

**MEETING SCHEDULER WITH SECURED PRIVACY WEB  
APPLICATION**

BY

SAMANTHA LOK TIAN WEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS ENGINEERING (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2023

## REPORT STATUS DECLARATION FORM

Title: MEETING SCHEDULER WEB APPLICATION DEVELOPMENT WITH  
SECURED PRIVACY

Academic Session: JUNE 2023

I SAMANTHA LOK TIAN WEN  
(CAPITAL LETTER)

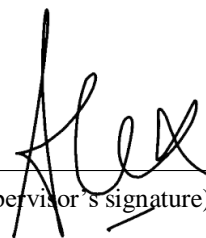
declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

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.35, Jalan KP 3/3,  
Taman Kajang Prima 43000,  
Kajang Selangor.

**Date:** SEPTEMBER 15, 2023.

\_\_\_\_\_  
Ts Dr Ooi Boon Yaik

**Date:** 15/9/2023

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

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**  
**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 15<sup>TH</sup> SEPTEMBER 2023

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that **Samantha Lok Tian Wen** (ID No: **19ACB01455**) has completed this final year project entitled “**Meeting Scheduler Application with Web Security**” under the supervision of **Ts Dr OOLBOON YAIK** (Supervisor) from the Department of **Computer Science**, 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,



\_\_\_\_\_  
(SAMANTHA LOK TIAN WEN)

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**MEETING SCHEDULER WEB APPLICATION DEVELOPMENT WITH SECURED PRIVACY**” 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 : \_\_\_\_\_

Name : SAMANTHA LOK TIAN WEN

Date : 15<sup>TH</sup> September 2023

## ACKNOWLEDGEMENTS

First and foremost, my deepest gratitude is directed towards Ts Dr. Ooi Boon Yaik, who graciously served as my guiding light throughout this journey. His invaluable direction, persistent counsel, and insightful critique have not only steered the course of this project but have also enriched my comprehension of the development process, requirement specifications, among other pivotal topics. His mentorship, particularly through the intricate phases of the software development life cycles, has been a cornerstone of my learning experience. His consistent feedback was both enlightening and instrumental in shaping the trajectory of my work. Lastly, the backbone of my journey has undoubtedly been the ceaseless support of my family. My heartfelt appreciation goes out to my parents, whose unwavering faith, sage guidance, and continual encouragement fueled my resolve and determination. Their life lessons and faith in my endeavors have been the driving force, pushing me towards the finish line. I am also deeply thankful to my friends who, with their constant support and belief in my vision, have made this study not just feasible, but a memorable journey.

## **ABSTRACT**

The scheduling and management of meetings in universities pose significant challenges. Therefore, a meeting scheduler web application with secured privacy has been proposed to address these issues. The system is developed system to automate the information systems for an arrangement of meeting between users alternatively. The system includes an integration data set as the Google authentication for users to give their credentials to allow the system to extract the information given. The system also developed to evaluate the use of CRUD (Create, Read, Update, Delete) in the web-based information systems as it is crucial that the systems extracted precise and accurate data. To develop the meeting scheduler web application with secured privacy, technologies such as Python, HTML, MongoDB, CSS, and JavaScript were used. The proposed system addresses the limitations of manual meeting management and suggests solutions to overcome them.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>REPORT STATUS DECLARATION FORM</b>	<b>ii</b>
<b>FYP THESIS SUBMISSION FORM</b>	<b>iii</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENTS</b>	<b>vi</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement and Motivation	2
1.2 Objectives	3
1.3 Project Scope and Direction	4
1.4 Contributions	5
1.5 Report Organization	6

<b>CHAPTER 2</b>	<b>LITERATURE REVIEW</b>	<b>7</b>
2.1	Theory of Literature Review	7
2.2	Reviews on Existing Systems	8
2.2.1	WhenAvailable.com Application	8
2.2.1	Summary of WhenAvailable.com Strengths and Weaknesses	9
2.2.2	Doodle Application	10
2.2.2	Summary of Doodle Strengths and Weaknesses	12
2.2.3	Calendly Application	13
2.2.3	Summary of Calendly Strengths and Weaknesses	13
2.3	Comparison and Limitations of all Existing Systems	15
2.4	Summary	16
<b>CHAPTER 3</b>	<b>SYSTEM METHODOLOGY</b>	<b>17</b>
3.1	System Design Diagram/Equation	17
3.2	Agile Methodology – SCRUM Process	21
3.3	Project Timeline	23
<b>CHAPTER 4</b>	<b>SYSTEM DESIGN</b>	
4.1	System Components	24
4.1.1	Hardware Specification	
4.1.2	Software Specifications	
4.2	System Architecture Diagram	25
4.3	System Design	26



## **CHAPTER 5            SYSTEM IMPLEMENTATION**

5.1	Software Setup	
5.1.1	Integrated Development Environment/Software Tools	27
5.1.2	Programming Language	28
5.1.3	Framework and Configurations	29
5.1.4	MongoDB: Revolutionizing Database Management in the Moden Age	31
5.1.5	Google Calendar API Integration	32
5.2	System Operation	38

## **CHAPTER 6            SYSTEM EVALUATION AND DISCUSSION**

6.1	System Testing and Performance Metrics	42
6.2	Testing Setup and Result	
6.3	Project Challenges	43
6.4	Objectives Evaluation	44
6.5	Concluding Remark	45

## **CHAPTER 7            CONCLUSION**

7.0	Conclusion	46
-----	------------	----

## **REFERENCES**

### **APPENDIX A**

A.1	Weekly Report	A-1
A.2	Poster	A-2

### **APPENDIX B PLAGIARISM CHECK TURNITIN SUMMARY**

B-1

## LIST OF FIGURES

Figure Number	Title	Page
Figure 2.2.1	Display of Whenavailable.com Web version	8
Figure 2.2.2	Web version of poll creation	8
Figure 2.2.3	Web version of adjusting the calendar.	9
Figure 2.2.4	Web Version of Doodle's Main Layout	10
Figure 2.2.5	Web Version of Doodle's Affiliates	11
Figure 2.2.6	Web Version of Doodle to Create a Booking	12
Figure 2.2.7	Web Version of Calendly Dashboard	13
Figure 2.2.8	Web Version of Calendly for Bookings	13
Figure 2.2.9	Web Version of Calendly Scheduled Events	14
Figure 3.1.1	Source Code Implemented with Interval Algorithm	18
Figure 3.1.2	Pattern Merge Interval for Start Time and End Time	19
Figure 3.1.3	APScheduler Running Backend to Retrieve New Data	20
Figure 3.2.1	SCRUM Approach Process	21
Figure 3.3.1	Gantt Chart for FYP 1 & 2	23
Figure 4.2.1	System Architecture Design	25
Figure 4.3.1	System Design	26
Figure 5.1.1	Logo of Visual Studio Code	27
Figure 5.1.2	Google Developer Console	32
Figure 5.1.3	Add New Project	32
Figure 5.1.4	Details of New Project	33
Figure 5.1.5	Enable API Services	33
Figure 5.1.6	Search for Calendar API	34
Figure 5.1.7	Configurations for OAuth Consent Screen	34
Figure 5.1.8	Generate credentials.json	35

## **LIST OF FIGURES**

Figure 5.2.1	Authentication for Credentials on Primary Account(middleman)	38
Figure 5.2.2	After obtaining the information from the user's Google Calendar	39
Figure 5.2.3	User Login System (Account Creation)	39
Figure 5.2.4	User Login System	40
Figure 5.2.5	User Information Stored in MongoDB	40
Figure 5.2.6	Dashboard Layout	41
Figure 5.2.7	Schedule Meeting Layout	41

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 2.3	Comparison and Limitation Table of All Existing Systems	7
Table 3.1.1	Hardware Requirements	24
Table 3.1.2	Software Requirements	24
Table 6.1.1	Functional Testing System	42
Table 6.2.1	Objectives Evaluation	44

## **LIST OF ABBREVIATIONS**

<i>HDL</i>	Hardware Description Language Create,
<i>CRUD</i>	Read, Update, Delete Visual Studio
<i>VS Code</i>	Visual Studio Code
<i>JWTs</i>	JSON Web Tokens
<i>BSON</i>	Binary JSON
<i>SSPL</i>	Server-Side Public License
<i>SPI</i>	Software Process Improvements
<i>AVL</i>	Adelson-Velskii and Landis

## **CHAPTER 1: INTRODUCTION**

In modern times, timely communication is of paramount importance, especially as life's tempo has accelerated. With this perspective, a communication tool should facilitate immediate file and message exchanges, experiencing little to no lag based on the transmission medium. To make such a system effective, it's essential to have a database that updates instantaneously, ensuring all transferred data is accurately tracked. Communication has always been a cornerstone of human interaction. Historically, as territories expanded and distances between people grew, the means of communication evolved. Traditional methods like letters and telegrams, while effective, were slow, often taking days for delivery.

However, the dawn of mobile phones paved the way for a faster solution: text-based messaging. In 1984, the foundations for SMS were laid by Friedhelm Hillebrand and Bernard Ghillebaert during the Franco-German GSM collaboration. A significant challenge they faced was the constrained message length of 128 bytes. By 1992, the first SMS was sent, and a commercial service was launched in Sweden by Aldiscon with Telia the next year. SMS became a communication mainstay in the 2000s, though its cost was a drawback. With the emergence of smartphones in the late 2000s, a plethora of messaging apps, such as WhatsApp, WeChat, Viber, and Snapchat, began dominating the scene, becoming instant favorites worldwide. [44-45]

Nevertheless, it is a frequent challenge to identify a convenient time for a meeting that involves multiple participants. This process includes exchanging details about everyone's availability and negotiating a mutually convenient meeting time, which can be tedious and lengthy. Although various software programs have been developed to facilitate the scheduling process, they may still demand significant manual input, or may not be appropriate for arranging meetings among individuals from different organizations. This project will propose the system that identifies appropriate meeting times by automatically examining users' calendar information to determine their availability, which is obtained directly from the device or system that they use to manage their calendar from GoogleCalendar server.

The determination will be done with an algorithm that automate searching in finding the soonest slots and get the available slots. The algorithm that will be implemented is Interval Algorithm which will be integrating Binary Search Tree to implement the equation. The meeting scheduling system provides generic interfaces that enable third parties to create support with users' credentials in accessing their calendar events information. In this chapter,

problem statement and motivation will be discussed to find out the key concern of this project.

