digital assistant besides from performing MS Teams functions. The UI requirements determine the AI capabilities should be implemented in the bot in the next phase such as NLP module is needed for the digital assistant can process user input in natural language or spoken language. Apart from that, the digital assistant is considered to be able to perform functions other than only MS Teams functions to improve user experience and productivity in performing their tasks by using MS Teams. For example, users need to be able to schedule their task to perform a specific Teams function or a combination of commands in certain time or period. This is to save their time and effort from calling the same command when they want to execute a function periodically and to automate repetitive tasks. Moreover, other useful commands to be executed by the digital assistant including its integration with third-party apps such as email and Microsoft apps to achieve seamless experience of users to connect Teams functions with the third-party apps functions via the digital assistant developed. For instance, users can choose to send an email consisting of specific information in a MS Teams channel to every member in this channel via command to the digital assistant. Therefore, the steps and parameters for the bot to perform other functions than MS Teams functions should also be collected in this phase. At the end of this second step of this phase, other data and requirements will be collected to prepare the data needed for next phases.

Machine Learning and Modelling

The modelling of this project can be divided into machine learning needed part and non-machine learning needed part. For the first part, the digital assistant should implement all functions those are unrelated to ML such as the assistant's function to perform all functions in MS Teams. ML is not needed in developing this feature of the digital assistant due to the assistant only need to collect necessary parameters from user and execute specific MS Teams function or other function corresponds to user command. The steps and logic to perform those functions are predefined by code and do not require ML algorithm while performing the functions. Furthermore, another feature of this assistant which is non-ML required is the feature to automate repetitive tasks. For example, user can perform multiple tasks by creating a task list or schedule their tasks to be performed at a specific time or period. No ML algorithm is needed as the digital assistant will only need to create a task list consisting of multiple commands to be run in sequence or add a specific task to its scheduler to be performed later. Therefore, the first part of the digital assistant is the normal algorithm to conduct every function it can perform which is non-ML required.

For the second part of the bot, ML is needed for the digital assistant's feature to provide advisory on steps according to objective three. In this phase, the AI assistant model will be trained by using machined learning based on the required data collected to guide users from choosing the right function to perform, predict and suggest the next possible function users may want to perform. ML is required to predict the most possible functions can be executed by users based on their preferences and previous performed functions by the suitable ML algorithm. Through the modelling process, the AI assistant should be able to study the way to perform the prediction to provide recommendations under certain user condition or based on users' commands. Moreover, to build a neural network of the AI model for it can achieve better performance during the prediction and recommendation. NLP should be developed in this phase for the AI assistant to recognize and understand users' command in human languages. Furthermore, the algorithms implemented in its neural networks should allow this AI assistant to understand and analyse the correlation between every function it can perform to increase its accuracy in guiding users to perform a single MS Teams function, or a function chain built by multiple functions in order to satisfy their task requirements.

Product Implementation

After the functions correspond to bot commands are developed and the model is trained, the digital assistant will be implemented in the last phase to build its UI and examine its abilities to assist user when performing tasks via MS Teams workspace app. The AI model will undergo testing and assessment to further modify the algorithms and data to train the model in deeper. The final AI product should allow users to execute every Teams function via simple command according to the task they would like to complete. Moreover, the AI assistant should be able to guide users on the next steps they may want to perform based on their last command or functions performed history.

Main Functions and Features of the digital assistant proposed in this project

Function 1: Execute single users' command in MS Teams

The digital assistant in MS Teams workspace can help users to easily execute almost all Teams' functions and some extra functions by simple type command. This project is to propose an AI assistant that should be able to understand users' command by recognizing natural languages typed by users. Besides, to ensure the command flexibility by the AI assistant can analyse the similar words and meanings from users typing command which lead to the same task and execute that specific task. Also, the proposed AI assistant should be able to lead the users to perform every small step if they want to execute a function which composed of many steps.

For the novelties or extra modification feature to the digital assistant proposed, users should be allowed to add new command into the existing command lists by inputting the name, command, and steps to perform the command. Besides from the commands to execute Teams' function, the digital assistant proposed also acts as a chat bot to able to response on users' text. Therefore, users should be able to modify the chat functions of this digital assistant to train the bot how to response on or answer users' specific word or question.

Function 2: Execute batch commands and automate workflow

The AI assistant should be able to read in and perform user commands in batch if they want to execute multiple commands in sequence. To execute commands in batch, users can create, access, edit and run their workflow by using this digital assistant. For example, users can build a task list involve different Teams functions to use, and the proposed assistant will be able to call each function after users had finished all steps in the previous function performed. In addition, to automate workflow, users can schedule their task to automate specific task or list of tasks in certain date and time or in certain period. For example, to conduct a meeting every Wednesday at 3pm. In result, to automate repetitive tasks and prevent user from calling the same bot command to perform a task periodically.

For its novelty, users will be allowed to back to the last function of MS Teams if they wish to re-execute the previous command in a workflow list. Moreover, in the workflow automation, users are allowed to set trigger to automatically execute a set of commands in a specific workflow when certain event happened.

Function 3: Create and use template

Through the proposed project, users can build and use their own templates for their most frequently used tasks to automate repetitive tasks. For example, the email template to make an appointment or the message to request people to fill in a survey. These templates created can be quickly used by the users when they would like to perform similar tasks to save their time. Some default templates will be provided by the proposed system for some common routine tasks to aid users in task automation. For example, the default templates including spreadsheet with defined columns for user to fill in each row of data so that the assistant can perform the automation process according to respective spreadsheet's function. To create their own template, users will need to use the system's macro feature in the RPA process to establish new automation workflow. The user defined or modified template can be applied in their created workflow to automate certain steps in their function chain. In result, the digital assistant can automatically execute every task inside the workflow and when a template involved in the current task, it will request users to enter parameters needed in the template.

Function 4: Include advisory on steps

The digital assistant should provide recommendations to guide users from choosing the right tool or function to perform via user commands or inputs by performing ML algorithms. For example, the digital assistant should suggest user to perform the MS Teams functions to create a team, invite specific members and schedule a meeting when the user wants to schedule certain event. Moreover, the assistant can recommend multiple functions those are most related to the keywords inputted by user or user task requirement for users to choose their desired function to perform in MS Teams. Besides, the AI bot can also predict and suggest the next possible function to be conducted by the user according to the last executed function via ML algorithm or by perceiving the existence of last executed function in a predefined function chain.

1.7 Project Objectives

In this section, the project objectives will be presented to resolve the problem statements defined in the previous section.

1. **To reduce steps required**

   This project aims to simplify the steps required by users to perform a task in MS Teams. The digital assistant developed will be able to reduce the steps users need to perform before they can complete a task by users can input simple command to execute a Teams' function. This objective is important for the digital assistant proposed to simplify the task steps used to accomplish the too many functions in MS teams. In result, users can save the time they need to execute certain functions by accessing different user interface in MS Teams when they would like to configure and use a function in MS Teams. In this project, a smart digital assistant will be developed and integrate with MS Teams for users can save time and steps to configure and use functions or combination of functions in MS Teams. In result, to reduce users' time spent to complete a task to increase their productivity and for the users to feel more convenience when using MS Teams' functions.

2. **Automate repetitive tasks**

   This project's purpose is also to automate repetitive tasks of users and organizations who is using MS Teams as their workspace app. The task automation implemented using AI and machine learning can help to remove the tedious process and repetitive steps performed by users to complete similar tasks to waste the non-productive time. The digital assistant proposed will be able to automate repetitive tasks by create a list of tasks to be executed in sequence or perform task scheduling to execute specific task or list of tasks at a specific time or period. Moreover, the task automation can be done by using templates created and managed by users to execute those steps automatically in the flow chain. Moreover, user customization is allowed in the automation process by using macro features of this system for them can manage the steps and situations to perform a specific routine task. In result, to save users' effort and time to perform repetitive steps for similar tasks to further improve their productivity and efficiency.

3. **To include advisory on steps**

   The digital assistant proposed in this project will be able to provide advisory on steps to users' command. This feature is to avoid users from using unsuitable function to perform a task which may lead to the task fail or inaccurate task result produced. Moreover, to save the time users spend to search for the suitable function can be used by them to achieve their desired task in MS Teams which consist of too many functions and similar tools. To achieve it, the digital assistant proposed should be able to provide users with multiple related options based on users' inputted keywords to perform certain task and allows users to choose their desired option to execute that function. This feature applied in the situation when the command inputted by users can lead to different functions in MS Teams and the AI assistant will be able to predict and suggest the most near and correct function to the users based on their commands. Moreover, the digital assistant should be able to predict and advise the next possible steps which the users may want to perform next based on their last command or last executed function, especially when they want to execute a function in a function chain. In result, this digital assistant can ensure users to use proper tools and functions to perform their task correctly via MS Teams. Besides, to provide conveniency to unfamiliar or new users to MS Teams so that they can use the MS Teams' functions in an easier and more correct way to perform certain tasks without the needs to be trained hardly and learn clearly how every function in MS Teams works.

1.8 Report Organization

The report is organized into 7 chapters including Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology or Approach, Chapter 4 System Implementation, Chapter 5 System Evaluation and Discussion, and Chapter 6 Conclusion. First of all, Chapter 1 presents the background and technical knowledge, motivation, problem statement, impact, significance and contribution, project scopes, objectives and report organization of this project. In Chapter 2, the literature review on Microsoft Teams and previous works on digital assistants is conducted to evaluate the functions, strengths, limitations, and recommendations of each product. The third chapter discusses the system methodology and approach used to develop the project, followed by the details of the system design including some of the design diagrams. Furthermore, the fourth chapter is regarding the system implementation including the project setup, configuration, development, implementation issues and challenges. Moreover, Chapter 5 discusses the system evaluation and discussion involved system testing setup and result, followed by the project challenges and objectives evaluation based on the testing result. Finally, the last chapter of Chapter 6 presents the conclusion and recommendation on this project for future improvement in this project and the respective industry area.

Chapter 2
Literature Review

2.1 Microsoft Teams

2.1.1 Reviews on Microsoft Teams

Introduction

MS Teams is a collaboration platform can be used by anyone own a Microsoft account to help an organization to organize team and have virtual collaboration in one place [16]. MS Teams provides users with a free version or a paid version by limiting the number of team members can be added, maximum meeting participants and maximum meeting duration in different version of it. The main objective of MS Teams is to boost team productivity via its various functions and features.

Functions and Features

Activity

Through the activity feature in MS Teams, users can receive the alert of unread messages, @mentions, replies and more notification in real-time [16]. Besides, MS Teams allows users to customize alerts such as to choose the specific user or group of users to be notified [10]. This feature improves transparency in MS Teams by enabling every member in a team channel to be mentioned and notified via a simple @mentions. In result, every member in a team channel can be kept in the loop and know everything in real-time.

Teams

Users can create public or private teams or channels by using MS Teams and inside the channels, they can conduct a virtual conference, have conversations, and share files and resources. Teams' customization is allowed in MS Teams for users to set up and manage a team and collaborate in teams or channels those had been created by a project leader or team manager. Inside the team, users can easily invite people with whom they want to collaborate and conduct virtual collaboration through posting messages or reply to someone's messages. Moreover, they can share files and resources in a secured environment though the file sharing function in MS Teams including functions to upload and search files [16]. The teams and channels created or joined by users can be show or hide according to users' preferences which is important to avoid information overwhelm to them [10].

Meetings

Users are allowed to conduct or schedule a meeting in MS Teams and the calendar for scheduling will be synced with the users' Outlook calendar to avoid data missing. Through the schedule function, users can also set up and attend a live event scheduled in their calendar [16]. Users can organize a public or private live event such as conferences and talks which can accommodate up to 1,000 participants in MS Teams. During meetings, in-meeting chat messaging function is provided for users to chat when meeting. Besides, permissions of the participants of a live event can be customized by the organizer. To increase productivity, MS Teams now had provided a feature of meeting white board available during meeting for every participant to sketch together and share ideas and innovations using the whiteboard. The whiteboard content can be typed, drew, erased, and shared in real-time. To increase the meeting efficiency, MS Teams also provide functions of automatic transcription and live captions during meeting. Moreover, the cloud recording function in real-time to sync the users' recorded meeting content onto the cloud. To prevent distractions and noises during the meeting, hard audio mute feature is included in MS Teams for the meeting organizer to stop certain attendees from unmuting themselves [10]. Besides the video conferencing, MS Teams also provide a unique feature of audio conferencing which is not provided in many other workspaces' platform. Via audio conferencing, employees can join the meeting via their phone only which is more convenience, and they can join the meeting in any place.

Chat

Users can chat with anyone via Teams and for now, this function had been automatically integrated to Windows 11 to make the chat function become more convenience to be used [16]. Through the chat function, users can have one-to-one or group conversations by using the chat box or through video meetings and audio calls. [10]. Moreover, it includes the feature of immersive reader to read and speak the reply in a chat or text in a channel which the speeds and tones are customizable for the users can easily listen to a message when they are not able to read it. Another important feature to make users feel more convenience and comfortable when using MS Teams is the inline message translation function to translate a chat message into user's native language to allow smooth conversation between employees [10].

Calls

Users can call anyone including internal users and external users via Teams. [16]. To increase conveniency, MS Teams allows users to search for the specific contact of the person they would like to call [10].

Files

File sharing and screen sharing functions in MS Teams allow virtual collaboration of multiple users by sharing their data and resources. In MS Teams, information barriers are implemented to allow a project manager to determine which employees to collaborate in Teams to keep the data isolated when they deal with sensitive data and conversations. The conveniency of the files management function is that all files the users upload to MS Teams will be automatically synced to the cloud such as to the SharePoint to allow user collaboration and avoid data lost. Besides, MS Teams limits its file storage at 2GB for free MS Teams users and 1TB for paid users. [10]. The security of data can be ensured by users to customize the permissions on files those contain sensitive information. Also, the files and data will still be stored on cloud such as SharePoint even users had deleted their channels or teams [15].

Shifts

MS Teams allows an organization to manage first line employees by managing and arranging team schedules with Shifts. By using this function, a manager can easily manage shift requests in MS Teams and schedule the shift of every specific employee to increase the work efficiency of all employees [10]. Currently, digital workspace often been exclusively targeted on white-collar employees, or called knowledge workers which initially involved in traditional paper works. However, the trend of digital workspace should also involve digital transformation on the first line or so-called blue collar workers' tasks. Often, the blue-collar workers who engage in hard manual skilled or unskilled works still receive their information, schedules and tasks in paper format and rarely deliver their reports digitally. Therefore, it is important to digitalize information sharing and team collaboration for frontline workers to increase their work efficiency and minimize the efforts.

To assist the frontline workers, MS Teams allows them to securely consult and share information. Besides, the most important feature is that they will be able to consult and adapt their work schedules in MS Teams via the Shifts app integrated in. The digital shift to minimize and even remove the paper-based work in any business flow of an organization will be the future trends and the advantage of paper-based processes still can be maintained, for example, the benefit of low implementation cost can be achieved by workspace automation.

Command box at top of MS Teams

Moreover, there is a command box at top of the MS Teams for users to search for certain people, file or launch an app [16]. Slash commands are used in the command box to navigate MS Teams efficiently. For example, some general slash commands can be type into the command box such as /Files to lists the recent files shared with the users and /GoTo to immediately lead users to a specific channel [10].

Tabs

Tabs in MS Teams function to store frequently used apps. Moreover, Tabs in MS teams can display interactive web content to the channels and include two kinds of tabs which are the static tabs and the configurable tabs [10]. Static tabs support individual user such as to create their own notes as they work, while the configurable tabs support teams' channel such as to offer ongoing information and guidance to teams [10].

Apps and tools

MS Teams allows user to explore useful apps and tools in MS Teams to increase the efficiency of business flow. To increase productivity, every user in MS Teams can build their personal apps space by installing apps to the MS Teams. For example, users can view all approvals and tasks assigned to them from apps like Planner in their personal apps space [10].

Strengths

Strength 1: Increase productivity

Teams increases productivity and enrich communication by **combining all collaboration tools in only one platform** including chats, calls, online conferences, file sharing, tasks, event schedule, frequently used tools, and others. Therefore, to save the users' time to look for specific data or information via different tool. Moreover, MS Teams **allows seamless move of digital workspace onto the cloud** and Teams is a fully cloud-based solution to benefit organizations of using a digital workspace in cloud [14]. For example, the organizations using MS Teams can enjoy the pros of big data processing in cloud and the AI in cloud can simplify their tasks and make them more efficient. To increase productivity, **democratization of low code** has now been integrated in MS Teams. The low code in MS Teams is expanding beyond traditional front-end or back-end development. For example, the low code for front-end development supported by Power Apps in MS Teams while the low code for back-end development supported by Power Automate. There are some other low-code platforms can be used by an organization within MS Teams such as Power BI to support data analytics and reporting, an integrated AI Builder for implementing intelligence fusion, Power Virtual Agents to plug in smart chatbots, and Power Automate to achieve RPA [20].

Strength 2: Seamless virtual collaboration

Through MS Teams, **users can enjoy seamless and quick video or audio calls with peoples inside their organization or the external business partners**. This feature makes MS Teams into a great collaboration platform to provide users with simple, user-friendly, and real-time collaboration by using the file and screen sharing options. Via the main components of MS Teams which are the teams and channels, users are allowed to communicate with a defined group of people to have meaningful and seamless conversations with them. Via the video conferencing tool, an organization can provide their staffs with a training program including learning sessions. Besides, solutions such as file-based collaboration and screen sharing function allow employees to have fruitful discussions, share inputs and provide innovative ideas in real-time remotely.

Strength 3: Better focus on users' work

MS Teams allows users to have better focus on their work by eliminating the time spent by users to scroll down the emails and messages. MS Teams separates conversations, meetings, files, and documents according to channel. Only the related data and information will be showed in a specific channel to ensure the proper organization of information. Therefore, users in MS Teams can focus their attention to a specific channel where they collect their task requirement or perform the work. This advantage saves users' time to get side-tracked with random email or inbox and allows them to only focus on specific channel to look for specific topic and information. In result, users can have their discussions via Teams to remove the necessary of them to retrieve tasks or important information from the email threads to keep them can focus on a specific task in MS Teams [14]. In result, MS Teams boosts the productivity by saving users' time to schedule meetings, allocate task, track, and manage files to achieve highly focused collaborations in Teams [14].

Strength 4: Integration with Third-party tools

MS Teams now has **integrated with several useful third-party tools** such as the Microsoft 365 integration to integrate with Microsoft apps such as Outlook, Word, SharePoint, and OneNote. This feature allows the easy connection of users to the tools they need to complete their tasks and for easier file sharing and management. For example, to connect SharePoint to a channel and add SharePoint sites and pages to the channel. In result, it can save users' time to access the third-party tools exclusively. Also, MS Teams integration provide flexibility for users to even **integrate with cloud storage solutions** such as Google Drive or Dropbox, and to be taken into social media platforms such as Twitter [14]. Besides, MS Teams can also **integrate with MS Teams bots built on Bot Framework Composer to take benefits of automation features**. These bots are online all the time and can help users to handle boring and routine tasks. For example, integration with Polly Bot to vote among employees and Stats Bot to generate scheduled reports via sources such as Google Analytics. In result, the AI bot for task automation can help users to complete their functions in a more effective way and easier to analyse data using data analytic tools. The integration feature also provide benefit of **reduce the time spent by users to switch between different app and tab** when performing their tasks to free up the productivity [10]. The integration of useful tools in MS Teams had made the smart workspace app act as a central hub for users to use multiple app functions such as Planner and Outlook within the MS Teams platform. This advantage also allows users to integrate third party tools into their chats and channels such as for them to manage the files without leaving a chat or schedule a meeting within a channel.

Strength 5: Guest Access

MS Teams allows users to have conversation with individuals outside of their organization and even with individuals who are not using MS Teams. This is an important strength **for employees can communicate with their customers** by inviting them to join a chat as a guest via their email account and can perform the meeting and file sharing options. Besides, to share a link to the external individuals for them can join the users' Teams workspace. The prerequisite requirement is that both internal and the external users need to switch on their external access for users can invite the external individuals to join their teams and channels using guest access. An external individual can only access the files and information shared in a channel once the individual is made into a guest. MS Teams allows users to share channels, talk and chat with anyone including internal and external users via Teams Connect which was announced at Microsoft Ignite in March 2021 [10]. Teams Connect is still in private preview currently but predicted to be roll out by year 2022.

<u>Limitations</u>

Limitation 1: Too many similar tools

The too many similar tools in MS Teams **easy to confuse users about what tool they should use to complete a specific task under different situation and circumstance**. This is due to the reason that way too many tools that offer similar or even overlapping features in MS Teams to confuse users from choosing the right tool to use [15]. In result, the organization may need to provide their employees with a training lesson to those tools which is time and resources waste, and the users will also need to spend extra time to get familiar with the multiple various tools in MS Teams. The various apps and tools available in MS Teams allows high degree of customization according to the organization business needs but those flexibility and customizability can be a disadvantage and be too complicated to some of the users.

Limitation 2: No unified search on product

MS Teams has a limitation of **no unified search tool** in this platform for user to search for certain content for all Microsoft 365 conversation. This disadvantage causes users hard to search and locate a particular chat especially when they want to look back to the manager's requirements which they do not remember where they had the chat before.

Limitation 3: Insufficient notifications

Even MS Teams provides the Activity function to notify users for some mentions and broadcast news, the **notifications function in MS Teams is still imperfect** and this lacking feature will cause some difficulties in users' works. For example, users will not receive alert message or notification when they want to create a new team with the name same to an existing team, and finally they may end up with two or more teams with duplicate names. In result, confusion between team members will be caused and the resources in Teams will also be wasted as everyone in Teams can create a team or channel except their permission is restricted by a team manager. Thus, to solve this issue, users will need to **spend extra time** to search the name of all existing teams via the search bar before they create a new team.

Limitation 4: Confusing file structures

The disadvantage of MS Teams includes **confusing way to store files in Teams** which will confuse the users. This is due to all files uploaded to Teams conversations will be dumped into the channel root folder without proper organization. Therefore, to resolve the complicate file structure's problem for hard to find and manage a certain file, users will try to reorganize the files in MS Teams. Once the users try to reorganize the files by moving it into properly named folders, it brings the bad effect of the broken file links in the conversation [15]. Although MS Teams allows users to search for a file, this con will also force users to leave the unstructured files or need extra solutions to solve the issues of broken file links.

Limitation 5: Limited flexibility

The **limited flexibility** in MS Teams may lead to **users' time waste to do manual replications in Teams**. For example, the limited flexibility such as users are not allowed to move channels between teams may cause them to need to copy the channel content manually to a new created channel in another team when they would like to reorganize or modify their teams structure [15]. Besides, users not able to replicate a team in MS teams which lead to the need of manual replication which is time wasting and the productivity is lowered.

Limitation 6: Others

MS Teams provide a limitation of **uncomplete integration with functionalities of Skype**, which the online meeting experiences in MS Teams are still not achieve optimal user experiences. For example, the features are still not intuitive and do not support some capabilities such as users cannot take notes in OneNote during meeting in Teams. Besides, **limited functions and features of MS Teams in mobile devices** may provide bad user experiences to users who are accustomed to the desktop version which contains full functions of Teams. Currently, MS Teams still faced the problem of unimpressive mobile version although the MS Teams supports cross-platform for different device [14].

Recommendation

Some recommendations can be suggested to resolve the main limitations of MS Teams. First, to solve the problem of too many similar tools in MS Teams which will confuse users and waste their time to choose the right tool, a smart assistant can be implemented in MS Teams to guide users form choosing their right tool to use. The smart assistant should be able to suggest the possible tools can be used to help users from completing their task and allow users to choose from the option provided. To further improve users' experiences, the smart assistant can predict and suggest the next possible tool may need by the users to allow them to complete their task step by step and in correct procedure. Secondly, the smart assistant can implement the unified search option for users to search for specific content in MS Teams via chat with the assistant bot. The smart assistant can also be used to enhance the notification function in MS Teams such as allows users to customize their notification and increase more useful notifications in Teams. Besides, to resolve the limited flexibility problem in MS Teams, the smart assistant should have the ability to help users from preventing the manual work such as move channel and replicate channel. Therefore, to save users time and hide the MS Teams problem of limited flexibility. Lastly to resolve the confusing file structures problem in MS Teams, the suggestion provided here is the MS Teams can improve its file structures. For example, the first suggestion is MS Teams can organize files smartly into proper structure and easy for search and view. Or else, MS Teams can allow users to easily reorganize files without the broken of file links inside a team channel. Moreover, a future innovation can be considered to improve MS Teams productivity is to add the functions to take note in OneNote during meeting to increase meeting experiences in MS Teams.

2.1.2 Summaries

Table 2.1 Summaries of Literature Review on MS Teams

MS Teams	Summaries
Functions & Features	Activity, Teams, Meetings, Chats, Calls, Files, Shifts, Command box, Tabs, Apps & Tools
Strengths	Increase productivity
	Seamless virtual collaboration
	Better focus on users' work
	Integration with third-party tools
	Guest Access
Limitations	Too many similar tools
	No unified search on product
	Insufficient notifications
	Confusing file structures
	Limited flexibility
	Others: • Uncomplete integration with functionalities of Skype • Unimpressive mobile version of MS Teams

Recommendation	Proposed a smart assistant in MS Teams which can: • Suggest possible tools & predict next tools to use • Implement unified search option • Enhance notification function in MS Teams • Help to increase MS Teams flexibility, automate tasks such as move channel and duplicate channel
	Improve files structures: • Auto arrange files in proper structure • Allows users to reorganize files without broken of file links inside team channel
	Add function to take note in OneNote during meeting

According to the Table 2.1 to summarize the functions and features, strengths, limitations, and recommendations of MS Teams, we discovered that although MS Teams had become the most common digital workspace app used by most of the enterprises for remote working now, but it still consists of some functions and features those can be improved by some external tools or software. To improve the pros and to solve the cons of MS Teams, different type of tools and applications had been proposed in MS Teams Store or external source to aid users when using the MS Teams. Also, some bots and assistants can be installed in MS Teams to increase its productivity and to make users' works become more efficient. In the next section, the previous works on digital assistants including general intelligent personal assistant, digital assistants or bots in MS Teams, and digital assistants for two other workspace apps will be reviewed to their functions and features, strengths, limitations and to provide recommendations to those solutions. Finally, to analyse the possible functions, features and innovations can be implemented in the proposed solution in this project.

2.2 Previous works on digital assistants

2.2.1 Intelligent Personal Assistant (IPA)

Introduction

IPA is a software developed to assist individuals based on their commands or queries which implemented from the features of advance AI such as NLP and semantic web [6]. Usually, an IPA nowadays had implemented NLP for user-machine interaction to allow users to communicate with it via voice. For example, the common IPA for now are Amazon Alexa, Google Assistant and Apple Siri. With NLP, an IPA will be able to translate human intention into commands and functions to be executed through speech recognition [6]. The main objective of an IPA is to provide users the hands-free capabilities to perform task and to achieve a specific goal. In this section, a review on IPA will be conducted to **investigate the general functions and features can be applied in our proposed digital assistant for workspace app** and to **analyse on its architecture and framework**.

Functions, Features & Strengths

Retrieve data and information

Users usually use their IPA to retrieve information such as weather, map, and calls which needed by them in their daily life. The general function of an IPA is **to share essential information and data to users and provide them with guidance and decision support**. Via the basic operation of an IPA to observe users' context, analysing and responding on their queries, the AI feature of an IPA can generate specific information to plan and reason on users' queries to return the specific outputs to them. The idea of user-centric context was first proposed by Schilit and Theimer to indicate user location, environment, emotional state, and others [21]. Based on the context, the AI assistant can relate user's tasks to those contexts to produce a context-aware solution for better planning and reasoning on users' command. An IPA should also automatically update any new information related to users' interested topics and provide them with notifications to ensure users can always receive the newest data about their favourite topics.

Search and Chat

Other common functions provided by a personal assistant are **search and chat function** [9]. Normally, an IPA allows users to perform web search and chat via voice command to gain specific results or outputs from the assistant. For example, general assistants such as Google Assistant, Microsoft Cortana and Apple Siri can help users to search content by using their default search engine based on users' keyword and return the searched results to them. It can save users' time to quickly search for a specific content without the need to open the web browser and perform the search function. Moreover, an IPA offers chat function to answer users' queries or response on users' text in human-like speech or text. The chat bot function inside an IPA can facilitate the interaction between human and the assistant to improve their experiences. By applying machine learning, the IPA should be able to chat with single user according to the user's preferences and interests.

Task and time management

The capabilities of an IPA including to **schedule joint activities, monitor, and notify users of specific timepoints and task** [5]. The time management function can help users especially the workers to schedule and manage their meetings or appointments and the task management help them to easily generate reminder and balance workloads. A user's productivity can be increased via a personal assistant to deal with a task or schedule with high complexity. A personal assistant with time management function allows users to schedule an event and meeting in a digital or cloud calendar and provide an automated calendar management. A personal assistant can also implement a task manager to accept problematic requests from users, provide suggestions and guide them to resolve the problems via different sub tasks [13]. This feature can help users to register, schedule and perform their task step by step. Furthermore, an intelligent assistant must be able to determine the feasibility of a task to help users' decisions in task management. Moreover, notification function is available to remind users at a specific time, schedule, or to perform any sub function when performing a task.

Control apps

An **IPA used in mobile devices** usually has the capability to **control certain logic of the mobile apps**. For example, users can access or manage some of their mobile apps via the personal assistant in their mobile device such as Google Assistant or Siri from Apple. Sometimes, this function also available in users' desktop such as some desktop assistants can help users to control a few numbers of application. An event-driven approach can be implemented for the personal assistant can execute a specific function to achieve certain goals when triggered by an external event or change. For example, an IPA can automatically open certain application when the timer started by users is stopped.

Personalization

A feature of an IPA is it **allows users personalization** on it. An IPA should be able to adapt to the specific preferences and characteristics of a user as the work styles are dynamic and highly personal depends on its user. Therefore, users must be able to customize their preferences over the actions can be executed by the IPA including notification, user-machine interaction way, task, and time management [13]. Besides, an IPA with AI technology can also automatically predict and personalize user preferences through machine learning unobtrusively to provide users with suggestions in daily life. Thus, the personal assistant can be more user-friendly and adapt to users' working styles to increase their productivity.

Limitations

Absence of visual aid

The popular speech-based IPA nowadays such as Cortana, Siri from Apple and Google Assistant are facing a limitation of **absence of visual aid** to further causes hard to retain data presented through a sequential-only audio channel [12]. Besides, a weakness also be brought by its absence of visual aid to increase the difficulty in returning visual outputs in human-like speech format for a personal assistant that allows two-way communication.

Recommendations

Improve human-assistant interaction

Besides from general solution of voice assistants, the human-machine interaction can be improved by **the assistant can analyse and translate both users' speech and gestures into an executable command** or a query [6]. AI technology can be implemented to achieve this feature for users to interact with their personal assistant from anywhere and anytime even during situation that they are not convenience to type or speak their command. To enhance the human-assistant interaction, an **UI** can also be developed for users to choose to interact with their assistant in voice or type command. For example, a 2D or 3D avatar can be designed for the personal assistant to have its UI and provide users with better experiences such as it can perform different face expression or gesture based on user's input. To provide a more immersive experiences to users, VR or AR technology can be also developed for an IPA to interact with users in a more direct and intuitive way. Therefore, users can interact with their personal assistant by using VR or AR for users to feel it like a real assistant to assist them. It can also provide users with a more real and immersive experiences when user chats or interacts with the personal assistant via an intuitive UI.

Architecture & Framework

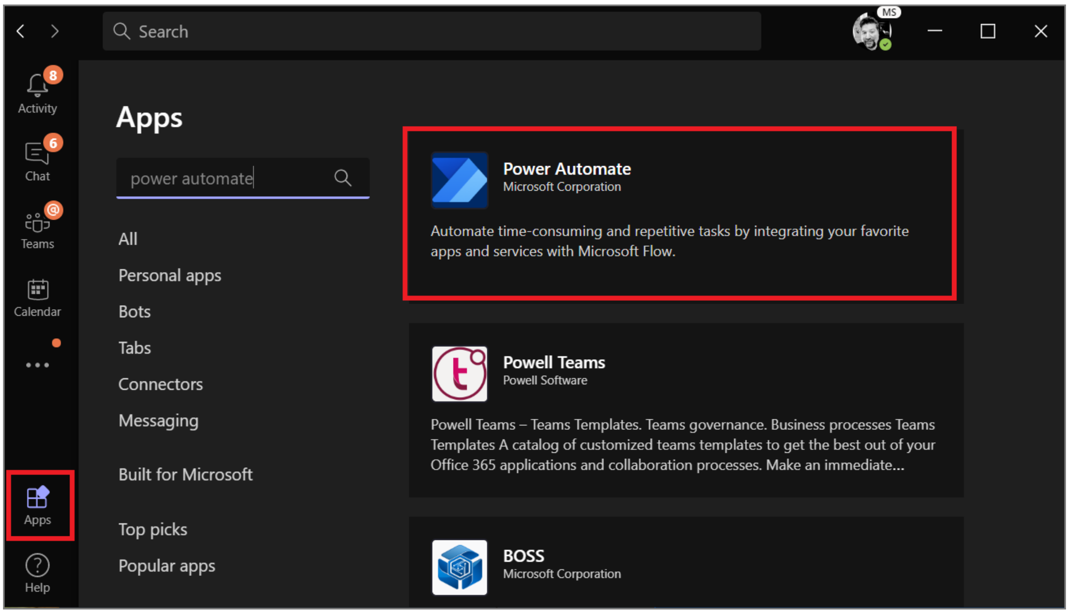
Croatti and Ricci had conducted a paper to develop personal assistant agent to be run on mobile and wearable devices using JaCa-Android framework based on JaCaMo platform [5]. Based on the paper, the framework to design a personal assistant should be possible to exploit agent-oriented programming and multi-agent programming. An agent-oriented programming model proposes programming constructs for developing agent belief, goals, basic capabilities, and a defined set of reasoning rules. According to their paper, a level of abstraction in modelling and exploiting will be prepared and hide the low-level details including mechanisms context and complexity of task implementation by expose API to the users. In the architecture proposed, the personal assistant can interact with users via observing user's inputs and extract the important context which is relevant to user's tasks. An artifact is designed to compose a set of operations can be done on the assistant side for users to carry out an action by instantiating the artifact. For its benefit, it can reduce the system complexity. The artifact also aims to expose a set of observable attributes modelling the observable environment state for the personal assistant can perceive users' external changes dynamically. Its architecture has also applied event-driven approach to trigger an action and has the capabilities to auto-determine subgoals and create a sequence of actions. This will be a preferable solution in our proposed digital assistant to guide users on performing actions in a sequence to achieve their final goal. The reviewed paper also proposes asynchronous execution of actions can be executed by the personal assistant via different artifact.

To relate the architecture and framework reviewed to our proposed system, the agent-oriented programming concept can be implemented in our proposed system to develop the digital assistant with basic capability to execute users' commands, a goal to provide correct outputs, and a set of rules on how to answer users' queries and perform certain functions in MS Teams. Instead of mobile framework, our proposed system will be developed by using desktop framework to model and exploit the digital assistant in a desktop platform. On the perception view, an artifact can be applied in the proposed solution to detect external events of a user to always predict the most accurate command and result. Moreover, the event-driven approach and sequence of actions will also be implemented in the system framework of this project with modifications which will be designed in Chapter four.

2.2.2 Power Automate in MS Teams

Introduction

Power Automate in MS Teams is a cloud-based workflow engine developed by Microsoft to automate users' process or task in MS Teams. Power Automate in Teams can be installed as an app in MS Teams by select the Apps icon and search for the Power Automate App [18].



Power Automate
Microsoft Corporation

Automate time-consuming and repetitive tasks by integrating your favorite apps and services with Microsoft Flow.

Powell Teams
Powell Software

Powell Teams – Teams Templates. Teams governance. Business processes Teams Templates A catalog of customized teams templates to get the best out of your Office 365 applications and collaboration processes. Make an immediate...

BOSS
Microsoft Corporation

Figure 2.1 Add the Power Automate App to MS Teams (Microsoft, 2021)

Functions and Features

Create and Access flow

Users can create cloud flows from a custom or a blank template in Power Automate App by accessing the 'Create' tab under the Power Automate App. After that, users in MS Teams can access the flows they have created by accessing several tabs in MS Teams. For example, the 'Home' tab to access the summary of flows and 'Business process flows' tab to view their business process flows. Besides, users can view the detail information about Power Automate via the 'About' tab in MS Teams [18]. A flow is used to trigger certain actions from Teams messages or from other situations defined by the users.

Manage approvals

Users can manage the approval requests they had received or sent via the Power Automate App in MS Teams by accessing the 'Approvals' tab. Power Automate in Teams allows users to create an approval flow and view all approval requests those are pending action from the users and the approval history. This function helps the users to easily list and manage their approvals and to avoid time wasted to search for specific approvals in their emails [18].

Chat

Users can access the ‘Chat’ tab to interact with an always online Power Automate bot in MS Teams. Via the chat function, users can easily perform an action in Power Automate App by typing and send the command to the bot in Chat. If users are not familiar with the possible commands they can use, they can easily get the list of commands the Power Automate bot can handle by typing the ‘learn more’ command. This can simplify their job to list all flows, view, and run certain flows [18].

Adaptive Cards

Adaptive Cards are platform-agnostic user interface snippets of used created in JSON format provided by Power Automate in MS Teams for users to share information or to collect data. Users can create their own Adaptive Card and use those cards in a flow to post to Teams or channels, or as a flow bot to a user. This function allows users to collect data by posting the Adaptive Card to a team or channel and wait for other users to response to the card.

<u>Strengths</u>

Many useful templates provided

There are **a group of templates and sample automation flows provided by Power Automate App in MS Teams for user to create, use or modify a flow from the templates**. It benefits users across the world who have similar problems in MS Teams to create and use a flow based on existing templates to complete a common repetitive task. Besides, the advantage is that users can easily search a related template by typing the keywords in the search bar. In result, it can **save users’ time to create and use a flow**, and for them to quickly perform a job by using the useful templates provided.

Automate task in MS Teams

Power Automate in MS Teams allows users to trigger specific actions in their flow created via keywords mentioned in a chat or channel conversation to execute some MS Teams functions. For example, to create a team or chat, get messages and post messages. Power Automate App provides many possibilities to users such as to automate approval processes, track customers opportunities, set up notifications and others. With this feature, Power Automate App **makes the processes in MS Teams more efficient and reduce users' workloads**. Also, to eliminate human error especially for those repetitive steps to complete similar tasks. By using the automate flows in Power Automate, the reliability and the correctness of users' tasks increased. Besides, the Power Automate RPA allows users to automate the business process by performing the repetitive tasks.

Can integrate with many other apps or tools

One benefit of Power Automate App in MS Teams is that **it can integrate with many other apps than MS Teams including third-party apps** such as Microsoft 365, Planner, and Microsoft Form. In result, it provides users with the benefit to create a more dynamic and useful flow. The Power Automate RPA in MS Teams uses many connectors to allow the APIs of third-party software can communicate with MS Teams to automate business process. For example, users can create a flow to notify Teams when there is an action triggered in another apps such as a file is added to the SharePoint. This feature saves users' time to shift between apps to check and manage certain actions and data. Moreover, it also allows easier ways for users to exchange data in MS Teams with data in cloud. In result, save time and increase productivity.

<u>Limitations</u>

Not suitable to build complex and dynamic workflow

Power Automate in MS Teams **does not support complex workflow** such as a scenario to span multiple departments and have a set of rules to select approvers. It is more suitable to be used for simple workflow only such as to send an approval to a specific department. In result, it causes the limitation of Power Automate in MS Teams which cannot support more complex users' requirements to automate their tasks. This will be a problem to the organization which want to maximize their productivity in MS Teams by creating and using a more dynamic workflow according to their business scenarios. Moreover, the limitation includes **users need to analyse all scenarios before they actually implement a workflow** by using Power Automate as once the flow is run, users will not be able to change it in the processes. This may lead to the difficulties for some users who do not have an enough intuitive ideas in their mind to create and run a flow. Besides, it restricts users' possibility to create a dynamic workflow and need to spend extra time to analyse the business model to make decision on how they should create a desired workflow in Power Automate.

Only support sequential workflows

Power Automate in MS Teams **only supports sequential workflows** which do not allows users to back to the last step in a workflow chain. This limitation causes users may not be able to create their desired workflows which is more dynamic and near to their real scenarios in MS Teams. In result, the conveniency and experiences of users to create and use a workflow via Power Automate App are decreased.

Not completely low code

Another disadvantage of Power Automate in MS Teams is that **it is not completely low code** to force the users to still need some technical knowledges about coding to use this app. For example, the users need to have extra knowledges and do more research when they want to create a larger and more complex workflow by adding in more functions in it. The processes for users to create their desired workflows in Power Automate in MS Teams are still not user-friendly enough even templates are provided. For now, the lacking support agent to help them when users are creating and managing their flows will still be a problem in Power Automate in MS Teams.

Limited actions in MS Teams

The last limitation is the **limited commands can be run by the Power Automate bot to execute the MS Teams functions** which is not enough to cover all functions in MS Teams. This limitation results in lower productivity of MS Teams users as they cannot include certain MS Teams actions in their workflows created to automate their tasks.

Recommendation

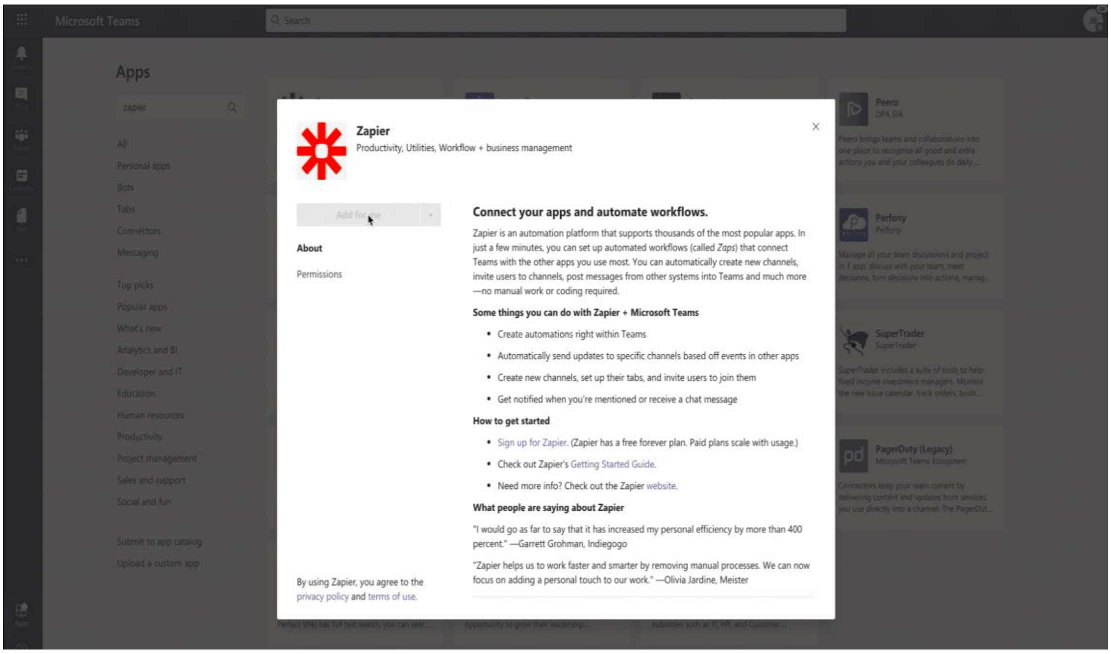
In conclusion, the four main limitations of Power Automate in MS Teams are not suitable for complex and dynamic workflow, only supports sequential workflows, not complete low code, and limited commands to be used. To solve these limitations, the first recommendation can be provided is to simplify its steps to build a workflow. Therefore, users without coding knowledges can also build their workflow with simple steps and procedures to save their time. Besides, the workflow function in Power

Automate can be improved by allowing more complex and dynamic workflows to be built by users. For example, to allow users can return to the previous step in their workflow chain or allow them to select options or make changes when the workflow is running for them to create and execute workflows dynamically. Lastly, the problem of limited actions of the Power Automate App in MS Teams can be resolved by increasing the commands and actions of Power Automate in Teams such as to manage shifts.

2.2.3 Zapier for MS Teams

<u>Introduction</u>

Zapier is an integration platform designed for MS Teams to integrate with more than 4,000 third-party applications. Same to most of the automation tools, Zapier works on a trigger and action basis such as users can receive a Teams notification when they trigger certain action in Teams or in other apps. To use Zapier in MS Teams, users can search Zapier in the Microsoft's app store and add it to the MS Teams [23].



Connect your apps and automate workflows.

Zapier is an automation platform that supports thousands of the most popular apps. In just a few minutes, you can set up automated workflows (called *Zaps*) that connect Teams with the other apps you use most. You can automatically create new channels, invite users to channels, post messages from other systems into Teams and much more —no manual work or coding required.

Some things you can do with Zapier + Microsoft Teams

- Create automations right within Teams
- Automatically send updates to specific channels based off events in other apps
- Create new channels, set up their tabs, and invite users to join them
- Get notified when you're mentioned or receive a chat message

How to get started

- Sign up for Zapier. (Zapier has a free forever plan. Paid plans scale with usage.)
- Check out Zapier's Getting Started Guide.
- Need more info? Check out the Zapier website.

What people are saying about Zapier

"I would go as far to say that it has increased my personal efficiency by more than 400 percent." —Garrett Grohman, Indiegogo

"Zapier helps us to work faster and smarter by removing manual processes. We can now focus on adding a personal touch to our work." —Olivia Jardine, Meister

By using Zapier, you agree to the privacy policy and terms of use.

Figure 2.2 Add Zapier App to MS Teams

Functions and Features

Schedule & Automate works

The main function of Zapier is to automate repetitive tasks or routine tasks by it can trigger an action in prescheduled period of users such as every day, every day of the week or every day of the month. Users are allowed to schedule and automate their works by using Zapier with MS Teams via several steps. First, to authenticate the schedule by Zapier and MS Teams. Secondly, to build an automation flow by choosing one of the apps as the trigger. After that, users can pick a resulting action from the other applications. The fourth step is that users need to select the data and information they want to send from one application to the other. Basically, these four steps will take less than 3 minutes of users to create and automate their workflows. The final workflow created by a user is called Zap. Zapier provides various type of triggers in MS Teams such as when a new channel or team is created, when a member is mentioned in a channel, and when a new message is received in a chat. The type of triggers provided by it can satisfy most of the users' requirements in MS Teams to connect to other apps. Besides, Zapier also provides many templates those can be used to automate works in MS Teams which is useful to increase users' productivity.

Strengths

Simple steps to schedule task

Users can **easily schedule their task** via Zapier in MS Teams such as to post message to MS Teams channel or chat by only four simple steps as introduced in the last part. It **saves users' time to schedule a task** and it is convenience for them to automate a task. Another benefit is that all users can schedule their task in low code environment which do not require any background of the programming knowledges.

Automate workflow to send notification

Zapier **allows notifications to be automatically send to the chat or a specific team or channel in MS Teams** when triggered by an action in MS Teams itself or other apps. This feature increases conveniency of team members to follow up on a specific message and users will never miss or lost certain data and messages in other apps. Besides, this feature of Zapier also provides benefit for an organization to maintain its customer relationships as employees can easily follow up with customers' messages in any platform and real-time update the business process. For example, Zapier can send a message to a specific channel in MS Teams when new invoices are paid in Teamleader app. This function allows a team to easily manage their business and allows the message to be viewed by every member inside the channel in real-time to avoid the need to forward the message.

Many integration to third party apps

Besides, another strength of Zapier is that **it allows MS Teams to connect to thousands of the most popular apps** for users to automate their works with no code knowledges required. This strength can help to **save users' time** to switch between different apps to collect different data which can increase their productivity, and the integration feature allows users to simply receive data from the other apps in only the MS Teams platform. For example, the popular apps those MS Teams can integrate to such as Google Calendar, Slack, Spotify, Discord, Gmail, and others. This advantage also allows users to focus on their work in MS Teams by automating the information transfer processes between different web apps. Also, the large amount of integrated app options increases the flexibility for users to schedule and automate their task dynamically based on different scenario and case. For example, Zapier allows users to notify Teams when new Google Calendar events start.

<u>Limitations</u>

Not suitable to schedule complex workflow

The advantage of Zapier includes to schedule a task in short steps and less time. However, this can also be a limitation to cause **users hard to schedule a complex task or workflow by using Zapier** such as to involve more than two apps or insert loop condition inside the workflow. This restricts users from automate their complex workflows and routine tasks and decreases their conveniency to use Zapier for task automation. In result, time waste will occur if users would like to schedule multiple tasks to achieve one task which consist of multiple workflows such as tasks in a big chain or chain involve loops.

Limited triggers or actions in some apps

Zapier has a limitation of **limited triggers or actions in MS Teams and some apps it can connect to although the app provides many API available**. For example, it lacks function to create new team or channel in MS Teams and mostly Zapier is only used to receive message and notification in MS Teams. The lacking actions in MS Teams may lead to inconveniency of users to execute specific MS Teams function in their workflows.

Need to turn off the whole Zap when users want to modify it

Zapier also includes a disadvantage of **not allow users to modify a Zap process when it starts running**. This may cause users to **waste time** to turn off the Zap before they can modify the processes and then to turn it back on whenever they would like to do a slight modification to a Zap. Also, it **lowers the dynamicity of a workflow** can be created in Zapier and makes the users not convenience when editing Zap.

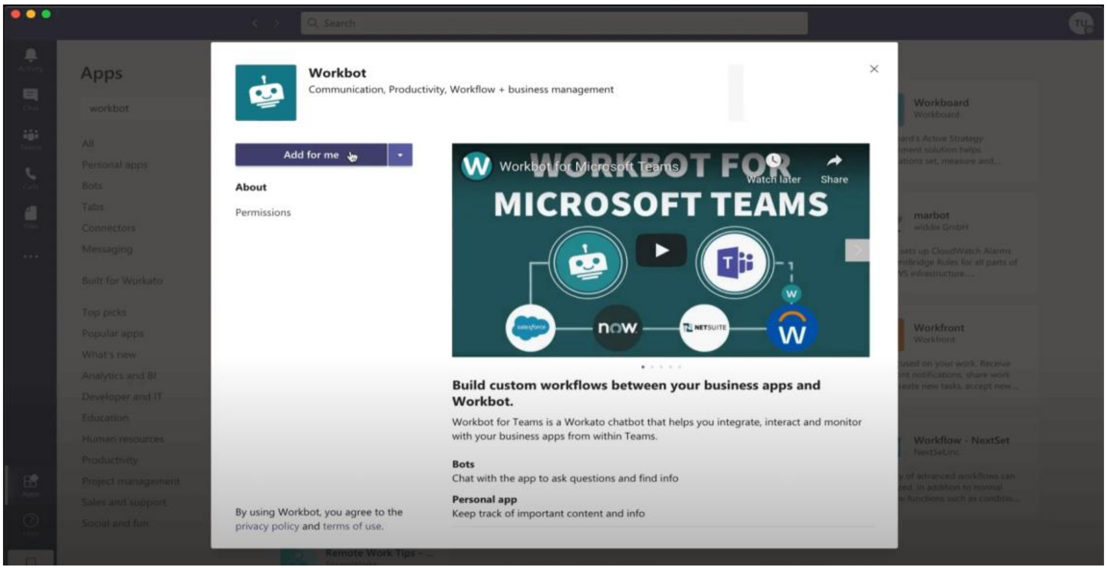
Recommendation

One limitation faced by Zapier is it is not suitable to schedule complex workflow. The contradiction is that the increased simplicity to use Zapier had lower its function to build a more complex and dynamic workflow such as workflow including loop. To solve this problem, the suggestion provided is that Zapier can allow users to insert a loop condition in their workflow and allow them to involve more possible routes or apps in their workflows. Thus, it can increase users' flexibility to build a workflow which is more able to meet their task requirements in MS Teams. Secondly, the triggers and actions of Zapier can be improved to increase the triggers and actions can be applied in some third-party apps and MS Teams. This is to increase users' ability to execute more actions in their needed applications to increase their conveniency and productivity when performing tasks. Also, the dynamicity of Zapier can be increased by allowing users to modify a Zap process when it starts running. In result, users can have better experiences when using Zapier to create their workflows to automate tasks in MS Teams.

2.2.4 Workbot in MS Teams

Introduction

Workbot for Teams is a Workato chatbot designed to provide instant access to data from apps within MS Teams and allow users to monitor and interact with their business apps in real-time. For example, users can configure Workbot to send messages to a team channel or chat when certain events happened in MS Teams or from other apps. Workbot for MS Teams allows users to complete their works within the chat space and choose the notifications they want to receive it as a direct message. To use Workbot in MS Teams, users can search the Workbot in the MS Teams Apps store and add this application to their MS Teams.



Workbot
Communication, Productivity, Workflow + business management

Build custom workflows between your business apps and Workbot.

Workbot for Teams is a Workato chatbot that helps you integrate, interact and monitor with your business apps from within Teams.

Bots
Chat with the app to ask questions and find info

Personal app
Keep track of important content and info

By using Workbot, you agree to the privacy policy and terms of use.

Figure 2.3 Add Workbot App into MS Teams

Functions and Features

Create & Modify workflows

Users are allowed to list, create, and modify workflows in Workbot, and the workflows created are known as Recipes in this bot. Recipes are automated workflows created by users to access and manage data across all users' apps directly with MS Teams. A Recipe consists of one trigger and one to many actions that can be carried out by the same or different apps when the trigger event occurs. For example, when users enter the command of view users to Workbot in MS Teams, it will trigger the actions to search for users in Salesforce and post the results to Workbot in MS teams Chats. Through the workflows created, users can easily collaborate on business information available in multiple apps by Workbot to fetch the records from external applications directly into the chat window in MS Teams. Besides, users are allowed to approval workflows in Workbot for MS Teams. Users can approve or reject a request sent to a specific team channel or chat from other apps to quickly handle an approval. For example, this function can be used in the situation when to approval a customer purchase in e-commerce platform.

Chat options

Workbot allows users to interact with it by typing commands in the chat with it. In the chat to Workbot in MS Teams, users can receive message or notification from other apps based on the workflows they had created, and they can even perform an action on that message such as to approval or reject it. Moreover, users can type in their commands to perform actions such as to list certain information. Also, Workbot in MS Teams provides functionality to auto follow-up certain messages such as to notify a user or a team channel when an issue is raised, assigned, or updated from a monitoring tool. Moreover, Workbot can give alert to users based on their defined thresholds or settings by posting the related alerts to a specific chat or a relevant team channel.

Strengths

Automate workflows

Workbot **increases productivity by automate workflows** created by users in Workbot for MS Teams. In Workbot for MS Teams, one or several actions can be auto executed once a trigger event set by users is triggered. For its pros, Workbot can notify users with two different types of messages in Teams chat to automate their business process. The first are messages generated in respond to specific requests. For example, for users to generate data collected in other apps such as sales data from Salesforce.com Inc. The second types of messages are the messages auto created based on user activity pattern such as to sync certain data to a specific app periodically. The Workbot can also help to transfer data between MS Teams and other different apps to save users' time to manually collect and transfer the information in different apps.

Suggest next steps to help user's decision

Workbot in MS Teams executes the Assist technology which is a learning engine that able to understand the way users interact with it and the interaction context. Besides from proactively gives users the information they are looking for based on their command, Workbot can also **help users act on the information by predicting and suggesting next steps**. This strength of Workbot in MS Teams helps users to make their decision on the next step they may want to conduct to reduce the users confuse on the various functionalities they can perform in the next step.

Limitations

Limited actions can be done in MS Teams

There are **limited actions that Workbot can done in MS Teams** such as to receive a Teams notification or to post a message only. The not enough actions can be executed in MS Teams may lead to the **inconvenience of users** to manually carry out certain actions when they receive some tasks and notifications in MS Teams. In result, this limitation makes users to not be able to create a completely automated workflow to fully satisfy their business requirements which the workflow created may need their manual interaction inside the flow. For instance, the lacking actions to be done by Workbot in MS Teams such as to create a new team or invite new member to a team, and to share data and resources to a specific team or all teams.

Limited integration apps

The weakness of **limited integration apps in Workbot for MS Teams** provides limited app options for users to manage their data and services across various platforms they are frequently used in their daily works. In result, it makes this integration platform become less powerful and decrease the productivity as users may need to spend extra time to switch between the third-party apps and the Workbot to manage different task and data. Moreover, the creation of a Recipes will also be limited as users may want to involve an app that is not integrated in Workbot to make them not able to create their desired workflows.

Recommendation

The recommendations to improve Workbot for MS Teams are to increase its actions can be done in MS Teams and increase the integration apps in Workbot for MS Teams. The first recommendation can be achieved by implementing more functions of Workbot for it to execute more Teams functions by using the API provided by the MS Teams. In results, to help users to create more powerful workflows and automate more tasks in MS Teams. Secondly, the suggestion to increase integration apps in Workbot is important to allow users to create their workflow which involve more applications such as the third-party apps to save the time they need to switch between different apps manually. Besides, users can create a more complex and dynamic workflow which is nearer to their real-task scenario to automate their tasks efficiently.

2.2.5 Thomson Routers Elite (Workspace assistant)

Introduction & Functions

The company Thomson Reuters Elite had launched a workspace assistant available at Amazon Alexa store for lawyers to increase their work efficiency. The workspace assistant allows lawyers to access and manage their time schedules at digital workspace apps by using the Amazon Echo or other Alexa enabled device [22]. The assistant proposed had prepared a complete solution for business management by allowing users to manage different business activities in one place with intuitive view. This is the first solution to lead the enterprises to execute all their business operations in one place including customer and staff management, risk management and financial management.

Strengths

Seamless business management functions

The workspace assistant **allows users to access and manage all business processes in only one workspace** for lawyers to be more responsive to the requests of their client under the more cost-effective environment. This solution had provided lawyers with a full 360 degree of view of their customer relationship in an instant. The business operations can be done by lawyers via the workspace assistant including to control time entry, track their billable hours, provide legal services to their customers, and others. This function increases lawyers’ conveniency and simplicity to deal with their customers especially the workspace assistant can automatically calculate and post the time spent on a specific customer matter based on user inputted billable hours. Therefore, it reduces and automates the task initially to be done by the lawyer and the lawyer will be able to query details of the billable hours via interaction with Alexa. This feature is critical for lawyers to resolve the challengers they faced effectively and enhance their customer’s experiences.

Integration with third-party functions

Via the integration feature, users will be allowed **to perform multiple business process at only one workspace platform to save their time** to switch between different apps. In this solution, the workspace assistant had integrated with the existing Elite Workspace platform to allow users to access most of the Elite solutions to improve their efficiency and productivity. In result, the workspace assistant had provided users a unified access to tools and solutions they need to help lawyers from completing their tasks smoothly and spending less time. It can also benefits lawyers form increasing their productivity by saving their time to perform different operations.

Intuitive UI

The workspace assistant **allows users to interact with it via voice-only inputs** to question Alexa on the time management function. For example, lawyers can request Alexa to start or stop a timer via simple voice command to record the number of billable hours. The voice interaction can replace the traditional typing method for human to interact with machine in a more intuitive, direct, and quicker way. The workspace assistant provided by the Thomson Reuters Elite also give benefit of allowing users to access the solutions from desktop browsers and mobile devices. This feature had increased users’ flexibility to launch their workspace app with their prefer devices and easier to conduct works via simple voice interaction to the assistant. Therefore, they can simply perform a task no matter are they familiar with the device they are using to increase business efficiency. Moreover, it also reduces the complexity for lawyers to manage business, matter and time and even reduce the costly node integration between systems.

Limitations

Limited performances

The limitation of the workspace assistant solution includes **limited performance to perform task**. For example, the user reviews on this solution had reported that it is time-consuming to upgrade certain data and for them to reload the page. This limitation may be caused by its feature of integrate multiple business operations and data into only one place to increase the server's load especially when many users are simultaneously accessing the system. In result, the unified access to Elite solutions may not be much powerful for lawyers to conduct their business efficiently due to the time consuming when loading the page and to process data retrieved from multiple sources. This limitation can cause **extra time needed for users to process their tasks**.

Need improve on the reporting capability

Another weakness of this solution is that **limited reporting capabilities are provided**. Even though the solution had allowed users to access all operations and data at one place, the restriction of its reporting capability can also limit users from generating dynamic, useful and structured data for their reports. Therefore, one con of this solution is the improvements are needed for the workspace assistant to export report in more dynamic and customizable format. This limitation can cause users' inconveniency to export their data or force them to use third-party tools to restructure and export their needed data in their desired format. Besides, the strength of unified access to tools can be removed in this solution due to the lacking reporting capability.

Recommendations

Algorithms should be implemented to the workspace assistant for it can serve every single user based on their preferences and customization. Instead of process and load all data that consume time, a recommendation to this system is that AI ability can be implemented in this solution to calculate and retrieve only specific data needed by users to perform their current task. Via the smart AI to automatically collect needed data only from different sources, the system performance should be increased and reduce the page loading time. Besides, the algorithm should be able to extract only important data from various data sources to avoid the server to host too many data and lower its performance. Another suggestion is that the AI technology can be used to improve the task performance on reviewing the time allotted to different customers even to a very small-time amount. By investigating and applying suitable algorithms, the response time of the system can be increased to provide users with better experiences. Moreover, the improve in reporting function is also an important suggestion for a law firm to facilitate a completely paperless workflow system. By increasing its reporting capabilities, the organization can reduce or remove their paper usage by exporting everything including bills electronically to speed up digital transformation. Therefore, this solution should improve its reporting system such as to increase its report customizability, dynamicity, and the way to export a report.

2.2.6 Google Assistant (Workspace assistant & Voice assistant)

Introduction & Functions

According to Google Workspace Updates, the developer teams of Google Workspace had allowed users to access to its workspace via Google Assistant which available to users who enable the “Search and Assistant” additional service on Google [3]. Google Workspace is a workspace developed by Google in year 2006 which include collaboration tools such as Gmail, Calendar, Drive, Meet, Docs, Slides, and others to improve productivity. Besides, Google Assistant is also an AI assistant developed by Google that is mainly available on mobile and smart home devices to access Google functions, control users’ smart home devices, open mobile apps, and others. Now, users will be available to use Google Assistant to access Google Workspace to perform their tasks on this workspace platform such as to schedule meeting and invite participants via email, dial into a meeting and receive notification of next meeting. The aim of this solution is to provide users with a more seamlessly access to and perform tasks in Google Workspace via voice commands.

Strengths

Increase productivity when using Google Workspace

The main strength and objective of this solution is **to provide a more user-friendly method for users to access Google Workspace**. Via this solution, users can easily perform certain tasks in Google Workspace via voice command such as to reschedule or cancel a meeting in Google Calendar. Therefore, it can reduce their needs to access to Google Workspace functions and input data via normal GUI which is extra time needed. Besides, Google Assistant is efficient to answer 59.80% of daily based questions in their survey conducted [7]. Its strength consist of it has **high performance to receive and translate human's natural languages in changing levels of foundation noise** same as Siri. Therefore, it can easily retrieve users' commands from their voice and able to provide a more accurate result to perform certain tasks in Google Workspace. Thus, Google Assistant helps to increase workspace productivity by allowing users to perform task quicker and more accurate.

Voice assistant feature

Google Assistant is a popular voice assistant powered by AI to assist users with data searches, decision making and executing specific commands via natural language in a spoken format [11]. The benefit is that **it can facilitate human-computer interaction in a more intuitive and direct way** compared to traditional typing method. According to Zwakman, Pal and Arpnikanondt, a survey conducted in their paper had indicated that 52% of users would like to use a voice assistant over websites and tasks due to its conveniency and capability of work hands-free [11]. The voice assistant's hedonic strengths and its usability had become a critical reason in this era to facilitate users from adopting the assistant. The usability of Google Assistant in Google Workspace includes to allow users from make questions and execute commands via voice to the assistant. The high human-computer interaction between users and Google Assistant also benefitted from Google to promote two-ways communication for the assistant can response users in human-like speech. Besides from it, a voice assistant also provides benefit of capability to understand non-conversational cues to increase users' command flexibility. For example, the AI-based Google Assistant can auto detect non-conversational cues in users' spoken command to analyse the correct command or to perform multi-functions by one spoken command separated by non-conversational cues.

In result, Google Assistant has strength to reduce time waste for users to perform tasks by using Google Workspace and increase their conveniency on using the workspace functions.

Limitations

Google AI limitation

According to Zwakman, Pal and Arpnikanondt, the general limitation of an AI-based voice assistant with speech-based interfaces is **lagging** [11]. This may also be one limitation of Google Assistant to response users' voice commands in few seconds. Although currently there is still lacking a well-defined metrics to measure a voice assistant's usability and performance, the response time in seconds can still become a large weakness of this solution compared to a GUI-oriented object which can response in milliseconds. For Google Assistant, some previous works still measured that there is possibility of lagging in its system to get some answers or execute commands. Besides from the response time, **the accuracy of outputs provided by a Google Assistant generated by its AI algorithms also consist of limitations and can be improved**. For example, the survey conducted by Tulshan and Dhage stated that Google Assistant has limited capabilities in human free interaction same as other voice assistants such as Alexa and Cortana [7]. Therefore, it needs improvements to achieve more human free interaction in the future for users to fully remove their actions of input parameters via keyboard or touch screen and to provide them with a more immersive experiences when using Google Workspace.

Limited functions

The usage of Google Assistant to access Google Workspace is facing the problem of **limited functions can be done by users**. For example, users can only deal with a few solutions in Google Workspace such as Google Calendar and Google Meeting via Google Assistant. Besides from time and meeting management, the Google Assistant should have more capabilities to execute different type of commands from users to satisfy various of their requirements, and to further improve user's experiences. Thus, the current solution is still lacking actions can be done to maximize the Google Workspace productivity by the Google Assistant. Moreover, the **lacking integration**

with third-party apps will also be a limitation of Google Assistant in Google Workspace. For example, users will need to spend extra time to shift between apps if they would like to perform task or retrieve data from different third-party app.

Recommendations

The first recommendation to Google Assistant to aid users in Google Workspace is to further improve its strength of efficient voice recognition feature to sustain its value compared to other common voice assistants. For example, to add more emotions in its speech to response on users' commands to provide them with better interaction experiences. Another benefit of this feature is to allow users can access to Google Workspace solutions quicker and more convenience via Google Assistant. A suggestion provided by Zwakman, Pal and Arpnikanondt is that a genetic algorithm can be created and used to maximize the accuracy of automatic voice recognition to generate different parameter values based on a specific environment surrounding the user [11]. In result, to improve the assistant's voice recognition accuracy in different type of environments and its capability to understand commands from both native and non-native English speakers. This is also to provide a more accurate outputs to its users. Besides, another recommendation is the Google AI can be improved in deeper to provide more immersive experiences to all Google Workspace users especially for desktop users. For example, to provide the capability of touch-free meetings for users can completely focus on their task only based on the smart assistant's guidance and suggestion. In result, by improving the capabilities of a smart assistant on a workspace app, it can further adopt an organization's workers into hybrid work mode of this era's trend. To improve its productivity, Google Assistant can also add more functions for user to execute and perform their tasks quicker and with high accuracy. For example, to integrate the assistant with some general third-party apps and allows users to execute more functions in Google Workspace.

2.3 Summaries & Critical Remarks

Table 2.2 Summaries of Literature Review on Digital Assistants

O – Optional feature, P – Partially implemented

	IPA	**Power Automate**	**Zapier**	**Workbot**	**Thomson Routers Elite**	**Google Assistant**	**Proposed System**
Basic Functions							
Retrieve & Output data	√	√	√	√	√	√	√
Perform search options	√	√			√	√	√
Chat function	√			√		√	P
Time management	√	√	√	√	√	√	√
Task management	√	√	√		√	√	√
Notify user	√	√	√	√	√	√	√
Access third-party apps	√	√	√	√	√	√	O
Provide guidance	√			√			√
Personalization	√					P	P
Coding knowledge needed		P		P			
User Interaction							
Chat interaction	√	√	√	√		√	√
Voice interaction	√				√	√	O
Graphical User Interface	O	√	√	√	√		O
Adaptive cards		√	√	√			
Supported Platforms							
Mobile platform	√	√				√	
Desktop app	P	√	√	√		√	√
Web-based app		√	√	√	√		√

General Actions in Workspace Apps							
Access data		√	√	√	√	√	√
Manage data		√	√		√		√
Schedule event		√	√			√	√
Send alert		√	√	√		√	√
Create & Manage Workflow in MS Teams							
Create flow		√	√	√			√
Edit flow		√	√	√			√
Access flow		√	√	√			√
Run flow		√	√	√			√
Support workflow with if-else condition / loop		√	√				
Support workflow backward							√
Provide templates		√	√				√
Actions in MS Teams							
Send notifications		√	√	√			√
Send / post messages		√	√	√			√
Execute Teams app’s function		√					√
Create team / channel		√	√				√
Schedule meeting		√					√
Create & start calls		√					√
Create & access files		√					√
Create & manage shifts							√
Create & manage tabs		√					√
Manage approvals		√		√			√

2.4 Discussions

According to Table 2.2, six digital assistants had been reviewed and summarized according to their basic functions as a digital assistant, user interaction method, supported platform, general actions in workspace apps, ability to create and manage workflow in MS Teams and actions in MS Teams. The six digital assistants to be compared and contrasted in the chapter are legacy IPA, Power Automate, Zapier and Workbot for MS Teams workspace app, Thomson Routers Elite for workspace app, and Google Assistant for Google Workspace.

The first category to compare these reviewed solutions are the basic functions can be conducted by a digital assistant. Based on Table 2.2, all the six systems can perform function to retrieve and output data to users, notify users, help them to access third-party apps and manage time. All these four functions are important for a digital assistant to save users' time to search for a specific data and its update, access third-party apps and schedule time manually. Therefore, the proposed system is planned to implement the functions to provide necessary data and notification to a user who is using MS Teams, and the function to access third-party apps will be an optional innovation feature of the proposed system due to the time limitation and its complexity to develop this feature. Besides, other common functions of a digital assistant are the task management and the function to perform search options which implemented by four to five digital assistants reviewed. Thus, the proposed system will also develop these functions to help users to manage their task effectively in MS Teams workspace app. Besides, to allow users to easily search for a specific template, data, or command in MS Teams including chat and team channel.

According to the table, the function to provide guidance only supported by general IPA and Workbot, but it will also be an important function to be developed in this project to aid users in performing various MS Teams function. For example, the proposed system will be able to predict and suggest the possible next step or command to guide users from completing their tasks by using the correct function in MS Teams. Besides, only IPA, Workbot and Google Assistant support chat function of the digital assistant to chat with users, while only IPA and Google Assistant support user personalization on the digital assistant. This project will develop a simple chat function for users to have a simple conversation with the digital assistant in MS Teams and will allows users to do several easy customizations on the assistant to match their preferences when performing tasks to increase user experiences. Also, the proposed system is in a low-code environment and users can use this system without any coding knowledges. This feature can resolve the limitation of Power Automate and Workbot which required few coding knowledges of users to manage a complex task.

The Table 2.2 also shows that most of the digital assistants support chat interaction with users to receive users' command via typing exclude Thomson Routers Elite workspace assistant. Besides, the second general interaction way between a digital assistant and a user is that users can use the graphical UI provided by the system to perform their functions. The graphical UI supported by some IPA and all the other digital assistants exclude Google Assistant while IPA, Thomson Routers Elite, and Google Assistant allows users to interact with them by using voice command. Moreover, Power Automate, Zapier and Workbot allows users to input parameters or perform choice via Adaptive Cards in MS Teams. The proposed system will mainly develop the chat interaction method between users and the assistant, while the voice interaction and graphical UI are possible interaction ways to be developed in this solution. In the next category, the Table 2.2 shows that most of the digital assistants reviewed support platforms of desktop app and web-based app, while only three systems support mobile platform. Therefore, the proposed system will be developed to be used in both desktop and web-based environment for users can use the system for MS Teams in both desktop app and website.

Besides, the general actions of the digital assistants can perform in workspace apps had been summarized in Table 2.2. Exclude the personal assistant, all other systems can help users to access data in a workspace app to increase their conveniency. Moreover, the send alert function is supported by four solutions while the other functions to manage data and schedule event supported by three solutions. As all functions can be performed by more than half of the reviewed systems, all the four functions will be implemented in the proposed system for users to access and manage data, schedule event, and receive alert in workspace app. Also, the table had compared and contrasted the actions of Power Automate, Zapier, and Workbot as Teams Apps to create and manage workflow and execute certain function in MS Teams. According to Table 2.2, all the three systems allow users to create, modify, access, and run their workflows for task automation in MS Teams. To automate routine tasks and for users to easily execute MS Team functions, the proposed systems will also have the functions to create, edit, access, and run the workflow. Furthermore, the table shows that only Power Automate and Zapier can support workflow with if-else and loop conditions and provide templates to users for them to easily create a workflow. Also, no system supports workflow backward and this will be a limitation of current systems. Therefore, the proposed system will implement the feature to allow users to back to the previous step in their workflow chain to increase their flexibility to perform task. Besides, the proposed digital assistant will also provide some templates for commonly used tasks for users to automate routine task while users can also create their own template via the macro features of the proposed solution.

According to Table 2.2, Power Automate in Teams supports most of the actions in MS Teams while Zapier and Workbot supports only four functions in Teams. To solve the first two problem statements in Chapter One, the proposed digital assistant will be able to help users to execute most of the MS Teams functions. For example, to send notifications, send or post messages, create team or channel, create and schedule meeting, create and start calls, create and access files, create and manage shifts and tabs and manage approvals in MS Teams. In result, the steps for users to use every function of MS Teams can be simplified.

Chapter 3

System Methodology/Approach

The project methodology, system requirements, and project timeline will be discussed in this chapter to produce a planning plan to develop this project. Besides, the implementation issues and challenges in the project development are presented in the feasibility analysis of each proposed method to implement the project.

3.1 Project Schedule Planning

3.1.1 Project Methodology

Agile methodology is used to develop this project. It is a project management methodology by developing the product using continuous iteration of short cycle that allows for rapid production and constant revision. This methodology is chosen to be used due to its benefit of it promotes project development in incremental and iterative approach to develop small project quickly. Moreover, to reduce the time used for planning and need less documentation of this project.

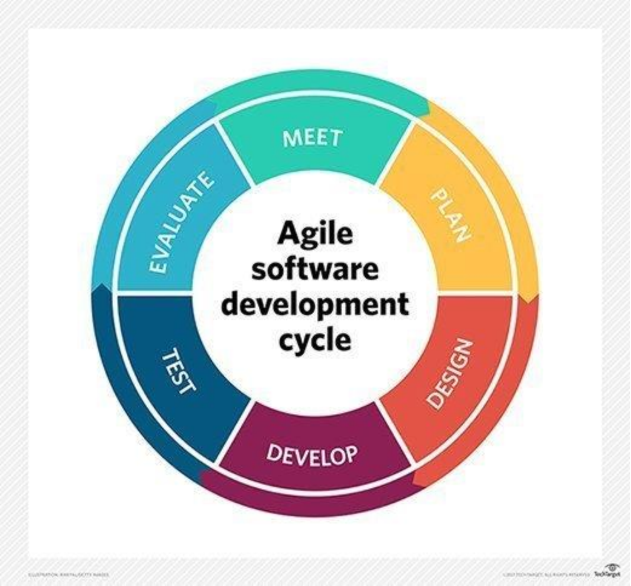


Figure 3.1 Phases of Agile Methodology

In this project, Extreme Programming (XP) technique will be used as a type of agile methodology to allow small team or individual member to produce a high-quality software and adapt to evolving and modifying requirements. It promotes frequently deploy of the product in short development cycles which inherently increases the product's productivity and code quality. There are five phases in the Agile XP methodology used to develop the digital assistant for MS Teams workspace app. The stages will be conducted in iteration to repeat the processes of add new requirements, design, code and test the program developed in the last iteration step by using unit testing and acceptance testing, and to deploy the last improved system, then enter a new iteration until the project developed is fully satisfy its requirements.

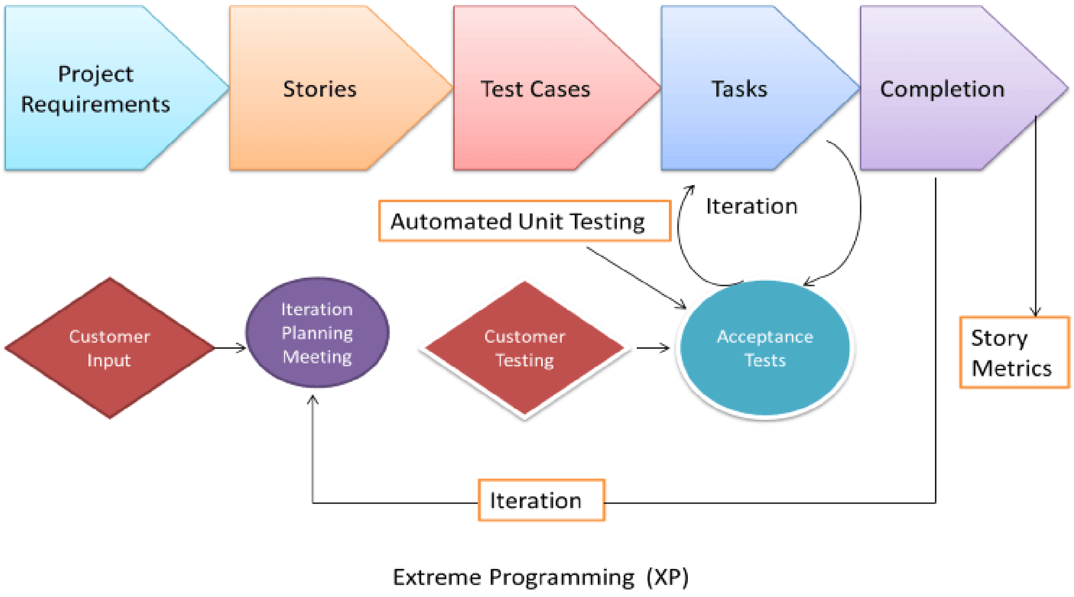


Figure 3.2 Phases of Agile XP Development Cycle

Planning & Analysis

In the first stage, requirements of the project will be collected and analysed. Moreover, to create a plan to break the project development process into iterations to cover the desired functionality part after part. A time schedule will also be created in this stage to predict the desired time for the whole project to be completed and can satisfy the requirements. This phase will be mainly conducted in the October 2021 semester to take the IIPSPW subject. Throughout the semester, topic and title of the project will be decided and to investigate its backgrounds and motivations. Secondly, this phase will conduct the literature reviews on MS Teams which act as main workspace app in this project to analyse its features, pros, and cons to recommend the solutions and innovations can be implemented in this project. Moreover, to review on the previous similar automation tools in MS Teams to collect and analyse their pros and cons to have better planning to the project. Besides, infrastructure requirements will also be collected and ensured in this phase to plan the software and hardware needed to be used to develop the project. According to the data and information collected, the project development processes can prioritize the main functions and sub functions to be included in this project, and to separate the function development process into several iteration stage according to its priority. The project’s functions and features planning will be decided based on the problem statements and objectives decided in this phase during the IIPSPW. The resources planning will also be done in this phase to ensure the type of resources needed in this project. Through the planning and analysis, a work plan in Gantt chart will be created to illustrate and plan the flow and time to develop the full project throughout the three semesters of IIPSPW, FYP 1 and FYP 2.

Design

The design phase will be conducted to bring the logic and structure design to the system will be developed in this project. The importance of this design phase is to build a clear and simplified structure for this project, and to avoid unnecessary complexities and redundancies in the structures designed. This phase can ensure the project built in proper and clear structure to avoid risks may happened in the coding stage if without a proper design on it. Also, a good design can save the developer’s time and effort when proceed to the next coding stage. The design phase will be more focused in FYP 1 and FYP 2 while in IIPSPW, the project framework will be simply discussed to have a foundation idea about the entire system design. In FYP 1, the design specifications and general work procedure will be discussed, and the issues or challenges faced will also be recorded. In FYP 2, the top-down system design diagrams will be created to depict the system’s components and flows in several useful diagrams. For example, architecture diagram, block diagram, deployment diagram and UML diagrams such as Use Case Diagram. The details of which and how specific diagrams will be designed and used will be investigated in this project. Besides, the types of testing and evaluation method performed to test the system will also be planned and evaluated in the FYP 1 to test the digital assistant and evaluate its performance to assist users when performing tasks in MS Teams workspace app. The test scenarios, data, and resources used to test the system will be collected and prepared to build the test plan to test the system in the future stage.

Coding

The coding phase will be implemented in both FYP 1 and FYP 2 in different iteration cycle to develop the actual codes for this project. The coding processes in this project including to develop both ML-required functions and non-ML required functions of the digital assistant. In FYP 1, the major features and functions of the non-ML feature in this project will be developed, while the rest of the functions including ML algorithm and AI model will be developed in FYP 2. In FYP 2 which need to develop the AI model, the AI assistant will be trained via machine leaning and to build the neural networks of it to assist users in a more intelligent way. Besides, a UI should be developed for users can install and launch the digital assistant in their desktop to input commands to the bot for it to perform MS Teams functions and other functions. There are two methods plan to use to integrate the digital assistant with MS Teams. First of all, to integrate the digital assistant itself into MS Teams such as to build a MS Teams app. Second of all, to build a desktop application as the bot UI for it can perform operations by manipulating the elements in the MS Teams desktop application. Both methods are further discussed and elaborated in the next chapter. Furthermore, the UI built in this stage is to allow user interaction with the AI assistant in a user-friendly and intuitive way. Therefore, the assistant needs to be developed the chat function to answer user's query via chat box. To achieve this function, NLP should be implemented in the coding phase for it can understand users' command in natural languages and able to execute certain MS Teams based on commands with similar meaning. In this coding phase, all production code will follow the coding standards and based on the flows and diagrams designed in the previous stage. Specific XP practices will be implemented in this stage to conduct the pair programming and continuous integration. The importance of this stage is to develop the system by using suitable programming languages and tools to match the desired functions, structures, and flows designed in the design phase. In result, codes of the project with desired functions and no mistakes and bugs should be developed after this stage.