### 1.1 Problem Statement and Motivation

First and foremost, the problem of **ineffective time management** is a constant problem for teachers and students to plan their daily tasks. How frequent should students meet their supervisor? Do you know the availability of other parties for each week to schedule a meeting? Time and date are by far the biggest problem to schedule a meeting neither with colleagues nor students. To schedule meetings, it is customary to employ a trial-and-error method, suggesting various times until one is ultimately chosen. Nevertheless, this is a huge time waster and might not provide you the best option.

The emergence of dynamic webpages led to the development of Web 2.0. To interact with users and make it easier for them to contribute information or conduct searches within websites, dynamic websites must be interactive. At this point, the significance of online application security really started to shine through. Hackers could insert malicious code that would enable them to steal sensitive information if a website allowed users to interact with it and enter sensitive data, such as usernames and passwords, if it is not configured properly. This depicts the **feeble deterioration of the implemented advanced technologies**. All the significant flaws, including XSS, SQL Injection, and LFI attacks, first appeared at this period.

Automated scheduling has a major drawback, which is **the absence of a standardized and organized communication protocol**. In addition to this, automating scheduling processes requires addressing other challenges such as developing a mechanism for decision-making and negotiation tactics. To arrange a meeting, participants need to reach a mutual agreement that fits their individual preferences and constraints, making the process time-consuming. To assist with this process, there are various calendaring applications available. However, these software solutions have not completely replaced the traditional scheduling procedure and instead serve as a tool to manage meeting-related information.

The intent behind this project development is **to develop an automated meeting scheduler web application with secured privacy by adhering the implementations of APIs authentication for web security and substantial utilization precedents for users**. The mission aims to assist the user in accessing real-time booking for meetings and user behaviour analysis. The application is designed to elevate users' experience in task and meeting arrangement, all while prioritizing privacy.

## 1.2 Research Objectives

The purpose of the thesis is to develop a website application that able to:

**To develop a web-based automated information system for meeting scheduling** that helps users consistently handle administrative duties like arranging meetings. This requires a delicate equilibrium between respecting individual privacy and preferences while maximizing the use of time and other resources. This defines the systems will consider individual users' preferences and privacy when scheduling meetings. As a result, automated scheduling allows operation seamlessly without the need for complex decision-making algorithms or negotiation strategies. Automated scheduling is an important aspect of this system, which allows it to operate with minimal human intervention to pick the appropriate meeting schedule.

**To integrate high security web-based information system with Google authentication module.** All the data and personal information will be accessed under individual's credentials. The integration of Google's module could bring ease for one to compile their calendars information and extract selected accurate information for user to make decision. This integration also provides a solution for protecting the privacy of an individual data by securing it without the need to make the calendars public. This approach of using individual credentials to access sensitive information ensures that only authorized individuals have access to this data, thus maintaining its confidentiality. This helps to protect against unauthorized access to personal data and helps ensure that privacy is maintained. The integration of a Google authentication module into a web-based information system can provide numerous benefits, including improved security, ease of use, and enhanced privacy protections.

**To evaluate CRUD in the web-based information system.** The primary of this project is to guarantee the precision of CRUD information system such as creating, reading, updating, and deleting data. Evaluating the results of the system is crucial to ensure that it can produce a precise and accurate output. The usability and complexity of the interface must be tested to establish a secure system that can operate smoothly and is easy for users to interact with to prevent unauthorized access or data breaches. All the system's features and modules should be tested to confirm that they are operational and free from errors. The project's goal is to strike a balance between usability and security to ensure that the system is secure while remaining user-friendly.



### **1.3 Project Scope and Direction**

This project's primary goal is to create a Meeting Scheduler Web Application with Secured Privacy that focuses on the problem stated in the problem statement. A personal calendar will be integrated into this information system, and it will use the user's authorized information to filter out availability dates and times to determine the best time for a meeting.

This project is based on a case study approach, with a focus on helping users make meeting arrangements by authorizing the sharing of certain information and keeping personal information secure.

### 1.4 Contributions

This project features assets can significantly improve the efficiency, collaboration, and transparency of individuals. Firstly, increased efficiency with an automated meeting scheduling system, users can save time and effort in coordinating meetings, thus increasing overall inefficiency by streamlining the meeting scheduling process, individuals can focus on their core tasks and responsibilities. Secondly, the lack of uniform structured communication protocol problem is handled by the enhancement of collaboration on which a meeting scheduling system can bring together individuals from different locations and time zones, facilitating better collaboration among teams. Thirdly, the feeble deterioration of implemented advanced technologies can be solved by allowing users to maintain the confidentiality of their personal information by using their individual credentials to access the app. This allow users experience by providing a secure and efficient way to schedule meeting while ensuring their data is secure and not accessible to unauthorized users with the integration of Google authentication.

### 1.5 Report Organization

This research comprises of seven chapters. The first chapter will provide an introduction, followed by a literature review in the second chapter. Chapter three will outline the proposed methodology and approach for the study, while the fourth chapter will discuss the system design. Chapter five on the other hand will handle the system setup and system operation. Finally, the sixth will be evaluating the project outcome and seventh chapter will present the conclusion.

**Chapter 1** of this report will serve as an introduction to the project, providing a review of relevant background information, problem statements, motivations, project objectives, scopes, impacts, significance, and contributions. The importance of this chapter is to emphasize on the overall concepts of this project. **Chapter 2** will focus on explaining the theory of existing systems, with a specific emphasis on literature reviews of similar systems available in the market. The objective of this chapter is to pinpoint the advantages and disadvantages of each system. By comparing them with the proposed system, we will highlight the similarities and disparities. Though all the evaluated systems are amazing, however there are some improvements to be made according to the comparison table.

**Chapter 3** will focus on the proposed method and approaches used in planning, controlling, and structuring the project development process. Most of all it will introduce the theoretical on implemented algorithms and equation used along with the retrieval explanation with figures. **Chapter 4** on the other hand will detail the functional and non-functional requirements, hardware, and software specifications, and include the system diagrams and system architecture diagram of the system. In contrast, **Chapter 5** will depict the system operations of the completed system, including software setup and image visualization and segmentation of each layout on the frontend.

**Chapter 6** delves into system assessment and verification, employing the black-box testing method. This approach emphasizes the system's final output rather than its internal mechanisms. Additionally, a comparative analysis between the project's outcomes and the objectives outlined in Chapter 1 will be conducted. This chapter also provides an in-depth account of the challenges faced during implementation and how they were addressed. Following this, **Chapter 7**, the concluding section of this project, offers a comprehensive summary and reflection on the entire work.

## **CHAPTER 2: LITERATURE REVIEW**

### **2.1 Theory of Literature Review**

In recent years, development of meeting scheduling applications has gained significant attention in both academia and industry. The literature on meeting scheduling has grown significantly, drawing from various disciplines such as computer science, operations research, psychology, and communication studies. One major focus of research in meeting scheduling has been on the development of optimization models and algorithms. Optimization models seek to minimize or maximize an objective function subject to constraints and have been used to develop algorithms for scheduling meetings efficiently. These models and algorithms consider various factors such as time availability, location, participant availability, and priority. For instance, some models use heuristic algorithms that consider participant availability, preferences, and conflicts in scheduling. Other models use integer programming techniques to find optimal solutions to scheduling problems.

Another area of research has been on understanding participant preferences and communication styles in meetings. Several studies have examined the impact of communication patterns on meeting scheduling and effectiveness. Some studies have found that scheduling meetings based on the preferences and communication patterns of participants can significantly improve their engagement and productivity.[2] Other studies have examined the impact of personality traits, leadership styles, and communication styles on meeting effectiveness. Overall, the literature review would provide a comprehensive understanding of the research on meeting scheduling, highlighting the different models, algorithms, and techniques used in this area. The review would help to identify gaps in existing research and inform the development of new and improved meeting scheduling applications that can accommodate various constraints, preferences, and communication styles. The development of such applications can have a significant impact on organizational productivity and efficiency.

## 2.2 Existing Systems Literature Review

### 2.2.1 WhenAvailable.com

Whenavailable.com is a cross-platform existing system that is already in use and has been designed as a mobile application as well as a web server version. This system makes it easier to plan out time for various activities. It is a platform where content may be shared with people all over all over the world regardless of the time zone. [4]