Testing

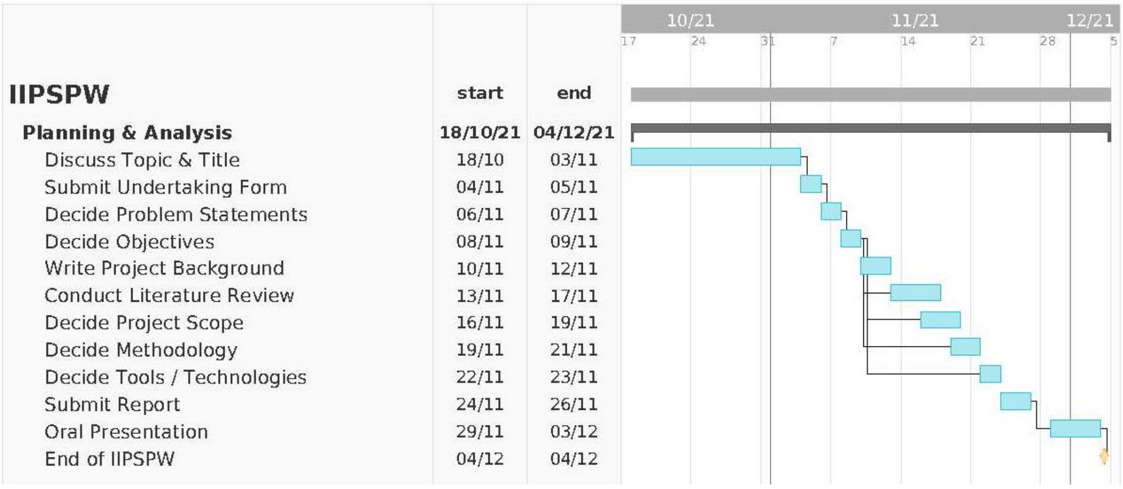
After the coding stage, the testing stage which is the core of XP methodology will also be conducted in both FYP 1 and FYP 2. This stage is important to test the flows and performance of the source codes implemented in the previous stage. Besides, to review the codes implemented in every iteration to ensure the product in this iteration functions well and can meet the iteration goals. Different test scenarios will be applied in this testing phase based on the test plan developed in the design stage. In this phase, the project will be tested by using the unit testing, integration testing and acceptance testing methods. Unit testing to test every function of the project in unit, integration testing to test the whole integrated system, and acceptance testing to ensure the product can meet the requirements of current iteration and had improved from the last iteration product. These tests will be run frequently, and all codes must pass all automated unit tests and acceptance tests before it can be released. Moreover, the system will be tested and evaluated according to the performance metrics designed for evaluate the system's performance. In FYP 1 and FYP 2, all functions of the AI assistant developed will be tested to execute users' command under different scenario and to test can it execute every MS Teams function correctly and achieve the objectives. Every bug found in these testing methods should be fixed to ensure the product outputted is error-free.

Wrapping

The last stage in XP methodology is to release the product to allow demos and reviews of users. The opinions and reviews of users will be collected to insert new requirements into the next iteration cycle, and to process improvements based on the comments on end of this iteration. This phase will be conducted in both FYP 1 and FYP 2 to release the product of the first version with foundation features in FYP 1, and the second version released in FYP 2 after improvements in its functions based on the new requirements added or commented in FYP 1. After the final deployment step of this project, the developed digital assistant should be able to achieve all objectives and can be integrated and used in real case scenarios in MS Teams by the workers who are using MS Teams platform as their workspace to increase productivity. The publish way of the digital assistant will be either published to Teams store as an app or published as a Windows App to be installed in user desktop depends on which way is more suitable in the process to develop the project. Two publish way are planned to reduce the risk to publish the app successfully. Finally, the project should end in FYP 2 after this final stage and after collect feedbacks and opinions from the supervisor, moderator, and other target users.

3.1.2 Project Timeline

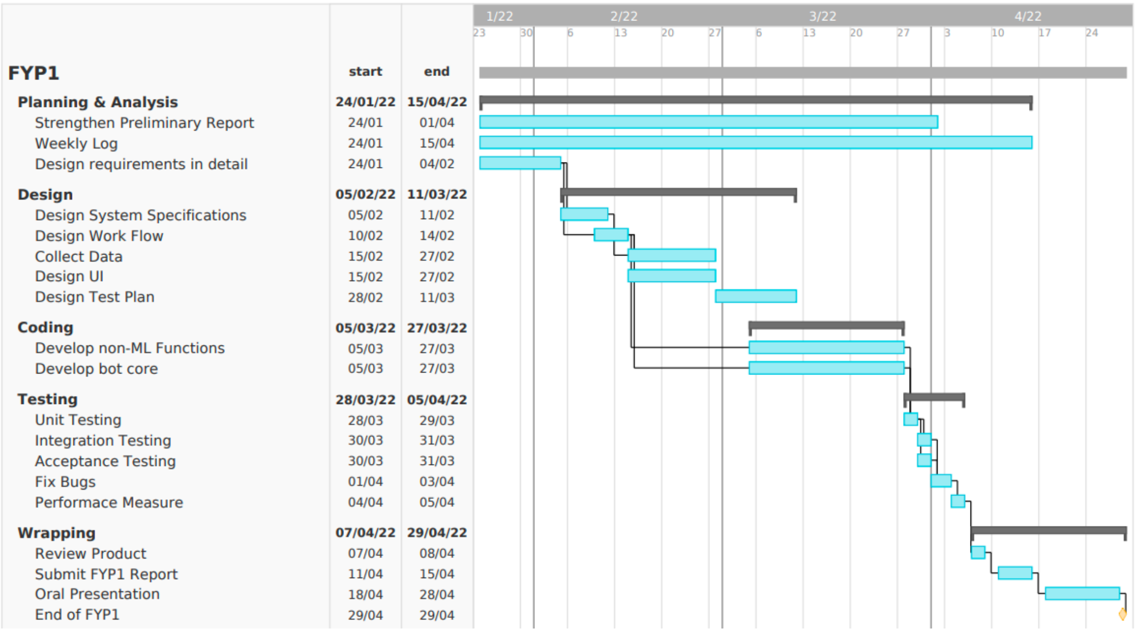
IIPSPW



IIPSPW	start	end
Planning & Analysis	**18/10/21**	**04/12/21**
Discuss Topic & Title	18/10	03/11
Submit Undertaking Form	04/11	05/11
Decide Problem Statements	06/11	07/11
Decide Objectives	08/11	09/11
Write Project Background	10/11	12/11
Conduct Literature Review	13/11	17/11
Decide Project Scope	16/11	19/11
Decide Methodology	19/11	21/11
Decide Tools / Technologies	22/11	23/11
Submit Report	24/11	26/11
Oral Presentation	29/11	03/12
End of IIPSPW	04/12	04/12

Figure 3.3 Gantt Chart of IIPSPW

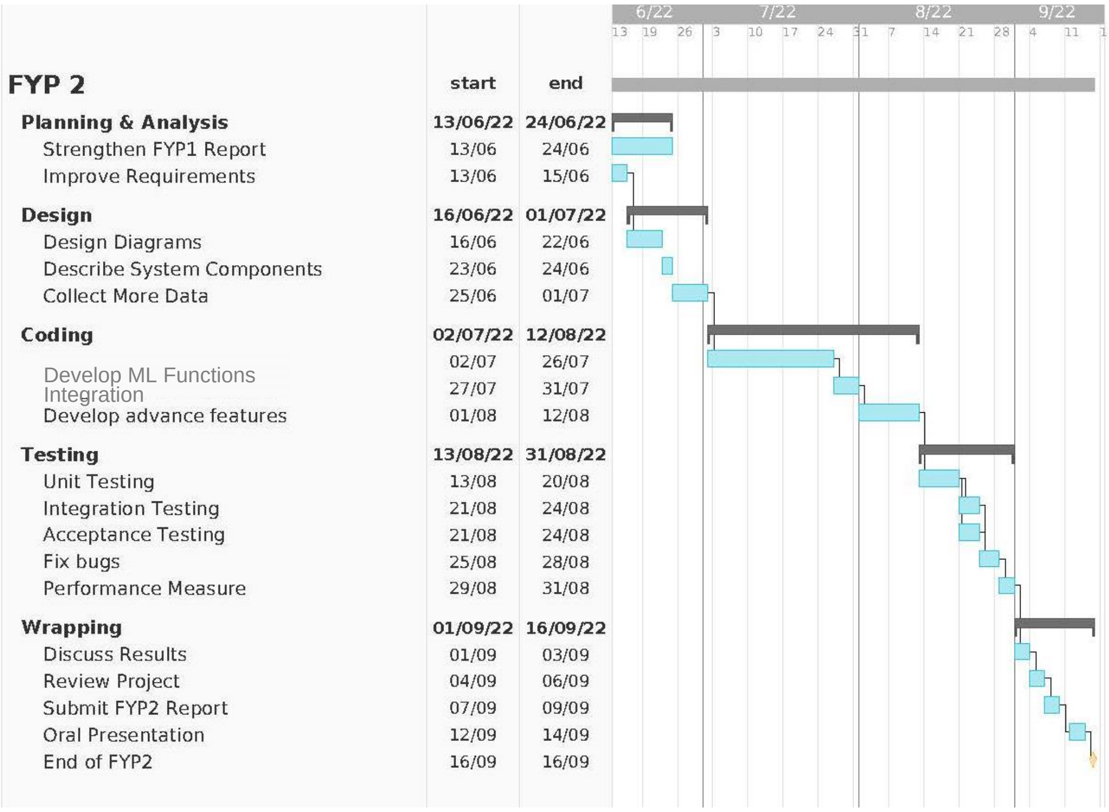
FYP 1



FYP1	start	end
Planning & Analysis	**24/01/22**	**15/04/22**
Strengthen Preliminary Report	24/01	01/04
Weekly Log	24/01	15/04
Design requirements in detail	24/01	04/02
Design	**05/02/22**	**11/03/22**
Design System Specifications	05/02	11/02
Design Work Flow	10/02	14/02
Collect Data	15/02	27/02
Design UI	15/02	27/02
Design Test Plan	28/02	11/03
Coding	**05/03/22**	**27/03/22**
Develop non-ML Functions	05/03	27/03
Develop bot core	05/03	27/03
Testing	**28/03/22**	**05/04/22**
Unit Testing	28/03	29/03
Integration Testing	30/03	31/03
Acceptance Testing	30/03	31/03
Fix Bugs	01/04	03/04
Performace Measure	04/04	05/04
Wrapping	**07/04/22**	**29/04/22**
Review Product	07/04	08/04
Submit FYP1 Report	11/04	15/04
Oral Presentation	18/04	28/04
End of FYP1	29/04	29/04

Figure 3.4 Gantt Chart of FYP 1

FYP 2



FYP 2	start	end
Planning & Analysis	**13/06/22**	**24/06/22**
Strengthen FYP1 Report	13/06	24/06
Improve Requirements	13/06	15/06
Design	**16/06/22**	**01/07/22**
Design Diagrams	16/06	22/06
Describe System Components	23/06	24/06
Collect More Data	25/06	01/07
Coding	**02/07/22**	**12/08/22**
Develop ML Functions	02/07	26/07
Integration	27/07	31/07
Develop advance features	01/08	12/08
Testing	**13/08/22**	**31/08/22**
Unit Testing	13/08	20/08
Integration Testing	21/08	24/08
Acceptance Testing	21/08	24/08
Fix bugs	25/08	28/08
Performance Measure	29/08	31/08
Wrapping	**01/09/22**	**16/09/22**
Discuss Results	01/09	03/09
Review Project	04/09	06/09
Submit FYP2 Report	07/09	09/09
Oral Presentation	12/09	14/09
End of FYP2	16/09	16/09

Figure 3.5 Gantt Chart of FYP 2

3.2 System Requirement

3.2.1 Hardware

The hardware involved in this project is a personal computer used to develop the AI assistant in MS Teams workspace app. The computer should achieve the basic requirements on minimum hardware specifications to run MS Teams and to train an AI bot to be integrate with the MS Teams. The computer also used to conduct machine learning and develop NLP function of the AI assistant for it to understand human languages and able to achieve our objectives in smart way. Moreover, a project compatible to Windows environment will be created and the project testing will be performed under the Windows environment.

Table 3.1 Specifications of Laptop

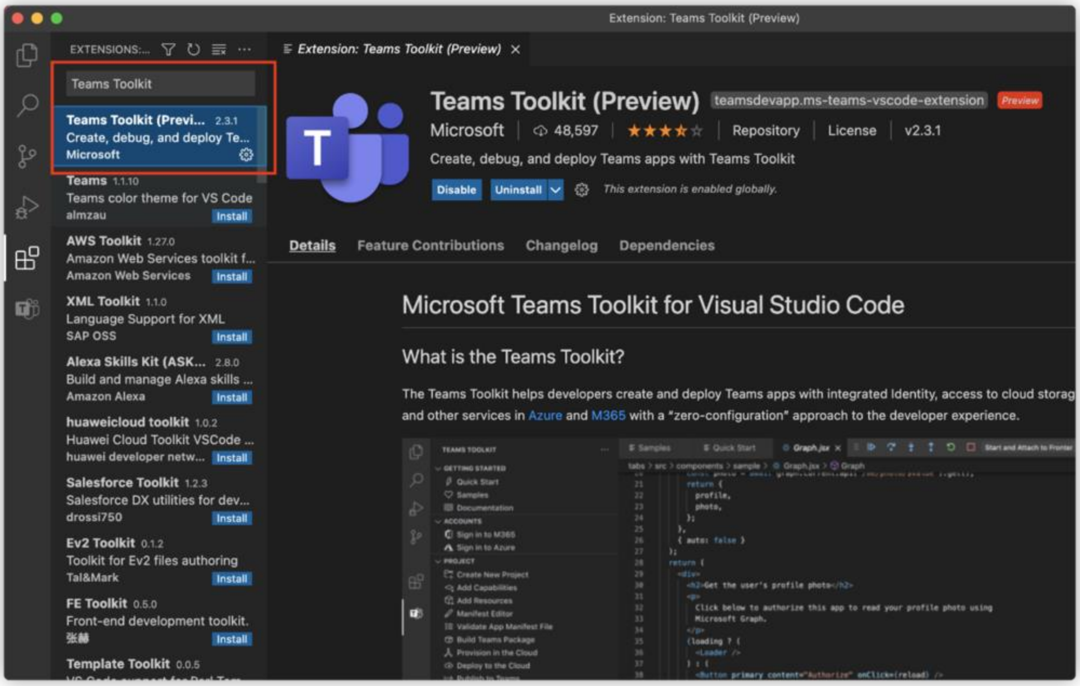
Description	Specifications
Model	Level 51 Forge-15S 1650 Edition
Processor	Intel Core i5-10300H
Operating System	Windows 10 Home 64-bit
Graphic	NVIDIA GEFORCE GTX 1650 4GB GDDR6
Memory	16GB (8GB X 2) DDR4 3200 MHz RAM
Storage	500GB LEXAR NM610 SSD
Internet	INTEL W-iFi 6 AX201 (AX/AC3000 + BLUETOOTH 5.0)

3.2.2 Tools Involved

In this section, the tools and software planned to be used to develop the digital assistant are presented. According to the coding phase of Chapter 3.1.1 for project methodology, two different plans are designed to develop the project which are first, to build a MS Teams app for integrating the digital assistant, and second, to build a desktop application or tool for the digital assistant can perform Teams functions outside the MS Teams. The consideration is to prevent the unavailability of either method due to safety or privacy issues of each solution. Therefore, software and tools used to develop both methods are investigated and presented.

3.2.2.1 IDE

Visual Studio Code with Teams Toolkit Extension



Create, debug, and deploy Teams apps with Teams Toolkit

This extension is enabled globally.

Microsoft Teams Toolkit for Visual Studio Code

What is the Teams Toolkit?

The Teams Toolkit helps developers create and deploy Teams apps with integrated Identity, access to cloud storag and other services in Azure and M365 with a "zero-configuration" approach to the developer experience.

Figure 3.6 To Install Teams Toolkit Extension in VS Code

Based on the first solution, VS Code is chosen as the IDE to be used to develop the digital assistant in this project. For a brief introduction, VS Code is a source-code editor developed by Microsoft to support various OS such as Windows, Linux, and Mac OS. VS Code functions as a powerful IDE for users to build different type of application or write source code in different programming languages such as C#, C++, Python, and others. It includes many useful functions of support debugging, syntax highlighting, smart code completion, snippets, code refracturing and embedded Gits which is suitable to be used in this project for a quicker and more comfortable project development process and environment. Moreover, the Teams Toolkit extension in VS Code can help users to develop and deploy a Teams application with better developer experience. This is due to the reason that this extension supports most of the services in Azure and Microsoft 365 for users to integrate those services in their app developed and allow their app to access to cloud storage and data from Microsoft Graph to simplify the process to develop an app in MS Teams. The Teams Toolkit extension can be easily installed in VS Code by search it in the Extensions and install it.

In this project, VS Code with the Teams Toolkit extension will be used to build, configure, and publish the app to be integrated with MS Teams. The application developed will be run in VS Code and debugged with VS Code debugger. To test the application, the project will be run in debug mode in VS Code and the built application will be run in the browser window automatically for testing the app run locally. For example, the built application in VS Code will be run to open MS Teams in web app form to run the app. To implement the UI of the web application, VS Code will also be used to develop the front-end web application with respective HTML, CSS, and JS files to be developed and its live server plugin allows user to easily monitor the web changes and efficient deployment of the web app.

In conclusion, this IDE is chosen for its popularity as a source-code editor and its high productivity with the intuitive keyboard shortcuts, easy customization, and support multiple programming languages which will be used to develop the same project. Besides, the Teams Toolkit is chosen in this project for its function as a Teams app developer tool to build and deploy Teams app quicker and easier.

PyCharm

Figure 3.7 PyCharm

One of the IDEs designed by JetBrains is chosen to be used in developing this project due to the complete solution and user-friendly UI provided. In our solution, PyCharm from JetBrains is selected to be the IDE used to develop the Python based solution for implementing and integrating the AI feature. PyCharm is an IDE designed for professional developers to develop a Python project including all Python tools in one place. This IDE is chosen to be used in this project to develop and train the digital assistant model by using machine learning in Python language. It provides benefits of many productive features such as intelligent code completion, on-the-fly error checking and quick debugs, and simple project navigation. For example, the benefits of this IDE including the complete and useful live code inspections up to 2,200 inspections to give a good performance on code error analysis and quick-fixed automation to save the time to develop the Python program. Besides, the code quality can also be boost by using this IDE to develop the project.

Moreover, the many ML related libraries available to be installed in PyCharm increase user's conveniency and efficiency to develop the ML based solution. For example, PyCharm includes a useful feature to simply install PyTorch in PyCharm. PyTorch is an open-source machine learning library for Python based on the Torch library. Mainly, it can be used to develop deep learning applications such as NLP. In this project, PyTorch is considered to be installed to PyCharm to develop and train the digital assistant by using machine learning to build the AI model for it to predict the most possible user's next steps or specific command based on user's inputs. Besides from training the AI model, this IDE is also planned to be used to develop the core and logic of this digital assistant due to Python language will be the main programming language chosen to develop this project.

Unity

Figure 3.8 Unity

Another IDE chosen to be used in the future development of the project is Unity designed by Unity Technologies as a cross-platform game engine used to develop product of 2D, 3D, or other interactive simulations on desktop, mobile, console and virtual reality platforms. The main programming language used to develop a Unity project is C# and, in this project, Unity is considered to be used to develop the UI of the digital assistant which is more interactive and attractive compared to normal web app UI for future improvement. The example of advantage of Unity consists of it provide good performance in produce animation and 3D experience product with unprecedented artistic freedom and quicker implementation. By using this software, the digital assistant is planned to develop a 2D, 3D, or even VR based UI as a desktop application and users will be allowed to interact with the assistant's UI in an easier, user-friendly, and more attractive way. Moreover, the software will be used to design and develop the animation that can be performed by the digital assistant UI correspond to different user's input or command, to provide more immersive and attractive user experiences. In result, this tool is chosen to be used to develop the assistant UI in more interesting way to increase the ways of user's interaction with the bot for increasing its accessibility, performance, and productivity.

3.2.2.2 Programming Language

There are few programming languages and technologies plan to be used to develop the digital assistant based on their own strengths on developing different part of this project.

Python

Python language can be used to develop the core logic of the AI-based digital assistant in this project. It is a popular interpreted high-level programming language to enhance productivity of applications compared to other programming language such as Java. Python is considered to be used in this project to train the AI model for it can understand users' command and provide suitable responses and suggestions. This is because Python is the most popular programming language used for AI development and machine learning. The benefit of Python includes its simplicity, readability and flexibility which make it become a more dynamic programming language and easier to develop complex algorithms needed for an AI. Besides from the AI feature development, Python will also be used as the main programming language to implement the assistant's logic and flow including the back-end code to handle user's requests, information processing, and provide respective outputs back to the front-end environment. The Flask framework also considered to be used for building a web application and handling the back-end Python code processing for this project.

Node.js & JavaScript

Node.js is an open-source and cross platform JavaScript runtime runs on Chrome's V8 JavaScript engine to execute JS code outside a web browser. In this project, JS can be used to develop the bot as the MS Teams app for it to integrate to the MS Teams in the solution one. For the first solution, the app built in this project will use JS to build the tab portion of the app and the rest of the app features will be developed based on the Node.js. The benefits of Node.js and JS to be used in web applications development include high performance, quick and easy development, and others. Moreover, in the second solution to implement the project via RPA process, JS will also be used to develop the script of the web application and be used as the main language to develop the solution by using Google Apps Script as JS is the main programming language to be interpreted in web environment. The ajax requests will also be sent to the back-end environment in the script written in JS language. In result, to be able to run the digital assistant in web environment for the MS Teams web version, JS will be the best programming language used for the project development.

C#

C# is a multi-paradigm and statically typed programming language which has features of generic, object-oriented declarative and others. The benefit of C# language includes cross platform, backward compatibility, object-oriented, and others. In this project, C# is also one of the programming languages that be used to develop the bot as the MS Teams app for it to integrate to the MS Teams for the first solution. Therefore, it will be applied in the coding process in the situation which is not suitable to develop the code by using other programming languages for the solution one. This is due to the reason that the .NET solution is more suitable to be used to manipulate element in Windows application such as the desktop version of MS Teams. Also, C# is the main language used to develop Unity solution for our advanced UI implementation for the optional feature of the project that will be developed in the future.

3.2.2.3 SDKs

.NET Core SDK

.NET Core is a new open-source framework to support various programming languages such as C#, F# and VB.NET to develop applications. It has high performance, can run with fast speed, and can be deploy to and run-in different platform. The digital assistant in this project will be developed based on the .NET Core SDK to run the assistant bot developed using C# language.

TeamsFx SDK & MS Teams JS Client SDK

Based on the first solution, the TeamsFx SDK is the MS Teams framework with common functionalities and integration patterns to develop applications in MS Teams. It provides simple authentication for both client and server-side code tailored developers for MS Teams apps. Moreover, the MS Teams JS Client SDK, or teams-js SDK will also be used in this project to develop the Tabs and Task modules for the MS

Teams app. This SDK provides common SDK functions in MS Teams such as to get current state in the running page, get most recently used tabs by users and others. It can also help to create hosted experienced in MS Teams and to display the app content in an iframe for the first solution.

Microsoft Bot Framework SDK

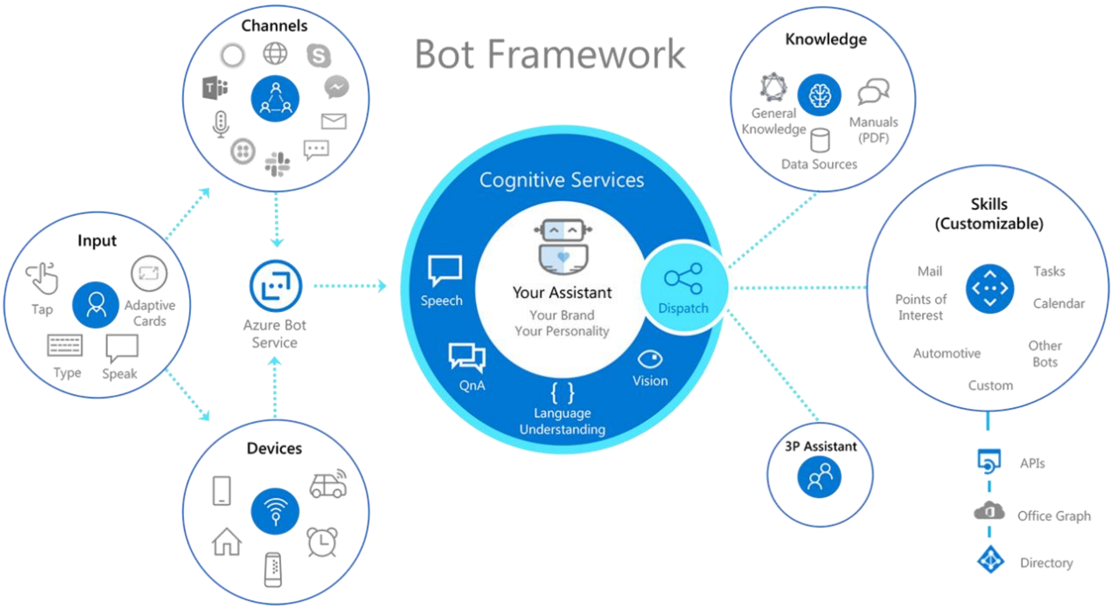
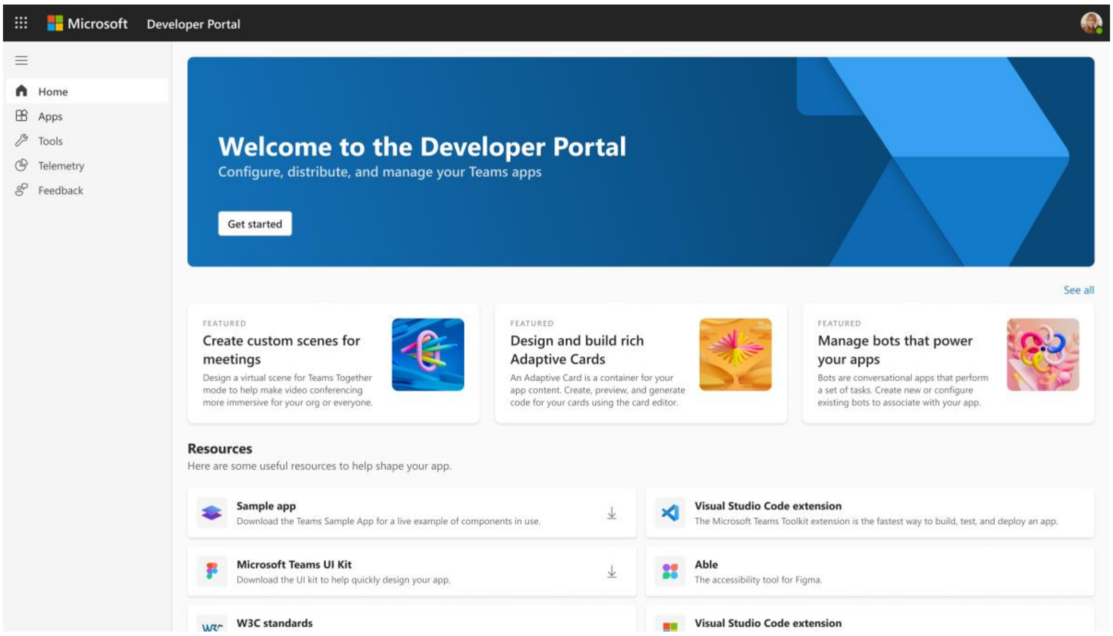


Figure 3.9 Microsoft Bot Framework

To implement a Teams app as planned in the first solution, Microsoft Bot Framework is a comprehensive open-source framework developed by Microsoft to build enterprise-grade conversational AI experiences. This framework is used to build an AI bot in this project to assist MS Teams users in smart way. The Bot Framework SDK supports various programming languages such as C#, JS, Python and Java to design and build conversational experiences with NLP and sophisticated composition of bot replies. The main objective of this tool is to connect the bot developed to client experiences. Besides, the Bot Framework Composer will also be used in this project which is an integrated development tool to develop bots based on the Microsoft Bot Framework. The benefits of this framework to be used include the bot built with ability to speak, listen, and know users' language using Azure Cognitive Services and others. Therefore, this framework will be integrated with the Azure Bot Service to build the digital assistant with ownership and safe data control.

3.2.2.4 Other Software

Developer Portal



Welcome to the Developer Portal

Configure, distribute, and manage your Teams apps

FEATURED

Create custom scenes for meetings

Design a virtual scene for Teams Together mode to help make video conferencing more immersive for your org or everyone.

FEATURED

Design and build rich Adaptive Cards

An Adaptive Card is a container for your app content. Create, preview, and generate code for your cards using the card editor.

FEATURED

Manage bots that power your apps

Bots are conversational apps that perform a set of tasks. Create new or configure existing bots to associate with your app.

Resources

Here are some useful resources to help shape your app.

Sample app

Download the Teams Sample App for a live example of components in use.

Visual Studio Code extension

The Microsoft Teams Toolkit extension is the fastest way to build, test, and deploy an app.

Microsoft Teams UI Kit

Download the UI kit to help quickly design your app.

Able

The accessibility tool for Figma.

W3C standards

Visual Studio Code extension

Figure 3.10 Developer Portal

Developer Portal is the main tool used by developers to configure, distribute, and manage their MS Teams apps. Furthermore, it includes a function of analyse the app’s usage such as daily or monthly active users. This portal is the newest developer tool designed by Microsoft for Teams apps development which can achieve same functions as the previous developer tool of App Studio. In this project, Developer Portal will be used to configure and manage the MS Teams app built and to set up its runtime environments. Moreover, this tool can also be used to test the app developed directly in Teams via the Preview in Teams button in the portal. The reasons of this tool choose to be used include the easier configuration and management of MS Teams app developed and it had provided some useful tools and resources for developers to design and build their MS Teams app. For example, the useful tools and resources such as Adaptive Cards Editor to create and preview own Adaptive Cards to include with our own created app, Microsoft Teams UI Kit to design the MS Teams app by using existing template and others.

Microsoft Teams

Figure 3.11 Microsoft Teams

For the first solution, MS Teams in desktop version will be used in this project to test, preview, and run the digital assistant developed to test its functionalities and its integration with the MS Teams. Moreover, the final product developed will be deployed to the MS Teams store. For the second solution, MS Teams in both web and desktop version will be used to test the digital assistant developed as a Windows application for it to perform every MS Teams function correctly and can provide ideal result without using any MS Teams related SDK but with the RPA software technology.

Figma

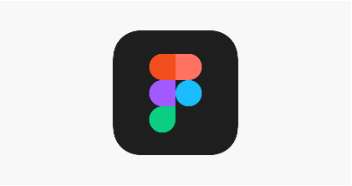


Figure 3.12 Figma

Figma is a design tool developed for vector graphics editing and is a famous software at the moment used for UI designing. In this project, the desktop application of Figma is used to design the UI of the digital assistant due to its advantages of simplicity, free, faster development, and useful plugins for UI design. The prototype of the project will be developed by using this software before proceeding to the real implementation.

UiPath

Figure 3.13 UiPath

UiPath is a solution designed for RPA to automate tasks by automation bot. Through the process of RPA, users can save their time and effort to manually perform a sequence of tasks by designing a workflow and automate the process by simple one click.

According to the first objective to simplify user's steps to perform MS Teams function, each Teams function is planned to develop a corresponding workflow for the automation of every Teams function by using UiPath. Moreover, other workflows besides from MS Teams function will also be developed by using UiPath and the workflow execution will be triggered by specific command inputted by users. In this project, UiPath is chosen for the RPA development because of it is in low-code environment, ensure compliance, provide user-friendly experiences and it supports a lot of RPA features to perform operation or manipulation on Windows application. Therefore, the processes inside MS Teams as a desktop application can be easily automated by using the workflow created in UiPath.

Dialogflow

Figure 3.14 Dialogflow

Besides from the Microsoft Bot Framework solution, Google also provides a bot solution of Dialogflow with built-in NLP feature to design and implement a conversational UI into bots and other platforms. In this project, Dialogflow is planned to be used to train and develop the conversational digital assistant for it can understand natural language inputted by user and provide suitable response based on the algorithm trained. The benefits of this solution are it's low-code and simple to train the conversational bot, supports multiple programming language including C# and Python to be used in this project, support multiple response methods including option card and interactive voice response, and it provides good performance in analysing user's input to decide which intent the input should be classified to and classify parameters inputted.

Google Apps Script

Figure 3.15 Google Apps Script

Google Apps Script is one of the services developed by Google for rapid application development in websites. It allows users to easily create business software in this platform and the software can be easily integrated with G Suite and other Google APIs including Google Calendar, Google Spreadsheet, Gmail, etc. to increase business productivity. This platform is planning to be used in the second solution to develop the code that can integrate with the Google Calendar for achieving part of our project objectives for the AI recommendation and automation process. This platform will be used to implement the script that can automate the scheduling process in Google Calendar to save users' time to manage their schedule related to different workspace apps. Therefore, Google Calendar will also be involved in this project development for schedule automation and integrate some of the project's AI features.

3.3 System Design / Overview

This section will present some diagrams to provide an overview for the system design including the system framework, use case diagram, system architecture diagrams, and system flowchart. Moreover, respective tables are also presented in this section after the use case diagram to list the planned functions of the virtual assistant can perform and automate in MS Teams platform.

3.3.1 System Framework

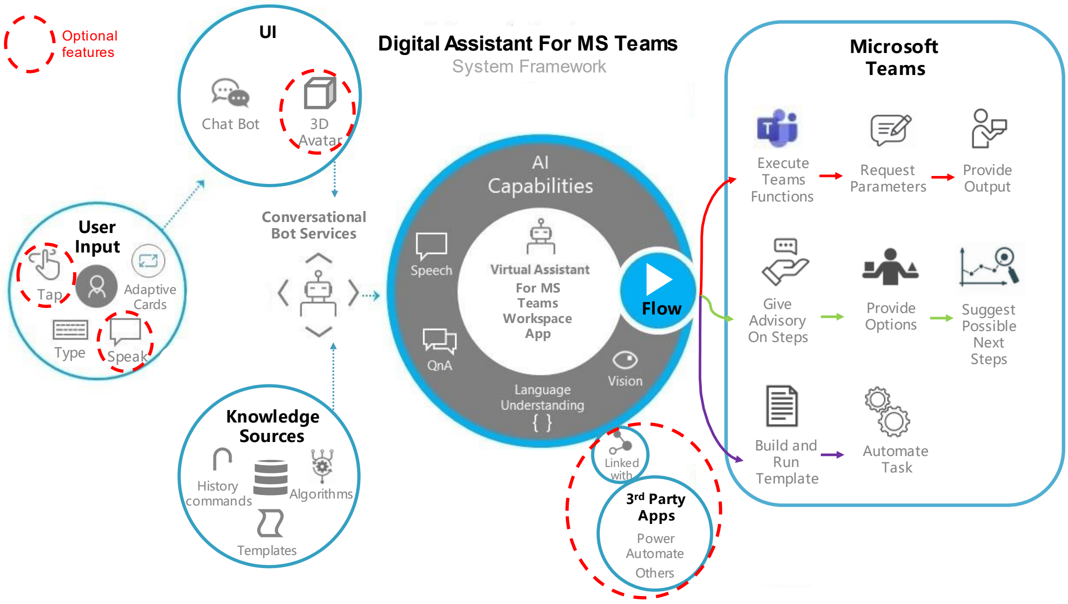


Figure 3.16 System Framework: Digital Assistant for MS Teams

Based on the above system framework, a virtual assistant for MS Teams workspace app will be developed with the AI capabilities of question and answering, language understanding and processing, vision for context understanding. Users can interact with the assistant developed via the main interaction method of typing, and other interaction methods such as tap and speak are optional to be developed in the future. For the UI, the AI assistant will present as a chat bot and allow users to chat with the assistant to obtain certain responses that can help to perform their workspace apps operation. The knowledge sources of the virtual assistant including the history commands of user and algorithms to provide advisory steps to the user, followed by templates that allow user modification for workflow automation. In result, the assistant can perform operations as depicted in MS Teams to assist users for increasing their productivity and efficiency to use MS Teams.

3.3.2 Use Case Diagram

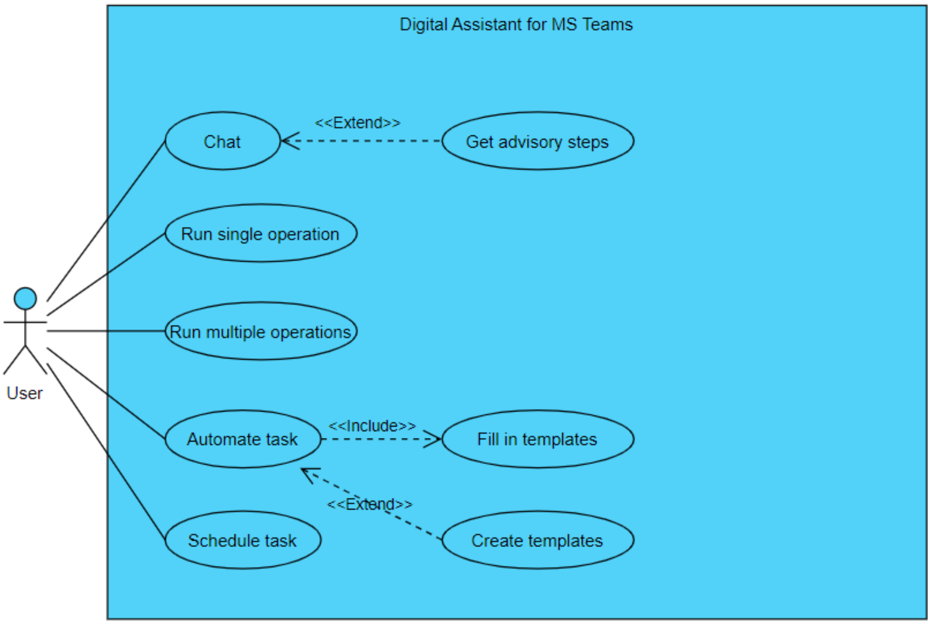


Figure 3.17 Use Case Diagram

Based on the use case diagram designed, a user can perform use cases to chat with the digital assistant, run single or multiple operations in MS Teams via the virtual assistant, automate task by fill in respective template of spreadsheet provided, or choose to create user's own template, and schedule task. In the first use case, user can interact with the digital assistant via typing to chat with the bot and achieve its responses including questions and answers. According to the user's query or command, the assistant with AI capability may give advisory steps to the user for performing the best operations that fit user's intention. Second of all, via the digital assistant, user can choose to run single operation in MS Teams such as to create a team or join a team. Besides, user can also choose to perform multiple operations as workflow in a row such as to create a new channel after creating a new team. Furthermore, user can automate repetitive task by fill in necessary parameters into respective spreadsheet provided as template or create new template for other task automation in Teams. Finally, user can schedule their task in Google Calendar in a more efficient way with the AI recommendation and automation feature.

3.3.3 Functions to Be Performed

To further elaborate the operation can be performed by the digital assistant in the second and third use case for the previous use case diagram, this section will list the assistant's functions correspond to every command that can be performed by the bot, which is the exact operation that can be conducted by the digital assistant according to different command. The bot's functions had been briefly discussed in the project scope in Chapter 1 and the summarize section of Chapter 2 while in this chapter, the details of every command that can be performed by the digital assistant will be presented in the table form. The commands can be executed by the digital assistant including abilities to execute MS Teams functions, functions to automate repetitive task and other extra functions.

Functions to be performed by the digital assistant

Table 3.2 Functions to Be Performed by the Digital Assistant: Manage Teams

Manage Teams

No.	Command.	Operations.
1.	Create team	Request user to input team name (required), team privacy to create a new team.
2.	Join team	Request user to input team code (required) to join an existing team.
3.	Manage team (Description)	Request user to input team name (required) before user can perform manage team operations, including 'Hide team', 'Add channel', 'Add member', 'Leave team', 'Edit team', 'Get link to team', and 'Manage tags'
4.	Hide team	Request user to input team name (required) to hide an existing team that the user had joined.

5.	Add channel	Request user to input team name (required), channel name (required) and select channel privacy to add new channel to an existing team.
6.	Add member	Request user to input team name (required), member’s name, or group to add member to an existing channel.
7.	Leave team	Request user to input team name (required) to leave the team.
8.	Edit team	Request user to input team name (required), select new team privacy, or select new team avatar to edit the team.
9.	Get link to team	Request user to input team name (required) to generate a link of specific team.
10.	Manage tags (Description)	Request user to input team name (required) before user can perform manage tags operations, including ‘Create tag’, ‘Edit tag’, and ‘Delete Tag’
11.	Create tag	Request user to input team name (required), tag name (required), description, members to be added to create a new tag.
12.	Edit tag	Request user to input team name (required), new tag name (required), description to edit an existing tag.
13.	Delete tag	Request user to input team name (required), and tag name (required) to delete the tag.
14.	Add tab	Request user to input team name (required) and select a tab to add to the team.
15.	Manage channel	Request user to input team name (required) and select new channel permissions to manage the channel permissions.
16.	Show analytics	Request user to input team name (required) to generate team’s analytics to users.
17.	Arrange team	Request user to select preferable sorting method (required) of the teams.
18.	List all members	Request user to input team name (required) to list all members in the team

19.	Customize notification	Request user to input team name (required), input or select new notification settings option of the team, and save the new settings.

Table 3.3 Functions to Be Performed by the Digital Assistant: Inside Team Operations

Inside team operations

No.	Command.	Operations.
20.	Meet now	Request user to input team name (required) to create a new meeting.
21.	Schedule meeting	Request user to input meeting title (required), start date (required), start time (required), end date (required), end time (required), channel, attendees to schedule a new meeting.
22.	Team files	Request user to input team name (required) to list all files in the default channel of the team.
23.	Add folder	Request user to input team name (required) and folder name (required) to create a new folder in the team channel.
24.	Add file	Request user to input team name (required), file name (required), and file type (required) to create new document in type of word, excel, ppt, one note, forms, or survey in the team channel.
25.	Upload file	Request user to input team name (required), select a file or folder from devices (required) to upload file or folder to the team channel.
26.	Sync file	Request user to input team name (required) to sync files in team channel with One drive.
27.	Get link to file	Request user to input team name (required), and file name (required) to generate a link to the specific file.
28.	Download all file	Request user to input team name (required) to download all files in the team channel.

29.	Download file	Request user to input team name (required), and file name (required) to download specific file in the team channel.
30.	Add cloud storage	Request user to input team name (required) and select cloud storage provider (required) to add cloud storage to the team channel.
31.	Open share point	Open share point for user.
32.	Sort file	Request user to input team name (required) and select preferable sorting method (required) to sort file in the team channel.

Table 3.4 Functions to Be Performed by the Digital Assistant: Overall Team Operations

Overall team operations

No.	Command.	Operations.
33.	Assignment	Shows all assignments including assignment assigned and assignment completed.
34.	Assigned ass	List all assigned assignment’s summary
35.	Completed ass	List all completed assignment’s summary
36.	Submit assignment	Request user to select an assignment (required) and attach file from One Drive, link, teams, or upload from device (required) to submit work for the assignment.
37.	Add assignment file	Request user to select an assignment (required) and file type (required) to create new file in type of word, excel, ppt, or class notebook page to create new file in the assignment.
38.	Delete assignment file	Request user to select an assignment (required) and file (required) to delete the file.
39.	Turn in	Request user to select an assignment (required) to turn in the assignment

40.	Open assignment file	Request user to select an assignment (required), file (required), and the file open method to open the file in Teams or online.
41.	Download all assignment file	Request user to select an assignment (required) to download all files in the assignment.
42.	Download assignment file	Request user to select an assignment (required) and file (required) to download the file.
43.	Filter assignment	Request user to select an assignment (required) to filter out assignments according to team's channel.
44.	Calendar	Open calendar page.
45.	Add event	Allow user to schedule new meeting, webinar or live event by inputting multiple required parameters. (Too many to list, prefer users to input parameters via template provided or dialog box).
46.	Call [name]	Request user to input the member's name (required) to call the member via phone.
47.	Chat [name]	Request user to input the member's name (required) to chat the member.
48.	Email [name]	Request user to input the member's name (required) to email the member.
49.	Video call [name]	Request user to input the member's name (required) to video call the member.
50.	Contact	List all contacts of the user.
51.	Add contact	Request user to input the member's name or number (required) to add new contact.
52.	Delete contact	Request user to input the member's name (required) to delete the contact.
53.	Add [name] to speed dial	Request user to input the member's name (required) to add the member to speed dial.
54.	Files	Open files page.
55.	Add folder	Request user to input folder name (required) to create a folder under One Drive.

56.	Add file	Request user to input file name (required) and file type (required) to create file in type of word, excel, ppt, one note, forms, or survey under One Drive.
57.	Upload file	Request user to select file / folder from devices (required) to upload the file or folder.
58.	Manage file	Same operations as the file operations inside team channel, but the files to be managed are all files from all teams. The operations including 'Sync file', 'Get link to file', 'Download all file', 'Download file', 'Add cloud storage', and 'Sort file'.
59.	Open one drive	Open all files in One Drive.
60.	Create shift	Allow user to create new shift by inputting and selecting multiple required parameters. (Too many to list, prefer users to input parameters via template provided or dialog box).
61.	Add group	Request user to select shift (required) to create new group in the shift.
62.	Delete group [group name]	Request user to select shift (required) and group (required) to delete the group.
63.	Add shift people	Request user to select shift (required) and input name / email (required) to add member into the shift.
64.	Add shift	Allow user to add new shift by inputting and selecting multiple required parameters. (Too many to list, prefer users to input parameters via template provided or dialog box).
65.	Add activity	Allow user to add new activity by inputting and selecting multiple required parameters. (Too many to list, prefer users to input parameters via template provided or dialog box).
66.	Search shift	Request user to select shift (required) to search the shift.
67.	Add request	Allow user to add new request by inputting and selecting multiple required parameters. (Too many to list, prefer

		users to input parameters via template provided or dialog box).
68.	Add note	Request user to input note for specific day.
69.	Export shift	Request user to select shift (required) to export the shift.
70.	Print shift	Request user to select shift (required) to print the shift.

Table 3.5 Functions to Be Performed by the Digital Assistant: Others

Others

No.	Command.	Operations.
71.	[keywords]	Open default browser (Chrome) and search keywords through Google search engine.
72.	Create task list / Create workflows	Add new task (workflow / sequence) to a workflow list to automate them in sequence.
73.	Remind me	Remind user of a specific task in a specific datetime.
74.	Send msg / email	Request user to input name / email, msg / email contents Or request user to input group of name / email, msg / email contents to send message or email.

3.3.4 System Architecture Diagram(s)

Two system architecture diagrams are designed to illustrate the architecture patterns for both of the solutions as planned.

Solution 1

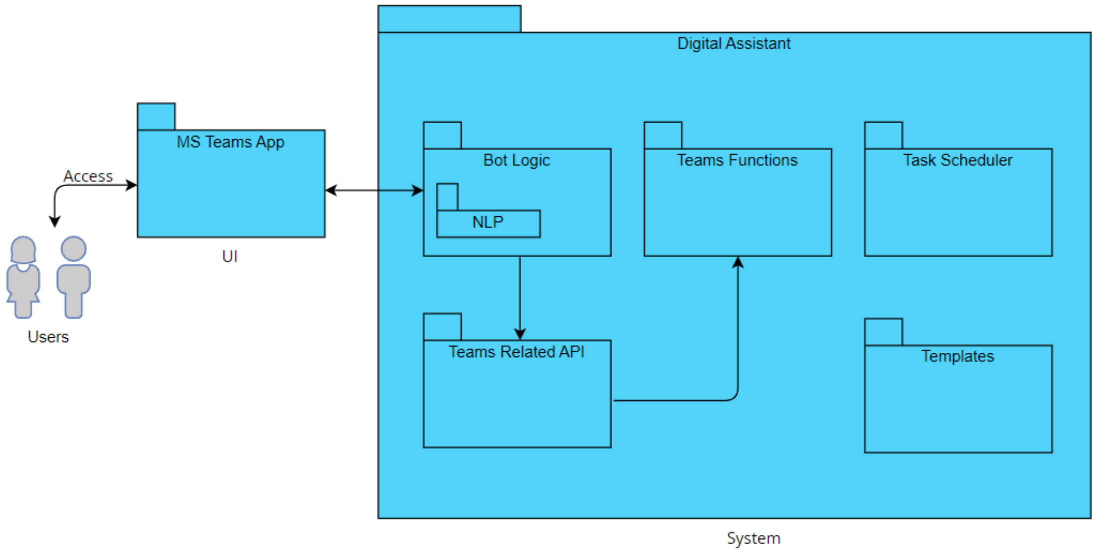


Figure 3.18 System Architecture Diagram for Solution One

According to the system architecture diagram designed, users can access the digital assistant developed via the GUI of MS Teams by installing it as a MS Teams App. Through the app deployed, users can provide inputs to the digital assistant and receive asynchronous response from it to achieve two-way communication. Inside the digital assistant system developed, it consists of bot logic module with NLP implementation to analyse user input and decide the actions to be performed according to user's intent. After user's intent is analysed and decided by the algorithms developed in the module, it will call functions from the suitable Teams related APIs to perform MS Teams function. Thus, users can perform Teams function via the conversational bot deployed as the Teams app and receive the result of the function performed without the need to directly manipulate the MS Teams application. Also, the scheduler module and template module should be included in the System designed for it to automate repetitive tasks by scheduling task and provide useful templates to users. For elaboration, the task scheduler module should be accessed by the digital assistant when it detected user's intent to schedule a task repetitively or periodically. Through the scheduler module, users should be able to schedule their tasks to be auto executed under specific time constraint without manually perform the function repetitively. In result, to avoid time waste due to repetitive process for users to execute similar tasks. Secondly, the templates module should be accessed by the system when it detected user's intent to perform a repetitive process which can be automated by using suitable templates provided. For example, a template or dialog box can be responded by the bot to the user for the user to input or select options in the template to perform specific task that includes repetitive parameters, such as to create channels with regular naming method, and create weekly meetings with regular naming methods and platform used. For its novelty, users should be able to create and modify their own templates for custom task automation. The feature of templates will be further presented in the next chapter of preliminary work to create and implement some use cases by using suitable templates developed.

Solution 2

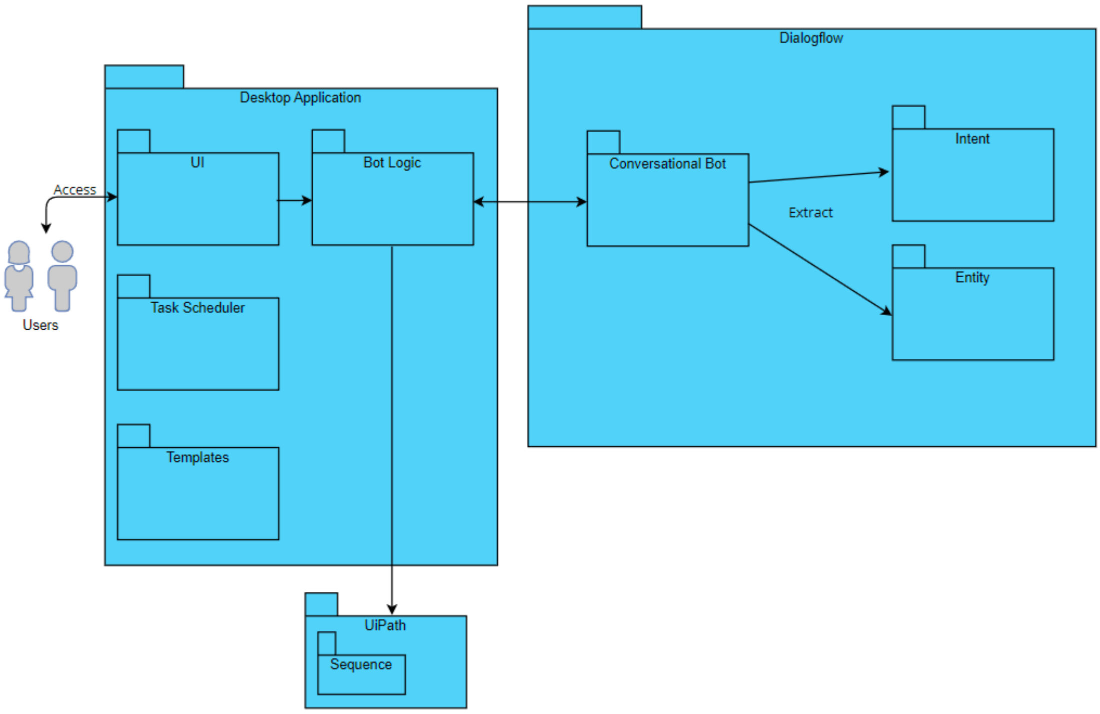


Figure 3.19 System Architecture Diagram for Solution Two

Based on the diagram designed, users can input their commands to the digital assistant via the UI designed in the desktop application. After suitable input pre-processing, user input will be sent to the conversational bot developed in Dialogflow for analyzation and classification into specific intent. Moreover, to extract parameters including in user's query correspond to specific entity inside Dialogflow and provide suitable bot responses as trained for every intent. For example, if user inputs a sentence of 'Schedule meeting after 5 minutes' and after the input text sent to the Dialogflow, the bot trained by using Dialogflow will be able to classify this user input into the intent of 'Schedule Meeting', and extract the parameter of 'after 5 minutes' where the parameter value will be the datetime after 5 minutes from when user send the query, and the parameter will belong to one of the built-in entity in Dialogflow named date-time. Therefore, by using this information extracted, the digital assistant developed can determine how to process the parameters extracted and provide suitable response to users based on its logic and algorithms developed.

For the process flow, the conversational bot built in Dialogflow will extract necessary information such as intent and entities from user's query text and the digital assistant developed can obtain the information by using Google API. After receiving the information, the digital assistant can perform different MS Teams function correspond to specific intent analysed via UiPath. Different from performing Teams function via the APIs provided by MS Teams, in this solution, UiPath as an RPA tool will be used to access and perform Teams operation externally without calling any APIs from Microsoft. Based on the previous section to plan tools and software to be applied in this project, UiPath is a tool for RPA process by allowing user to create and execute their own sequence for workflow automation. Therefore, in this project, UiPath will be used to create multiple sequences where each sequence corresponds to single specific command that can be executed by the digital assistant including commands to perform Teams function. For example, a sequence in UiPath will be created to execute the user's command to create a new team in MS Teams where the sequence will require the least parameter of team name. Therefore, the team's name parameter can be extracted from user's query text by the Dialogflow and returned to the system for it to pass the parameter value to the variable created in the sequence. A prompt message will be returned if the parameter doesn't contain in the user's query text by Dialogflow. After the parameter value of new team's name is obtained and assigned to its variable, the sequence can automate the process such as to launch MS Teams, click specific button or perform specific operation such as inputting necessary information to create a new team for users. The steps of sequences created for every command are predefined and are dynamic based on user inputs. Therefore, by creating different sequence to perform different operation in MS Teams, the bot function to execute Teams function can be achieved.

Furthermore, Scheduler and Templates modules are also implemented in the digital assistant to resolve the problem of repetitive task problem. The functions and flows of these two modules as same as described in the system design for solution one. In this solution based on Google solution, Google Apps Script will be used to integrate the system with Google Calendar for users can perform task scheduling in Google Calendar via automation process. Respective codes will be developed in the Google Apps Script platform to achieve the functions of first of all, automate the task or event scheduling process in Google Calendar via spreadsheet. Therefore, to increase user's efficiency and save their workloads to schedule repetitive task related to the MS Teams workspace app in Google Calendar. Second of all, to provide advice and recommendation on user's Google Calendar schedule via the digital assistant's AI capability. Third of all, to provide an interface or interaction method that can link both Google Calendar and MS Teams for user can easily navigate to different workspace apps such as MS Teams from Google Calendar by referring to their schedule. In result, to achieve our objective of include advisory on user's MS Teams operation and schedule based on related data and simplify user's steps to perform task in MS Teams.

Besides from the bot features that can be developed by using either one of the methods described above, AI capability should also be developed in this project to implement one of the proposed assistant's functions to provide advisory on steps. ML will be needed in achieve this feature to collect suitable data, conduct data analyzation and data pre-processing, develop algorithm for the ML, fine tune and evaluate the algorithm developed, and test the model. There are two objectives involved in AI field to achieve the project goal. First, the digital assistant should be able to find, advice, and suggest the most appropriate commands or functions to be performed corresponds to user's input task description. Second, the bot should be able to predict and recommend the most possible next steps or commands to be performed according to user preferences or the last executed commands. The two different objectives can be combined to one goal for the ML model to achieve, which is to predict the most suitable commands to be performed by users under different condition or use cases based on different data features. For example, the features used to train the model can include user's query text, user preferences, user's command executed history, and others. Furthermore, NLP feature should also be developed in the AI module in this project so that the digital assistant can interact with user in smart way and allow the question-and-answer interaction. The digital assistant should implement the NLP module and be trained to be able to response to different user's context accurately under different situation and environment. The final aim will be the digital assistant to act as a real assistant with human simulation which can provide response in normal talking language or through voice interaction feature. Moreover, user can also interact with the assistant in several ways including typing, tap, speak and others such as body language for advanced system development in future.

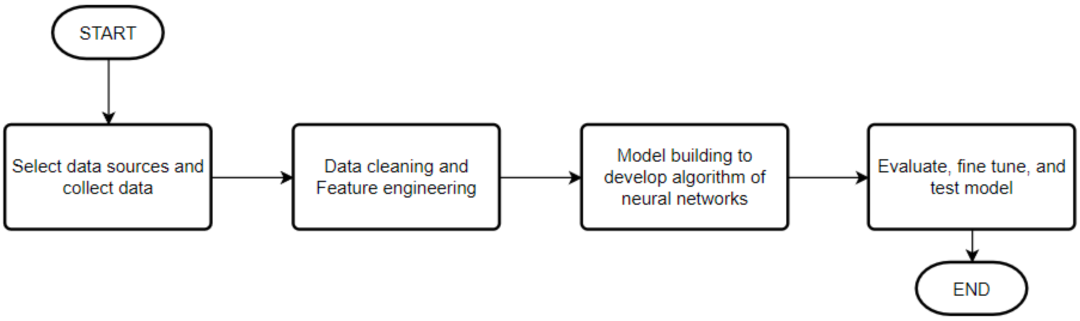

START
Select data sources and collect data
Data cleaning and Feature engineering
Model building to develop algorithm of neural networks
Evaluate, fine tune, and test model
END

Figure 3.20 Flowchart of ML Feature Development of the Digital assistant

A flowchart is designed to indicate the flows to develop the ML feature of the digital assistant in this project. Neural networks are considered to be developed as the algorithm to be trained in the ML model of the digital assistant. This is because the advantages of neural networks of its ability to perform well with insufficient knowledge, to perform self-learning and produce output that is not limited to the provided input, to discover interesting feature during the training process and its ability of parallel processing compared to other ML algorithms. According to the project methodology, the ML feature of this project will be implemented in FYP 2 to prepare and process the data, train, evaluate and fine tune the model, and to test its performance by using several ML testing methods.

3.4 System Flowcharts

System Flowchart: Non-ML Related Operations

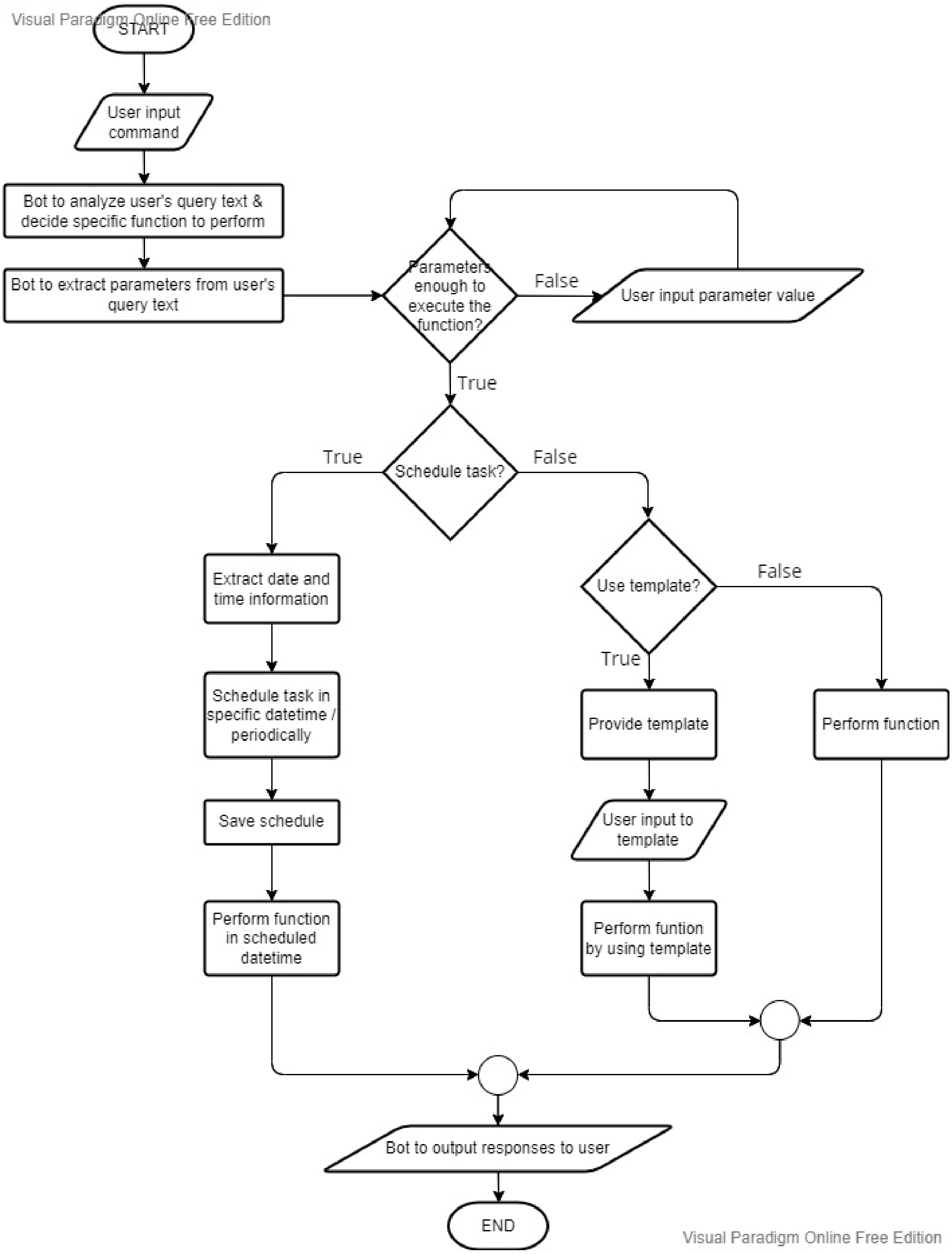

Visual Paradigm Online Free Edition
START
User input command
Bot to analyze user's query text & decide specific function to perform
Bot to extract parameters from user's query text
Parameters enough to execute the function?
False
User input parameter value
True
Schedule task?
True
False
Extract date and time information
Schedule task in specific datetime / periodically
Save schedule
Perform function in scheduled datetime
Use template?
False
True
Provide template
Perform function
User input to template
Perform funtion by using template
Bot to output responses to user
END
Visual Paradigm Online Free Edition

Figure 3.21 System Flowchart: Bot-User Interaction

The above flowchart shows the steps of user to input simple command to the bot and the bot to provide respective response and output back to the user.

System Flowchart: ML-Related Operations

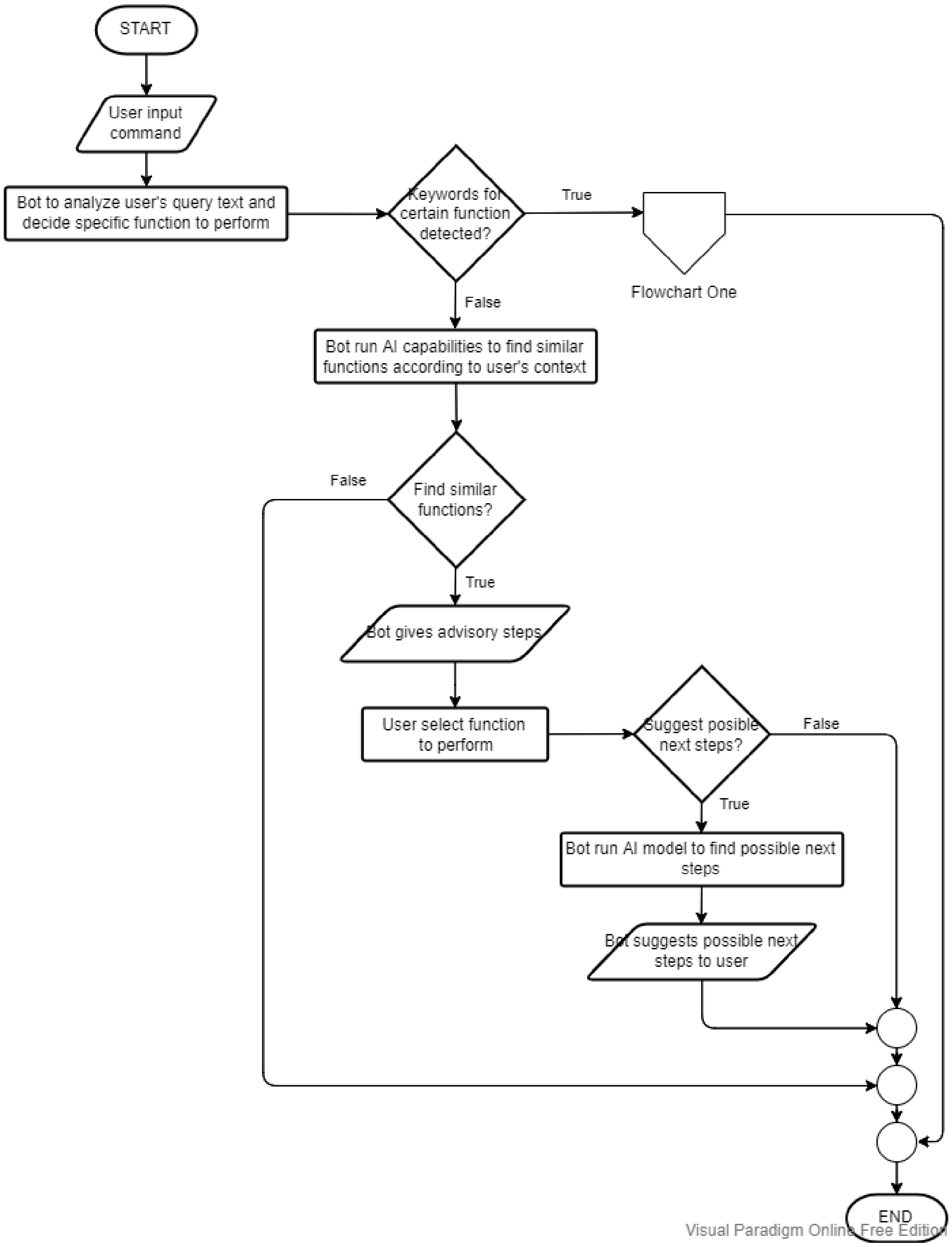

START
User input command
Bot to analyze user's query text and decide specific function to perform
Keywords for certain function detected?
True
Flowchart One
False
Bot run AI capabilities to find similar functions according to user's context
Find similar functions?
False
True
Bot gives advisory steps
User select function to perform
Suggest posible next steps?
False
True
Bot run AI model to find possible next steps
Bot suggests possible next steps to user
END
Visual Paradigm Online Free Edition

Figure 3.22 System Flowchart: ML-Related Operations

The above flowchart shows the ML-Related flows of the bot to provide advisory steps to user and it will perform the function selected by the user according to its recommendation.

3.5 System Components

The system components involved in this project can be mainly separated into the parts of: UI, digital assistant's core logic, and the AI related functions to build the bot that can interact with user, perform ML related and non-ML related functions. In this section, the system components are elaborated to break down the system into different part and module for designing the system development process.

3.5.1 System UI

First of all, a UI for this system is required for users can interact with the digital assistant by different methods of input such as tap, type, speak and via adaptive cards or option cards. The main method of interaction between users and the digital assistant to be developed in this project will be typing and user inputs to an option card or template created by the bot. For the first solution, UI is not necessary to be designed and implemented because the digital assistant will be directly integrated to MS Teams as a Teams app while for the second solution, UI is needed to build the desktop application that allows two-way communication between users and the bot. An attractive and interactive UI is important in this project to provide better user experiences and easier their way to communicate with the digital assistant, so that the bot can extract correct meaning from user's commands and provide accurate output. The UI design is conducted by using Figma to develop few webpages design that can achieve our desired way of interaction between user and the digital assistant. For the user can interact with the digital assistant bot, an input area should exist for user can type in command and send to the bot, while a display box is required for the bot to response to the user based on user's requirement. Two sample UI are designed which the first UI aims to allow users to interact with the digital assistant via chat function, while the second UI indicates the template provided by the bot for different use cases for automating repetitive task. The sample UI only for indicating the basic elements should be included for user-bot interaction and will be further modified during implementation process. In result, the actual implemented UI may be different from the prototype by simplify it or in different colour scheme based on actual implementation process. However, necessary elements will still be included in the actual development.

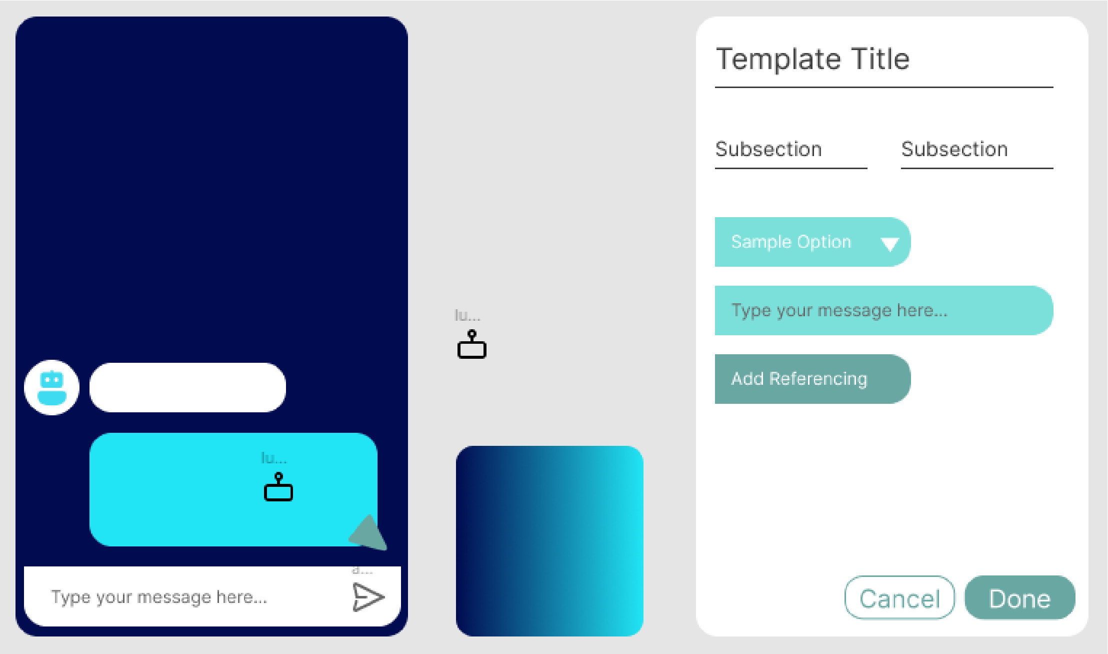


Figure 3.23 Sample Chat and Template Prototype

For the first UI, it allows users to input their commands to the bot by typing while other input methods will be further explored in future improvement to the system. Moreover, the bot is allowed to response to the user in the chatting space before or after executed the specific command. Second of all, another UI to be developed in this project is the built-in template in the digital assistant for users to carry on some specific commands or repetitive tasks. The input methods to the template dialog box should include typing, clicking, and selecting specific element. Besides, the add referencing column in the UI designed allows users to reference to a specific command that can be executed by the digital assistant for certain case.

Furthermore, based on previous sections, Unity is planned to be used in this project to develop more attractive UI as considered as innovative feature of this project after its elementary UI and operations are done. In system design, a 2D or 3D model is planned to be designed and developed to be the project interface for users can have more interaction to the digital assistant such as via tap or voice. In result, they can have more direct interaction with the bot as if they are interacting with a real human assistant. For example, the model developed can occupy a little space or hide as background application in users' desktop environment and responses to users' commands in a more diverse and interactive way such as animation and motion. This feature will be further explored in future project improvement to improve the quality and performance of the project UI.

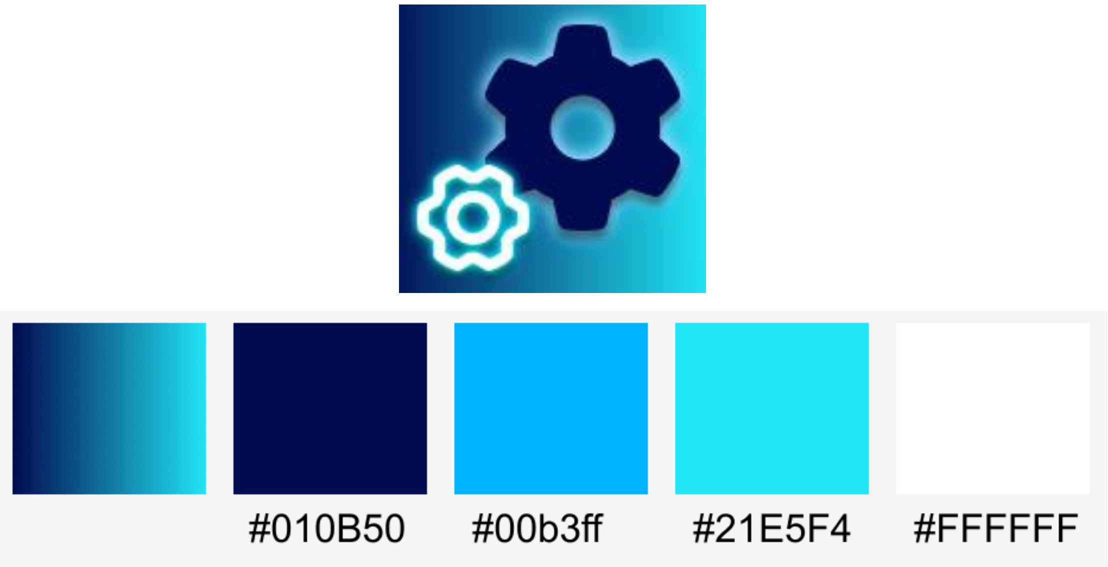


Figure 3.24 Logo and Colour Scheme

Also, a logo was designed for this project and a colour scheme is selected for implementing the UI during the actual implementation process.

Feasibility of the proposed method

UI is only needed in the second solution to build a desktop application for users can interact with the digital assistant. Based on the UI design, the most important component in the UI design consists of the elements needed for user can input their messages and receive responses from the digital assistant. Therefore, the UI can be developed as a Windows application or web application according to its purpose to perform operation on MS Teams desktop application or web application. According to the second solution, RPA process developed by using UiPath also allows the automation steps to be performed on both desktop application and web application of MS Teams according to different RPA sequence implemented. Therefore, a web application is planned to be implemented first for this project due to the simplicity and efficiency to develop a web app including handling web request. Besides, a web app can be easily deployed such as by using Flask framework for the project mainly developed in Python code. The webpages including HTML, CSS, and JS code will be developed by using respective programming languages to implement the web interface.

Moreover, for the template of dialog box to be provided to users for respective use case involved template, it is also considered as a workable solution as users can input their options through the most important elements of button and text input box, which both of them are basic elements in a web-based solution and easily to be implemented. Besides, other webpages for different functions of the system will also need to be developed as the UI such as for user to obtain, edit, and submit certain spreadsheet to perform automation process on repetitive Teams operation. Some forms are also planned to be implemented for user can submit their data through the form in the webpage UI to interact with the back-end logic and for the bot to process different request such as schedule task or event, perform automation, or give advisory steps according to user's input in different web elements.

For the challenges and issues involved in the System UI development, one of the challenges is to develop suitable UI solution for it can be integrated and compatible to the digital assistant successfully for ensuring it can achieve the user-bot interaction and the digital assistant's functions can perform properly. Based on investigation and evaluation, the web application to be implemented as UI able to receive user's inputs, run the bot logic code and interact with other components involved that will be discussed in the next section and return suitable response back to the user such as to perform certain operation on the web version of MS Teams. In result, the UI is considered feasible to be developed as a web application for this project.

3.5.2 Bot Core with AI Capabilities

The second important part in this project is the bot core with AI capabilities, which is also stated as the logic of the digital assistant to be developed in this project. According to the project scope, the bot developed should be able to execute single command, create and execute workflows, create, and use template, and include advisory on steps. Therefore, the procedures and the algorithms of how the bot can perform those functions are developed as the second part of this project. Python is planned to be used to develop the major logic of the digital assistant due to the less code to implement functions and its function diversity to achieve different operation.

First of all, the digital assistant should develop the most important function to be able to receive inputs from users, extract information and analyse the inputs collected, and to provide suitable response to interact with the users. To achieve the bot-user interaction, there are two solutions planned to be experimented in the next chapter to select the better approach to build the conversational bot. The first solution is to create the bot by using the tools and software from Microsoft solution which are the SDK related to MS Teams such as TeamsFx SDK and MS Teams JS Client SDK. By applying the related SDK, the bot developed by using Microsoft Bot Framework can directly perform MS Teams functions by using the related APIs to increase the accuracy to perform Teams functions. Moreover, the bot can be easily integrated to MS Teams as a Teams app to allow user who is using MS Teams to install the bot in their Teams application easily. To implement the bot, a bot should be built by using the Microsoft Bot Framework and deployed as a MS Teams app to be integrated to MS Teams after the functions are completed, followed by tested and maintained by using the Developer Portal and MS Teams. The detail system architecture diagram of this solution was depicted in the Figure 3.19.

Another method to implement the conversational bot that can interact with users will be by using the solution provided by Google, which is the Dialogflow, and extra software for RPA process which is the UiPath. Based on this solution, a desktop application will be created to enable the two-way communication between users and the digital assistant, and the user inputs will be sent to the conversational bot created by using Dialogflow for analyzation via suitable API provided by Google. After the interpretation of user input, Dialogflow will classify user’s input into specific intent and the returned extracted information in JSON format can also be obtained via Google API. For example, the extracted information including user’s original query text, intent, entities involved correspond to user’s keywords, and others. The benefits of using Dialogflow to create the conversational bot is it is in low-code environment where the bot training process can be done by using only the website GUI, and it has built-in NLP capability for users to easily train the bot to understand different inputs with the same meaning. Therefore, more time and effort can be saved for the project development. To build the conversational digital assistant by using the second solution and for it to be able to perform Teams functions, the system architecture diagram is shown below. The detail system architecture diagram of this solution was depicted in the Figure 3.19.

Moreover, one of the important functions of the digital assistant is to provide advisory steps to user to suggest them to possible function to perform and possible next steps according to user's context. To achieve the smart advisory function, an AI model should be developed in the bot core so that it can collect necessary data for it to run its algorithm and come out the result of advice and recommendation. This AI model implementation and fine tuning will be involved in both solutions for developing the ML feature of this system. Apart from that, Google Calendar is planned to be involved in this project to further increase user's productivity to schedule their task in the MS Teams workspace app with the automation function developed. Moreover, the advisory steps are planned to be provided also in the Google Calendar via script to suggest user on their future schedule based on its data collected and the result of AI algorithm ran. In result, user may auto schedule their Google Calendar based on the digital assistant's advice to automate the repetitive steps for them to schedule the task every time that can save time and cost. Besides, the bot core logic including function to integrate Google Calendar with MS Teams so that to achieve our first objective to simplify user's steps to perform MS Teams function. This is because via the integration, user can easily navigate to MS Teams or perform any operation on Teams by simply referring to their Google Calendar schedule and interact with our system. Therefore, it saves user's time and effort to only schedule task on Google Calendar and run their desire task via the Google Calendar interface to interact with our system and perform tasks without the necessary to manually open MS Teams and figure out the operations.

Feasibility of the proposed method to perform MS Teams operation

Based on the second solution preferred in the last section to build the bot core of the digital assistant, UiPath will be used to create a sequence for each function listed in the table. This method is considered feasible due to UiPath supports RPA on multiple platforms including Windows application and browser. Therefore, the MS Teams functions and other extra functions are able to be conducted by UiPath to detect and operate specific elements on different platforms involved. The sequences are predefined and will require users to input necessary parameters to automate specific sequence. The parameters can be inputted through text, or input into the templates or dialog box provided by the system to increase user experiences and allow smooth operations of the sequence flow without errors.

However, to increase the feasibility of the proposed method due to the risk of limited time issue, the functions of the digital assistant will be developed in priority according to their importance and frequency to involve in real use cases. For example, some common functions that will be used frequently by users to solve their tasks such as create channel and schedule meeting, which will be developed first before other functions to increase the system's usability. Therefore, all functions planned may not be fully developed in this project, but some important functions will be implemented in prior to ensure the basic functions of this digital assistant and its performance in solving real-life use cases.

<u>Feasibility of the proposed method to achieve AI capabilities</u>

The main advantage of solution one is it can perform MS Teams functions by using APIs provided by MS Teams workspace app itself to increase the function accuracy. In addition, the compatibility of this solution will be higher than the second solution as both Microsoft Teams and the method used in the first solution are both based on Microsoft solutions. However, it consists of limitation of the Microsoft solution to build a bot and integrate the bot into MS Teams app involved too many tools and services, which can make the project development process become more convoluted and time-consuming. For example, the tools and services required to achieve the first solution involved examples of IDE with Teams toolkit extension to build a MS Teams app, Microsoft Azure services including Azure bot service, QnA maker and LUIS to build the conversational bot with NLP ability, Microsoft Bot Framework and Bot Framework Emulator for the bot analysis, interpretation, and testing. Moreover, extra tool is needed to evaluate and test the bot after it has deployed to the MS Teams as a Teams app such as Developer Portal. The too many tools involved to achieve the solution results in the issue of difficult process to design the project development procedures and flows, and the project's difficulty highly increases due to large amount of the tools' features and functions should be learnt and applied in the given time period. Besides, another challenge of this solution is some of the Azure services cost and it will increase the project budget if the solution is chosen to be used to develop the project based on the project cost feasibility analysis.

Besides, for the second solution, its benefits including less tools involved to implement this solution compared to the first solution, and it saves time to develop some project feature such as the NLP capability as Dialogflow provide built-in NLP feature to train the conversational bot. Second of all, it also performs better in the cost feasibility analysis as only free or free version of software and tools are chosen to be used to implement the solution. There are also some disadvantages of this solution such as the bot created by using Dialogflow cannot be deployed and integrated to MS Teams, and the operations on MS Teams can only be performed by the third-party software of UiPath which may raise error when performing the function. Thus, this solution provides less accuracy in performing MS Teams functions but the simplicity to develop the project is a lot higher than by using the first solution.

In conclusion, solution two will be the preferred method to implement the digital assistant due to the higher cost feasibility in this FYP project. Moreover, it is less convoluted than the first solution with its built-in NLP feature and low-code environment to develop the bot. Nonetheless, both proposed methods to develop the bot core of the digital assistant will be performed in the next chapter of actual implementation to remove the method that cannot be performed and decide the final method used in the project development.

Chapter 4
System Implementation

In the Chapter 4, the system implementation and results will be conducted and documented by first to experiment the two planned development methods and decide the final approach to apply. Besides, the procedures to implement the system will be discussed including to set up the software of this project, develop the UI, bot core and logic, and AI parts of this project. Furthermore, some use cases are also prepared and presented in this section, followed by the system results with respective screenshots. Finally, the implementation issues and challenges is also included in this chapter with its proposed solution.