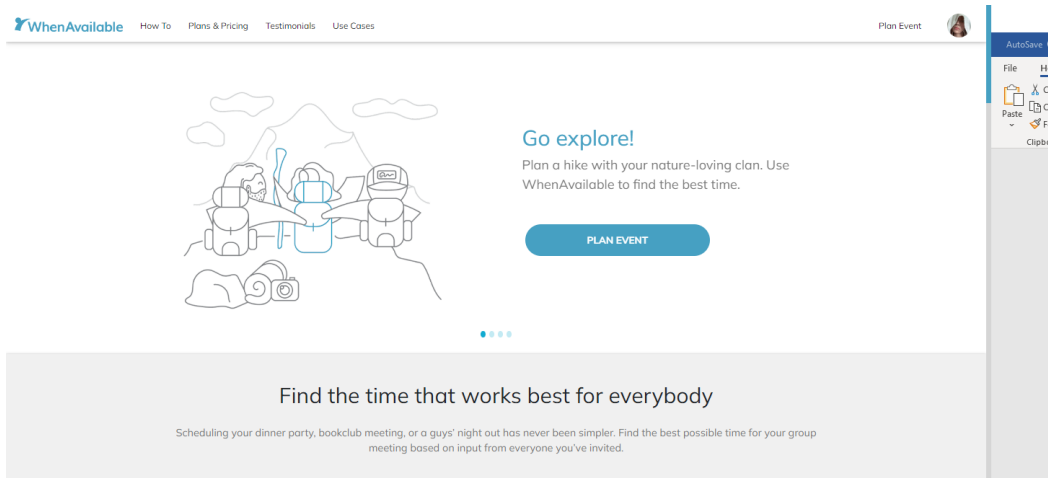


Figure 2.2.1 Display of Whenavailable.com Web version

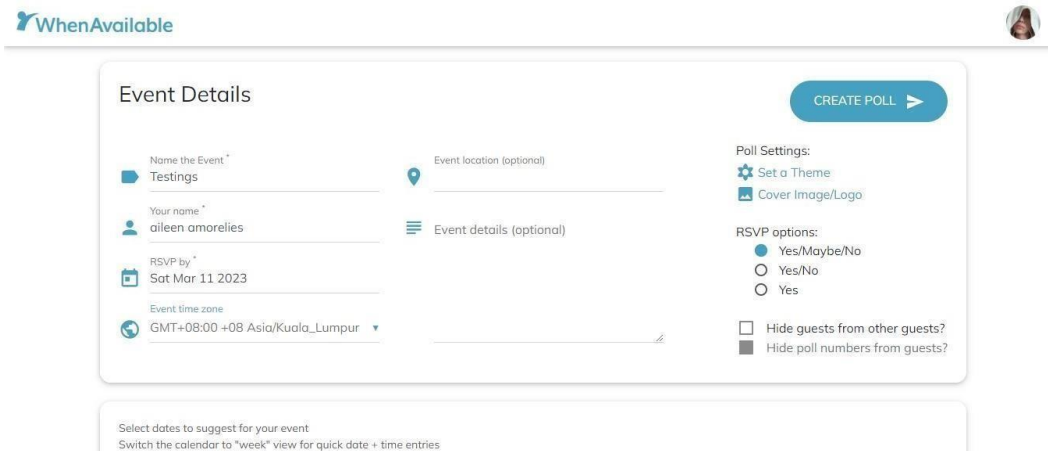
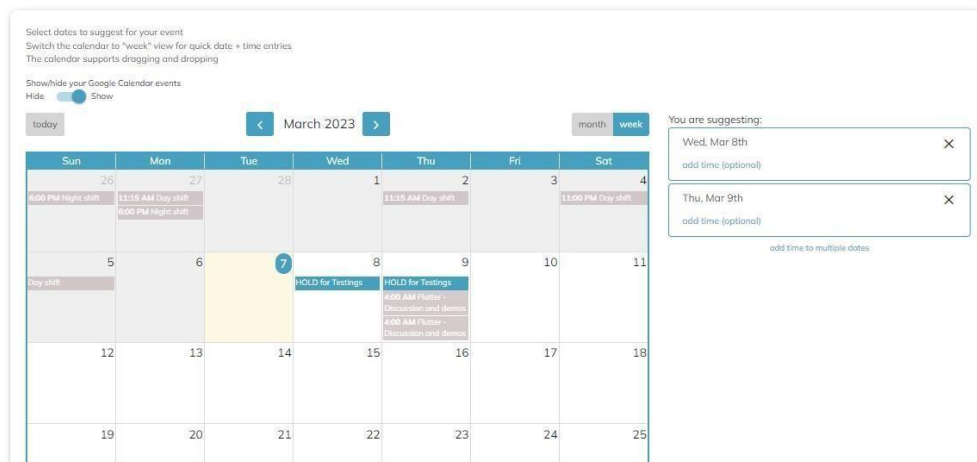


Figure 2.2.2 Web version of poll creation

Users can enter all the necessary information to build plans, including the location, event title, creator, date, time, and time zone selections. Further information is optional. The settings for the event's poll are displayed on the right sidebar. By picking from available options, user might establish a theme. Nevertheless, the ability to alter the cover with images or logos for the event is a premium package; as a result, users may only access this function through in-app purchases.



*Figure 2.2.3 Web version of adjusting the calendar.*

Figure 2.3 shows that even while users may upload their Google calendar into the systems for better planning, they could choose to conceal or reveal it to distinguish between meetings and their personal calendar. After done creating the poll, invitations will be sent through emails directly from the poll that is created, or it could also be sent with the shareable link. The limit of participants for one poll is 20 guests. To invite more participants, user is required to upgrade and purchase their plan.

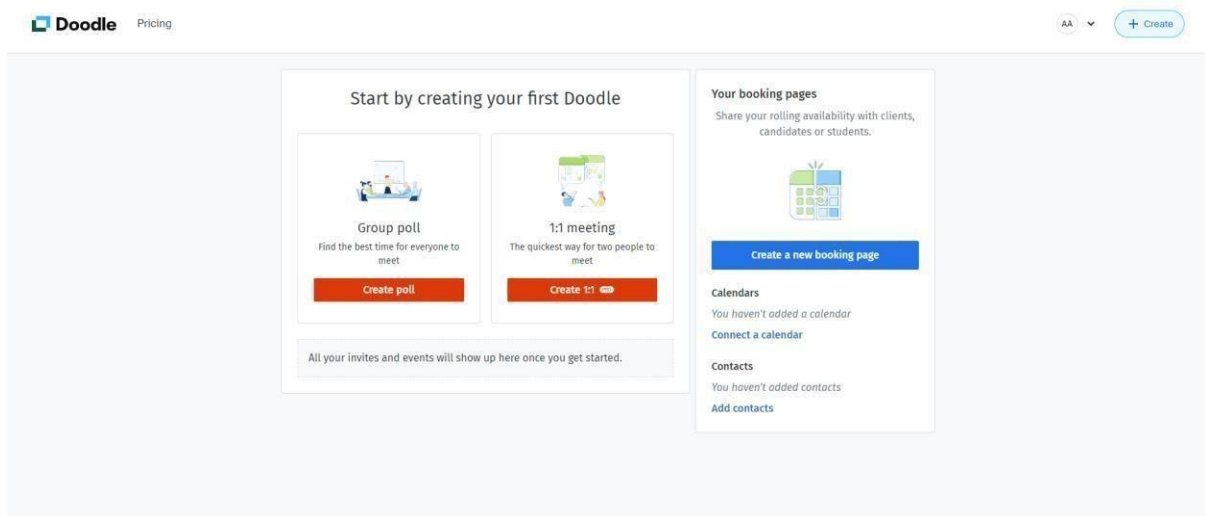
### **2.2.1 Summary of Strength and Weakness of Whenavailable.com system tool**

Whenavailable.com emerges as an efficient scheduling tool marked by its simplicity and intuitiveness. Its user-friendly interface makes it accessible even for those who might not be tech-savvy, ensuring a hassle-free experience. Beyond this, the tool provides a rich tapestry of customization options, notably the ability to track members' attendance dynamically. This feature, which allows the addition and removal of features, grants administrators or event organizers increased flexibility in managing their events or meetings. Furthermore, its integration with Google Calendar enhances its utility, providing users with a cohesive scheduling platform that interfaces smoothly with one of the most widely used calendar systems in the digital sphere.

On the flip side, Whenavailable.com has faced certain challenges that impact its overall efficacy. One significant issue pertains to the mobile application, where glitches have been reported. Some users found themselves unable to access the application despite a successful registration, which can be a source of frustration and might deter potential users from adopting the platform. Another weakness, which appears to be shared with other scheduling platforms, is the limited control over time zone settings. In an era where global coordination is essential, this limitation might introduce challenges in aligning schedules across different geographical zones. These areas of concern highlight the need for continuous improvement in the tool to better cater to its user base.

### 2.2.2 Doodle

Doodle is a web server which is also a cross-platform system where it could be found in App Store and Google Play Store. More than three million unique people per month from around the world utilize Doodle, the top online scheduling tool in the world. It makes scheduling practically painless and takes the hassle out of figuring out the best day and time for a group of people to get together. Doodle's user-friendly interface enables users to establish their availability for a particular group meeting and let each member select the dates that work best for them. The host will select a variety of dates and times for the meeting, distribute them to the participants, and ask them to vote on the times that suit them the most. User can programmatically schedule a meeting in the premium edition for the chosen time in everyone's calendar.



*Figure 2.2.4 Web version of Doodle's Main Layout*

Moreover, with the premium package, it is included that users could do 1:1 meeting with the proposed time and the other party could choose the alternatives of the available dates with the help of the meeting tool Doodle, users can quickly determine a time that suits their group's needs without going back and forth for hours. The primary feature of Doodle is its ability to create polls for meetings.

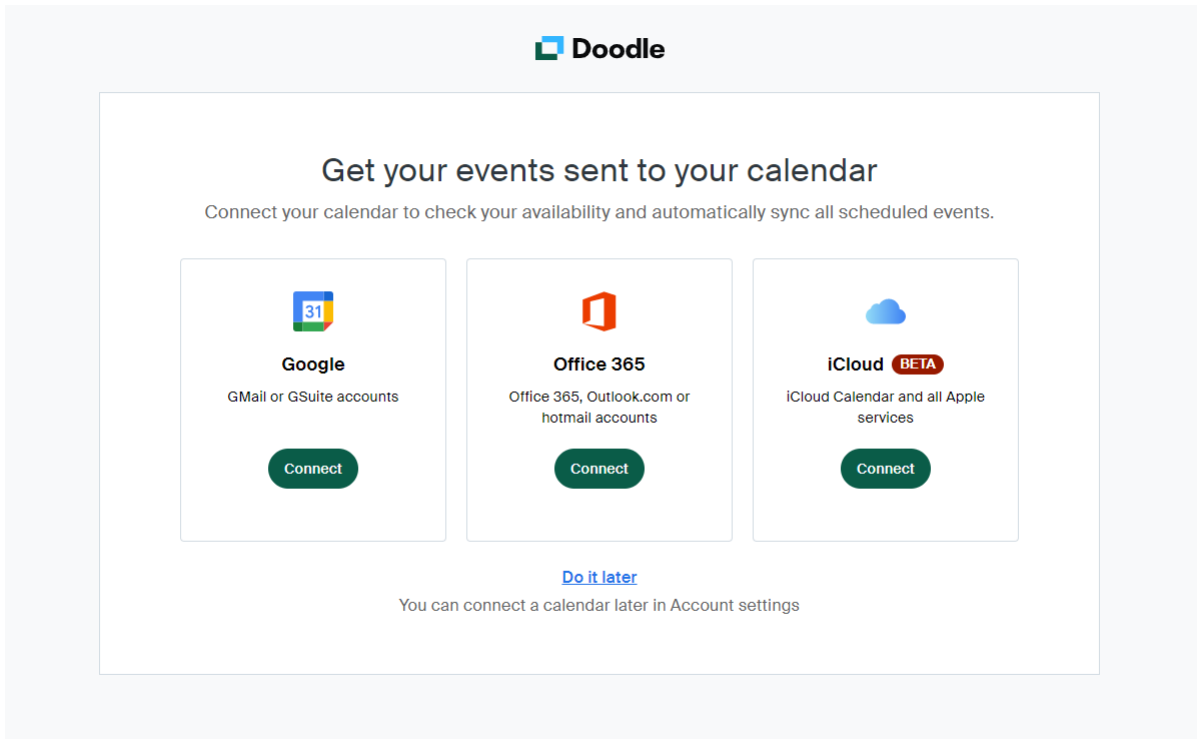
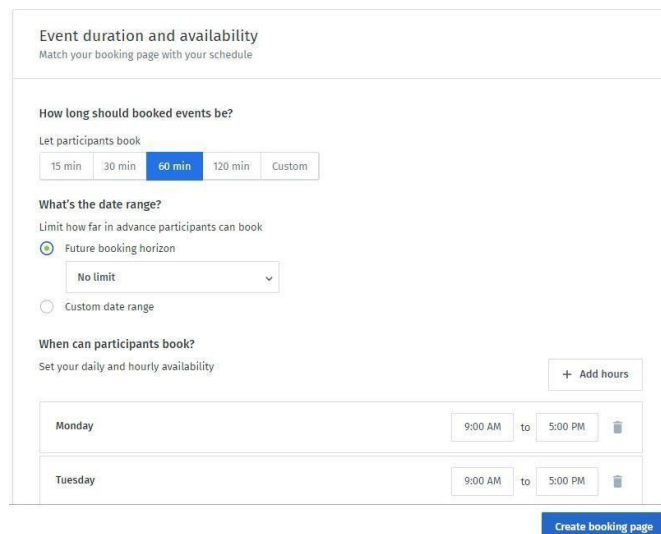