4.1 Experiment of Two Planned Development Methods

Based on the last chapters, two different proposed methods to implement the project are stated. To recap the proposed methods, the first solution is to build a MS Teams app by using the tools of Visual Studio Code with Teams Toolkit Extension, MS Teams related SDK of TeamsFx SDK and MS Teams JS Client SDK, and the developer portal to directly integrate the digital assistant into MS Teams software through Microsoft integration solution. Besides, the second solution is to build the digital assistant by using Dialogflow for creating conversational bot and UiPath for creating RPA sequence and enabling user interaction with the bot via a desktop application developed.

The steps involved in the first solution are building a Teams app and implement the bot by using the Microsoft Bot Framework. To implement the solution, we have installed the extension of Teams Toolkit into the Visual Studio Code and created a project according to the template provided by Microsoft to create a Teams app. Moreover, the dependencies required to build the Teams app were installed to the project according to the related documentation from Microsoft Teams. After a basic Teams app was created by referring to the template, the app is planned to be configured by integrating it into MS Teams and test the user interaction between MS Teams and the app created. To achieve it, the app built should be installed and configured into MS Teams by running the application in Visual Studio Code. The app built should ideally be opened by the desktop application of MS Teams or through browser according to user’s choice. However, there is a problem occurred during the app configuration process and an error was prompted by the Visual Studio Code:

Error: *[Teams Toolkit] Your Microsoft 365 tenant admin hasn't enabled sideloading permission for your account. You can't install your app to Teams! Please applying for a Microsoft 365 developer tenant by joining Microsoft developer program or contact your tenant admin to resolve the issue.*

The error was prompted due to our Microsoft account based on UTAR email used to develop the Teams app do not have enough permission from the IT lab admin to create and install an app in MS Teams. Therefore, the operations to install and configure the Teams app by using MS Teams were unallowed by the Teams Toolkit extension and error prompted. Some solutions were figured to resolve the issues. For example, we have tried to use personal Microsoft account besides from UTAR email to implement the Teams app. However, the limited functions of MS Teams for free Microsoft account are unable to be used to implement and test the planned features of the digital assistant to be developed. For example, no Teams function for free Microsoft account in MS Teams which causes it cannot perform important operations for team collaboration such as create team, create channel, and others. To save budget for this FYP project, the free version of Microsoft Teams that can be accessed by our personal account is not considered to be updated to a higher paid version. In addition, by considering the efforts and time needed to request required permissions from the IT lab admin, the first proposed method is decided to be labelled as solution that cannot be implemented and discarded due to suitable solution can be applied to resolve the issues.

Furthermore, the second solution was also experimented to avoid the risks of both proposed methods are unable to be used to implement the project. To test the feasibility of the proposed method, the software and tools will be used to implement the method were installed or accessed. After that, several steps were conducted to test the functionalities of each software and tool to evaluate the feasibility of the second solution to be used to implement the product. For example, a simple conversational bot was created by using Dialogflow successfully and a Python project that was created by using PyCharm was able to access the analysed results returned by the conversational bot via suitable APIs provided by Google. The result shows that the digital assistant bot was able to be created by using both Dialogflow and the Python project for its feasibility. As the Python project was able to interact with the conversational bot built in

Dialogflow, it can conduct different logic and algorithm according to different analysed results returned from the bot. For it to be able to perform RPA process for MS Teams functions for achieving our objectives, a simple sequence was created by using UiPath and in the Python project, the sequence was tested to be executed by the UiPath bot under certain Python code segment. After all experiments to test the usability of the second method, we decided to use the second proposed method to implement our project. The details of our preliminary work procedures and its results are presented in the next sections.

4.2 Setting Up

4.2.1 Software

Before starting to develop the second solution, there are several software and tools needed to be installed and set up in my laptop as listed below.

1. **PyCharm 2021.3.1 (Professional version)**

   PyCharm 2021.3.1 in professional version was installed and authenticated by the educational license distributed by JetBrains company to build a Python project in this project. The Python project built should be able to connect to the user interface to interact with users and create logics and algorithms for the bot can analyse user's text and perform different operation according to user's intent. Moreover, to implement other code logics for the bot functions to perform repetitive task, schedule task, and other AI related functions.

2. **Dialogflow**

   Dialogflow was accessed and used via browser to build and integrate a conversational bot with built-in NLP ability in this project. It is used to train the conversational bot to be able to analyse and differentiate user's intent according to user's input text. Moreover, it will be integrated to the Python project for the Dialogflow to analyse user's intent from user's query text that obtained from the Python project and return the result to the Python project. In result, the Python project can understand user's intent and perform correspond operation.

3. **UiPath**

   UiPath Studio was installed into the laptop to create RPA sequences for performing MS Teams functions and other necessary RPA processes in order to implement the project scope.

4. **Microsoft Teams**

   MS Teams application was installed into the laptop and can be accessed via browser to test the function of the digital assistant created to perform MS Teams operations.

5. **Visual Studio Code**

   VS Code was installed to be used to develop, evaluate, and test the sample UI in the form of web application for the preliminary work.

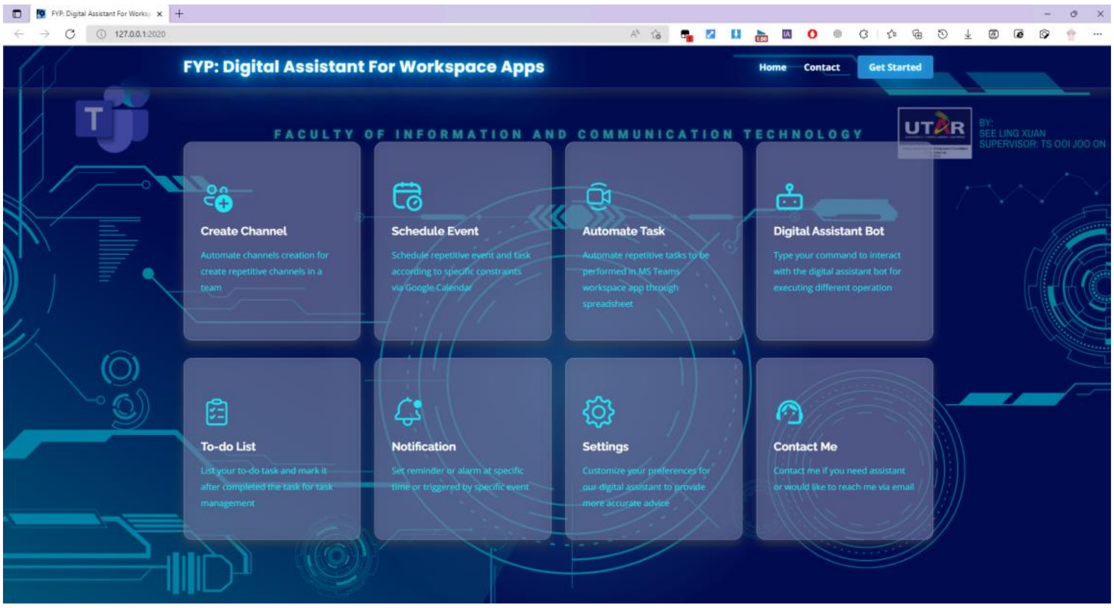
6. **Google Apps Script**

   Google Apps Script is accessed to develop the code related to Google Calendar.

4.3 UI Development

Based on the project scope and system design, a web application is planned to be develop for UI so that users can interact with the digital assistant via accessing the web application. In this FYP project, the elementary function of the digital assistant is more focused than its UI design for achieving objectives. Therefore, the web UI as implemented may be modified in future to further improve its accessibility and a desktop application is also considered to be developed in future for the bot can interact with the desktop version of MS Teams. Moreover, for the bot can be developed in a more attractive and provides various interaction ways to its users.

An UI for web application was developed instead of a desktop application in this project due to a web application is easier to be developed and integrate a Python project compared to a desktop application. For example, Flask framework was installed and used to integrate the Python project and the web application developed. By applying this framework, we can simply design and implement the front-end web by using the programming language of HTML, CSS, and JS while the back-end logic was developed by using Python code. Flask framework helped us to easily transmit data between both front-end web and back-end web for the digital assistant can easily obtain user's input via the front-end UI, process the input and generate correspond output and operation of it, and to easily return the outcome that can be rendered to the front-end web UI. In contrast, more efforts will be needed to achieve the integration between our Python project which is written in Python language and the desktop application in .NET solution which is written in C# language. As more time and effort are needed to integrate the two solutions in different programming language, an easier method was selected in this project to achieve the UI which is to develop a web application. However, a desktop application by using the .NET solution or by using the Unity software will be investigated and developed in future for the project improvement as desktop application also provides its own advantages for improving our digital assistant. For example, desktop application allows users to perform operation on the desktop version of MS Teams and the application can access user's local data in the user's machine to install the desktop application. Therefore, the desktop application increase the system's storage ability and flexibility in processing user's local data. However, based on our web application solution, the user's data need to be processed by the system such as the spreadsheet to automate workflow will be collected through the file submit method. In result, it makes the web application development become a feasible method in completing this FYP system proposed. The following figures show some UI developed for this project while more screenshots will be presented in the Section 4.7 of System Results.


FYP: Digital Assistant For Workspace Apps
Home
Contact
Get Started
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UTAR
BY:
SEE LING XUAN
SUPERVISOR: TS OOI JOO ON
127.0.0.1:2020

Create Channel

Automate channels creation for create repetitive channels in a team

Schedule Event

Schedule repetitive event and task according to specific constraints via Google Calendar

Automate Task

Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Digital Assistant Bot

Type your command to interact with the digital assistant bot for executing different operation

To-do List

List your to-do task and mark it after completed the task for task management

Notification

Set reminder or alarm at specific time or triggered by specific event

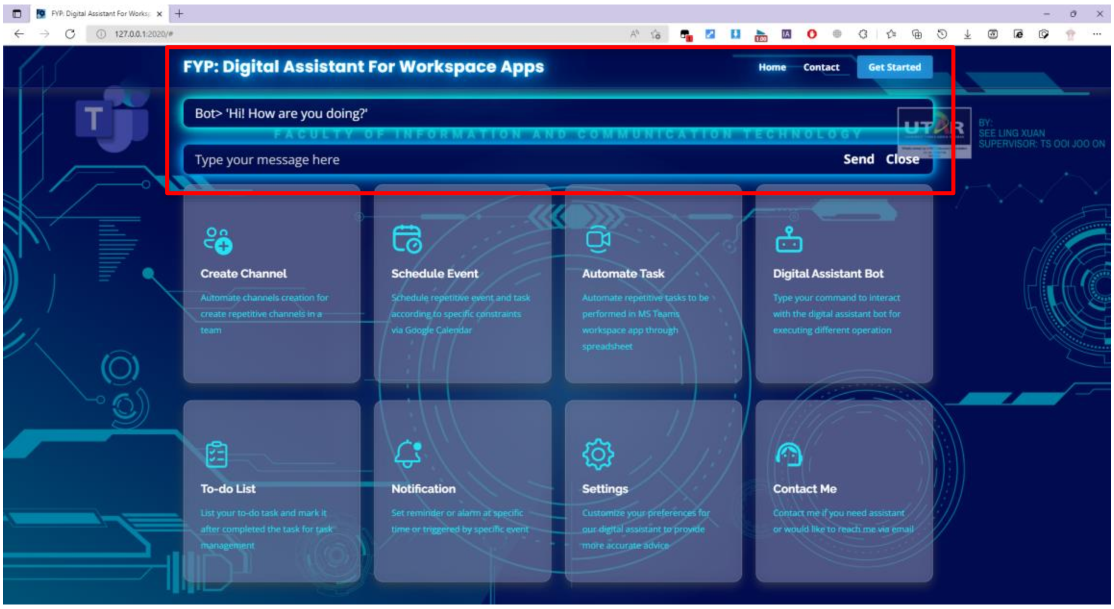
Settings

Customize your preferences for our digital assistant to provide more accurate advice

Contact Me

Contact me if you need assistant or would like to reach me via email

Figure 4.1 UI Developed: Home Page



Bot> 'Hi! How are you doing?'

Type your message here

Figure 4.2 User Interaction with the Digital Assistant Via Chat Box

The web UI developed in this FYP project including a website deployed by using Flask framework in the Python project. Several HTML, CSS, and JS files are coded in respective language of HTML, CSS, and JS to achieve the front-end web development of this project. Besides, some external plugins and technology such as JQuery, JQuery Plugin, Bootstrap, and other APIs or vendor plugins are imported and be applied in this project to develop a more attractive and high-level UI according to our decided color scheme. Also, all icons involved in the webpages' UI are collected from the website of iconify framework that offer various attractive and useful open source vector icons. The website was implemented by modifying the free template based on Bootstrap framework, which is a popular used front-end web framework to develop responsive website nowadays. The UI developed including the main parts of title, sample navigation bar, and a grid layout including eight cards in frosted glass design corresponds to eight functions those are elementary or novelty of the digital assistant to be developed.

On the website, a background with the same theme as the FYP poster is designed by using Canva and used. Besides, the icon and logo used on the website is the logo as designed in the last chapter and the home page of this website including eight functions showed in card form with half-transparent characteristic. The eight functions showed in the UI including the first two functions of create channel and schedule event. These two functions are the solutions to the use cases selected those will be present in the next sections to solve real-life problems and will be explained in more details in the following sections. Besides, the second function allows user to schedule their task on Google Calendar based in a more automated and efficient way. The third function of automate task is used to automate the bot's function to perform MS Teams' operation as a part to achieve the objective two. The fourth function of digital assistant bot is used for users to interact with the digital assistant to achieve our objective one. Besides, all four functions in the second row are novelty functions to be developed as advance features of the system if time and resources available. Those four functions are not the main functions to achieve our project's objectives but to also improve user's productivity in using workspace apps such as MS Teams. For example, the fifth function of To-do List allows user to list the to-do tasks and mark it after the user had completed each of the task listed. This function enables user to manage their tasks to do in the workspace apps more efficiently and easier the task management process. Furthermore, the sixth function of Notification enables user to set reminder or alarm at specific time or triggered by specific event that happened in the workspace apps. To implement this function, listener and trigger event need to be implemented to the digital assistant in respective workspace apps that consume time and effort so is left to be developed in future. However, the benefit of this function is that user can be easily reminded when certain event is triggered in the workspace app to not miss any important case. Also, the settings is the advanced feature to the AI and ML-related function for this digital assistant to improve the system's AI performance by collecting user's preferences and other configurations to provide more flexible and accurate advice to the user. Due to time consideration, the AI and ML related function in this system not yet allows user's configuration but will only involve default collected data in its algorithm to provide advisory steps to user. Lastly, the Contact Me section is also an extra card for the system's user can contact the developer if he or she faced any problem when using the digital assistant or would like to reach the developer. According to Figure 5.2, if users clicked on the fourth card to interact with the digital assistant, an input text box will be showed for users to input their query text to the bot and receive bot's response that will displayed in another text box after it had processed the request.

Several steps are conducted to achieve the website UI:

1. **Develop the front-end web by modifying Bootstrap template.**

   A suitable Bootstrap template was downloaded from the Internet and modified for it to fit the system UI requirements. Some modifications were done to the index.html file as the main page of the web application, the style.css as the main design file of the main web page, and the main.js as the main file to control the front-end web logic. The software including in this step is the VS Code while the programming languages used are HTML, CSS, and JS. More HTML files are then created to develop the UI for other webpages than the system's home page for achieving different subpages those can be navigated to from the system home page.

2. **Deploy the website by using Flask framework**

   Flask framework was installed into the Python project to deploy the web application to the default port number of 2020. All webpages developed including the index.html file was stored to a new folder named templates inside the root folder of the Python project for Flask framework can find and deploy the web page, while all assets used in the web page including images, CSS files, JS files and others were stored to a new folder named static for the sake of another Flask's requirement to locate the resources of the web page to be deployed. After this step, the website developed was deployed successfully and can be accessed by the website localhost:2020.

The development of the back-end logic of the digital assistant will then be presented in both non-ML related part and ML related part in the following sections. This is due to the reason that for easier understanding to the system's AI related and non-AI related parts by separating the implementation process into several small modules. First, bot development of non-ML related functions of the bot will be described in the following section 4.4. As an overview, the **combined steps to develop the entire bot core of the digital assistant including steps of**:

1. Create conversational bot by using Dialogflow. (AI related)
2. Create RPA sequences by using UiPath. (non-AI related)
3. Develop bot functions. (non-AI related)
4. Develop AI model. (AI related)
5. Integrate the UI, Python project, conversational bot built, sequences built, AI model, and MS Teams. (non-AI related)

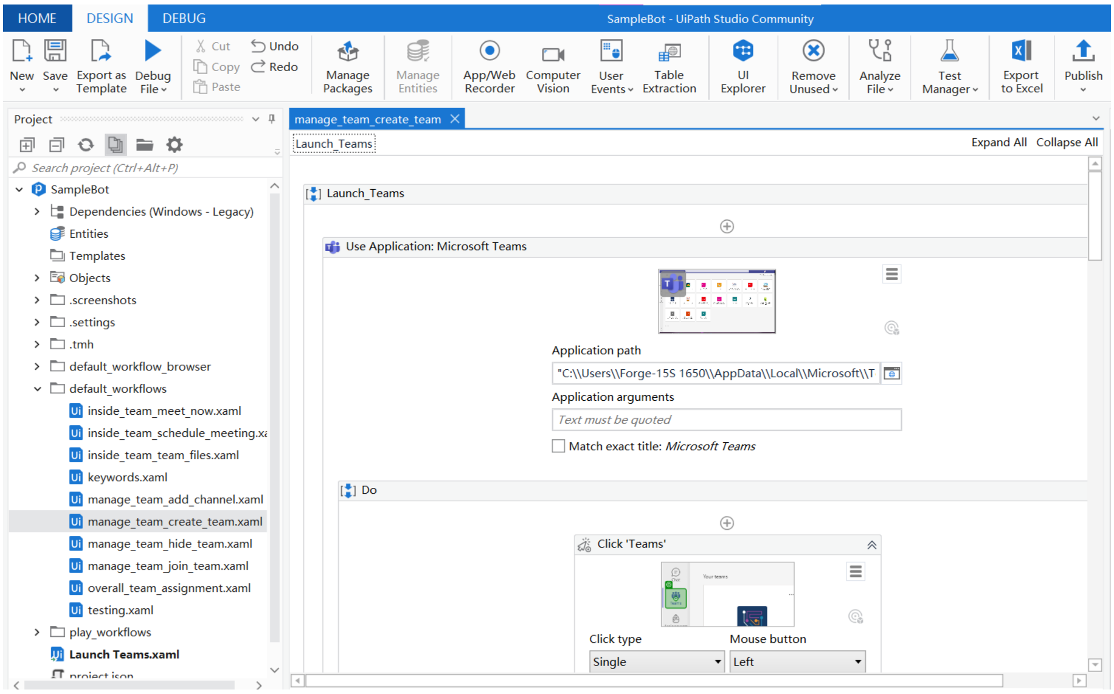
4.4 Bot Development

After the front-end web UI for the preliminary work was developed, the bot core of the digital assistant was then be implemented in the Python project including some of its non-ML related functions to achieve our first and second objective. The bot development including to build respective RPA sequences for task automation, and to develop bot functions those are not ML-related.

Several steps are conducted to achieve the bot development of non-ML features:

1. Create RPA sequences by using UiPath.
2. Develop bot functions.

4.4.1 Create RPA Sequences



Application path
"C:\\Users\\Forge-15S 1650\\AppData\\Local\\Microsoft\\T

Application arguments
Text must be quoted

□ Match exact title: *Microsoft Teams*

Click type: Single

Mouse button: Left

Figure 4.3 Project Structure in UiPath

UiPath is used in this step to create RPA sequences correspond to each intent we have created in Dialogflow. In other words, a sequence means the operation that will be performed by the digital assistant based on specific user’s intent. Therefore, the same number of sequences were created as the number of intents created in UiPath and each sequence was executed, tested, and evaluated by using the Run File function in UiPath. The sequence created will be executed by the UiRobot.exe file in the UiPath programme to test the automation process of certain function. In UiPath, the sequences created to execute specific operation based on user’s input were stored in both folders named default_workflows and default_workflows_browser where the former is to store the sequences to automate MS Teams functions via its website, while the latter is to store the sequences to automate MS Teams functions via its desktop application. In our project, the automation of Teams functions was preferred to be executed via MS Teams website due to the easier element recognition and selection by UiPath compared to the desktop application. This should not be a concern topic in this project as if the Teams function can be performed successfully by the UiPath, our objectives can then be achieved no matter the automation process was conducted via website or desktop application of MS Teams. Also, as we select web application as the UI of our project to be implemented, it will be a suitable solution to develop the RPA sequences those can be run in web browser for easier integration with our web application implemented. The advantages and limitations of both automation methods will be analysed in future to improve the product’s performance. Besides, suitable variables were created and assigned to a default value for each sequence including required variables such as team name and optional variables such as team description and team privacy to create a new team in MS Teams.

For one example of the sequences that we have created in UiPath to automate MS Teams function via its website, the steps of the sequence to create a new team in MS Teams including to first launch or open the browser and access the MS Teams website. After that, the Teams icon in the navigation bar will be clicked followed by the ‘Join or create team’ button to be clicked in the MS Teams website by the UiRobot.exe file of UiPath. After auto-clicked the button ‘Create a team’, the team’s name and team description inputted by the user will be auto-key in into the specific input columns to create a team in MS Teams, followed by the team privacy will be auto-selected according to user’s input. After that, the ‘Next’ button and the ‘Skip’ button will be clicked in sequence and a new team was created by using RPA process of UiPath to automate all steps we have configured in the specific sequence. After necessary sequences were built in the UiPath and each sequence’s steps were configured properly, the second step to create correspond RPA sequences by using UiPath was completed, and the sequences are able to be called by our project for execution.

4.4.2 Develop Bot Functions

According to the project scope and system design, there are several functions to be implemented for our digital assistant. In this project, the required functions of the bot were implemented by writing suitable code and logic in our Python project. In the Python project, several classes were implemented to achieve object-oriented programming and achieve better code structure such as the main class to deploy the web application and achieve the interaction between front-end and back-end web by using Flask framework, the ChatBot class to process and execute user’s command, the ScheduleTask class to perform the digital assistant’s function to schedule task, and others. The steps involved in the required bot functions of the preliminary work are listed:

Function: To execute Teams operation according to user’s command

1. **Receive user’s input from front-end web UI.**

   A function was created in the main file of the Python project for it can receive and process the requests from front-end web by both POST and GET method. The Python project is able to receive requests from front-end web via the Flask framework when user submit the form or other data via the web application UI. In the stage to develop system UI, necessary forms and structure were created for users to submit necessary information to interact with the bot and to perform different use cases. User’s input for different function was submitted to different route and processed by different function in our Python project to perform suitable operation and return data or web page to be rendered to the front-end website. The interaction process between the front-end and back-end web was achieved by the Flask framework. By using the framework, our Python project as the system back-end is able to receive user’s input from the front-end web UI to perform specific operation according to user’s input that can be obtained from the request in dictionary form. In this function, user’s input is the single user’s command to execute specific Teams function by the digital assistant.

2. **Send user’s input to the conversational bot and obtain analysed results.**

   After user’s command was obtained from the front-end web request, it can be sent to the conversational bot via Google APIs for Dialogflow for it to process and analyse user’s input. Besides, to return the analysed results including user’s intent, parameters involved in the user’s query text, and bot’s response to the function in the Python project to receive the information in json format. In this step, a function was created in Python to send user’s text to the conversational bot, receive the analysed result returned by the bot, and save the data returned as a dictionary object in Python. The keys of the dictionary object saved including user’s intent name, parameters involved, bot’s response, and other necessary information for further usage.

3. **Provide suitable response to the user or execute specific Teams operation according to user’s intent.**

   The conversational bot will analyse user’s input and provide suitable response back to the user. However, if user didn’t input enough parameters for the intent as trained in the previous stages, the conversational bot will return a message to request user to input necessary parameters to satisfy the intent's requirement. The satisfaction of intent's required parameters can be detected by an information in the bot returned result to indicate are all required parameters presented in the user's query text for specific intent it categorized to. If users didn't input enough parameters for their specific intent, no Teams operation will be conducted and only prompt message for them to input more parameters will be returned from the Python project to the front-end web UI. In the Python project, a function for parameters validation was created to detect is user inputted all required parameters for a specific intent and if user's input text can pass the validation function, a specific Teams operation can be performed by the Python project.

To perform a specific Teams operation, user's parameters to be used for the Teams operation such as parameter of team name used for Teams operation to create a new team will be passed to the sequence in UiPath with objective to create a new team. To achieve this step, we have created a function in Python to modify xml file by using the cElementTree module from xml framework in Python. This is due to the reason that in UiPath, every sequence was saved in the xml form to store every step and variable used in the sequence. Therefore, user's input parameters were passed to the function to replace the default value of variables in a specific UiPath sequence based on the user's intent. The modification was done to a copy of the original sequence and the xml file after processed was stored to another folder named play_workflow to store the sequence to be run by the system. No changes are done to the original sequence created in UiPath. The file path to store the sequence to be executed was passed to another created function to execute the sequence by using the os.popen() command in Python. After the digital assistant had executed the sequence to automate certain Teams operation, bot response of the specific intent will be returned by the function to process the requests and to be rendered to the front-end web page. Therefore, we implemented the function of the digital assistant to execute a single Teams operation according to user's command.

Function: To execute user's command in batch

We have created a function in our Python project to detect user's intent to create a workflow or a task list to execute multiple commands in batch and perform the operations in sequence. After received user's input command from the front-end web requests via Flask framework, user’s text was passed to a function to detect if the user’s intent is to execute multiple commands in sequence or not. If the intent detected, user will be asked to input their commands to be executed in batch in sequence and type ‘done’ to stop enter more command. We processed every command inputted by the user and stored the analysed result for each command to a list in Python. After the command ‘done’ was typed by the user, the commands stored in the list will be executed in sequence by passing each command stored to the function to replace variables’ value in xml file of the specific sequence, and to the function to execute the sequence by using terminal command in Python. Therefore, we implemented the function of the digital assistant to execute multiple user’s commands in batch. This function is currently done by the user-bot interaction via chat and dialog while in future, a UI dialog box is planned to be implemented for user can input or select their commands to be executed in sequence by using UI elements to increase user’s experiences and simplify the work.

Function: To schedule user’s command

We have created several functions in our Python project to detect user’s intent to schedule their task and schedule user’s command to be executed on specific time or interval. To schedule user’s task, we have created a class named ScheduleTask to extract date and time related information from user’s text by using nltk module in Python and schedule user’s task by using apscheduler module in Python. By using the background scheduler in the apsheduler module, user’s task can be scheduled to be executed on a specific time, interval, or according to a specific time constraint in background to not block other processes executing in user’s device. For the steps involved in implementing this function, user’s input text was passed to a function to detect if the user’s intent is to schedule task or would like to run the task in real-time. If the intent detected, user’s text will be passed to another function to extract datetime information inside user’s query text such as year, month, day, hour, minute, second, day of week, and others. According to the extracted time information and keywords related to task scheduling such as ‘every’ and ‘after’ from user’s input, user’s task will then be added to the scheduler provided by the apsheduler module to be performed later. In the preliminary work, user can perform this function via two different ways to schedule their task by chat with the digital assistant or input their schedule information via a dialog box UI developed in the web application.

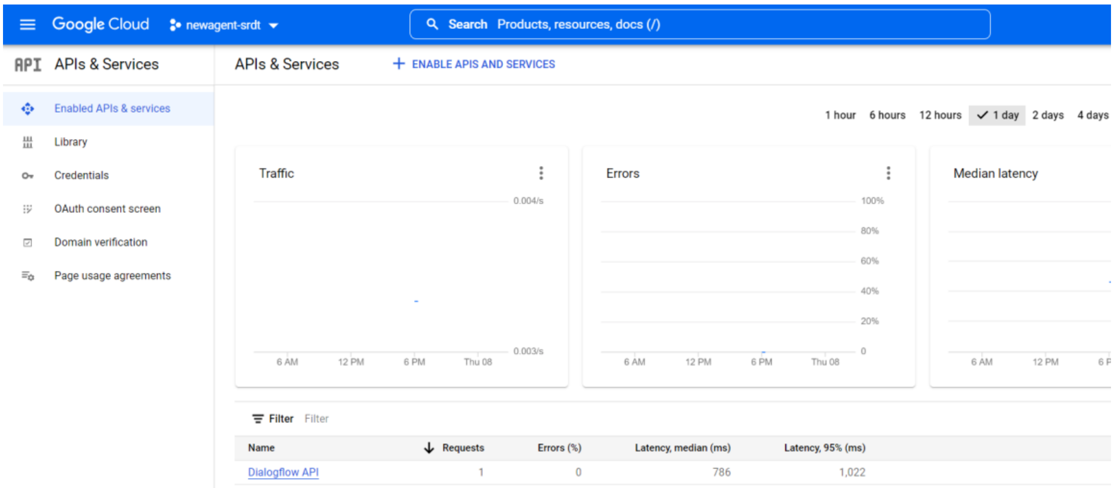
4.5 AI Development

The development of the AI features of this project including to develop and apply its NLP ability and build an AI model for predicting and providing the advisory steps to user. Therefore, the AI development part of the digital assistant can be explained in two separate parts to develop its NLP feature and the AI model development part. Besides, the AI model built is to fulfil the third objective as our planned. First of all, the implementation, training, and application of the NLP technology of the digital assistant will be described where the software used to achieve this ability is the Dialogflow provided by Google solution.

4.5.1 NLP Capability Implementation

To implement the NLP capability for the digital assistant can understand user's query text in natural language, analyse user's intent and parameters included in the text, and provide suitable response back to the user, Dialogflow is used to build a conversational bot that is the first step to perform in the summarized procedures to develop the entire bot core. The benefit of using Dialogflow including the simplicity for training the NLP ability of the digital assistant and APIs provided by the Google solution so that our digital assistant developed in Python language can easily send request to the conversational bot built in the Dialogflow platform and obtain results analyse by the bot trained with NLP ability and returned to.

To implement this step, a new project was created in Google Cloud before the bot can be created by using the Dialogflow and the Dialogflow API should be enabled in the APIs and Services section so that we can build a Dialogflow solution and access the solution via deployment.



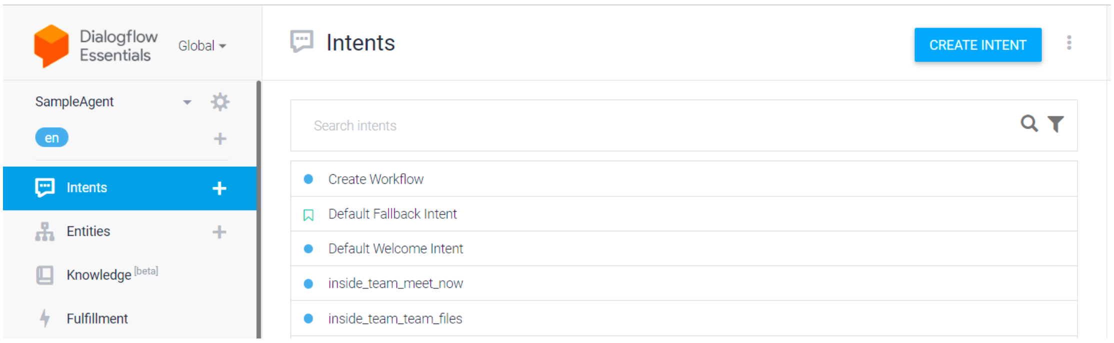
Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
Dialogflow API	1	0	786	1,022

Figure 4.4 Enable Dialogflow Service in Google Cloud

The steps to create a conversational bot by using Dialogflow were then be performed by referring to the documentation provided by Google. After that, a new agent was created, and a basic conversational bot was now built in the Dialogflow. We were then started to train our conversational bot for it to analyse user's intent to perform different bot function and give suitable response. For example, our bot functions including to perform MS Teams operations such as to create team, create channel, schedule meeting, and others.

Several steps are conducted to develop the conversational bot in Dialogflow:

1. **Create intents.**

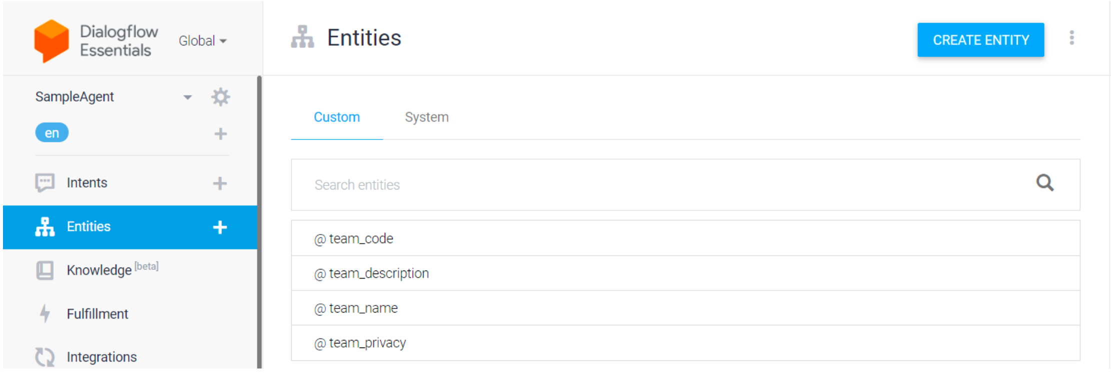


- Create Workflow
- Default Fallback Intent
- Default Welcome Intent
- inside_team_meet_now
- inside_team_team_files

Figure 4.5 Intents in Dialogflow

An intent in Dialogflow categorizes user's intention for a single conversation dialog. In this project, user intents including every function that can be performed by the digital assistant as listed in the table in Chapter 3. Therefore, an intent is planned to be created for each function to be conducted by the bot as listed in the previous section in Chapter 3 after the use case diagram. According to the functions listed in the tables, each function will have their respective intent for the bot can understand user's context and predict accurate function that the user would like to perform. The intent will be trained in the followed steps to build the NLP ability of the digital assistant for analyzing user's intent based on the keywords and entities including in user's query. For example, if user inputs text of 'Create a new team' to the bot, the bot can categorize user's input to the intent of creating a new team in MS Teams. Therefore, it build the conversational ability of the bot to understand user's context, extract keywords, and categorize user's query into specific intent. The next step will be to create entities needed for some intents to perform MS Teams operation.

2. **Create entities.**

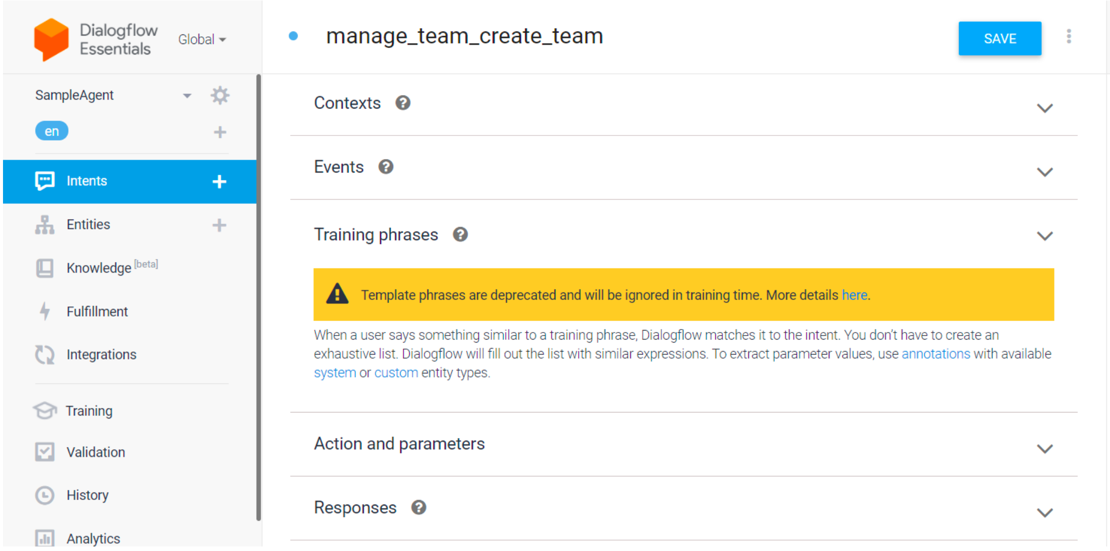


@team_code
@team_description
@team_name
@team_privacy

Figure 4.6 Entities in Dialogflow

Secondly, necessary entities were created in the Dialogflow. Entities are parameters those can be included in user's query text such as datetime, location, and others. In this project, some new entities should be created such as team name and team privacy that can become a required or optional parameter in user's query text to achieve specific Teams operation. Entities is important so that the digital assistant can develop its NLP ability to request specific entity based on the intent's needs after the intent was trained. In this step, the entities will be created according to the parameters needed by each function to be performed by the digital assistant in MS Teams as listed also in the previous tables. After intents and entities are prepared for each operation of the digital assistant can perform, each intent will then be trained for the bot's ability to understand user's query, conduct analysis, and response in human language as its NLP ability.

3. Train intents.



Template phrases are deprecated and will be ignored in training time. More details here.

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

Figure 4.7 Train an Intent

After created necessary intents and entities, a single intent can be trained for the conversational bot built can analyse user's query text, categorize user's intent, extract parameters or entities included in user's text, and provide suitable response. The results analysed by the bot will be returned in a JSON file for further analyzation after the integration between our Python project and the bot. The training process of an intent involved to first add user expression that can lead to the intent into a specific intent. For example, user expressions such as 'Create a new team' and 'Build new team' were added to the training phrases of the intent to create team in MS Teams so that user's text that has similar meaning to the training phrases can be categorized to this intent. For the second step, the parameters those can be included in user's query text were specified such as the intent to create a new team in MS Teams need required parameter of team name and optional parameters of team description and team privacy to conduct the operation. Therefore, the bot will be able to extract parameters from user's query text and categorize them into correspond entities in Dialogflow. In the last step, specific response sentence was added to the intent for the bot can provide suitable response back to the user. For example, 'Will create a new team [team name]' was provided from the bot to the user if the user's intent is to create a new team in MS Teams with a team name provided by user as a parameter.

4. **Conclude.**

   After necessary intents and entities were built in the Dialogflow and each intent was trained properly, the first step in the overview steps to build a conversational bot by using Dialogflow was completed, and the conversational bot with NLP ability is able to be integrated to our project for further operations. The NLP capability of the digital assistant allows it act as a smart assistant to interact with user in normal language of English. Since the NLP ability was developed for the digital assistant for it can detect what function user would like to perform according to the keywords inputted in their context, the next AI feature to be developed for the assistant will be to build an AI model that can give advisory on user's steps to perform with higher accuracy. Therefore, user can choose whether to perform the recommended function suggested by the digital assistant or not to proceed their next operation. The objective of the AI model is to predict user's next operations for the digital assistant can give advisory steps to user.

4.5.2 AI Model Implementation

Data preparation is the first step in developing the AI model before actual training the model as the AI model developed will need to feed in data for it can perform its algorithm and predict the desired outcome. The AI model built can further increase the flexibility of the digital assistant to process users' input and give suitable recommendations on user's action so that user can interact more dynamically with our digital assistant even the user has less knowledge in using MS Teams. To give advisory on user's steps, the data prepared for the AI model can run its algorithm and predict outcome including the sequence of history commands performed by the user before and the data that can analyse the relationship between MS Teams operations and functions for investigating the relationship between different command in MS Teams. Therefore, the relationship between different command in MS Teams can be investigated and based on the relationship, the model can predict the next possible command that can be performed by the user and suggest the related commands with higher interaction percentage to user's history commands or functions had performed.

The data preparation step can be conducted in several methods including to collect raw data from different online data source or prepare own fake data for training the algorithm. In our implementation, fake data is prepared due to the reason that the data required such as user's history data is hard to find the data source through platforms such as websites or online databases as it considered as personal data. Therefore, a sample dataset of user's history command was prepared to be used to train the AI model in the following training process. Moreover, the data of every function that can be performed in MS Teams and their details is collected to discover the dependency between different functions and find out their interaction and relationship to know what function easily led to what function to be performed by the next step. The data preparation step is implemented by creating the dataset X of the data the model can use to predict the outcome, and the dataset Y as the outcome of each row in the dataset X used to train the model, which is the actual next step performed by the user in our solution. As the data collection process is conducted by preparing own dataset with fake and sample data, only few amounts of data are prepared by considering the time and resource constraint. Therefore, the data collected only in a limited and small amount which may affect the algorithm's performance but, in this project, the elementary function of the AI model to give advisory on steps will be more taking into account than pursuing the high accuracy of the model that can cost too much time in the model implementation. In result, the small amount of data should not be considered as a problem if the AI model can be successfully trained and obtain desired output with minimum of 75% of accuracy. The AI model developed can then be further modified its dataset used for training and its algorithm in future improvement to increase the model's accuracy and performance.

For model training, several algorithms of the AI model can be developed are investigated and analysed such as clustering, decision tree, random forest, logistic regression, linear regression, etc. The framework can be used to develop the AI model in Python languages are also explored such as PyTorch and TensorFlow. After evaluation, TensorFlow is selected to be the framework used to build our AI model by considering its advantages including the guided curriculum and resources provided, ML supported feature to easily build ML solution based on its existing solution, and various number of tutorials provided by the TensorFlow framework for references. Also, the TensorFlow is considered easier to be applied than PyTorch in our proposed solution due to its functions and features. Therefore, the environment needed to apply the TensorFlow framework is set up in our Python project structure so that this framework can be used to develop and deploy our AI model used to predict and provide user with the advisory on steps.

To set up the environment, TensorFlow is installed in out PyCharm as the IDE used to develop our Python project by installing the TensorFlow package into it so that our function can use the functions provided by this package to conduct a ML solution based on our requirement. After the required environment to use the TensorFlow package in building our AI model is set up, we now need to train a new AI model by developing its algorithm we would like to use to run the model. The ML algorithm selected to be applied in our AI model training is the regression algorithm to predict our desired output based on the input data we fed into the algorithm developed. Besides, the regression model is suitable to fit in out dataset prepared including dataset X and dataset Y for conducting the prediction purpose. The algorithm can be easily implemented by using the modules and functions provided by the TensorFlow package according to its tutorial to build a regression model. The model's objectives are to find out the best functions to suggest to the user as their future possible step to perform with percentage for each suggestion based on the prediction confidence.