Figure 2.2.5 Web version of Doodle’s Affiliates

Figure 2.2.1 depicts the affiliation between Google Calendar, iCloud, and Microsoft Office 365 with Doodle. The system accepts Google Calendar when it is imported, and there are other capabilities that are available that allow users to do things like simultaneously modify two calendars on one platform. The Doodle calendar will automatically update in Google Calendar after a user makes some modifications.





*Figure 2.2.6 Web Version of Doodle to Create a Booking*

Doodle allow users to customize the event durability and attendees' availability by timing. Host can also set the participants could make the bookings by which day of time. If you've found that there are some of the attendants haven't booked any event time, the system will automatically remind them through notifications. To notify the members on the official booking page is created, there are two ways where one of it is to share the link provided or host could email the participants.

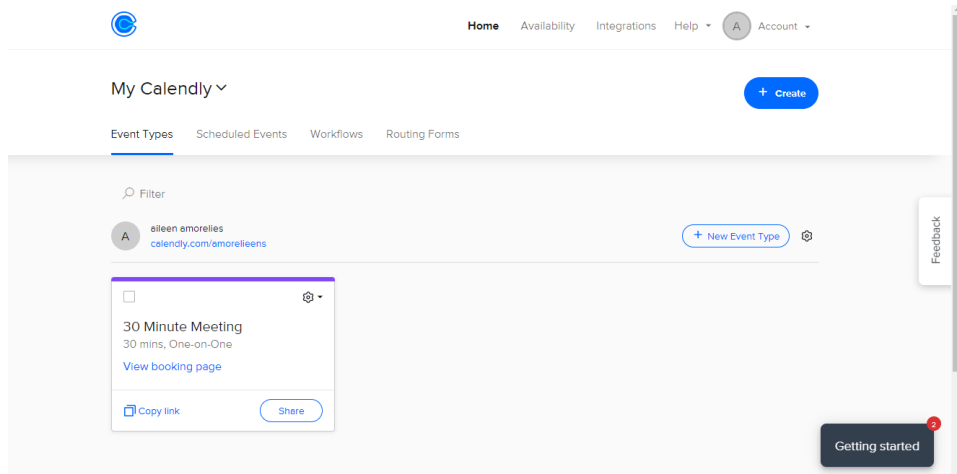
**2.2.2 Summary of Strength and Weakness of Doodle system tool**

The Doodle system tool stands out as a comprehensive scheduling platform with several notable strengths. First, it boasts a plethora of customization options, enabling users to tailor their scheduling experience to specific needs and preferences. This personalized touch is further augmented by the system's capability to send automatic reminders and notifications to participants, ensuring punctuality and attendance. Another hallmark of Doodle is its seamless integration with a multitude of calendar platforms, which provides users with a streamlined experience and minimizes scheduling conflicts. Furthermore, its capacity to efficiently coordinate and schedule group meetings with multiple participants makes it indispensable for larger gatherings and collaborative endeavors. Together, these features underscore Doodle's versatility and user-centric design in addressing modern scheduling challenges.

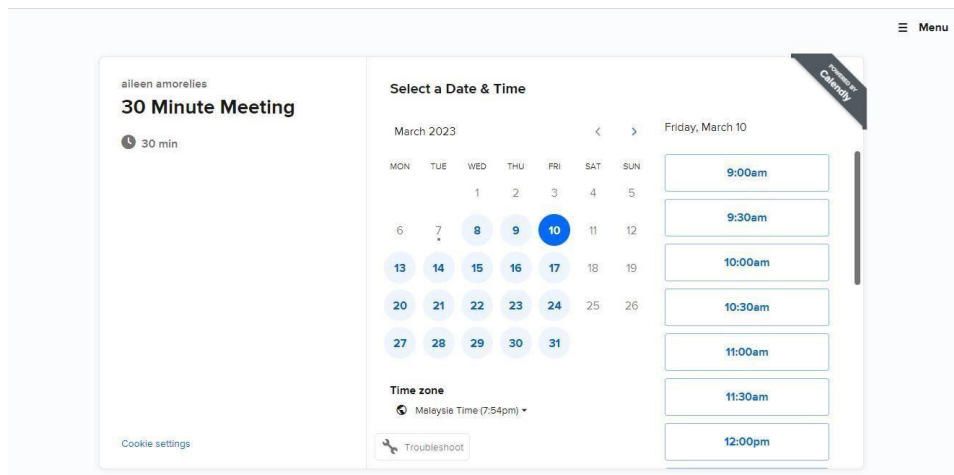
While the Doodle system tool offers a myriad of strengths, it isn't without its shortcomings. For some users, especially those less acquainted with digital tools, navigating the interface can be somewhat confusing, leading to potential missteps and a steeper learning curve. Another area where Doodle might fall short is in its control over time zone settings. The limited flexibility in this domain can occasionally lead to scheduling hiccups, especially when coordinating meetings across different regions. Despite these weaknesses, it's essential to recognize that Doodle remains a widely used application that addresses many of the modern scheduling challenges, but like all tools, it has areas for potential improvement.

### 2.2.3 Calendly

Calendly is a go-to resource for setting up meetings effectively and professionally, doing away with the burden of back-and-forth emails so users could resume their work. Calendly, which is already known as the leader in easy scheduling for more than 50,000 businesses and 10 million users worldwide, now provides enhanced capabilities for teams with strong collaboration capability, solid administrative controls, and enterprise-grade security and compliance.

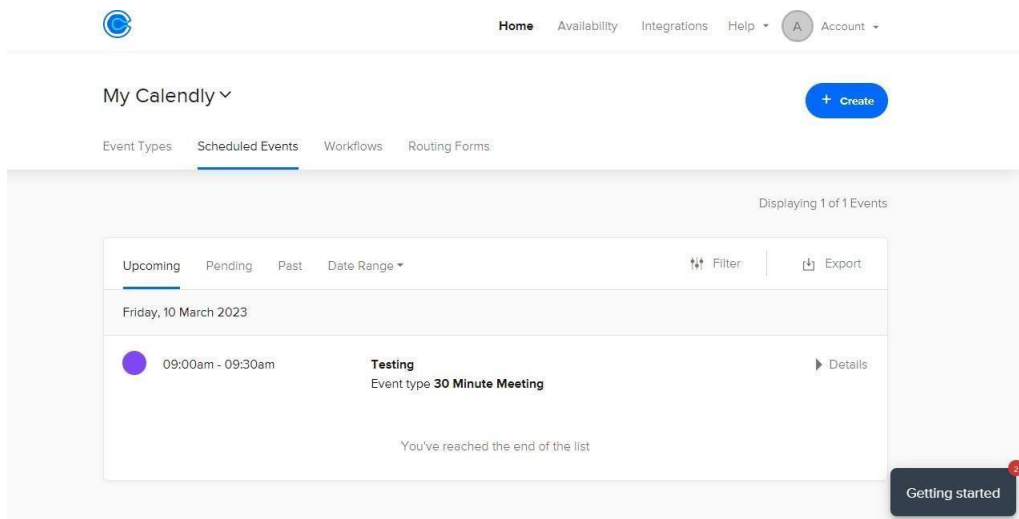


F *Figure 2.2.7 Web Version of Calendly Dashboard*



*Figure 2.2.8 Web Version of Calendly Calendar for Bookings*

To automatically verify availability and assist users in connecting with their most valuable contacts, prospects, and customers, Calendly can link with up to six of other calendars. Other meeting scheduling tools focus more on keeping your calendar tidy, preventing double bookings, and preventing overcommitting.



*Figure 2.2.9 Web Version of Calendly Scheduled Events*

User can automate all the work to be performed in relation to events, including sending text messages when reservations are made and emails with reminders. They can either start with a typical workflow or design their own. Routing forms enable you to request information from users up front and route them to a certain booking page or website in response to their input.

### **2.2.3 Summary of Strength and Weakness of Calendly**

Calendly stands as a robust scheduling solution tailored to modern professional needs. One of its primary advantages lies in the broad spectrum of customization options it presents, allowing users to adapt and mould their scheduling environment in alignment with their preferences and requirements. Moreover, Calendly doesn't just stop at scheduling; it extends its capabilities to offer insights with analytics of recent activities, ensuring that users stay informed about their scheduling patterns and engagements. This feature is particularly beneficial for businesses and professionals keen on monitoring their client or team interactions. Further enhancing its usability is its ability to seamlessly integrate with various calendar platforms, creating a unified and integrated scheduling experience for users across different systems.

However, no system is devoid of challenges, and Calendly is no exception. One of the sticking points for many users is the limited control it provides over time zone settings. In an increasingly globalized world, where cross-border interactions are commonplace, this limitation can sometimes cause scheduling misalignments or misunderstandings. Additionally, the need to manually fill up certain information can be a bit cumbersome, especially for users seeking a more automated and streamlined scheduling process. These nuances underscore the fact that while Calendly offers a range of impressive features, there's still room for improvement in specific areas to enhance user experience.

### 2.3 Comparison and Limitation of all Existing Systems

Application name	Application Features			
	Calendar Integrations	Automated scheduling	User-friendly Interface	Customizable meeting settings
WhenAvailable (Tope Awotona, 2020)	√		√	√
Doodle (Contarini Ventures, LLC, 2019)	√		√	√
Calendly (Doodle AG Werdst., )	√	√	√	√

*Table 2.4.1 Comparison and Limitation Table of All Existing Systems*

Table 2.4 depicts the summary of comparison on all existing application. There are a few dissimilar features for each reviewed existing system. Significantly that WhenAvailable.com did not proposed automatic scheduling a meeting by identifying available time slots for all participants. Doodle application has almost all the features calendar integrations, user-friendly interface, and customizable meeting settings except for automated scheduling where their application still requires user to manually setup the meetings by polling. Calendly however on the other hand applied most of the feature reviewed.

## 2.4 Summary

The reviewed systems mostly have almost similar features that can be integrated in the implementation specifically features of allowing the users to schedule a meeting with several participants in groups. In conclusion, users may simply book meetings using the online scheduling tool Calendly. and appointments with others. Users can create custom booking links that can be shared with others, and the recipient can then choose from available time slots that are displayed on a calendar. Calendly integrates with a wide variety of calendar platforms, including Google Calendar, Outlook, and iCloud. WhenAvailable.com is a free online scheduling tool that allows users to easily schedule meetings and appointments with others. Users can create custom booking links and share them with others, who can then select from available time slots on a calendar. WhenAvailable.com integrates with Google Calendar and can also be used with other calendar platforms through iCal feeds. Doodle is a website that allows users to effortlessly plan meetings and appointments with others. Users can create a poll with different date and time options, and then share the poll with others to vote on their availability. Doodle integrates with many calendar platforms, including Google Calendar, Outlook, and iCloud. These are the summaries made after reviewing all three scheduling tools. However, these scheduling tools on the other hand consists of similar weaknesses as well such as limited time zone settings. The limited time zone settings in a calendar app usually refer to the ability to view and set time zones for events or appointments within a specific range of time zones. For instance, a calendar app might allow you to select time zones within a certain region, such as the Americas, Europe, or Asia-Pacific. Or it might limit you to only viewing time zones within a certain number of hours ahead or behind your local time zone. These settings can be helpful when scheduling meetings or events with people in different time zones, as they allow you to easily see what time it will be in each person's local time zone and avoid scheduling conflicts. It's worth noting that some calendar apps may also allow you to manually input any time zone you like, regardless of whether it falls within the app's limited settings.

## **CHAPTER 3: SYSTEM THEORETICAL & METHODOLOGY**

### **3.1 Equation Implemented**

Implicit Interval algorithms is implemented in the systems. In computer science, binary search trees (BSTs) are fundamental data structures that offer efficient sorting and straightforward update operations due to their linked node structure and implicit key sorting. When they are balanced, they perform at their best. A regular linked list is more effective than both balanced and unbalanced binary search trees. Numerous strategies for balancing Binary Search Trees were suggested in the recent literature. The goal of earlier studies was to find the optimum balanced binary search tree approach for a given data management scenario; however, recently proposed solutions have not yet been thoroughly investigated. To help software developers select the most effective balanced Binary Search Trees method, we have compared the traditional and most modern approaches in this study. [11]

The intricacies of interval trees with received queries compromising intersections of scope point and extract all the scope from the original assembly. There are several aspects we can discuss regarding the algorithm. In an interval query, this project strives to detect all the intervals from the tree that cross paths with the designated span. If any of the interval is considered from the tree, so then it will intersect with the given query interval, either the start or the end of the interval from the tree should lie within the query interval or it should completely enclose the query interval. A binary search tree, composed of the start and end points of intervals within the interval tree, can be built to determine the intersections. This allows for easy lookup of intervals with start or end points within the query interval. Constructing a balanced binary search tree (e.g., AVL or Red-Black Tree) will typically take  $O(n \log n)$  time when given  $n$  elements. This is because, for each element, you're inserting into a tree and ensuring it's balanced, which takes  $O(\log n)$  time per insertion. Additionally, one must search the interval tree for intervals that may surround the query interval. [13-14]

Furthermore, deleting an interval from an interval tree is more involved than simply removing a node from a standard binary search tree. Given that intervals can span across multiple nodes in the tree (since the nodes may intersect and overlap multiple 'center' points), deletion is a must ensure that the tree's structural and functional integrity is maintained. However, deleting an interval might lead to empty nodes, and these nodes must be removed from the tree. This involves adjusting other nodes and ensuring intervals are appropriately stored in the correct nodes. Deletion may involve promoting another node to take the place

of the deleted node, ensuring that all intervals are appropriately shifted and stored. [15-16]

Balancing is also an aspect that must not be overlooked, as it is essential for maintaining the efficiency of an interval tree. Specifically, AVL trees and Red-Black trees, rotations might be required to balance the tree but rotations in interval trees are more complicated due to the overlapping nature of intervals. During rotations, one must ensure that intervals are correctly positioned, considering the tree's core property – intervals overlapping a node's center point must be stored in that node. In a nutshell, while interval tree is a powerful data structure allowing efficient querying for overlapping intervals, operations like insertion, deletion and balancing require careful handling due to the overlapping nature of stored intervals. [42-43]

$O(n \log n)$

```
def find_soonest_slot(busy_slots, duration, after_time=None):
    merged_slots = []
    for slots in busy_slots.values():
        merged_slots.extend(slots)
    merged_slots.sort(key=lambda x: x['start_time'])

    current_time = datetime.utcnow()
    if after_time:
        current_time = max(current_time, parse(after_time))

    for i in range(len(merged_slots) - 1):
        gap = merged_slots[i+1]['start_time'] - merged_slots[i]['end_time']
        if gap >= duration and current_time <= merged_slots[i]['end_time']:
            return merged_slots[i]['end_time'], merged_slots[i]['end_time'] + duration

    # If no slot is found within the busy times, return the time after the last busy slot
    if merged_slots:
        return merged_slots[-1]['end_time'], merged_slots[-1]['end_time'] + duration
    else:
        return current_time, current_time + duration
```

Figure 3.1.1 Source Code Implemented with Interval Algorithm

The 'for' loops that populates 'merged\_slots' is linear with respect to the total number of slots across all users or entities used the operation of  $O(n)$ . The sorting step 'merged\_slots.sort(...)' implements the complexity of  $O(n \log n)$  for sorting  $n$  intervals.

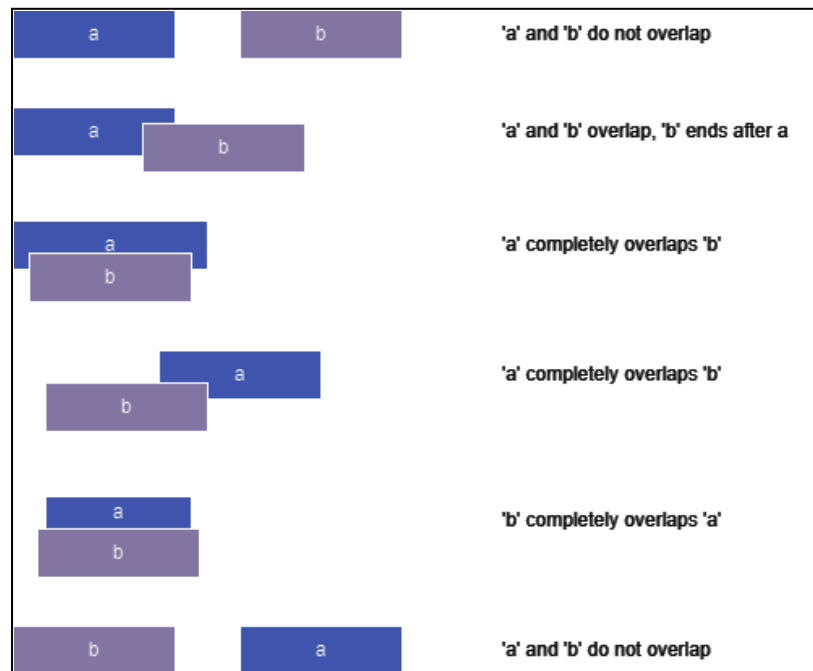


Figure 3.1.2 Pattern Merge Interval for start time and end time between users.

The process of finding overlapping intervals as discussed revolves around identifying intervals that intersects with a query interval. In the project, this concept is mirrored by the search for the next available time slot within the busy slots' intervals with the function 'find\_soonest\_slot'. The method aggregates all slots from 'busy\_slots' and sorts them based on their 'start\_time'. Each time slot can be visualized as an interval, and the function's task is to find a gap (or interval) between these 'busy\_slots' that is large enough to fit the requested duration by user in an iteration. The idea of constructing a binary search tree of interval start and end points is mentioned in the provided description. Similarly, in the code, all the busy slots are merged into a single list and then sorted based on the start time. This sorting is crucial as it allows for a linear check to find the next available slot.



```

INFO:apscheduler.executors.default:Job "main (trigger: interval[0:10:00], next run at: 2023-09-13 22:25:31 +08)" executed successfully
INFO:apscheduler.executors.default:Running job "main (trigger: interval[0:10:00], next run at: 2023-09-13 22:35:31 +08)" (scheduled at 2023-09-13 22:25:31.160079+08:00)
INFO:googleapiclient.discovery_cache:file_cache is only supported with oauth2client<4.0.0
INFO:apscheduler.executors.default:Running job "main (trigger: interval[0:10:00], next run at: 2023-09-13 22:35:43 +08)" (scheduled at 2023-09-13 22:25:43.117740+08:00)
INFO:googleapiclient.discovery_cache:file_cache is only supported with oauth2client<4.0.0
INFO:apscheduler.executors.default:Job "main (trigger: interval[0:10:00], next run at: 2023-09-13 22:35:31 +08)" executed successfully
INFO:apscheduler.executors.default:Job "main (trigger: interval[0:10:00], next run at: 2023-09-13 22:35:43 +08)" executed successfully
INFO:apscheduler.executors.default:Running job "main (trigger: interval[0:10:00], next run at: 2023-09-13 22:45:31 +08)" (scheduled at 2023-09-13 22:35:31.160079+08:00)
INFO:googleapiclient.discovery_cache:file_cache is only supported with oauth2client<4.0.0
INFO:apscheduler.executors.default:Running job "main (trigger: interval[0:10:00], next run at: 2023-09-13 22:45:43 +08)" (scheduled at 2023-09-13 22:35:43.117740+08:00)