In the ML algorithm developed, all data prepared in the first data preparation step is now be fed into the algorithm to train it so that it can come out the prediction. The dataset is not split into both training and test set in this stage due to the limited dataset prepared in our first stage. In the system implementation of this AI model, suitable functions provided in TensorFlow to build a regression model is called in the function we developed to run the ML algorithm with necessary parameters given to the function. The respective function provided by TensorFlow to build and train the regression algorithm with our prepared data will then run the algorithm and return back the predicted result to us. Therefore, we can evaluate the model trained and finetune it for higher performance and accuracy of the model can be achieved.

After the AI model was trained to output the suggestion of user's possible next functions by investigating the relationship and dependency between different function in MS Teams and based on data of user's history command and function performed, the model built can undergo the fine-tuning process to increase its accuracy. The finetuning process including to discover the score metric of our developed model that is one of the metrics to identify and evaluate the model's performance and accuracy.

The score metrics used to evaluate the model's accuracy in this project are the confusion matrix, f1, and recall those can be simply calculated from the functions provided by TensorFlow to know current model's performance based on the scores and results obtained. To perform the finetuning process, cross validation is implemented so that the model's accuracy can be increased in a little amount as the data prepared by us is not huge enough to highly increase the model's accuracy even after finetuning. However, the performance of the model can be evaluated via the score metrics and undergo finetuning process after the model training and evaluation to complete the model. The training and finetuning processes are both be implemented by calling and using suitable functions provided by the TensorFlow package. After this last step to develop this component involved ML development of the project, the AI model is implemented by using TensorFlow in Python language to predict and suggest user's next steps based our dataset prepared and feed into the algorithm.

4.6 Integration

After the UI and the bot core including ML-related and non-ML related functions are completed, the system UI, conversational bot built by using Dialogflow, sequences built by using UiPath, and AI model built were integrated to our Python project. So, the digital assistant can receive user's input via the system UI, process user's input to obtain user's intent, parameters, and suitable response text from the conversational bot, and execute specific RPA sequence to automate certain Teams function in MS Teams. Besides, the digital assistant can run its algorithm to perform prediction and give advisory steps on user's fuzzy context or not involve direct keyword of certain function in their context.

Several steps are conducted to achieve the integration:

1. **Use Flask framework to integrate system UI with the Python project.**
   As described in the first step to develop system UI by creating a web application, the web application created was deployed by using Flask framework installed in our Python project. In the Python project, specific web page file was returned to a specific route and the front-end requests were processed by creating different functions to process the requests such as form submit from the front-end web.
2. **Use Google API to integrate the conversational bot with the Python project.**
   Due to the conversational bot was built by using Dialogflow based on solution provided by Google, Google has provided suitable APIs for our Python project to perform several operations to the conversational bot that we have created and trained. Therefore, integration is allowed for our Python project to send user's input text to the bot in Dialogflow and receive the bot processed results in JSON format via Google APIs. To use Google API, several dependencies were installed in the project based on Google documentation. Therefore, the digital assistant will then be able to perform different operation according to the user's intent and parameters as returned by the Dialogflow bot and render the bot's response to the front-end web to show to the user.
3. **Execute terminal command with Python to execute specific RPA sequence.**
   Due to UiPath and UiRobot.exe inside the UiPath programme are applications installed to our desktop, the application can be launched, and the sequence created by using UiPath can be executed via the terminal command of '[Path of

UiRobot.exe file] execute –file [Path of the sequence to be executed]'. Therefore, the RPA sequence can be easily executed in our Python project by running this command in terminal via the Python function of os.popen(command) where the command to be executed via terminal will be passed to the function. In result, the integration between our Python project and the sequences created by using UiPath was achieved, and the sequences are able to be executed by the UiPath programme installed in our desktop to perform Teams operation via browser.

4. **Integrate AI model built to the Python project.**

   As the AI model was built in Python language as same as the Python language used in our project structure to develop the digital assistant, the model can be simply integrated into our project by creating suitable module and functions. The functions created allows the AI model can read the data prepared in excel sheet and feed the data into its algorithm. After that, the AI model should run the algorithm and return its result after function call so that the back-end function can retrieve the AI model's result and render it back to the user's screen. In result, users can get the digital assistant's advisory steps and choose to perform the step or not.

5. **Conclude.**

   After the steps were performed, the integration of our Python project, web application as the system UI, conversational bot, RPA sequences to be executed by UiPath to perform operations on MS Teams, and the AI model built was achieved. The result of the complete system will be presented in the section 4.8 with screenshots to present the completed system's results and flows.

4.7 Use Cases

This project had done some research and perform analysis to collect several real-life use cases for implementing the digital assistant's functions those can be useful in solving real-life problems and achieving our objectives. The use cases including two use cases collected in the preliminary work and one more industrial use case prepared in the FYP 2 work for solving industrial problems. The two use cases collected during the preliminary work are to create channel and schedule event. In the preliminary work, the basic digital assistant's functions including to execute single or multiple commands, and to schedule task on a specific time or interval. The use cases will be implemented by using, modifying, and integrating these three implemented digital assistant's functions with other tools and functions, and providing templates to user in the form of dialog box to simplify user's tasks in real-life. Moreover, another use case involved in the FYP 2 work including to provide advisory steps on user's Google Calendar. The use case will be implemented by integrating Google Calendar with our digital assistant and MS Teams for users can easily schedule their task on MS Teams via Google Calendar based on the bot's advice and recommendations.

First, the three selected use cases are introduced:

1. **Create channel.**

   According to our research and information collected from various sources, one of the use cases in real-life is to create multiple channels in a team with a regular naming basis based on the organization's standard rule and requirement. For example, an organization may need to create different channel for members in different team in the company to work and cooperate together and the naming rule of each channel created can be the example of [Department name – Team name – Number of team member, the duration of the team to be formed, or the goal of the team to work on]. Therefore, it can be a strict process for a department manager or a programme manager to create multiple channels with a standard naming rule whenever a new team is created, or a new project is announced periodically or under a specific time constraint. Moreover, time waste if a project manager needs to undergo full steps to create a new channel with very similar naming and description as the previous project conducted but only with different team members, especially when the task should be done periodically. For another example, by implementing the online learning mode, the lecturers and tutors of UTAR need to create channels with similar naming and description for every subject or class they teach in every semester but only with different students to add to the channel. This repetitive process should then be automated to save the lecturer's time to create similar channels with a standard naming rule for every semester if they keep teaching the same subject. Therefore, the objective of this digital assistant to solve the use case is to automate the process of channels creation in MS Teams according to the naming rule inputted or selected by the user. Furthermore, it is important to automate repetitive work to save user's time and effort.

2. **Schedule event.**

   The second use case in real life is for user to schedule repetitive event or meeting in their calendar. For example, a worker in an organization may need to partake in a regular meeting with different team members, co-workers, or staffs from higher department to report on their work progress or to conduct a regular discussion. For another example, the lecturers and tutors of UTAR need to schedule every class they are teaching in the semester which makes them need to pay the amount of time and effort to build a convoluted schedule. Therefore, it can be a strict process for the worker to schedule every repetitive event by performing similar steps to schedule task on a specific interval such as weekly or month basis on their calendar. In result, time and effort are wasted due to the repetitive steps to schedule every single event and most of the tasks may consist of the same detail information but only in different date and time. To resolve this real-life use case, Google have provided its solution for users to schedule their periodic tasks via Google Calendar by inputting information such as the day and time interval to schedule their task, the meeting room link in the Google Meeting platform to be set up for the event, and other information to be included in the event scheduled. However, there are some limitations on the Google solution such as user will still need to waste time to input every detail information for their task to be scheduled periodically. To reduce the time and effort wasted, auto fill in ability should be implemented to the solution for user to schedule an event in an easier and faster way. Moreover, not enough columns in the Google Calendar solution for user to schedule their events. For example, user may want to add in other information for their event scheduled such as the link to other meeting platforms besides form Google Meeting platform, important notes, and others. Therefore, the solution should allow users to add in more column to their event scheduled to simplify their task to schedule an event. In result, the objective of this digital assistant to address the use case is to automate the process of event scheduling by using online calendar. Besides, it is important to reduce time wasted, avoid user’s repetitive steps, and simplify their work to schedule event or task in the MS Teams workspace app.

3. **Provide advisory steps on user’s Google Calendar**

The third use case indicate the problem of users to waste time and effort in manage their Google Calendar schedule based on their daily or periodically task that need to perform by using the workspace apps such as MS Teams. For example, time wasted for user to manually create the schedule of every task on Google Calendar even with the function developed to solve the second use case as user still need to fill in the excel sheet for schedule automation. For the third use case, we would like to develop the digital assistant’s function that can suggest the possible future schedules user may want to create on their Google Calendar based on the result calculated from the AI model built. For example, an employee works in certain organization may schedule an event on Google Calendar in certain period of after scheduled another related event. In this use case, we proposed a solution for our digital assistant with AI ability to predict and suggest the future events those the user may want to schedule after they have scheduled certain events or during certain situation. Therefore, to further eliminate the user’s workload and time to edit and modify the spreadsheet used for schedule automation on Google Calendar. This use case proposed is useful for user to build their schedules more efficiently and productively according to the bot’s advice. Therefore, to also increase their work productivity and remove the needs of repetitive and tedious task operations.

Secondly, the steps involved to achieve the three selected use cases are presented:

1. **Create channel.**

   A navigated webpage was created in the web application to accept user's input of the necessary information to create one or multiple channels in a team of MS Teams. The standard and rule to name the channels are inputted, selected, and can be defined by the users. After user submitted the form in the webpage to create channels with a standard naming rule, the form data will be transmitted to our back-end Python code via the Flask framework. After that, the data will be processed by our implemented function to create every channel defined by the user in the MS Teams by using the sequence built in UiPath to create channel. After all channels are created according to user's requirements or error occurred, suitable response will be returned by the back-end logic to the front-end web application.

2. **Schedule event.**

   A navigated webpage was implemented in the web application for users to input the detail information of the task they want to schedule on a specific date, time, or interval under certain time constraints on Google Calendar. The information can be inputted by the user involved the time-related information to schedule user's event, the platform user would like to hold the event or meeting, and other information that is important to schedule an event. Moreover, users are allowed to add in other information to the event scheduled by inputting it to the specific column of the webpage provided. After user submitted the form in the webpage, the form data will be sent to our back-end Python function to process the data and add the event scheduled to the online calendar. In result, user's single event or periodic event can be scheduled to the online calendar with proper information involved in the event. To improve user experiences, auto fill in feature was implemented in the form for user can schedule their event with shorter time and more productively. Furthermore, users are also allowed to add in more text to be auto filled in into their future schedule event, and the text inputted by user will be processed by our implemented Python function and saved to the system. After the events were scheduled, suitable response will be returned to the front-end web page.

   As this second use case implementation involved Google Calendar as the online calendar used to schedule user's event, a respective script is written in JS in the

Google Apps Script platform for it can perform operations on Google spreadsheet and Google Calendar via APIs provided by the Google solution. In the implementation of the project's function to solve this use case, the script developed in Google Apps Script platform request users to fill in information needed for schedule automation in respective webpage developed for adding the data into the google spreadsheet created. Then, the script will traverse each data row in the spreadsheet to extract the information for scheduling each row as a task onto Google Calendar by calling its API functions to add new schedule onto user's Google Calendar. As a new feature to Google Calendar, a function is also developed to add MS Teams channel link in user's schedule created. So, user can navigate to MS Teams from Google Calendar so that they can easily access the website of MS Teams if the event or task scheduled required to be performed in MS Teams workspace app.

Moreover, the interaction method between user and the digital assistant via UI of the webpages implemented was also developed for the bot's function to schedule user's command. This is to improve user experiences and to increase the accuracy for users to schedule their task on the correct time or interval as the accuracy to extract date and time information from user's natural language cannot be ensured completely by using the nltk module for NLP. Via the dialog box developed, users can input specific time constraint to schedule their task and the task scheduled will also be added to the jobs of the scheduler according to the last step to develop the digital assistant's function to schedule task. The bot function to schedule task can also be considered as one of the use cases in this project for user to automate their repetitive routine task by a single command or operation to schedule task. Therefore, to avoid user's repetitive steps to perform the similar task during a routine datetime.

3. **Provide advisory steps on user's Google Calendar**

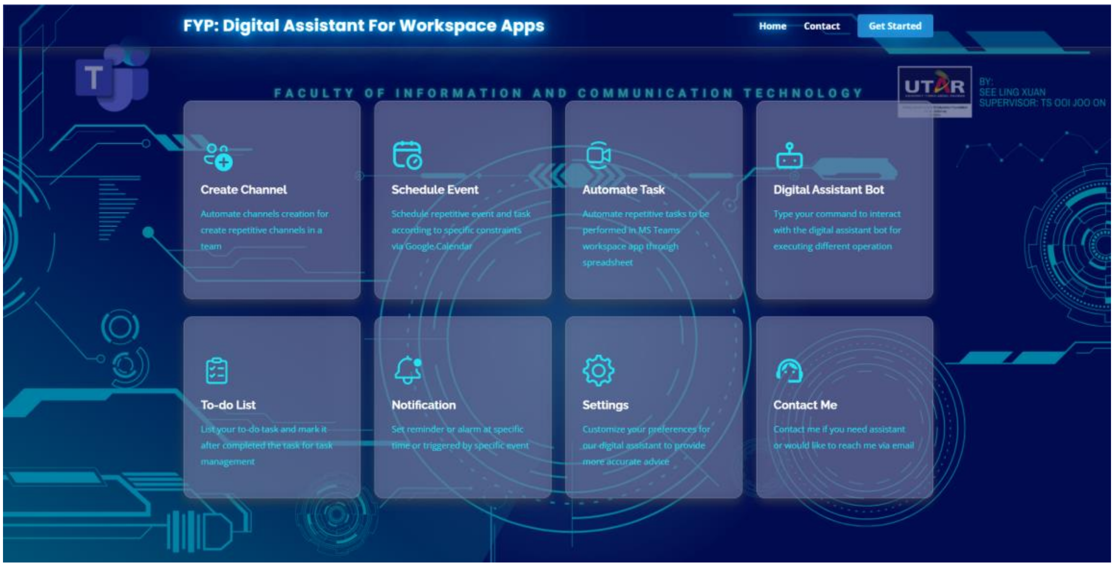
In the third use case, advisory steps should be provided by the digital assistant to recommend user the possible schedules to make on the Google Calendar. The AI model is trained and develop its algorithm so that it can predict the possible schedules those can be created by the user in future under certain situation. The prediction of the AI model will be based on user's previously created schedules and events to forecast what schedule the user will need to perform after by taking account into the task characteristics and relationship with other task of the history schedules. Therefore, the AI model is built according to the same steps and procedures we used to develop an AI model in this project including data preparation, AI modelling, model fine tuning and evaluation for completing the AI model that can run algorithm and get the desired output. For the desired output of recommended schedules to be created can be presented to user and create related schedules in Google Calendar, Google Apps Script will be continued to use as in the solution for the second use case for achieving the integration between our digital assistant and Google Calendar by implementing scripts. Respective scripts are developed in JS language for the advice can show to the user and user can choose to create respective schedules according to advice given by the digital assistant in the Google Calendar interface. In result, the developed AI function to provide advisory steps on user’s Google Calendar can achieve our second objective to automate repetitive task on creating schedules related to MS Teams, and our third objective to give advice on user’s operations to create respective schedules in their Google Calendar.

4.8 System Results (With Screenshots)

The software of PyCharm, Dialogflow, UiPath, MS Teams, and VS Code were installed in this project to implement our proposed solution. Based on our FYP work, a system UI in the form of web application was created and be deployed by using the Flask framework in our Python project created. The web application provided the way for user to interact with our digital assistant to achieve the functions to perform specific user's command, schedule task, and resolve two real-life use cases to create channels and schedule event. Moreover, the bot core of the digital assistant was also developed in this project by implementing different component of it including ML-related and non-ML related components followed by integrating all components. Based on our system implementation work as described, we have first created a conversational bot by using Dialogflow. Necessary user intents and entities correspond to the bot's functions planned in the system design were created and the bot was successfully integrated to our Python project.

Besides, we have created necessary RPA sequences to operate Teams functions correspond to the intents created in Dialogflow by using UiPath. The sequences created were successfully integrated to the project and the sequence can be executing successfully via the terminal command in Python. In addition, we have developed some classes and functions in our Python project to achieve the basic bot functions to execute single or multiple Teams functions according to user's command and schedule task. Also, the digital assistant's functions to create repetitive channels and schedule repetitive tasks were implemented so that it can resolve the two use cases collected in this preliminary work. Moreover, we have developed an AI model for it can provide advisory on steps for the rest objective to be achieved in the FYP 2 work. Therefore, with the AI model developed, the third use case is also achieved via the implementation of the system's function to provide advisory steps on user's Google Calendar.

In conclusion, we have created a digital assistant that can receive user's input command or form data via the web application, perform specific operation correspond to user's input, and provide suitable output back to the web application. Besides, the digital assistant can also help user to automate work and provide advisory on steps via different module and function implemented in this project. Moreover, user can schedule their work that need to be performed in MS Teams workspace app via the schedule event function in our digital assistant developed. The result of our FYP work can satisfy our planning of the system methodology and system design as explained in the previous chapter and sections. From now on, system screenshots will be showed to present the system result as achieved including the flow of how the system can work.



Automate channels creation for create repetitive channels in a team

Schedule repetitive event and task according to specific constraints via Google Calendar

Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Type your command to interact with the digital assistant bot for executing different operation

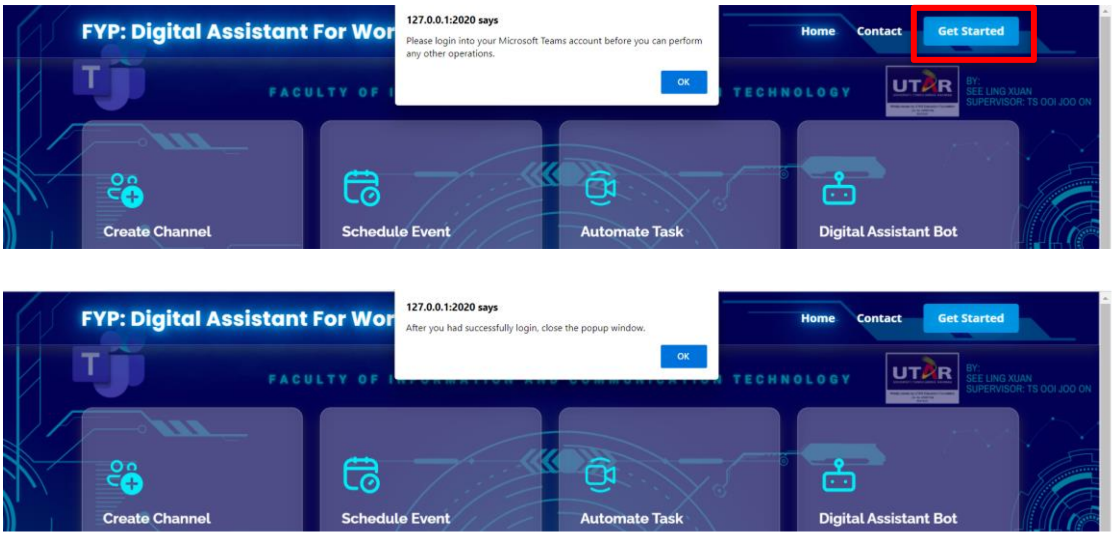
List your to-do task and mark it after completed the task for task management

Set reminder or alarm at specific time or triggered by specific event

Customize your preferences for our digital assistant to provide more accurate advice

Contact me if you need assistant or would like to reach me via email

Figure 4.8 Homepage



127.0.0.1:2020 says

Please login into your Microsoft Teams account before you can perform any other operations.

127.0.0.1:2020 says

After you had successfully login, close the popup window.

Figure 4.9 Homepage: Get Started Button

The Get Started button in the system homepage allows user to login to their MS Teams account before they can perform any operation via the digital assistant in user’s MS Teams account. Authentication is important for the RPA sequences created can be performed successfully in the MS Teams website that required user to login before performing any of the operations on it. A popup window is showed after user had closed the two alert boxes and login to their MS Teams account before closing the popup window.

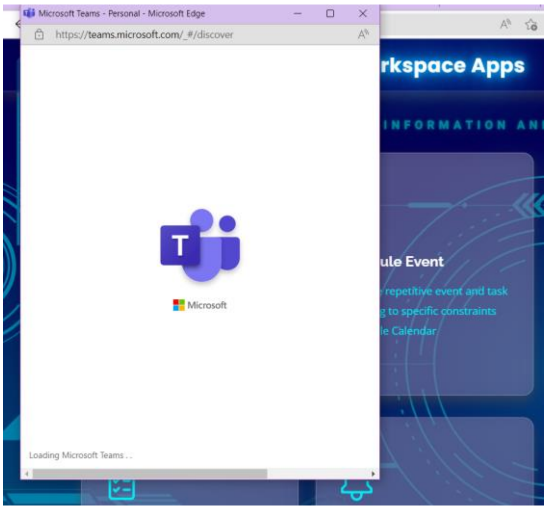
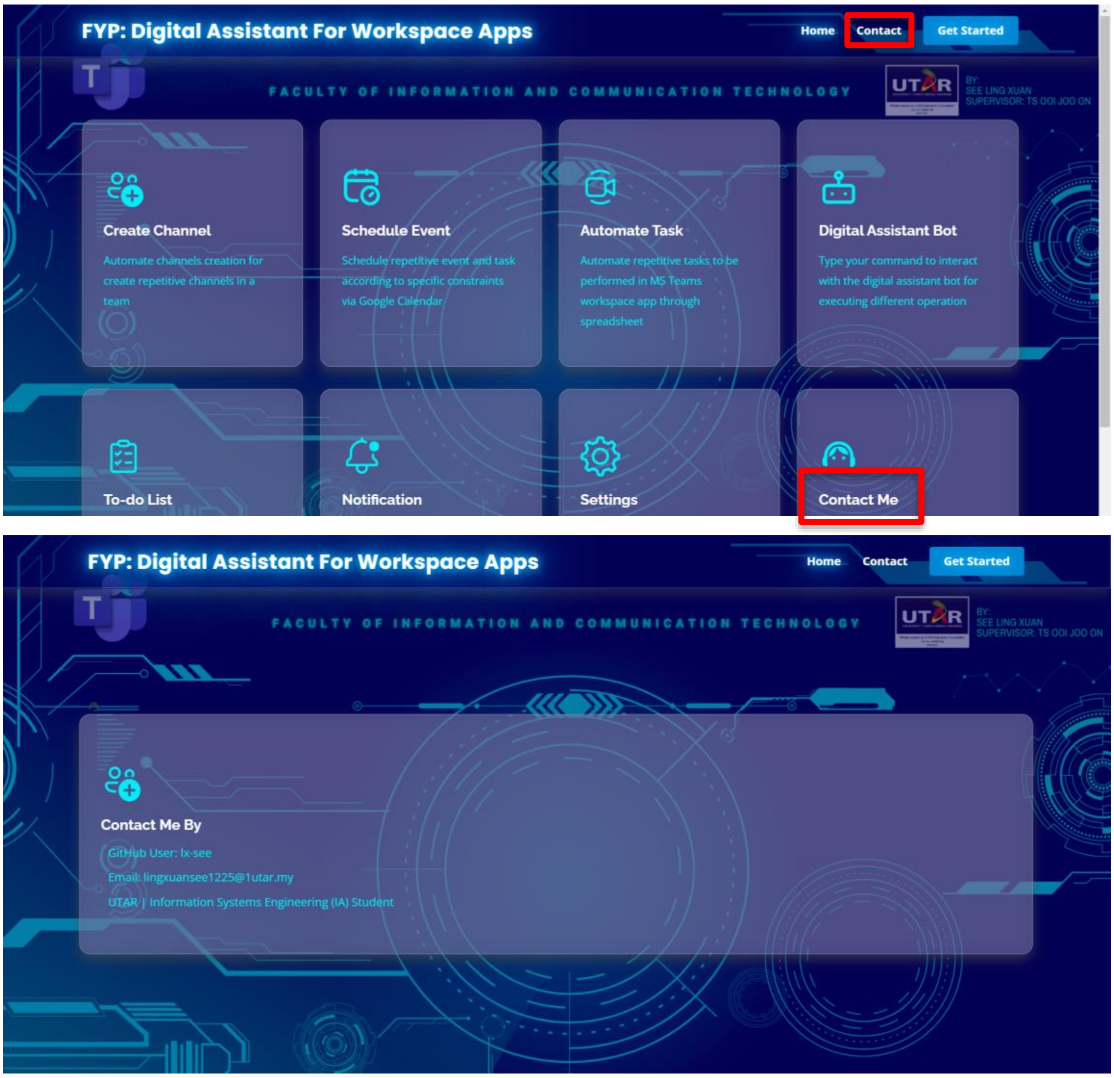

Microsoft Teams - Personal - Microsoft Edge
https://teams.microsoft.com/_#/discover
Microsoft
Loading Microsoft Teams ...
rkspace Apps
INFORMATION AN
ule Event

Figure 4.10 Homepage: Popup Window for MS Teams Login


FYP: Digital Assistant For Workspace Apps
Home
Contact
Get Started
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UTAR
BY: SEE LING XUAN
SUPERVISOR: TS OOI JOO ON
To-do List
Notification
Settings
Contact Me

Create Channel
Automate channels creation for create repetitive channels in a team

Schedule Event
Schedule repetitive event and task according to specific constraints via Google Calendar

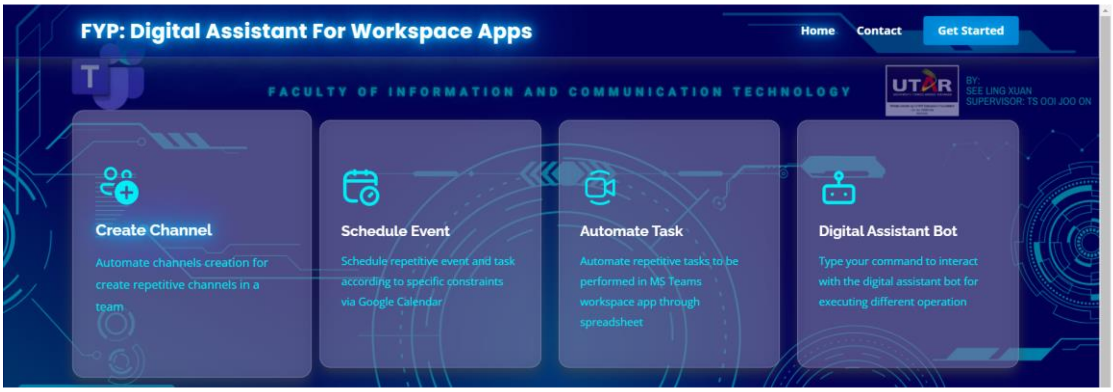
Automate Task
Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Digital Assistant Bot
Type your command to interact with the digital assistant bot for executing different operation

Contact Me By
GitHub User: lx-see
Email: lingxuansee1225@1utar.my
UTAR | Information Systems Engineering (IA) Student

Figure 4.11 Contact Page

By clicking on the Contact in the navigation bar or the rightest card in the second row of the project's functions, user can be navigated to the contact page and see respective contact information.



FYP: Digital Assistant For Workspace Apps

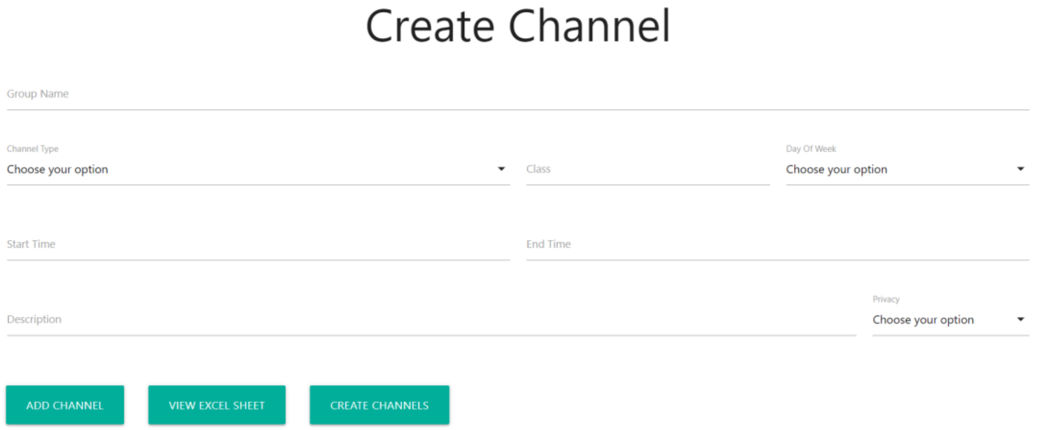
BY:
SEE LING XUAN
SUPERVISOR: TS OOI JOO ON

Create Channel
Automate channels creation for create repetitive channels in a team

Schedule Event
Schedule repetitive event and task according to specific constraints via Google Calendar

Automate Task
Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Digital Assistant Bot
Type your command to interact with the digital assistant bot for executing different operation



Create Channel

Group Name

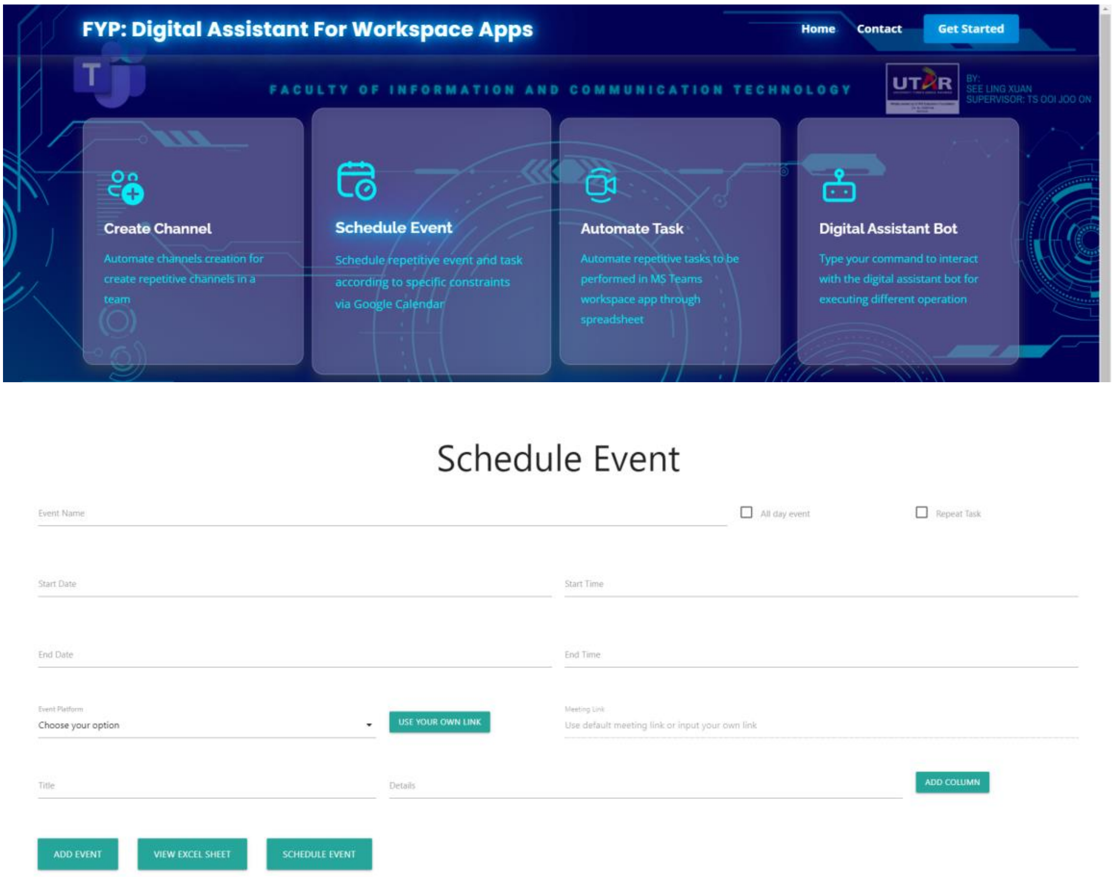
Channel Type: Choose your option
Class
Day Of Week: Choose your option

Start Time
End Time

Description
Privacy: Choose your option

Figure 4.12 Create Channel Page

User can be navigated to the create channel page that is the solution of the first use case planned by clicking on the first card in the homepage. The page consists of a form for user to input respective data needed to automate the steps to create repetitive channel in a MS Teams group created. Users can click on the three buttons in the webpage to add the data of new channel to be created into the spreadsheet, click the view excel sheet button to be navigated to the excel sheet, and click on the create channels button to run the automation process to create all channels according to each row data in the spreadsheet. The function of this create channel and every button exist in the webpage will be tested in the next chapter of system testing.



Automate channels creation for create repetitive channels in a team

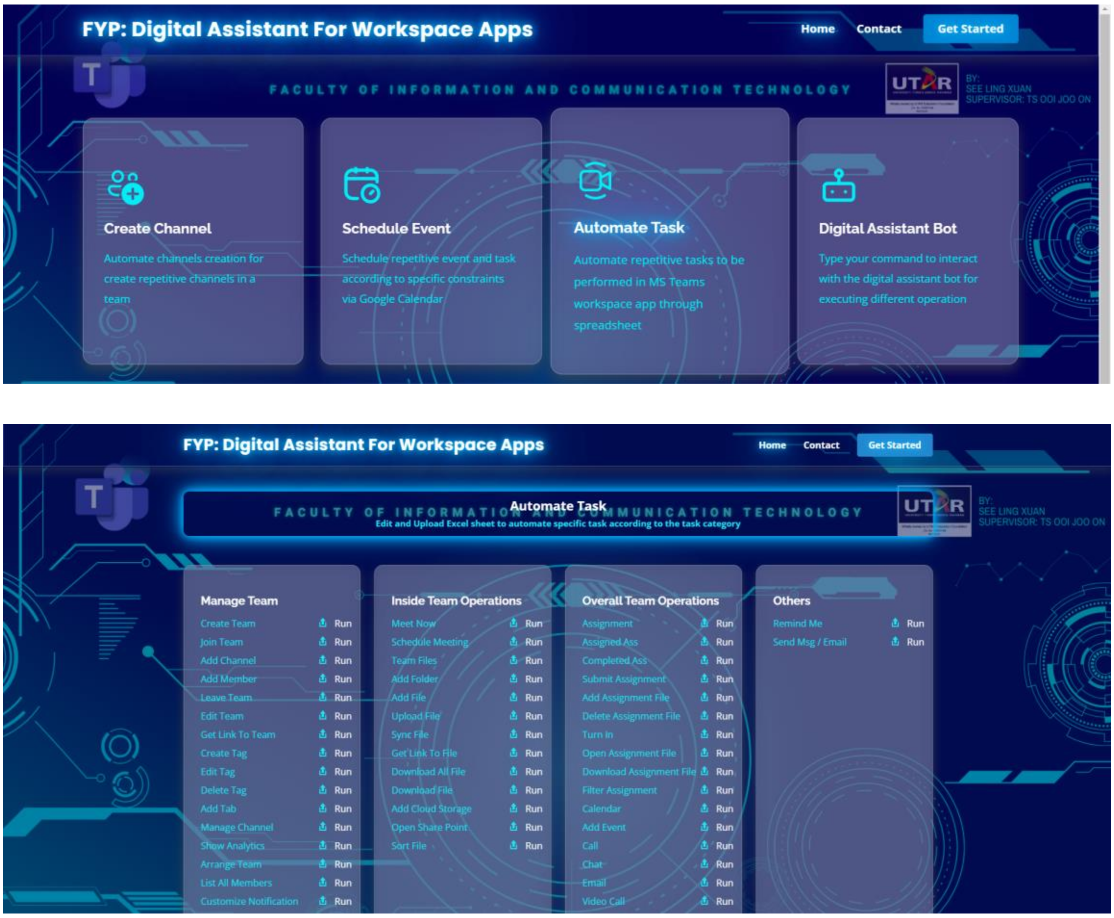
Schedule repetitive event and task according to specific constraints via Google Calendar

Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Type your command to interact with the digital assistant bot for executing different operation

Figure 4.13 Schedule Task Page

User can be navigated to the schedule event page that is the solution of the second use case planned by clicking on the second card in the homepage. The page consists of a form for user to input respective data needed to automate the steps to schedule repetitive events and tasks in Google Calendar. Users can click on the three buttons in the bottom of the webpage to add the data of new schedule to be created into the spreadsheet, click the view excel sheet button to be navigated to the excel sheet, and click on the schedule event button to run the automation process to schedule all events into Google Calendar according to each row data in the spreadsheet. The function of this schedule event and every button exist in the webpage will be tested in the next chapter of system testing.


FYP: Digital Assistant For Workspace Apps
Home
Contact
Get Started
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UTAR
BY:
SEE LING XUAN
SUPERVISOR: TS OOI JOO ON

Create Channel
Automate channels creation for create repetitive channels in a team

Schedule Event
Schedule repetitive event and task according to specific constraints via Google Calendar

Automate Task
Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

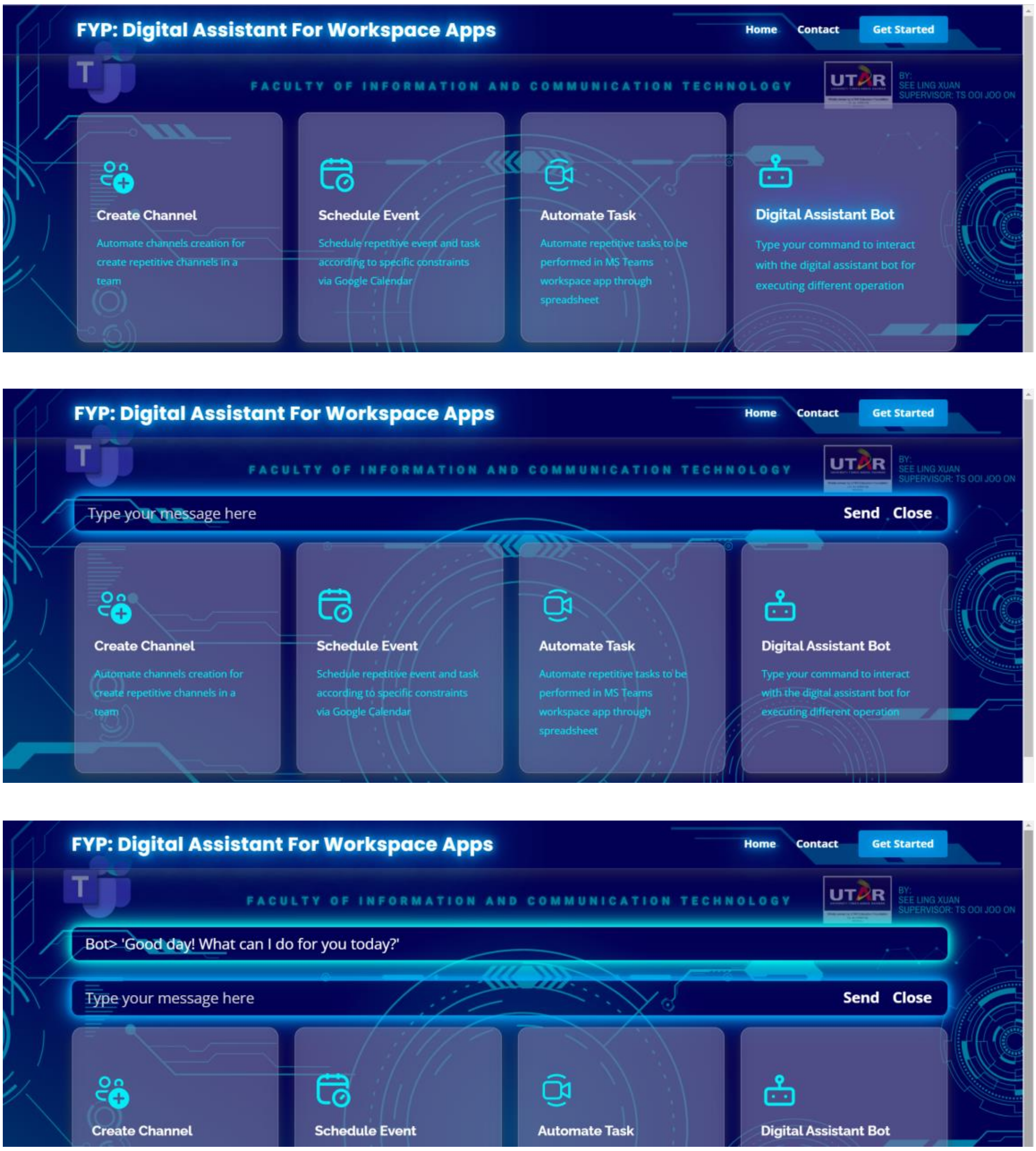
Digital Assistant Bot
Type your command to interact with the digital assistant bot for executing different operation

Automate Task
Edit and Upload Excel sheet to automate specific task according to the task category

Manage Team		Inside Team Operations		Overall Team Operations		Others	
Create Team	Run	Meet Now	Run	Assignment	Run	Remind Me	Run
Join Team	Run	Schedule Meeting	Run	Assigned Ass	Run	Send Msg / Email	Run
Add Channel	Run	Team Files	Run	Completed Ass	Run		
Add Member	Run	Add Folder	Run	Submit Assignment	Run		
Leave Team	Run	Add File	Run	Add Assignment File	Run		
Edit Team	Run	Upload File	Run	Delete Assignment File	Run		
Get Link To Team	Run	Sync File	Run	Turn In	Run		
Create Tag	Run	Get Link To File	Run	Open Assignment File	Run		
Edit Tag	Run	Download All File	Run	Download Assignment File	Run		
Delete Tag	Run	Download File	Run	Filter Assignment	Run		
Add Tab	Run	Add Cloud Storage	Run	Calendar	Run		
Manage Channel	Run	Open Share Point	Run	Add Event	Run		
Show Analytics	Run	Sort File	Run	Call	Run		
Arrange Team	Run			Chat	Run		
List All Members	Run			Email	Run		
Customize Notification	Run			Video Call	Run		

Figure 4.14 Automate Task Page

User can be navigated to the automate task page by clicking on the third card in the homepage. This webpage listed all functions to be performed by the digital assistant where some of the functions can be automated according to spreadsheet data such as to automate the process to create multiple teams while some of the functions cannot be performed as repetitive task such as the get link to team operation to only get link for specific team. Users can click on the function name they want to automate the task and an alert box will display if the task cannot be automated through spreadsheet. Else, user can download a spreadsheet with defined columns as parameters needed by respective function by clicking on the button with function name. After add record into the spreadsheet according to the defined columns for every spreadsheet, users can click on the button with upload icon to upload the spreadsheet to the system and click on the run button to run the automation process. Each row in the spreadsheet will be used to automate user's selected task such as to join into multiple teams and add multiple members into one or multiple teams. The detail of it will be tested in the next chapter.