```

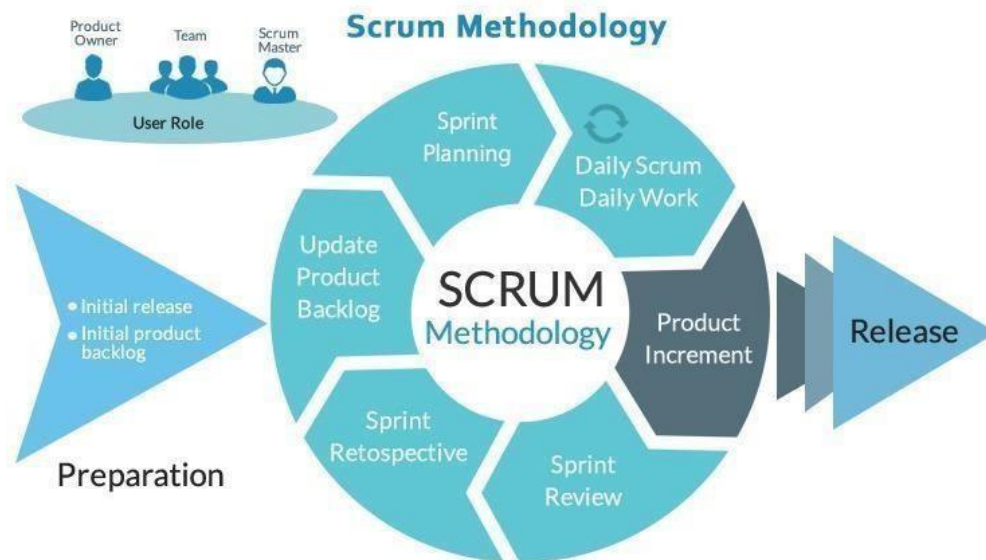
*Figure 3.1.3 APScheduler Running Backend to Retrieve New Data*

The deletion of intervals from interval trees is complex. This complexity arises from ensuring the tree remains balanced and maintains its structural integrity especially in the project the data are extracted from Google Calendar. The project handles the deletions in the configurations of data cleaning in the collection, so it only stores the most recent slots for all users whenever APScheduler is refreshed (Figure 3.1.3). Another scenario is handled to balance the tree where no suitable slot is found (i.e., no overlaps with available time). The function by default will return the time after the last busy slot.

The Advanced Python Scheduler (APScheduler) appears to be running a job called "main" at 10-minute intervals. This can be inferred from the consistent "trigger: interval [0:10:00]" mentioned in the logs. Every time this job runs, it's set to execute again after the next 10 minutes, as indicated by the "next run at" timestamps. The scheduler appears to be functioning correctly, with each job's execution being logged as successful upon completion.

Each time the "main" job is executed, there's a consistent log entry from the `googleapiclient.discovery_cache` which states: "file\_cache is only supported with `oauth2client<4.0.0`". This warning indicates that the version of `oauth2client` currently being used is either at 4.0.0 or a newer version, and hence, the file cache functionality might not be supported. This could potentially affect the performance or behavior of Google services integration within the application, as caching can play a vital role in the efficiency of API requests.

### 3.2 Agile Methodology – SCRUM Process



*Figure 3.2.1 SCRUM Approach Process*

SCRUM approach will be deployed in this project as the software application development methodology. The rationale for using this technique model (Figure 4.3) is due to the continuous evolution of technology and working algorithms and procedures that are constantly changing. The SCRUM approach is deemed the most appropriate approach for the project since it is highly adaptable and allows for the reduction of issues during and after development. The primary objective of the SCRUM approach is to satisfy the customer's needs. It is also easy to modify after development, and amendments can be made without much difficulty. The SCRUM approach is an agile framework that allows for flexible and iterative project management. It is especially useful for software development projects where changes are inevitable, and customer satisfaction is of utmost importance. In contrast to the traditional waterfall methodology, the SCRUM approach is highly adaptable and allows for continuous improvement throughout the development process. This makes it the ideal choice for this project, where changes in technology and procedures are common.

The project planning will prioritize the objectives as a balance of the value and the cost thereof. This includes determining iterations and deliveries. The project team must be agile and flexible in developing products as the market demands higher quality and faster delivery at lower costs. Short development cycles that meet customer demands without compromising the quality of the results are required. This is a straightforward methodology for teams that are seeking rapid results and is popular among several developers. Project planning is an essential part of the SCRUM approach, which prioritizes objectives based on their value and cost. This ensures that the most important tasks are tackled first, and resources are allocated appropriately. The team must strike a balance between the value of each objective and its cost, ensuring that the project remains within budget. The SCRUM approach emphasizes the need for agility and flexibility in product development to meet customer demands. Customers today demand high-quality products delivered quickly and at a lower cost. This requires short development cycles that can meet these demands without compromising the quality of the results. The SCRUM approach is a straightforward methodology that is popular among many developers as it provides rapid results. [7]

The development of the project begins with the identification of the things that need to be built and a list of detailed characteristics that are arranged in priorities. This comprises the initial release and product backlog that the product owner desires. Sprint planning meetings will be held at the start of each sprint, with participation from all team members, including the product owner, scrum master, and development team. The team must grasp the objectives that can be accomplished in each sprint, including the sprint goal. The development team should construct a work plan, including the workload, to meet the goal. This plan should be noticeable in two to four weeks. The client will then present the desired result for each sprint, along with their requirements for the deliverable product. The team must then evaluate which of the elements provided by the client can be delivered. The product owner and scrum master must declare their requirements. Finally, the development team must organize the team's work to achieve the sprint goal.

### 3.3 Project Timeline

This Gantt chart will depict the estimated timeline for deliverable and milestones of my Final Year Project 1. My objective is to complete my Final Year Project over the course of two semesters. The documentation for FYP1 will be focusing on requirements, available functionalities, system flow, and interface design during the first semester. This will make the development process more efficient and effective. Furthermore, in FYP1, the first prototype will be constructed to execute functions such as login, logout, account registration, and integration of Google Authentication.

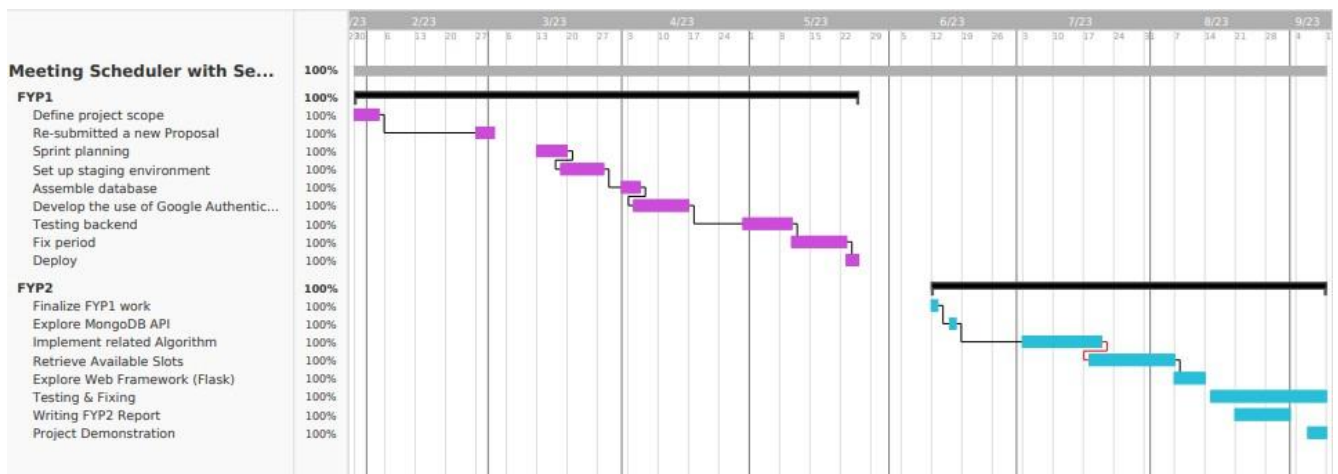


Figure 3.3.1 Gantt Chart for FYP 1 & 2

**CHAPTER 4: SYSTEM DESIGN****4.1 System Components Specifications Requirement****4.1.1 Hardware Requirements**

The hardware required to develop the Meeting Scheduler Web Application with Secured Privacy are as follow.

<b>Hardware</b>	<b>Specifications</b>	<b>Description</b>
Processor	Intel® 7 <sup>th</sup> Gen or newer CPU 3GHz or higher	How fast the computer can perform the function
Memory	16GB	Used to store the file in computer
Graphic Card	NVIDIA GeForce GTX1650	To display the graphic, user interface
Input Device	Mouse, Keyboard	To control the input data

*Table 3.2.1 Hardware Requirements*

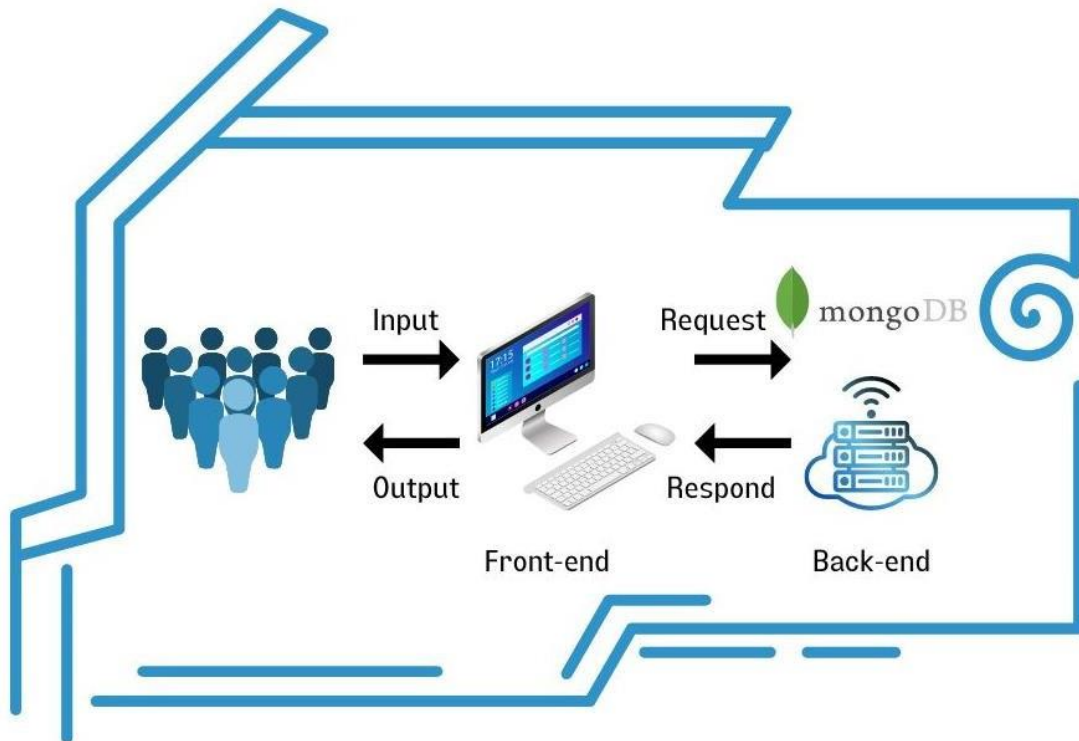
**4.1.2 Software Requirements**

The software required to develop the Meeting Scheduler Web Application with Secured Privacy are as follow.

<b>Software</b>	<b>Specifications</b>	<b>Description</b>
Operating System	Windows 10	Provide service for application software
Coding	Python, Flask	Used to develop the web application backends
Database	MongoDB	Used to store the data of the user's information
Monitor	16:9 (Desktop)	Screen-to-body ratio

*Table 3.2.2 Software Requirements*

## 4.2 System Architecture Diagram



*Figure 4.2.1 System Architecture Design*

A web application's architecture describes how its many elements, including user interfaces, databases, and servers, communicate with one another. Essentially, it depicts the logical connections between the client-side and server-side elements of the application, resulting in an improved user experience [5]. The diagram presented above illustrates the system architecture of a website application meeting scheduler information system. To access and utilize the system's functionalities, users must make use of a computer or laptop as an input device. This input device retrieves or modifies data from the database located at the backend of the system. Once the database has retrieved or updated the required data, it will be displayed on the user's computer or laptop screen.

### 4.3 System Design

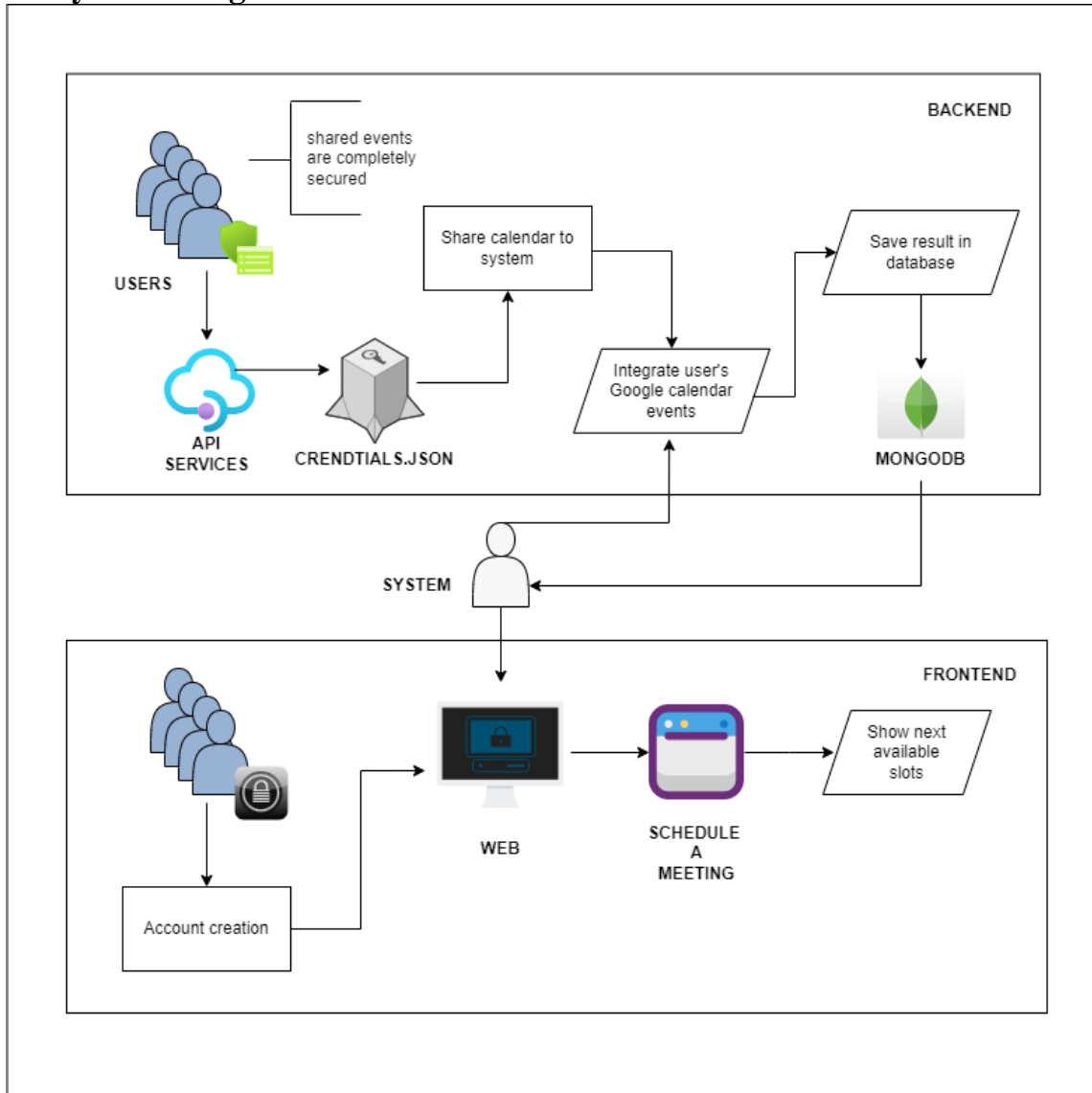


Figure 4.3.1 System Design

The process from the diagram above depicts of a user interacting with a system that allows for account creation and meeting scheduling. The system will begin with the user sharing their calendar to the primary account whereas it'll act as the middleman to share everyone's calendar with the system. The calendars events are totally secured having the option of logging in or creating a new account. After logging in, the user is taken to the dashboard, where they can then further to schedule a meeting. When users want to conduct a new meeting, the system automatically show the soonest available slots for all the emails of the favorable member. The system then shares availability slots for the meeting. The figure above emphasizes the significance of user identification and security by needing Google credentials for calendar integration, ensuring that the solution is user-friendly and practical.

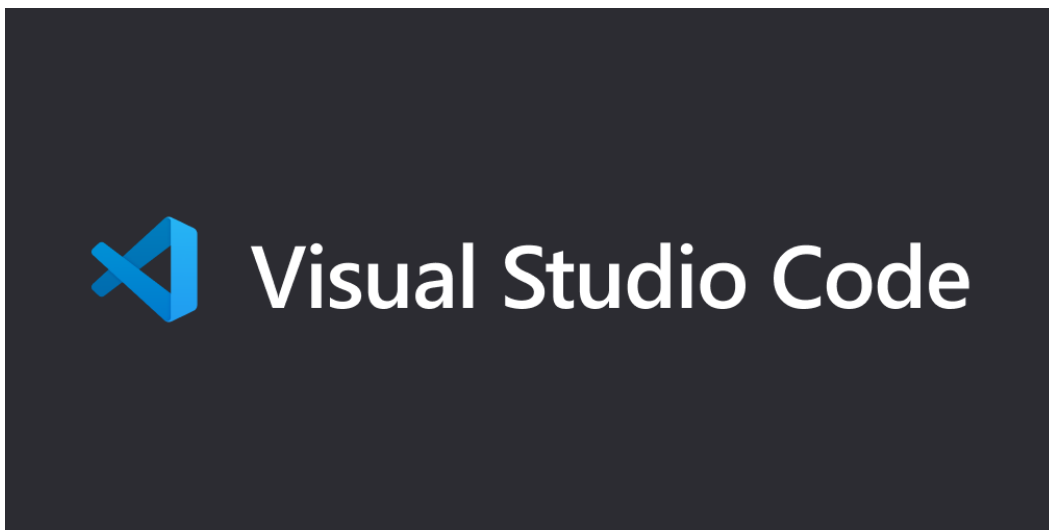
## **CHAPTER 5: SYSTEM IMPLEMENTATION**

### **5.1 Software Setup**

#### **5.1.1 Integrated Development Environment/Software Tools**

An integrated development environment or in short IDE is a software application which provide a comprehensive tool to the programmer for software development.[3] The integrated development environment that selected for this project is **VS Code** which is also known for Visual Studio Code.[24] The free source-code editor VS Code, created by Microsoft, was initially made available in April 2015.[25] It can run on Windows, Linux, and macOS. The developer can therefore continue their work on a new platform.[26]

VS Code is an IDE that offers a wide range of essential features. It provides a lightweight and customizable interface for developing and debugging several programming languages such as Python, Java, JavaScript, et cetera.[28] Visual Studio Code supports various extensions that can enhance its functionality and features, making it a popular choice for developers across various industries. It is also often used for web development, especially for front-end development using frameworks such as React, Angular, and Vue.js.[27]



*Figure 5.1.1 Logo of Visual Studio Code*



### 5.1.2 Programming Languages

Programming languages are utilized by programmers to provide computer instructions. These languages are also known as high-level languages, which are understandable by computers. They allow developers to create software programs and scripts, as well as write, test and debug codes. The world has hundreds of programming languages, including Python, HTML, JavaScript, and C#, to mention a few.

The chosen programming language for this project is Python, which was created by Guido van Rossum in 1991 and is one of the most widely used programming languages worldwide. Python is applicable for web development, software development, system scripting, artificial intelligence, and scientific computing. It is compatible with various platforms such as Mac, Linux, Windows, and others. Compared to C#, Python has a more extensive library system with over 137,000 libraries, making it easier for developers to manage databases, web browsers, unit testing, and more.