FYP: Digital Assistant For Workspace Apps
Home
Contact
Get Started
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UTAR
BY:
SEE LING XUAN
SUPERVISOR: TS OOI JOO ON
Send
Close

Create Channel
Automate channels creation for create repetitive channels in a team

Schedule Event
Schedule repetitive event and task according to specific constraints via Google Calendar

Automate Task
Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Digital Assistant Bot
Type your command to interact with the digital assistant bot for executing different operation

Type your message here

Bot> 'Good day! What can I do for you today?'

Figure 4.15 Homepage: Digital Assistant Bot

The digital assistant bot function allows user to interact with the digital assistant by clicking on the fourth card in the homepage. After that, an input box will display for user can input their command or query and send to the bot by clicking on the send button or press the enter key. Another text box will then show above the user’s input message box to display the digital assistant’s response to the user. Through the chat box interaction, user can request the digital assistant to perform MS Teams function or perform other operations as implemented in the System Implementation chapter. Also, user’s advisory steps can be obtained from this chat box interaction. The details of it will be further discussed in the next chapter of system testing.

4.9 Implementation Issues and Challenges

4.9.1 Implementation Issues and Challenges

The implementation issues and challenges will be described for both preliminary work and FYP 2 work as different modules are developed in separated phase. In the preliminary work, we had faced some implementation issues and challenges to develop the project for achieving the first and the second objective. First of all, the issue includes the difficulty in integrating the multiple components of the digital assistant. For example, it is a challenge to integrate all components from different solution including conversational bot from Google solution, RPA sequences from UiPath, MS Teams from Microsoft solution, Python project in Python language, and the web application in HTML language. Due to the reason that all components to build the digital assistant are from different programming language, structure and platform, different integration method need to be figured and implemented for the successful project integration and for the sake of the digital assistant can perform its functions properly by linking and accessing all components of it. Therefore, the issue is time consumed and effort consumed to search the suitable way to integrate all components of the bot developed.

Another issue faced in the implementation process is the difficulty to increase the accuracy of the analyzation of user's command performed by the conversational bot developed in Dialogflow. This is because no 100% accuracy can be ensured for the bot to categorize user's intent and extract every user's input parameter from user's text correctly. The user's text analyzation in this project was conducted by the conversation bot built and trained by using Dialogflow with built-in NLP ability. However, we can only train the conversational bot by adding more user's expression that can lead to the specific intent. Therefore, if user inputted some expressions those are not used to train the intent or user's input has implicit meaning which is difficult for the bot to analyse and categorize, wrong intent will be categorized to and wrong result to be returned to our Python project. In result, wrong operation will be performed by the digital assistant to lower the assistant's accuracy. Besides, the extraction of user's parameters is also an important challenge due to the possible issues of the bot cannot recognize certain parameters inputted by the user, it categorizes user's input parameter to a wrong entity in Dialogflow, and it recognize uncomplete value of parameters inputted by the user. In result, the operation to be performed by the digital assistant will also failed to be performed due to incorrect or insufficient parameters provided. Therefore, the challenge includes to improve the digital assistant's accuracy to perform certain operation based on user's command.

Lastly, another challenge that also be faced in the preliminary work is the difficulty to deploy the project developed. This is due to the reason that too many components and solutions from different platform and programming language are involved in building the digital assistant in this project to cause high coupling between each solution. The most important issue is one of the solutions used to develop the RPA process of this project is by using UiPath, which is a desktop application that has been installed to our desktop. As the RPA sequences created by using UiPath needed to be executed by the UiRobot.exe file of the UiPath programme, this can be the limitation of our clients need to install UiPath in their desktop device before they can use the digital assistant. In result, more time, effort, and desktop space will be wasted by our target users to install and access both UiPath and our digital assistant. Therefore, it is hard to find an efficient method to deploy our digital assistant for it can be easily installed or accessed by our target users to automate their tasks by using MS Teams workspace app.

Besides, some challenges also faced in the development during FYP 2 to build, train, and integrate the AI model for achieving our third objective for the digital assistant can provide advisory on steps. The issue faced including to design and plan for the data need to be collected and prepared to feed in the AI model's algorithm for it can run suitable algorithm and predict the functions can be performed by the user for saving their time and effort. The challenge is that hard to decide suitable features and columns of the data those can used for the prediction with higher accuracy and performance of the AI model. Therefore, a large amount of research is needed to investigate the suitable data with features in suitable data types and find out the way to collect or prepare the data such as to create own data as simulation data to train the model. The large amount of research needed can cause time needed in preparing the data than to train the model that can affect the project finished time. So, ensure the data preparation followed by the rest processes for AI model implementation can be completed by time will be the biggest challenge in the FYP 2 work development.

Moreover, another challenge involved in the FYP 2 implementation is the risk to implement the AI model. The risks including can the application of prepared data be really fed into the algorithm in the AI model and used to predict our desired output.

This is due to the data collected is based on our own evaluation and assumption so that the usefulness of the data cannot be fully ensured. Therefore, if the data prepared and applied into the algorithm cannot match our purpose to implement the AI model, extra time and effort may cost for re-preparation of the data to feed in the algorithm. This can cause our project implementation cannot met the project deadline and fail the project. Besides, another risk of the AI model implementation including the algorithm selected may not be the best algorithm to produce output with desired performance and accuracy. In result, we may need to select another algorithm and retrain the AI model that is time and effort cost with the risk of the data prepared may not match the requirement of the new selected algorithm. So, both data and algorithm selection are important in implementing the AI model for our work as wrong choice can cause high risk to the project that can decrease the whole project's efficiency and alter our planned project methodology and phases. So, high attention needs to be paid for choosing suitable data and algorithm to build the AI model to present the risk involved.

4.9.2 Proposed Solutions

The proposed solutions are also presented for solving the challenges and problems encountered in both preliminary work and FYP 2 work implementation to complete the entire project. First of all, to resolve the issue in preliminary work to integrate multiple components from different programming language, platform, and solution to build our digital assistant, different integration method was searched, investigated and implemented to integrate each component separately to our Python project. In result, our Python project as a centre core can link to each component integrated to it via suitable integration method. For example, Flask framework to integrate the web application with the Python project, Google APIs to integrate the conversational bot with the Python project, and Python command to integrate the sequences built from UiPath with the Python project.

Besides, the second issue is the accuracy of the conversational bot to analyse user's intent and parameters correctly should be increased. To improve the performance and accuracy for the digital assistant to categorize user's intent and extract parameters included correctly, a dialog box in the web application can be developed for user to specify the Teams operation they would like to perform and input necessary parameters for executing the operation. Through the UI interaction, user’s intent and parameters can be clearly identified to improve the accuracy of the command to be performed by the digital assistant. As the process for the bot to analyse user’s input in natural language to output analysed information is removed via the user to directly key in necessary information to the System UI and the information will be passed to the Python project for further processing, the digital assistant can surely to perform correct function according to the data inputted by the user. Moreover, input validation should be implemented in the System UI to ensure the digital assistant can collect correct data from users to perform specific operation.

Lastly, the last issue to be addressed for preliminary work is the challenge to deploy our digital assistant in a more effective way for users can install or access our digital assistant more easily. To resolve the issue, the proposed solution is to develop other function for our Python project can execute the RPA sequence created by using UiPath in xml file format to remove the necessary to execute the sequence by the UiRobot.exe file in the UiPath programme. Therefore, user do not need to install UiPath in their desktop for the sequence execution to automate Teams operation. For the secondary method if the first solution cannot be achieved, another way can be figured and implemented for the Python project to perform automation process on the MS Teams website via browser automation. To achieve the second solution, an online extension or a script should be developed or utilized for the website automate process can be performed to the MS Teams website according to the request from our Python project to use the extension or script to perform RPA process. In result, the elements in MS Teams website can be recognized and operated by the suitable extension or script developed or utilized to perform specific Teams function. Therefore, no installation of software is needed in our target user’s device as the project can be deployed as a web application to automate Teams operation via the RPA process to be performed on MS Teams website. The important point to be noted for this solution is that authentication needed before our digital assistant can perform operations on the MS Teams website.

Furthermore, to resolve the problem of hard to plan for the data needed for model training that we faced in FYP 2 implementation, the proposed solution is that to do more research, searching, and evaluation to find suitable data to feed in the algorithm of the AI model. By conducting more research through thesis and reference papers, we can gain more knowledge on what data is related to our desired output of recommended functions to be performed by users for their next steps. Besides from the paper, the MS Teams should also be evaluated and investigated in more details to find out the relationship between functions those can be performed in MS Teams. Through the evaluation on MS Teams functions, user can know what function occurred or triggered by user has higher probability to cause another function to be performed. The relationship between MS Teams functions is important so that the digital assistant bot developed can predict user’s next functions according to their history functions had performed by investigate the relationship between those functions. To increase the accuracy and performance of the AI model, data collection is important so time and effort is recommended to be spend more in this stage than the model training part to ensure the project can be completed by time. In result, the first solution to this challenge encountered in the FYP 2 development is that to do more research and evaluation to find out the best data to be used for the model training with the correct data type that is supported by the algorithm selected. Else, data pre-processing is needed to convert the data into suitable data type that the AI algorithm can consume, run, and produce results.

The last problem to be solved is the risk to implement the AI model. To resolve this problem by reducing the risk involved, a detail planning should be conducted before making the decision of data prepared and algorithm selected for the model training. Through the planning process before actual implementing the AI model, the risk can be highly decreased due to the planning analyse the strengths and limitations of each algorithm to be selected to choose the best algorithm that can produce the best result that match our expectation. Also, to investigate the requirements of different algorithm can be run to get our desired output to prevent the data we collected is not match the algorithm requirement. Therefore, the extra pre-processing part of the data collected can be avoided to save project implementation time and ensure the implementation process can met our deadline. Moreover, as the algorithm was planned and investigated before actual implementation, in increase the accuracy and performance of the algorithm on data collected as the selected algorithm will be the best selection after comparing different algorithms in the planning phase. In conclusion, the comparison between different algorithm that can used in the model training should be performed and evaluate them before select one of them to carry out the model training process. Also, the data preparation should collect data according to the algorithm’s requirement to save time and effort in developing the AI model.

Chapter 5

System Evaluation and Discussion

In the fifth chapter, the system testing plans and performance metrices designed will be presented to evaluate the system. Moreover, the processes to setup the testing are elaborated, and the results obtained are presented. Last but not least, the objectives evaluation and project challenges will also be included in this chapter's explanation to evaluate the whole project's performance and its challenges.

5.1 System Testing and Performance Metrics

5.1.1 System Testing Plan

This section presents several testing plans develop in this project and the performance metrics designed to evaluate the system performance after completed. First of all, the system testing plan including the plan to conduct unit testing, integration testing, system testing, acceptance testing, and testing based on performance metrics will be discussed.

The first testing to be conducted in this chapter is the unit testing that will test the smallest testable parts of the application developed at function level. Each function developed in the Python program will be tested to ensure it can receive correct number and type of parameters from the function calling it and return suitable type of object back to the function who calls it. Moreover, each component in the web UI will also be tested by using unit testing to ensure each web component can work well as expected to perform its function. For the unit testing, black box testing will be conducted to both websites developed as the front-end environment and the function developed in the Python code for the back-end environment. Black box testing is a testing method derived from unit testing to test the system developed with no prior coding and implementation knowledge to the internal part of the system. The advantage of it is that by conducting black box testing, the system including the front-end web components and back-end function can be tested without predefined knowledge to ensure even a new user to use this system can understand how to use the system. Besides, the new user to use the system can perform every operation properly and get the desired output that should be provided by the system.

There are several techniques including in the black box testing such as boundary value analysis, equivalence partitioning, orthogonal I array testing, fuzz testing, graph-based testing, all-pairs testing, state transition testing and others. The techniques of black box testing that will be applied in our project will be the boundary value analysis and equivalence partitioning for checking the system's errors by inputting data of different boundary and partition in the forms those will provided in the system's webpages. In result, to test can every input in the provided forms work properly to avoid accepting unsuitable input that may destroy the system function or causes exception. Besides from these techniques, all inputs inside the proposed digital assistant will also be tested with different type of input such as text, number, etc. to investigate the accuracy of every input component displayed on the webpage. By conducting the black box testing, our system's web UI should be fully tested to ensure every web element performed as expected to provide expected output under different condition. Moreover, error messages should display if user inputs wrong type of data required or empty data that cannot be proceed by the system for the next operation. Therefore, the system's ability to treat unsuitable input will also be tested by using the black box testing technique.

Furthermore, after all elements on the website developed were tested by using the black box testing technique, each function developed in the back-end environment will be run one by one by providing it with required parameters provided for unit testing purpose. If any of the function developed cannot be run or cannot produce desired outcome as expected, bug will considered be found and solutions must be figured out and performed to solve the discovered bug so that the function can run as expected. This is due to the reason that every function we develop in the back-end environment is vital to perform the system's main functions and features and to interact with the front-end web environment by accepting and processing user's request from the web application. Therefore, each function should be carefully tested to ensure no error or exception can be occurred in different situation to run the function. For example, during the situation that if two or more than two functions are related and have high relationship or correlation. Therefore, the unit testing should ensure every function can perform as expected and one function's fail will not affect its related function or else, the whole system flow will be destroyed, and the system cannot continue to perform its desired operations.

The unit testing method test every component inside our proposed system including its web elements and the back-end functions developed to ensure every single element is working well on its own. After that, integration testing is planned to be conducted to run the software testing to the integrated system by combine some units of the system into a module or system’s function that required several function units. The module built by combing different unit or single function of the system will be tested to discover any error occurs when two single unit that work well separately are combined and work together. If any bug is discovered during the integration system such as after the unit combination, the combined units of certain module in the system occur bug and exception is discovered, solutions must be planned and performed to solve the error to finally produce every integrated module with no errors. This testing method is to ensure that the system can perform every module and function in combined units that is error-free and can produce desired outputs. After the modules are tested and ensure no errors, system testing will then be performed to test the whole system as a complete software.

The integration process of the whole system is described in the system implementation part to integrate different part and component of our system into a whole complete software. After the integration process, system testing will be conducted to now only a single element or module in the software, but to the entire system to discover the error exist after system integration. For example, the connection between each unit and module inside the integrated system need to be tested to ensure every unit and module is connected well and one unit can access other unit through function of APIs developed. Therefore, the system testing is important to ensure the system can perform with no errors in whole structure and after the units and modules were connected. If error found when testing the whole software, the errors need to be solved as soon as possible so that a complete error-free system can be delivered as the system output.

These are the three formal testing methods to be applied in our system testing including the unit testing, integration testing, and system testing. The testing process is conducted step by step to first test the unit, combined units, and finally the whole integrated system. After that, the acceptance testing will also be conducted in our proposed system to determine if all system’s requirements are satisfied and can work as expected. The requirements will be compared with the system’s outputs to ensure each requirement can be achieved by specific or combined functions of the system. The functions designed for achieving the system's objectives should be achieved and can run by the user to product expected outputs. The acceptance testing will be performed to evaluate all requirements of the system be developed properly with desired outcome produced after accurate processing. Therefore, by conducting the acceptance testing, the system can be delivered with all requirements satisfied and also satisfy our three objectives planned.

5.1.2 Performance Metrics

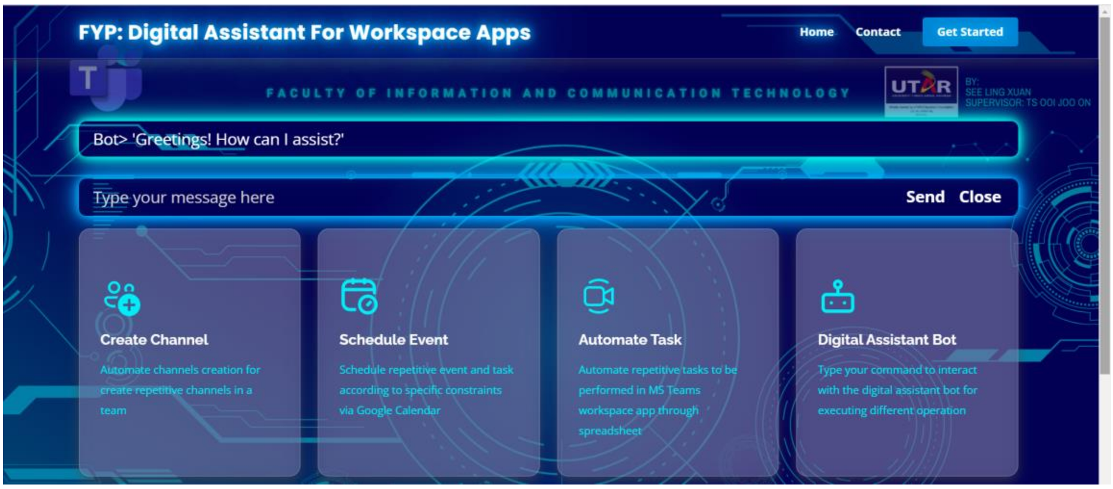
Software performance metrics are metrics to measure and evaluate a software's performance based on many factors such as CPU usage, average response time, error rates, etc. In our project, some performance metrics are prepared to test the system based on the performance metric for evaluating the digital assistant's performance after completion of the project. The performance metrics to be applied in the project testing including first, the average processing and response time of the system. This metric can also be used to indicate the user's satisfaction rate as user satisfaction increases as the average processing time of the system to decrease. To measure the system response time, the timestamp to start the system's operation and when the operation is completed will be recorded and the response time can be calculated as the difference between the start and end time. The same steps to measure the system response time for different system's function will be conducted for three times to calculate the average processing and response time for the system. Therefore, to increase the accuracy of this performance metric used to evaluate the system.

The second performance metric used to evaluate the system performance will be the CPU usage to determine the responsiveness of an application. If the system tested with different function that can be performed produce testing result of high CPU usage, it may indicate the system's function is performed in a high CPU usage threshold situation and the computing time and CPU resources are used more than expected. Therefore, the system's responsiveness may be decreased if the result collected for the metric is too high.

Finally, the memory usage will also be used as one of the metrics to evaluate the system's memory usage. This is also one of the important performance metrics used for system evaluation as the lower memory usage of the system indicate the less resource consumption of our digital assistant proposed. Besides, to decrease the risk of page faults those maybe occurred every time the system needs to access the memory consumed on the disk and a system with lower memory usage can decrease the disk access time for the system can perform any function. In conclusion, we will use average processing and response time, CPU usage and memory usage as our system's performance metrics to evaluate the system performance based on the metrics selected.

5.2 Testing Setup and Results

5.2.1 Unit Testing



FYP: Digital Assistant For Workspace Apps

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

BY:
SEE LING XUAN
SUPERVISOR: TS OOI JOO ON

Bot> 'Greetings! How can I assist?'

Type your message here

Create Channel

Automate channels creation for create repetitive channels in a team

Schedule Event

Schedule repetitive event and task according to specific constraints via Google Calendar

Automate Task

Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Digital Assistant Bot

Type your command to interact with the digital assistant bot for executing different operation

Figure 5.1 Homepage

In the first part of the unit testing, each web element inside our webpages will be tested as a single unit to ensure every element can perform as expected on its own. Therefore, every button and link existed in all webpages developed are clicked to test can it navigate user to the correct webpage or trigger certain web element to show. For example, each navigation element on the navigation bar including the Title, Home, Contact, and Get Started button is clicked to be navigated to the webpage of homepage, contact page, and the MS Teams login page for authentication purpose. After that, every title that is a link for the eight cards developed as the main content of the homepage is clicked one by one to be navigated to respective webpage. For example, by clicking on the title of the Create Channel card, user can be successfully directed to the respective webpage of create channel and perform operations. For the fourth card of digital assistant, by clicking on it, a text box is shown successfully for user to input their command to the bot and by clicking the Close button, user can successfully close and hide the box if the user do not want to interact with the digital assistant anymore. If the user clicks on the Send button after inputting certain message, the text box consist of bot's reply will be displayed so that user can know the bot's response. Based on our testing, if user didn't input any messages in the input text box and click on the Send button, his or her message will not be sent to the bot for processing due to empty message discovered.

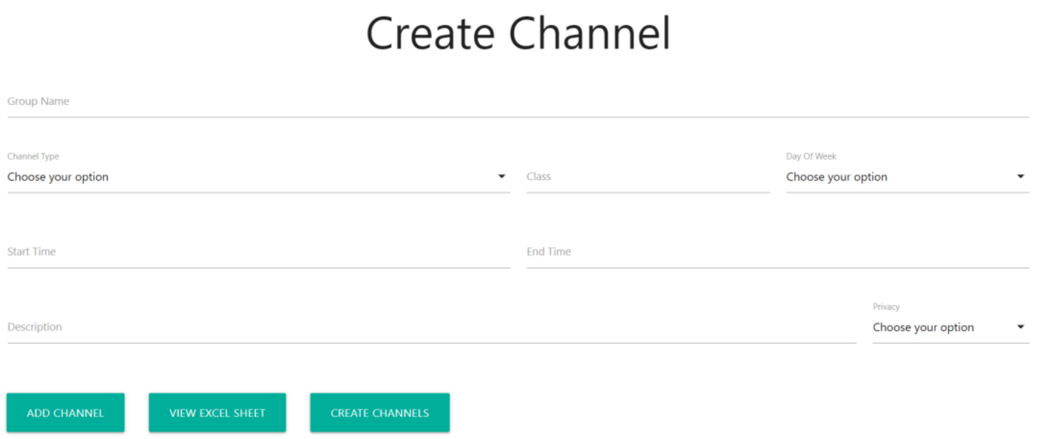


Figure 5.2 Create Channel Page

By clicking on the first card to be navigated to the Create Channel webpage, user can enter the above webpage to automate their task for creating repetitive channels in a specific MS Teams' team created. As this webpage consist of a form for user to fill in respective data needed and three buttons to conduct different purpose, each element inside the form is tested to know can it accept correct input from the user. For example, if user input empty message to the necessary data input element, an alert message will display to remind user to fill in their data before adding new channel into the spreadsheet. Moreover, the start time and end time elements can only accept user's input of correct time format after testing and for the channel type, day of week and privacy, user is now allowed to select other options that is not included in its default options to prevent manual error. The testing result of the unit testing by inputting data into every input column in the form indicate that each input element can accept data in correct data type.

The three buttons below are also tested one by one to add a new channel information into the spreadsheet by clicking on the first button. The new channel information consists of all data filled in by the user and to view the added channel, user can click on the view excel sheet button to be navigated to the spreadsheet that include all user's added channel information. Therefore, by clicking on the third button, user can automate the task of channels creation by the digital assistant will create channels according to every channel added to the spreadsheet. After conducted the unit testing, the result show that all element in this webpage works as expected where correct information can be inputted into each input element and the three button can perform its operation for different purpose.

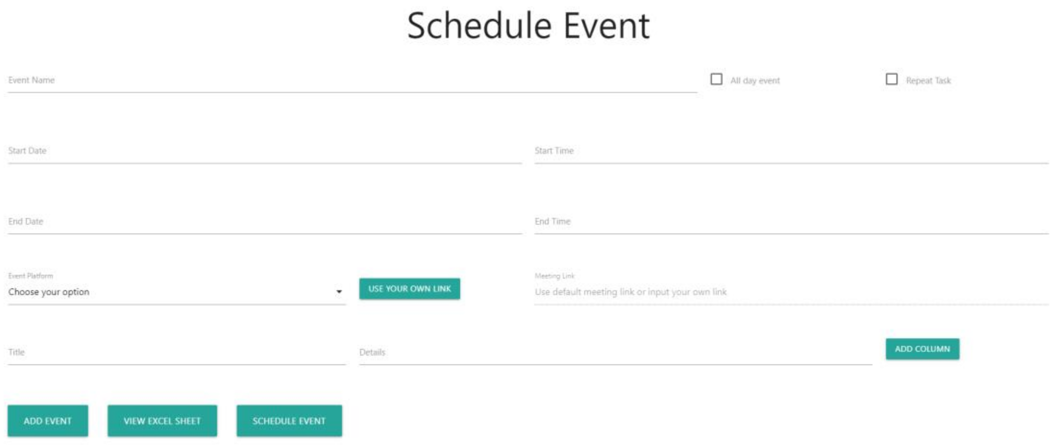


Figure 5.3 Schedule Event Page

By clicking on the second card to be navigated to the Schedule Event webpage, user can enter the above webpage to automate their task for creating repetitive events in Google Calendar. The webpage also consists of a form that including several input elements for user can fill in data in desired data format. Moreover, two buttons are including in the form to be tested. Based on our testing, the first button of use your own link can be clicked after user had chosen certain event platform to enable the user to input their own meeting link in the right meeting link input box. If user choose not to click this button, the meeting link input box will not allow user to edit the column and input any data. After that, the add column button is tested by clicking it and a new column including the input boxes of title and details will be added to the page for user can input more event title and details that user would like to describe the event's purpose. This webpage is tested as mostly the same steps to test the previous Create Channel webpage to test every single input element inside the webpage and each button in the bottom of the webpage. According to our testing result, user can successfully add event, view excel sheet, and schedule event via the three buttons located at the bottom part of the webpage.

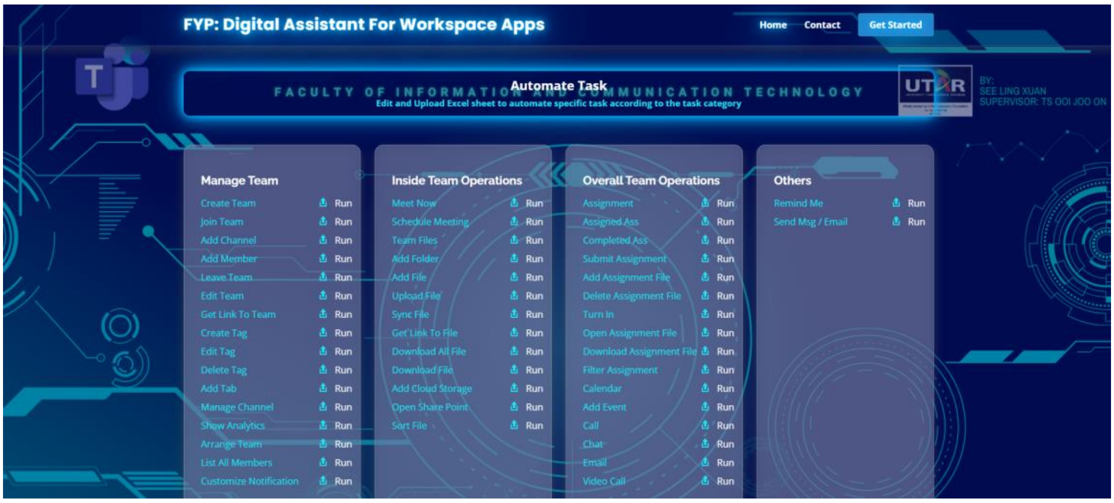
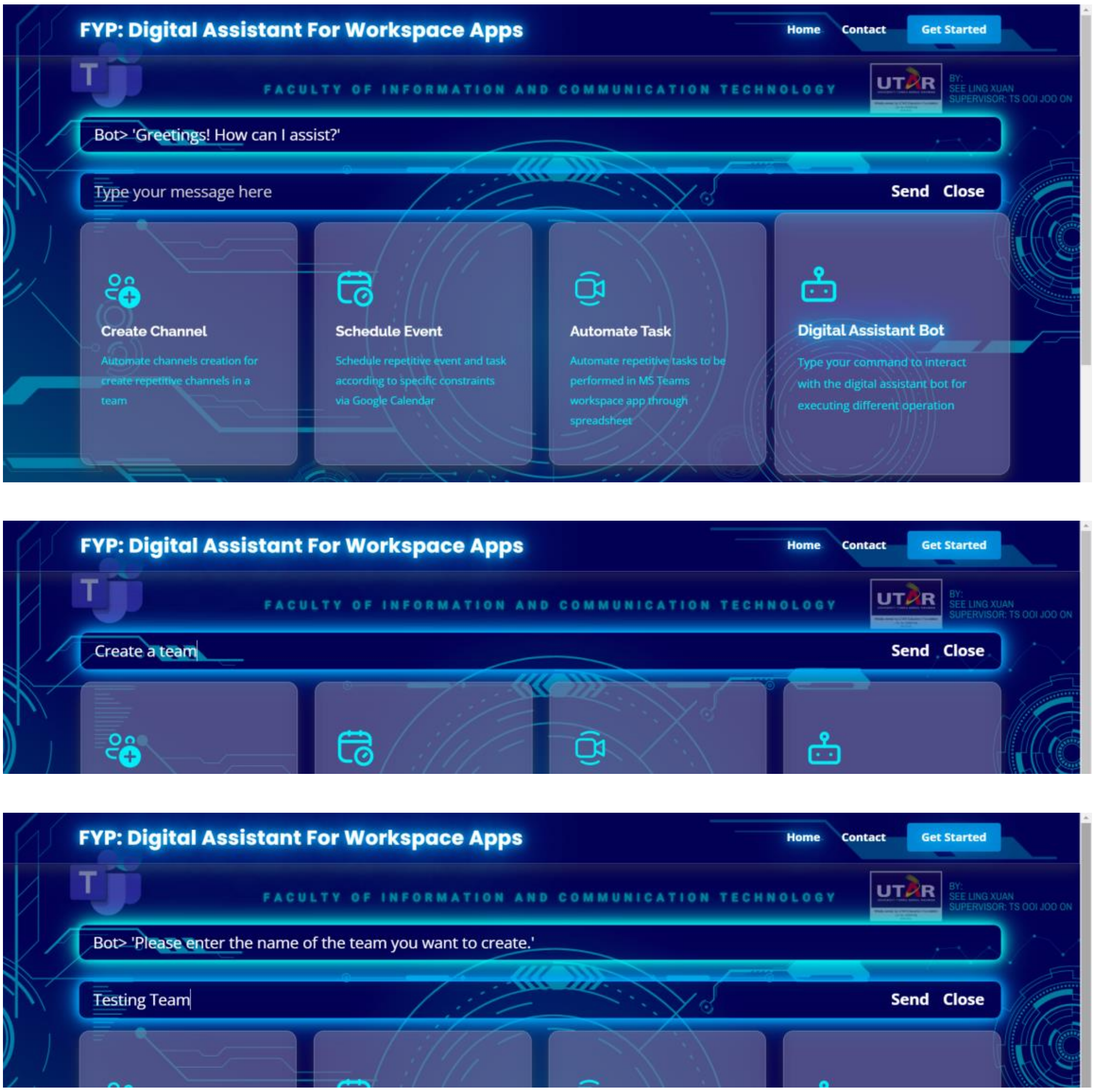


Figure 5.4 Homepage: Automate Task

By clicking on the second card to be navigated to the Automate task webpage, user can enter the above webpage to select and automate different repetitive task that can be automated in MS Teams such as to create multiple channels, join into multiple teams, and add multiple files to a specific team. In this page, the function listed in each category is created such as the Create Team under the Manage Team category. By clicking on the element, user is able to download a spreadsheet with no records but with defined columns that indicate the parameters user need to fill in to run the automation of the Create Team function. Therefore, to test the automation feature, we fill in each data row according to the columns stated and click on the upload button with the upload icon beside the function name to upload the filled spreadsheet. After that, we tested the Run button by click on it so that the automation function can perform according to each row we have filled into the spreadsheet and uploaded it. For example, if we have inserted three rows into the spreadsheet for create multiple teams in MS Teams with the necessary parameters of team name and team privacy, by clicking on the Run button, the digital assistant can auto create these three teams in the MS Teams according to each row's data.


FYP: Digital Assistant For Workspace Apps
Home
Contact
Get Started
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UTAR
BY:
SEE LING XUAN
SUPERVISOR: TS OOI JOO ON
Send
Close
Create Channel
Schedule Event
Automate Task
Digital Assistant Bot

Bot> 'Greetings! How can I assist?'

Type your message here

Automate channels creation for create repetitive channels in a team

Schedule repetitive event and task according to specific constraints via Google Calendar

Automate repetitive tasks to be performed in MS Teams workspace app through spreadsheet

Type your command to interact with the digital assistant bot for executing different operation

Create a team

Bot> 'Please enter the name of the team you want to create.'

Testing Team

Figure 5.5 Homepage: Digital Assistant Bot

The digital assistant bot function is tested to ensure user can input their message into the text box showed. According to the unit test result, the bot will give different response back to the user according to user’s command such as if user request the bot to create a team, the bot will request user to give necessary parameters and help user to create a new team in MS Teams. Also, the digital assistant can provide advisory on user’s steps via the same chat box if user didn’t include an obvious intention in his or her context that can easily categorized into a specific function by the system’s NLP ability. This feature is also tested in our system testing phase.

Besides from the web element to be tested one by one, the back-end Python functions are also conducted respective unit testing for testing each function can be run successfully with no exception. Moreover, each function should produce desired output as planned. The function developed and tested in the Python project including the functions developed to handle the front-end web requests by using the Flask framework such as to render certain webpage to the web UI according to the URL user had access to. Besides, other Python functions also developed in the project to process user's request, get specific intent and parameters needed through the conversational bot built with NLP ability, run specific RPA sequence on the website of MS Teams, performing the AI model algorithm for suggesting recommended steps, and handle other logic required for the system flows and functions as designed. In result, the unit testing has conducted for each front-end and back-end environment in our project and ensure every single item can work properly.

5.2.2 Integration Testing, System Testing, Acceptance Testing

The integration testing conducted in this project including to test the integrated units such as the forms including in both Create Channel and Schedule Event page that is the combination of many web input elements. Therefore, the forms of combined input elements will be treated as a single element to be tested in the integration testing for testing that can the form send all data including in it to the back-end Python code for it to carry out respective function and provide results. Moreover, several functions are combined to be tested as a module to perform the digital assistant functions such as to run the RPA sequence for performing operation on MS Teams. The functions combined to test the system's ability to perform single operation on MS Teams including the function to send the user's query text to conversational bot built with NLP ability to obtain user's intent and parameters needed, the function to choose and run specific RPA sequence with user inputted parameters according to user's intent analysed, and other related functions. After the integration testing conducted to test each system's function implemented in unit combination, the overall testing is performed to test the whole system after integration to ensure every system flow can be conducted as expected and every system's function is working well. Finally, through the acceptance testing, we evaluate is the implemented system can be accepted by the user by satisfying all of our system's requirements and objectives according to our planning and system design. This part will be further explained in the following section of objectives evaluation to evaluate the system's satisfaction to its requirements and ensure it pass the acceptance testing by achieving all of our objectives before it can be delivered to its end user in the future. After that, the next section perform the testing result of our system based on the performance metrics as selected to evaluate the system's performance on its processing and response time, CPU usage, and memory usage.

5.2.3 Test Result Based on Performance Metrics

Table 5.1 Test Result Based on Performance Metrics

Performance Metrics	**Tested Function**	**Round 1 (s)**	**Round 2 (s)**	**Round 3 (s)**	**Average time (s)**	**Manual time (s)**
System Response Time	Single MS Teams operation: Create team	15.315	16.899	16.991	**16.402**	**18.699**
	Automation: Create multiple teams (3)	54.088	62.037	53.112	**56.412**	**60.135**
CPU Usage (Start the app)	**8.1%**					
CPU Usage (After running for 3 minutes)	**2.0%**					
CPU Usage (Running function)	**45.0%** (The function of hide team is run. The percentage including 24.9% by Microsoft Edge as the browser, 0.6% by MS Teams, 17.9 by IDE of PyCharm to run the project, and 1.6% by UiPath to run the RPA sequence.)					
Memory Usage	**330 MB** (Including 20915 files and 2435 directories according to the File Manager information)					

5.2.4 Explanation on Testing Result

The table used to present the system's performance under some defined metrics such as first, the average system response time is measured and calculated to compare the time costed by the digital assistant and the time to manually performing the necessary steps. Based on the result, we can see that for the single MS Teams operation for the example operation to create a new team, the average system response time to perform the operation by using our digital assistant is less than the time for us to manually create a team on MS Teams. This is because the digital assistant which run the RPA sequence to automate the process can type in the parameters needed more quickly than manual typing. Therefore, to prove that the usage of the digital assistant developed can save user's time in performing MS Teams' operation for both single and multiple operations. For the automation process of repetitive work such as to create multiple teams, the time of the digital assistant is still less than the human's steps to manually create each team in the MS Teams due to the same reason of machine have higher typing speed. However, the difference between the assistant's time and manual time consumed to perform the operation is not very high due to RPA sequence is used for performing the operations by the digital assistant. Therefore, the original processing speed of MS Teams to the assistant's action to simulate human's action will not be affected. Besides, the CPU usage and memory usage of the system consider not very high and still in the performance that can be accepted. However, the CPU usage during the digital assistant to perform MS Teams operation is considered a little high but still less than half of the whole CPU usage. The CPU usage data is collected by combining the CPU usage of the IDE to run the application, UiPath to perform the task, and the web browser to run our web application and interact with MS Teams according to the task manager. Moreover, the memory usage of the project combines the memory consumed for the Python project structure and the RPA sequences built in UiPath.

5.3 Objectives Evaluation

As a recap to our project objectives, the objectives in this project including to reduce steps required, automate repetitive tasks, and to include advisory of steps. The explanation will be described in each paragraph according to respective objective for evaluate is our work satisfy the objective we have planned for this project.

1. **To reduce steps required**

   Our project implemented can successfully achieve the objective one by reducing the steps required for user to perform operations on MS Teams. The operations including single operation and multiple operations in a workflow. According to our project's function implemented, the digital assistant had developed a chat function for user to input their command or query to the digital assistant, and the digital assistant can analyse user's intent and request them of the necessary parameters for performing the intended function. After that, the digital assistant can help to auto perform the MS Teams function by helping user to auto trigger each element needed in MS Teams and fill in respective parameters for completing the function intended. In result, user do not need to manually operate on the MS Teams to perform certain operation but only to type in their command and provides the digital assistant with necessary parameters to perform the operation. This proves that user's steps required to perform any MS Teams function had been simplified compared to their original steps to manually access the MS Teams app and run every step by their own. Our digital assistant developed assists user to automate the MS Teams operations

2. **Automate repetitive tasks**

   Our proposed solution and project implemented can satisfy the second objective by automating the repetitive tasks including repetitive task to be performed in MS Teams and the repetitive step to create schedules related to MS Teams on Google Calendar. Based on our use cases implemented and function developed for automate repetitive task, our second objective is achieved by allowing users to fill in respective spreadsheet according to the task they would like to automate, and the digital assistant developed can automate that specific task by read in every row data in the spreadsheet and use the data to auto perform the same task repetitively. In result, the automation feature implemented for the digital assistant allows user to save time and effort in performing repetitive tasks manually by only fill in respective spreadsheets for the task automation. Through the automation feature that successfully be developed for the digital assistant, user's productivity can also be highly increased by eliminating the repetitive steps and tasks in user's daily life.

3. **To include advisory on steps**

   The third objective is to include advisory on steps. According to our system implementation, an AI model is developed for the digital assistant can analyse and produce the advisory on user's possible next steps to perform. Therefore, this objective is achieved by training the AI model for handling this aim of the project. Moreover, based on our third use case proposed, the advisory on steps to create future possible schedules on Google Calendar can also be achieved by our digital assistant developed. Therefore, the respective AI model and function implemented had achieved this objective for user can have the bot's advisory to steps to perform MS Teams operation and create schedules those need to be performed in the MS Teams platform. In result, users can easily select the function they would like to perform even they have less knowledge and skill to the MS Teams by chatting with the digital assistant. Furthermore, users can easily create their schedules on Google Calendar based on the digital assistant's advisory on steps to save their time and workloads needed to create the future schedules.

In conclusion, we can conclude that our FYP had successfully fulfil all listed objectives of reduce steps required to perform MS Teams operations, automate repetitive tasks in MS Teams, and the digital assistant to provide advisory on steps to its users. In the next section, the challenges in implementing this project will be discussed followed by the last chapter will gives recommendations on future improvement of this project developed.

5.4 Project Challenges

First of all, the challenge faced in this project including the time needed for a lot of functions to be developed for the proposed digital assistant as planned to achieve our objectives. For example, the function of the digital assistant including to auto perform the functions those can be performed by user in the MS Teams manually via command only involve the implementation of a lot of intents and RPA sequences correspond to each function in our project development. Moreover, time is also needed to plan for the AI model implementation that involve data analysis, data collection and preparation, model training and so on. The collection of use cases in our project also considers as a work that consume time but is necessary to develop the feature of our system to be applied in real-life industrial case for its contribution and novelty. In result, the time management will be a big challenge in this project implementation to plan and implement each system's function in respective time phase as planned in the system methodology and to ensure the system can be completed and integrated by time.

The second challenge of this project is the integration problem as stated in the previous chapter's section of the implementation challenges and issues. This challenge also consumes the developer's time and effort for analysing and performing the way of system integration due to a lot of components involved in the system. Besides, the involved system components consist of both local installed application of UiPath and the web application such as MS Teams, Google Calendar, and the scripts developed in the Google Apps Script. Therefore, to think of a realistic and resource saving way for integrating different components in this project will be a challenge to be resolved in future so that the system integration can be done in a more cost and resource saving way.

The last challenge in this project is to increase the dynamicity and flexibility of the digital assistant to be also applied for and supported other workspace apps. This is also the project's challenge due to this project only focus on MS Teams workspace app and only allow the digital assistant to work for the MS Teams app only. More research and investigation are needed to be conducted in future to increase the solution's flexibility so that user can use the proposed project to simplify their workload in other workspace apps than MS Teams and increase their productivity.

Chapter 6

CONCLUSION AND RECOMMENDATION

This last chapter conclude and summarize the project by presenting its conclusion and provides recommendation on how to improve this project in future implementation and the advanced functions and features those can be added to this project.

6.1 Conclusion

Workspace apps have become a trend that grow rapidly on the moment to achieve seamless collaboration of employees and for the workers can work beyond physical boundary in a more productive and comfortable method. In workspace apps, the usage of AI technology to automate task in a workspace platform is becoming a concern topic to simplify worker's task and reduce repetitive tasks. MS Teams nowadays had become the most popular workspace app to be used by most of the organization, but it may not be enough efficient to maximize employee's productivity due to the tasks overload, tedious tasks and increasing functions to be used in MS Teams. Therefore, this project is motivated to implement a digital assistant for workspace apps to increase the user's efficiency and productivity to use MS Teams by solving the problems in MS Teams of way too many functions to be configured or used, tedious process to perform repetitive steps for similar task which to cause time wasted, and unsuitable functions used by users. The project's objectives are to reduce steps required by users to perform MS Teams functions, automate repetitive tasks and to include advisory on steps. Our contributions of this project are allowing employees who are using MS Teams workspace app can complete their tasks with simpler step and less time, able to automate repetitive tasks, and for the organization can earn more competitive advantages.

To achieve our objectives, reviews on MS Teams and similar works on digital assistants were conducted in Chapter 2 and summarized their functions, strengths, limitations, and recommendations to design our proposed project functions and features. In Chapter 3, we have planned the proposed method to implement the digital assistant by using XP technique in the agile methodology as our project methodology and created project timeline in Gantt chart based on the methodology designed. Besides, we have planned the suitable hardware and software can be used to develop our digital assistant and also, we have designed the system framework, use case diagram, system architecture diagrams, and system flow charts to visualize the system design. After that, we also presented the system components including system UI and its bot core with AI capabilities in the same chapter. Each system component is explained with its feasibility analysis to investigate the feasibility of proposed methods and system design. Next, we have implemented the functions including ML-related and non-ML related functions in Chapter 4. The system implementation is discussed and explained each step we have performed to implement the digital assistant for it can fulfill our desired functions and objectives as discussed in the previous chapters. In the chapter, we have performed the steps to create System UI, conversational bot, and RPA sequences as the bot components to be integrated followed by developed its elementary functions and functions to resolve real-life use cases after the integration was completed. Besides from those non-ML related functions, we also develop the AI related features of this system for its NLP capabilities and AI algorithm to be run and provide user advisory on steps. In the AI model development, we undergo data preparation, model training, model fine tuning, and integration processes to implement the rest function of our system. Some industrial use cases are also prepared and elaborated in Chapter 4 to implement the digital assistant that can solve real industrial's problems for increasing its contribution to society. Lastly, system results with screenshots followed by the implementation issues and challenges faces are also included in the Chapter 4.

In Chapter 5, the system testing, evaluation and discussion are presented so that our digital assistant developed can undergo different testing method to evaluate its performance. Based on the objectives evaluation conducted in Chapter 5, it discovers that we have successfully achieve all of our project's objectives of reduce steps required, automate repetitive tasks and include advisory on steps. This evaluation result indicates that our FYP can be concluded without problems as requirements are fulfilled but still, some of the recommendations can be suggested for future project improvement based on the possible novelty that can be implemented in this project. The recommendations suggested for the project improvement are explained in the next section.

6.2 Recommendation

First of all, the UI of the project can be improved such as to implement more interaction methods between user and the digital assistant. For example, the interaction methods such as through voice interaction and tap method can be implemented so that the digital assistant's accessibility can be increased. Furthermore, since the digital assistant developed in this project can only interact with user in English language as default, the future improvement on interaction method can also implement more language supporting feature to this project so that user from different country can interact with the digital assistant in different language. Besides from modifying and improving the web application UI as developed in the project, a desktop application can also be developed for this digital assistant can assist user when using the desktop version of MS Teams. Therefore, the UI for desktop application will need to be developed by using the elements provided by the .NET solution. In result, user will have more way to interact with the digital assistant and can perform their operations on both web version and the desktop version of MS Teams app. Moreover, Unity is also suggested to be used to improve the project's UI as the animation and modelling features supported by the Unity allows the project's UI implementation to be performed in a more attractive and interesting way for attracting user's attention and more inclined to interact with the digital assistant. For example, the digital assistant developed in 2D or 3D UI form by using Unity can have different interaction way with users when performing workspace app's function or interacting with the user so that the virtual assistant can be seems more like an actual human assistant to aid users in their work.

Second of all, although in this project, only MS Teams is selected as the workspace app to be mainly focus on the project development and implement the digital assistant, the solution can be built for other workspace apps too so that the digital assistant can aid user that is using other workspace apps or using multiple workspace apps in their work environment. As RPA process is selected to be used to implement this project with no MS Teams APIs are integrated and used, the same RPA solution can be applied for other workspace apps too in future by considering the advantages of RPA solution. However, the performance of RPA solution needs to be investigated and improved in future development as this technology is still not matured for now. By implementing and integrating the digital assistant into more workspace apps platform by using and improving the methods as stated in this project, the digital assistant will be able to fit more user’s requirement and save their workloads followed by increasing their productivity in using different workspace apps for their daily work and task. Therefore, the second recommendation is to implement the digital assistant in more workspace apps by developing respective RPA sequence for reducing people’s time and effort used to handle various intricate task in a workspace app and performing repetitive tasks.

Third of all, the last recommendation provided to improve this project in future is that the digital assistant can develop its AI feature in advanced by designing more columns can be used to feed in the AI model and continue improve its algorithm to provide more accurate advisory on user’s steps to perform. More attributes of the data can be investigated, collected, or created to improve the AI’s performance by modifying and retraining current model. Besides from increasing the data attributes and number of data used for the AI model training, the algorithm of the AI model can also be improved such as develop a new algorithm that can provide more accurate prediction result, or develop multiple AI model and integrate them to produce the best final result as the bot’s response to the user. These are the three recommendations suggested based on the product’s performance and evaluation to improve the project’s functions and features in future. In conclusion, the AI technology and RPA technology should be further improved in the future to fully automate human’s workload when accessing and using workspace apps in their daily life with high performance. Therefore, to accelerate and finally complete the digital transformation for people in every industry can conduct their task more productively, easily and in a more efficient way. That’s all to conclude the project.

REFERENCES

[1] *“*2021 Report: The State of the Digital Workplace*,”* *www-cmswire.simplermedia.com*. https://www-cmswire.simplermedia.com/cw-cp-smg-dw-report-dw.html

[2] “Digital Workplace Trends for 2021 - Anunta Tech,” *Anuntatech*, Jul. 28, 2021. https://www.anuntatech.com/media-and-press-releases/digital-workplace-trends-for-2021 [Accessed Nov. 21, 2021].

[3] “Google Assistant,” *Assistant*, 2019. https://assistant.google.com/

[4] “What is Artificial Intelligence (AI)?,” *www.ibm.com*. https://www.ibm.com/my-en/cloud/learn/what-is-artificial-intelligence

[5] A. Croatti and A. Ricci, “Mobile Apps as Personal Assistant Agents: the JaCa-Android Framework for programming Agents-based applications on mobile devices,” *Autonomous Agents and Multi-Agent Systems*, vol. 34, no. 2, Jul. 2020, doi: 10.1007/s10458-020-09474-7.

[6] A. de Barcelos Silva et al., “Intelligent personal assistants: A systematic literature review,” Expert Systems with Applications, vol. 147, p. 113193, Jun. 2020, doi: 10.1016/j.eswa.2020.113193.

[7] A. S. Tulshan, and S. N. Dhage, “Survey on Virtual Assistant: Google Assistant, Siri, Cortana, Alexa,” *Communications in Computer and Information Science*, 2019, pp.190–201.

[8] B. J. Copeland, “Artificial intelligence - Methods and goals in AI | Britannica,” *Encyclopædia Britannica*. 2019. [Online]. Available: https://www.britannica.com/technology/artificial-intelligence/Methods-and-goals-in-AI

[9] Berdasco, López, Diaz, Quesada, and Guerrero, “User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and Cortana,” *Proceedings,* vol. 31, no. 1, p. 51, Nov. 2019, doi: 10.3390/proceedings2019031051.

[10] D. Kent, “Best Microsoft Teams Features: 55 Gems We Use in 2020,” *News about Slack,* Microsoft Teams, Webex & Team Collaboration. https://dispatch.m.io/best-microsoft-teams-features/

[11] D. S. Zwakman, D. Pal, and C. Arpnikanondt, “Usability Evaluation of Artificial Intelligence-Based Voice Assistants: The Case of Amazon Alexa,” *SN Computer Science*, vol. 2, no. 1, Jan. 2021, doi: 10.1007/s42979-020-00424-4.

[12] D. S. Zwakman, D. Pal, T. Triyason and V. Vanijja, "Usability of Voice-based Intelligent Personal Assistants," *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 652-657, doi: 10.1109/ICTC49870.2020.9289550.

[13] K. Myers, et al., “An Intelligent Personal Assistant for Task and Time Management,” *AI Magazine,* 28(2), p. 47, Jun. 2007, doi: 10.1609/aimag.v28i2.2039

[14] K. Simon, “Advantages and Disadvantages of Microsoft Teams,” Profolus, Jul. 08, 2021. https://www.profolus.com/topics/advantages-and-disadvantages-of-microsoft-teams/

[15] L. Rathnam, “Microsoft Teams: Pros, cons—and should you switch from Slack or Zoom?,” *TechGenix*, Nov. 23, 2020. https://techgenix.com/microsoft-teams-pros-and-cons/

[16] LolaJacobsen, “Welcome to Microsoft Teams,” *Microsoft.com,* Jun. 18, 2019. https://docs.microsoft.com/en-us/microsoftteams/teams-overview

[17] M. Rouse, “What is AI (artificial intelligence)? - Definition from WhatIs.com,” *SearchEnterpriseAI*, 2019. https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence

[18] MSFTMan, “Learn to create and manage flows in Microsoft Teams - Power Automate,” *docs.microsoft.com.* https://docs.microsoft.com/en-us/power-automate/flows-teams [Accessed Nov. 16, 2021].

[19] N. Heath, “What Is AI? Here’s Everything You Need to Know about Artificial Intelligence,” *ZDNet,* Jul. 23, 2021. https://www.zdnet.com/article/what-is-ai-heres-everything-you-need-to-know-about-artificial-intelligence/

[20] T. Laureys, “The digital workplace in 2021: Five trends you can’t miss out,” *www.linkedin.com.* https://www.linkedin.com/pulse/digital-workplace-2021-five-trends-you-cant-miss-out-tom-laureys/ [Accessed Nov. 12, 2021].

[21] T. Shintani, T. Ito, & K. Sycara, “Multiple negotiations among agents for a distributed meeting scheduler,” *In Proceedings fourth international conference on multiagent systems*, 2000, pp. 435–436.

[22] W. Richmond, “Thomson Reuters Elite Launches Innovative Workspace Assistant Product,” *www.linkedin.com.* https://www.linkedin.com/pulse/thomson-reuters-elite-launches-innovative-workspace-product-richmond/?trk=articles_directory [Accessed Nov. 25, 2021].

[23] Zapier, “Microsoft Teams Integrations | Connect Your Apps with Zapier,” *Zapier*. https://zapier.com/apps/microsoft-teams/integrations

[24] UiPath Inc, “What is Robotic Process Automation? - RPA Software | UiPath®,” Uipath.com, 2017. https://www.uipath.com/rpa/robotic-process-automation [Accessed Sept. 6, 2022].

[25] "What is RPA | Robotic Process Automation Software," Automation Anywhere. https://www.automationanywhere.com/rpa/robotic-process-automation [Accessed Sept. 6, 2022]

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S1	**Study week no.: 2**
Student Name & ID: See Ling Xuan (19ACB01373)	
Supervisor: Ts Ooi Joo On	
Project Title: Digital Assistant for Workspace Apps	

1. WORK DONE

Based on the previous work in FYP 1 that had achieved the first and second objectives of the project, an overall idea to develop ML-related functions for achieving the third objective is generated. Moreover, the materials related to the FYP 2 are downloaded from the FYP portal. Based on the FYP 2 schedule, an overall schedule to conduct weekly log and to develop the rest functions of the product based on the project methodology is generated. Also, some existing solutions are reviewed and analysed.

2. WORK TO BE DONE

The reviews and investigation in the way to develop the AI feature of the product should be conducted to decide the way for training the AI model. Moreover, suitable solutions, tools and software should be found and explored to propose suitable methods to implement the project.

3. PROBLEMS ENCOUNTERED

Not enough use cases prepared for the FYP 2. More real-life use cases than FYP 1 should be prepared especially to collect some industrial use cases for implementing the complete necessary functions of this system and ensure the work produced can be useful in real-life examples.

4. SELF EVALUATION OF THE PROGRESS

Progressing smoothly. A rough idea is generated, and some solutions, tools, and software are explored to further analysis their strengths and limitations to be chosen to implement the AI feature left for the project implementation. More tools to be evaluated and more use cases to be prepared for future weeks.

Supervisor's signature   Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S1	**Study week no.: 4**
Student Name & ID: See Ling Xuan (19ACB01373)	
Supervisor: Ts Ooi Joo On	
Project Title: Digital Assistant for Workspace Apps	

1. WORK DONE

Different methods, tools, and frameworks such as PyTorch and TensorFlow to implement the ML feature of the system are investigated and evaluated based on their simplicity to use and other factors. After evaluation, the tool and method to develop the ML-related function of this project is decided and explored to be implemented.

2. WORK TO BE DONE

Since the tool to develop the ML-related function of this project is decided, the processes to develop the function should be investigated and planned to include data collection, AI model training, model fine tuning, and model evaluation. The data collection phase should be started for collecting or creating enough data to feed the AI model before actual building the model. Continue searching for suitable use cases.

3. PROBLEMS ENCOUNTERED

Time and effort consumed to plan the application of the selected tool and framework in the project development. Moreover, the time also consumed to prepare more real-life industrial use cases due to a lot of research needed to select suitable samples and use cases for this project.

4. SELF EVALUATION OF THE PROGRESS

Progressing smoothly. A rough plan was prepared for applying the selected tool and framework to implement the project's left function. The plan to implement the AI feature was created by separating it into several processes starting with the data collection. More use cases will be collected in the rest weeks to increase the project's contribution to the industry area. Also, the implementation of the AI technology in this project will be further planned and modified in future weeks.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S1	**Study week no.: 6**
Student Name & ID: See Ling Xuan (19ACB01373)	
Supervisor: Ts Ooi Joo On	
Project Title: Digital Assistant for Workspace Apps	

1. WORK DONE

Some of the necessary data needed for the AI model training is collected and created. The plan to implement the AI model step by step was prepared to build the model and integrate it into the system before the FYP 2 deadline. More detail system design is conducted by designing more diagrams including Use Case Diagram and System Flowcharts. The related frameworks and tools to develop the AI model in Python language are installed and integrated to the project develop environment.

2. WORK TO BE DONE

The data collected and created should be evaluated and start to build the AI model to set up the environment for the AI related function development. Also, the use cases collected should be planned in more details to design the demonstration and desired final results of the FYP work.

3. PROBLEMS ENCOUNTERED

Hard to evaluate the data needed to train the AI model and to collect or create suitable data for the AI model training. The data columns and features need to be planned in more detail to ensure the data used can successfully train the AI model to achieve the objective of the digital assistant to give advisory on user's steps.

4. SELF EVALUATION OF THE PROGRESS

Progressing smoothly. The AI model is starting to be developed by implementing its prototype and find out the way to apply the selected technology on it. Data needed is continue collected and created until the AI model can perform its desired function based on the data collected. Also, the algorithm to be run by the AI model for achieving desired objective should be planned and implemented in future weeks.

Xuǎn

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S1	**Study week no.: 8**
Student Name & ID: See Ling Xuan (19ACB01373)	
Supervisor: Ts Ooi Joo On	
Project Title: Digital Assistant for Workspace Apps	

1. WORK DONE

Necessary data is collected. Implementation and coding process is conducting to develop the AI model by using the data. Project environment was set up properly for the AI model can be trained inside the project structure and be integrated to our system after all. 30% of the coding process was done. The FYP 2 report is updated in more details and modified according to the template downloaded and the FYP 2 guideline. UI was updated for the FYP 2 work.

2. WORK TO BE DONE

To continue the coding part and implement more parts of the AI related feature of this project including the algorithm to run and come out the desired outputs. Also, the way to integrate the AI model with the digital assistant developed should be planned for the final integration. Both UI and use cases should be confirmed and finalized. The FYP 2 report should add in more details and the FYP 2 related section inside the report should be updated.

3. PROBLEMS ENCOUNTERED

Time needed to plan and develop the algorithm used by the AI model to analyse data and provide output of advisory steps to user. Also, the AI model need to be trained and modified for many times to ensure its performance and output accuracy that is repetitive, and time wasted. Besides, a lot of testing methods available via research causes hard to select certain testing and performance metrics used to evaluate the system.

4. SELF EVALUATION OF THE PROGRESS

Progressing smoothly. Project implementation is progressing on the way. The testing plan and performance metrics will be finalized in future weeks for system evaluation.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S1	**Study week no.: 10**
Student Name & ID: See Ling Xuan (19ACB01373)	
Supervisor: Ts Ooi Joo On	
Project Title: Digital Assistant for Workspace Apps	

1. WORK DONE

Completed more coding part of the project. FYP 2 report is further updated with more sections completed. The AI model is developed and trained its algorithm on the way and more required project functions and features are implemented and modified in detail compared to FYP 1. Testing plan and performance metrics are finalized for system evaluation after the project is completed.

2. WORK TO BE DONE

The project should be continue implemented to complete the rest part of the AI function development and complete the project. The processes of model training, fine tuning, and evaluation need to be conducted. Moreover, the system needs to integrate its different part and module to make it a complete system that can conduct the testing as planned. Besides, the FYP 2 report should be further updated to contain more details in project implementation.

3. PROBLEMS ENCOUNTERED

Time should be managed properly to meet the project deadline and achieve all objectives of the FYP. Besides, the consistency and performance of the project developed including its AI function should be improved to ensure it can perform every function properly and accurately. Another problem encountered will be the difficulty to perform system integration due to various tools and software are used in the project development. So, respective solutions should be figured out.

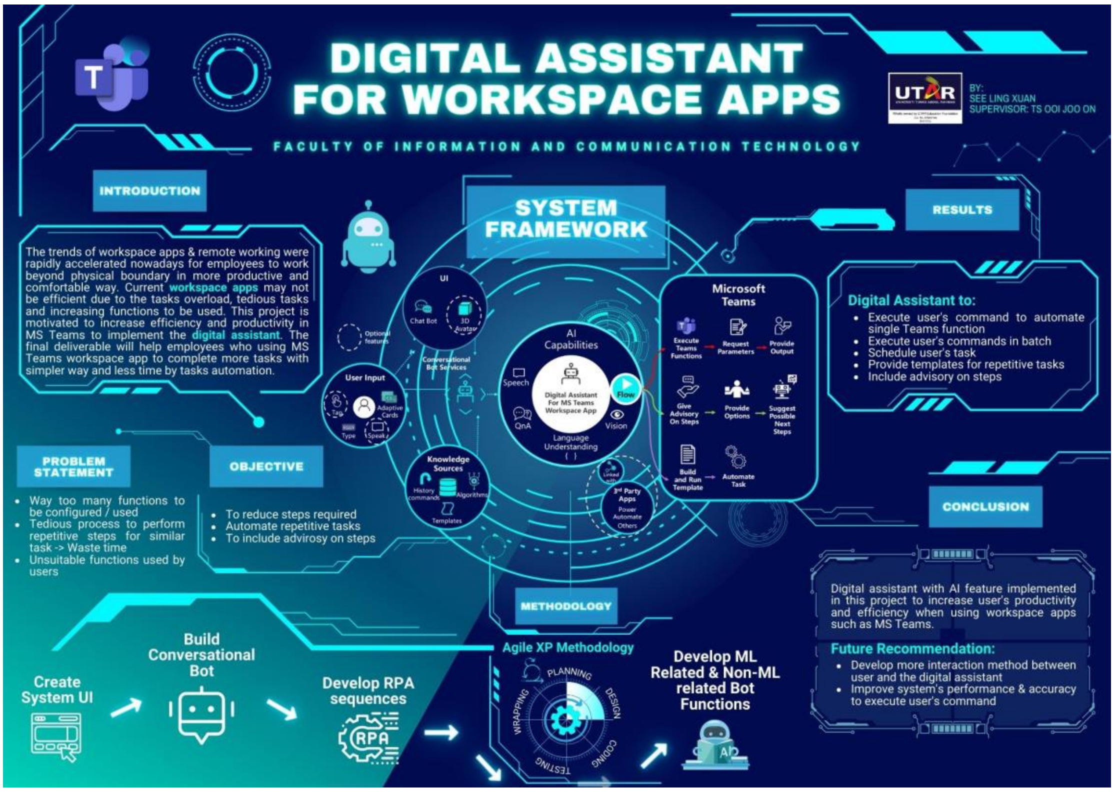
4. SELF EVALUATION OF THE PROGRESS

Progressing smoothly. The project implementation for the FYP 2 is considered to be done by Week 12 while the FYP 2 report is considered to be finished by Week 13. The system testing and evaluation will be conducted after the system is completed in future weeks.

Xuan

Supervisor's signature Student's signature

POSTER



DIGITAL ASSISTANT FOR WORKSPACE APPS

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

BY:
SEE LING XUAN
SUPERVISOR: TS OOI JOO ON

INTRODUCTION

The trends of workspace apps & remote working were rapidly accelerated nowadays for employees to work beyond physical boundary in more productive and comfortable way. Current **workspace apps** may not be efficient due to the tasks overload, tedious tasks and increasing functions to be used. This project is motivated to increase efficiency and productivity in MS Teams to implement the **digital assistant**. The final deliverable will help employees who using MS Teams workspace app to complete more tasks with simpler way and less time by tasks automation.

PROBLEM STATEMENT

- Way too many functions to be configured / used
- Tedious process to perform repetitive steps for similar task -> Waste time
- Unsuitable functions used by users

OBJECTIVE

- To reduce steps required
- Automate repetitive tasks
- To include advirosy on steps

SYSTEM FRAMEWORK

RESULTS

Digital Assistant to:

- Execute user's command to automate single Teams function
- Execute user's commands in batch
- Schedule user's task
- Provide templates for repetitive tasks
- Include advisory on steps

METHODOLOGY

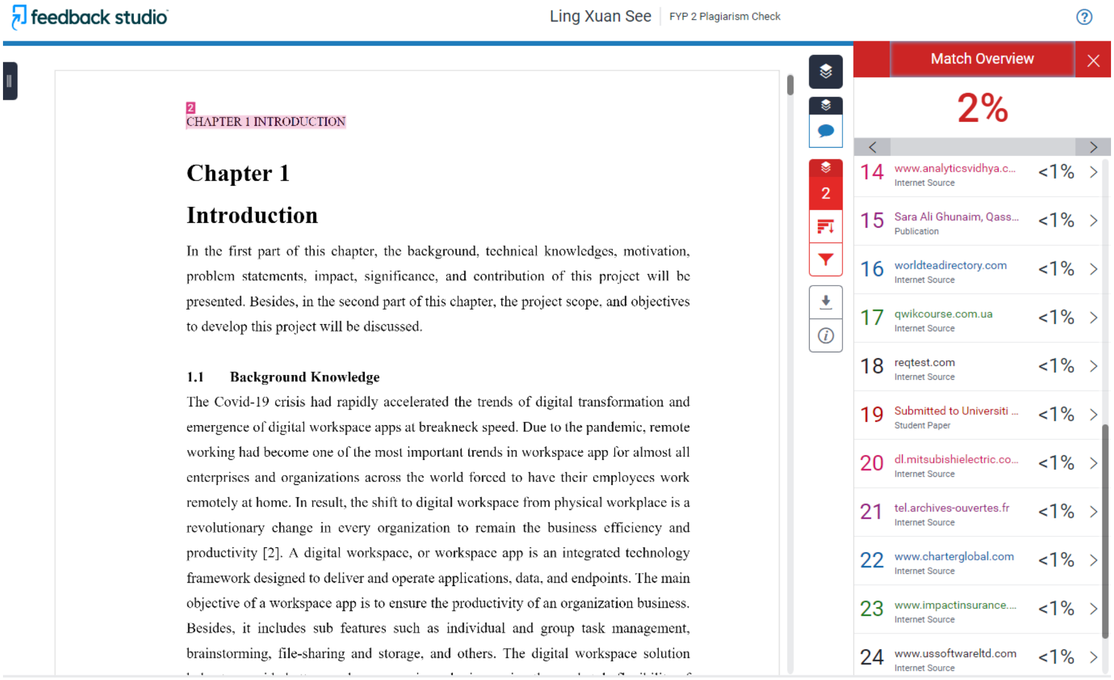
CONCLUSION

Digital assistant with AI feature implemented in this project to increase user's productivity and efficiency when using workspace apps such as MS Teams.

Future Recommendation:

- Develop more interaction method between user and the digital assistant
- Improve system's performance & accuracy to execute user's command

PLAGIARISM CHECK RESULT



Ling Xuan See | FYP 2 Plagiarism Check

2 CHAPTER 1 INTRODUCTION

Chapter 1

Introduction

In the first part of this chapter, the background, technical knowledges, motivation, problem statements, impact, significance, and contribution of this project will be presented. Besides, in the second part of this chapter, the project scope, and objectives to develop this project will be discussed.

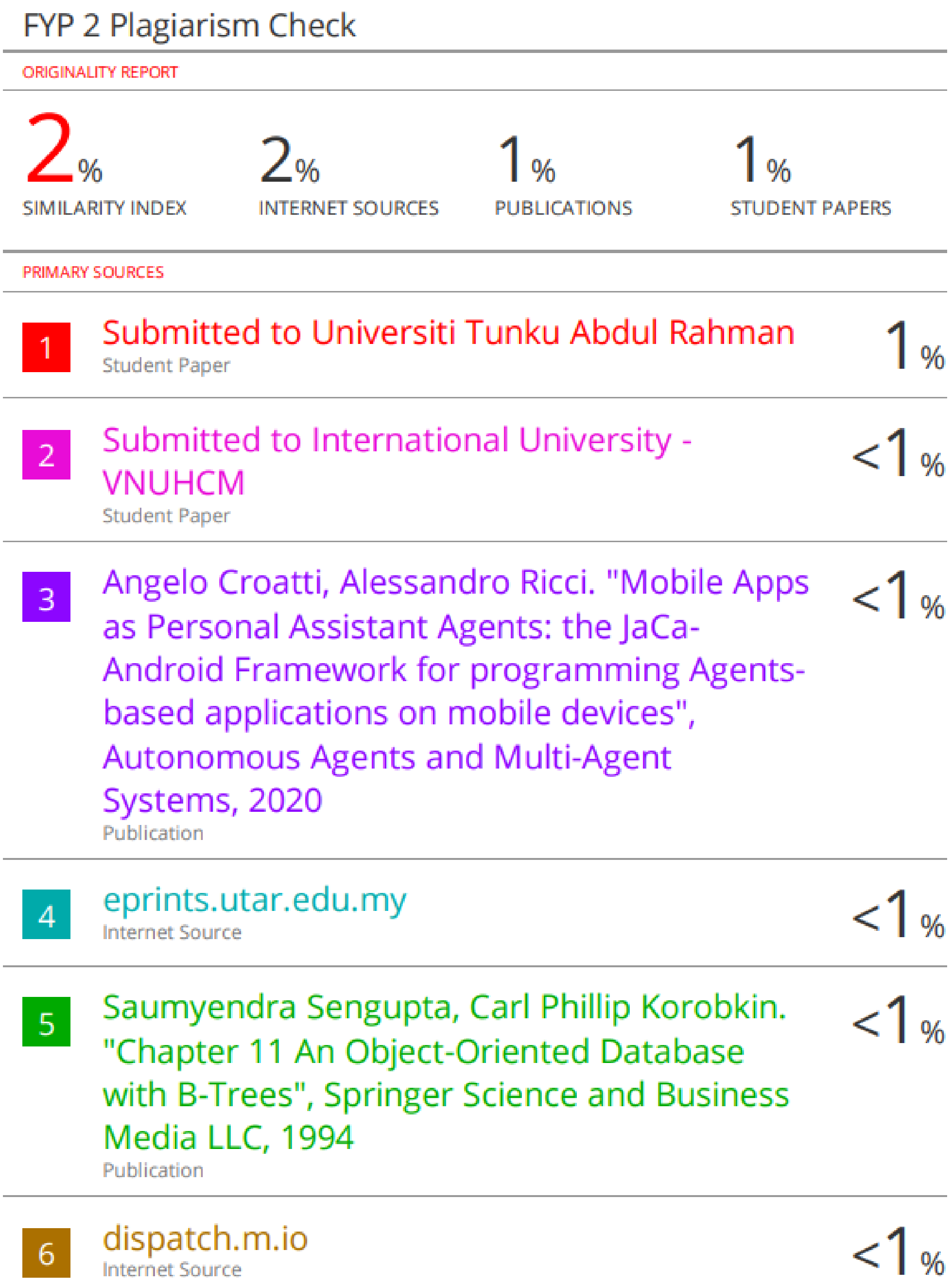
1.1 Background Knowledge

The Covid-19 crisis had rapidly accelerated the trends of digital transformation and emergence of digital workspace apps at breakneck speed. Due to the pandemic, remote working had become one of the most important trends in workspace app for almost all enterprises and organizations across the world forced to have their employees work remotely at home. In result, the shift to digital workspace from physical workplace is a revolutionary change in every organization to remain the business efficiency and productivity [2]. A digital workspace, or workspace app is an integrated technology framework designed to deliver and operate applications, data, and endpoints. The main objective of a workspace app is to ensure the productivity of an organization business. Besides, it includes sub features such as individual and group task management, brainstorming, file-sharing and storage, and others. The digital workspace solution

Match Overview

2%

14	www.analyticsvidhya.c... Internet Source	<1%
15	Sara Ali Ghunaim, Qass... Publication	<1%
16	worldteadirectory.com Internet Source	<1%
17	qwikcourse.com.ua Internet Source	<1%
18	reqtest.com Internet Source	<1%
19	Submitted to Universiti ... Student Paper	<1%
20	dl.mitsubishielectric.co... Internet Source	<1%
21	tel.archives-ouvertes.fr Internet Source	<1%
22	www.charterglobal.com Internet Source	<1%
23	www.impactinsurance.... Internet Source	<1%
24	www.ussoftwareltd.com Internet Source	<1%



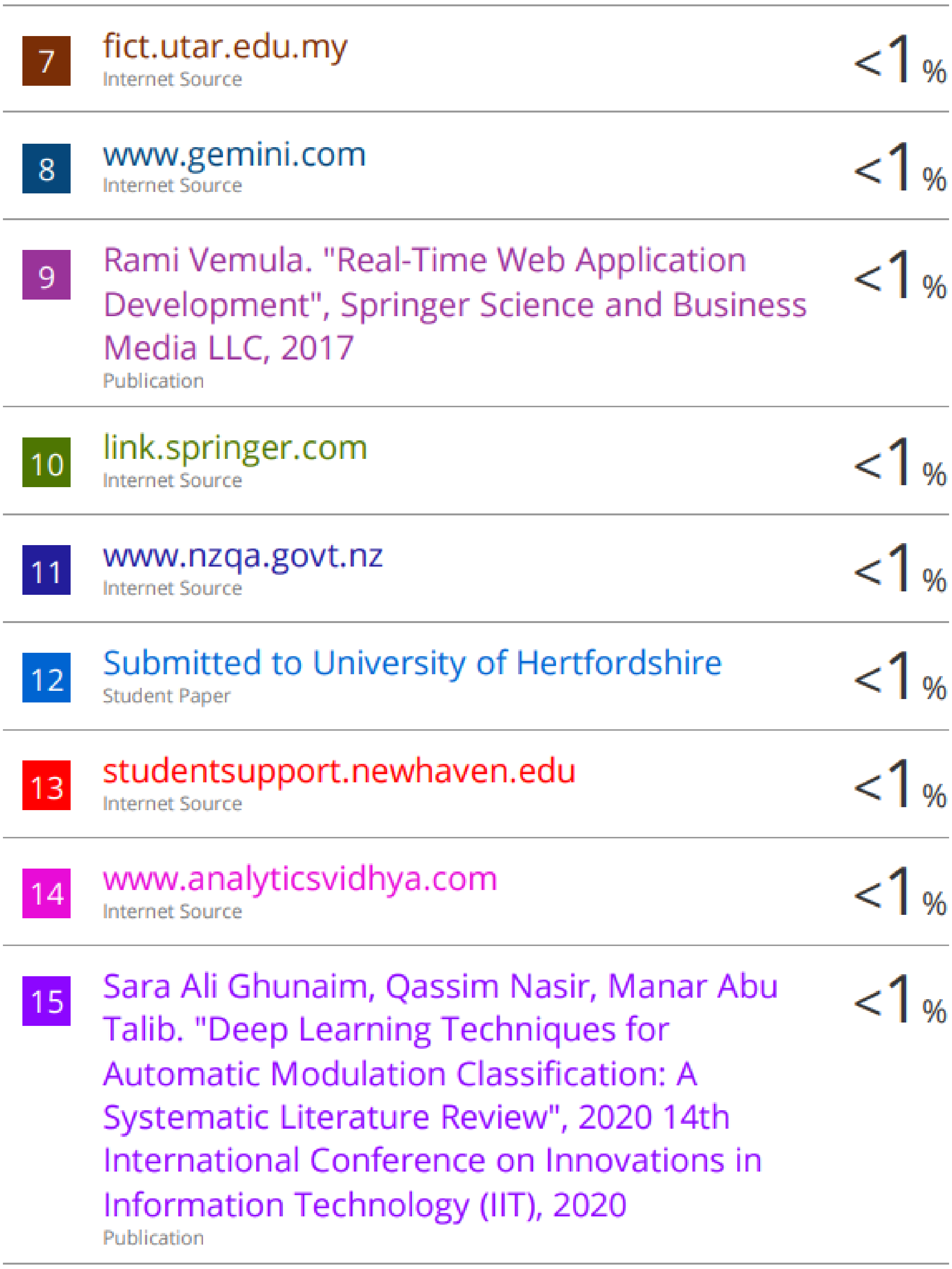
FYP 2 Plagiarism Check

ORIGINALITY REPORT

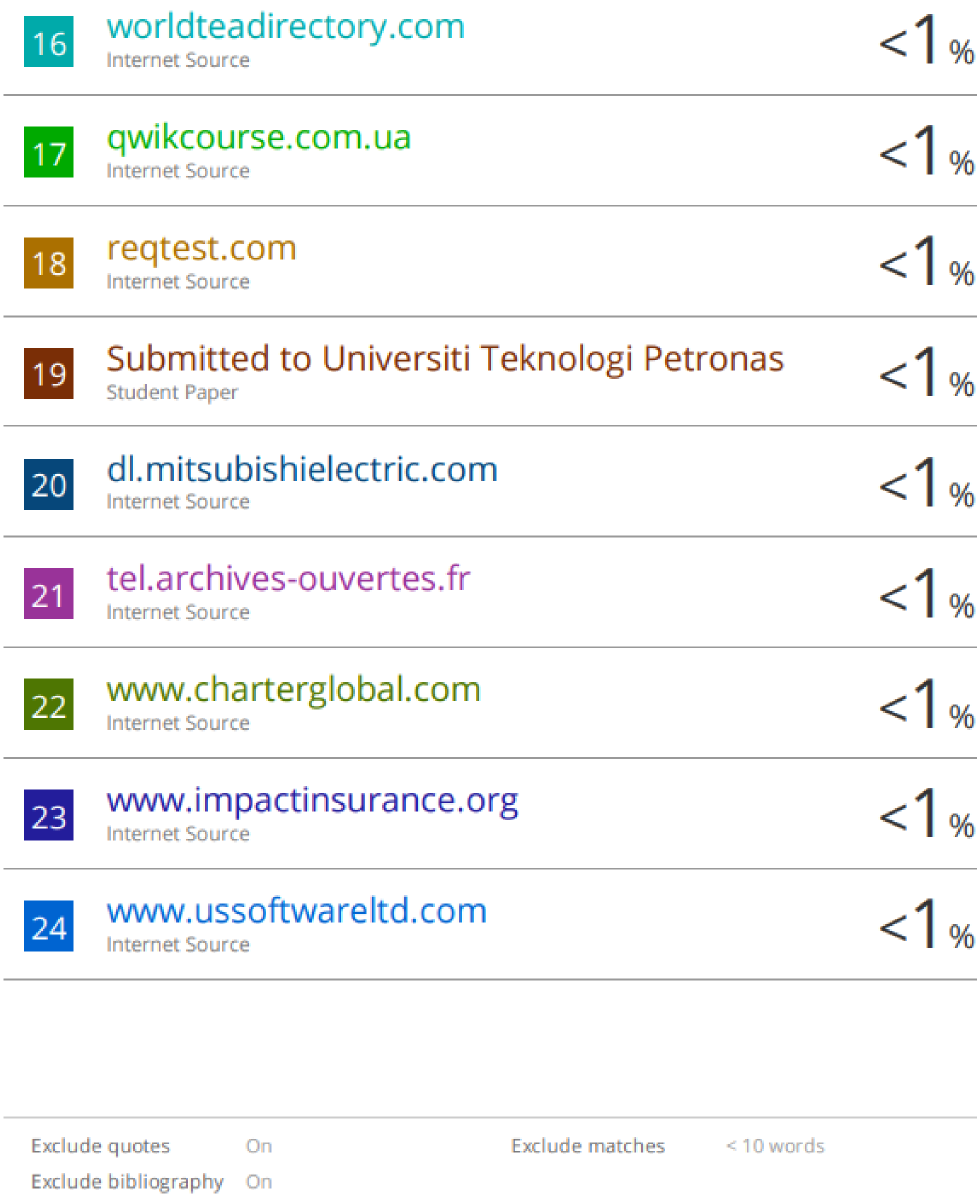
2%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
2	Submitted to International University - VNUHCM Student Paper	<1%
3	Angelo Croatti, Alessandro Ricci. "Mobile Apps as Personal Assistant Agents: the JaCa-Android Framework for programming Agents-based applications on mobile devices", Autonomous Agents and Multi-Agent Systems, 2020 Publication	<1%
4	eprints.utar.edu.my Internet Source	<1%
5	Saumyendra Sengupta, Carl Phillip Korobkin. "Chapter 11 An Object-Oriented Database with B-Trees", Springer Science and Business Media LLC, 1994 Publication	<1%
6	dispatch.m.io Internet Source	<1%



7	fict.utar.edu.my Internet Source	<1 %
8	www.gemini.com Internet Source	<1 %
9	Rami Vemula. "Real-Time Web Application Development", Springer Science and Business Media LLC, 2017 Publication	<1 %
10	link.springer.com Internet Source	<1 %
11	www.nzqa.govt.nz Internet Source	<1 %
12	Submitted to University of Hertfordshire Student Paper	<1 %
13	studentsupport.newhaven.edu Internet Source	<1 %
14	www.analyticsvidhya.com Internet Source	<1 %
15	Sara Ali Ghunaim, Qassim Nasir, Manar Abu Talib. "Deep Learning Techniques for Automatic Modulation Classification: A Systematic Literature Review", 2020 14th International Conference on Innovations in Information Technology (IIT), 2020 Publication	<1 %



16	worldteadirectory.com Internet Source	<1 %
17	qwikcourse.com.ua Internet Source	<1 %
18	reqtest.com Internet Source	<1 %
19	Submitted to Universiti Teknologi Petronas Student Paper	<1 %
20	dl.mitsubishielectric.com Internet Source	<1 %
21	tel.archives-ouvertes.fr Internet Source	<1 %
22	www.charterglobal.com Internet Source	<1 %
23	www.impactinsurance.org Internet Source	<1 %
24	www.ussoftwareltd.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	See Ling Xuan
ID Number(s)	19ACB01373
Programme / Course	IA
Title of Final Year Project	Digital Assistant for Workspace Apps

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 2 **%** **Similarity by source** Internet Sources: 2 % Publications: 1 % Student Papers: 1 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: Ooi Joo On

Date: 9/9/2022

____
Signature of Co-Supervisor

Name: ______________________

Date: ______________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB01373
Student Name	See Ling Xuan
Supervisor Name	Ts Ooi Joo On

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

Xuan
____

(Signature of Student)

Date: 8 September 2022