Python was chosen as the programming language for numerous reasons. To begin, Python provides a diverse set of frameworks that can help programmers save time and effort. Django, FastAPI and Flask, for example, are great for web development, whereas CherryPy are ideal for automated for real-time web applications. Furthermore, when compared to other programming languages, Python is known for being simple to understand and write, with a structure like that of the English language. As a result, Python programs are easier to maintain. In the event of an error, the programmer can concentrate on resolving the problem rather than dealing with the language's complexities, resulting in a more efficient debugging and maintenance process.

### 5.1.3 Framework and its Configurations

Python is the programming language used in this project; hence Flask is the framework that will be used in this project.[20] Flask is a lightweight, micro web framework in Python, offering the foundational tools required to build web applications without imposing additional functionalities. [19] The term "micro" underscores its core philosophy: providing the essentials while leaving other decisions, like choosing specific tools or extensions, to the developer. This approach gives Flask a unique balance of simplicity for beginners and flexibility for seasoned developers. At its core, Flask offers features like a built-in development server that updates in real-time and an interactive debugger. It's optimized for RESTful request dispatching, making it intuitive to build and expose APIs. One of its standout features is the integration of the Jinja2 templating engine, which facilitates the dynamic rendering of content in HTML templates.

While Flask starts minimalistic, its capabilities can be significantly expanded through a plethora of available extensions, from authentication mechanisms to form handling utilities.[21] Another noteworthy feature is its support for secure cookies, allowing session data to be stored directly on the client side, enhancing both security and performance. Additionally, Flask's inherent support for Unicode ensures that applications can handle a vast range of characters and symbols right out of the box. In essence, Flask's popularity in the developer community stems from its modular and scalable nature.[23] Developers have the freedom to start with a simple structure and expand it based on project requirements, avoiding the overhead of unnecessary components. Whether the goal is a basic web page or a comprehensive RESTful API, Flask has proven itself as a reliable and efficient choice in the modern web development landscape.[22]

Flask's beauty lies in its simplicity and the freedom it offers developers. By setting up a virtual environment, structuring the app judiciously, integrating necessary extensions, and adhering to best practices, one can harness the full potential of Flask. Whether a newbie or a seasoned developer, Flask's adaptability ensures everyone can craft web applications tailored to their vision and requirements.

## CHAPTER 5

### Flask's Anatomy and Structure

Once within this protective cocoon, Flask can be installed simply with `pip install Flask`. With Flask in place, one should pay heed to structuring the app properly. A typical Flask project might be compartmentalized into:

- A primary application package.
- A templates folder for Jinja2 templates, which powers Flask's frontend rendering.
- A static directory housing static assets like CSS, JavaScript, and images.
- A runner script, often named `run.py`, responsible for firing up the development server.
- This structure not only offers clarity but also supports scalability as the project grows.

### Configurations and Security

Flask provides an innate way of handling configurations: the `app.config` dictionary. While configurations set parameters for app behavior, they also play a pivotal role in security. For instance, the `SECRET_KEY` configuration, vital for sessions and CSRF protection, should be kept secure. To bolster security, environment variables, often managed with tools like `python-dotenv`, can store these sensitive details.

### Empowering Flask with Extensions

Flask's minimalist nature is by design, allowing developers to augment it with extensions suited to their needs. Common extensions include `Flask-SQLAlchemy` for databases, `Flask-Login` for user authentication, and `Flask-WTF` for web forms. These extensions, installed via `pip`, empower Flask with added functionalities, turning the 'micro' framework into a powerhouse.

### Versioning, Testing, and Deployment

Just like any software project, Flask apps benefit from version control. Tools like `Git` allow developers to track changes, collaborate seamlessly, and maintain a project's history. Moreover, Flask's compatibility with testing frameworks like `pytest` ensures the app remains bug-free and performs as expected. Once developed, the app can be deployed on various platforms like `Heroku`, `AWS`, or `DigitalOcean`, with each having its own set of guidelines and processes.

### 5.1.4 MongoDB: Revolutionizing Database Management in the Modern Age

In the ever-evolving landscape of software development, the tools, and technologies that developers use are of paramount importance. Among the plethora of tools available, databases form the bedrock of most applications. While traditional relational databases have served us well for decades, the surge in data volume and complexity in today's digital age calls for more flexible and scalable solutions. Enter MongoDB, a leading NoSQL database that has positioned itself at the forefront of this new wave of database technologies. [9] [6]

What sets MongoDB apart is its adoption of a document-oriented model. Unlike the rigid, table-based structures in relational databases, MongoDB uses JSON-like documents, termed BSON to store data. This design choice offers developers a high degree of flexibility. In traditional systems, any change in data structure might necessitate a thorough database redesign, a time-consuming endeavor. With MongoDB's schema-less approach, however, different documents in a single collection can have varied fields. Moreover, the data type for a common field can differ across documents. This dynamism ensures that applications can adapt and evolve swiftly, making MongoDB particularly suited for agile development environments. [18]

Performance optimization is another area where MongoDB shines. Efficient data retrieval is critical for applications, and MongoDB's robust indexing support ensures just that. Developers can create indexes on any field, even if nested within arrays or documents, ensuring swift data lookups. Furthermore, its rich query language, allowing data manipulation, aggregations, and intricate filtering, provides developers with a comprehensive toolset to interact with their data. [17] Lastly, MongoDB's open-source nature cannot be overlooked. By providing its source code under the SSPL, MongoDB fosters a spirit of community collaboration, allowing developers worldwide to contribute, customize, and integrate it into diverse applications.[8]

In conclusion, MongoDB represents a paradigm shift in how we approach database management. Its emphasis on flexibility, scalability, and performance addresses the core challenges faced by modern applications. As we continue to generate and rely on vast amounts of data, tools like MongoDB will undoubtedly play a pivotal role in shaping the future of digital innovation.

### 5.1.5 Google Calendar API Integration

1. Setting Up the Google Developer Console:

To start, you'll need to set up a project within the Google Developer Console. Navigate to the Google Developer Console and sign in with your Google account. Once logged in, create a new project by clicking on the dropdown next to the Google APIs logo, then on the + button. Provide a name for your project and proceed.

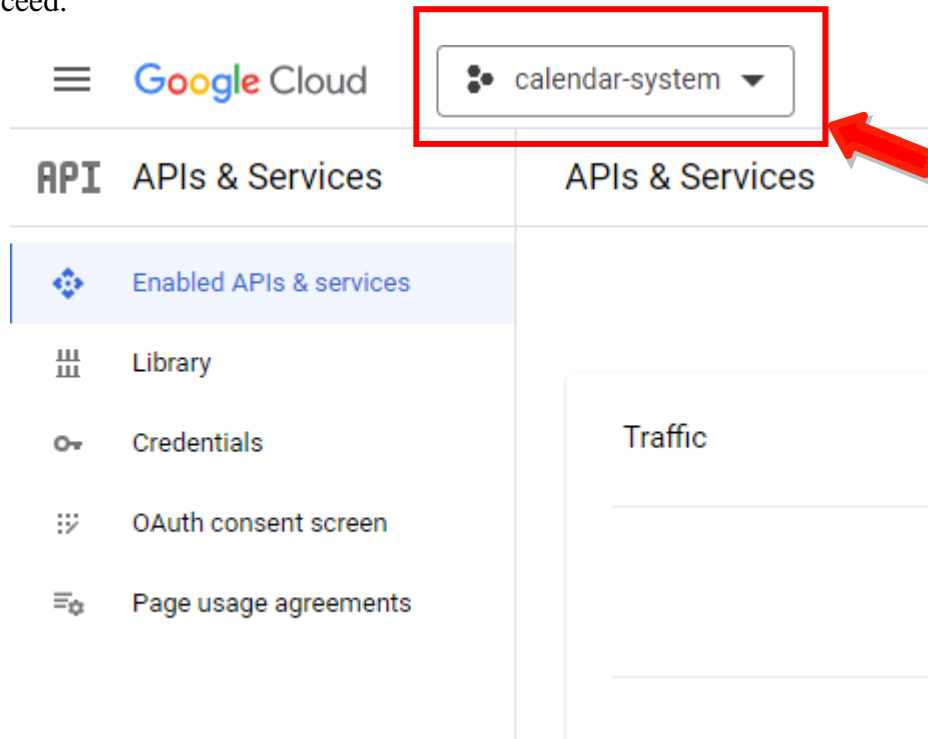


Figure 5.1.2 Google Developer Console

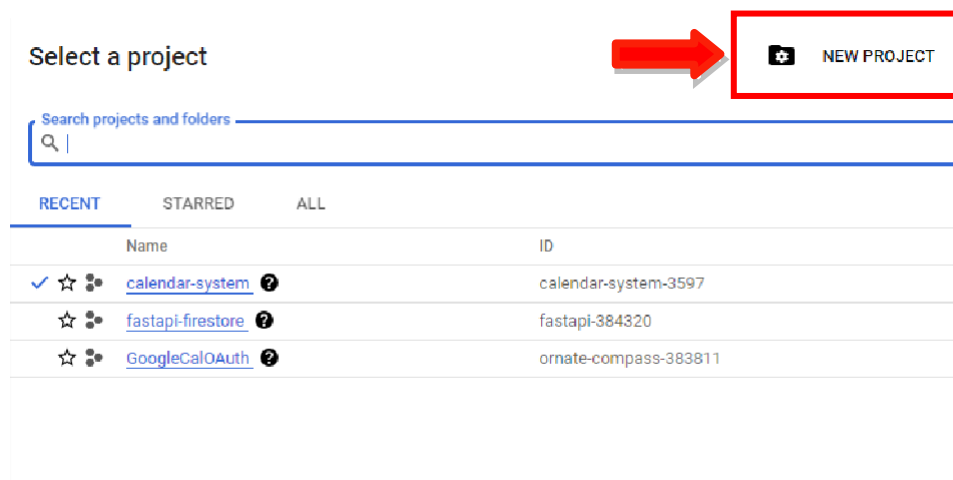


Figure 5.1.3 Add New Project

New Project

**Warning:** You have 3 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

**Project name \***  
 My Project 51622 ?

Project ID: adroit-petal-398917. It cannot be changed later. [EDIT](#)

**Location \***  
 No organization [BROWSE](#)

Parent organization or folder

**CREATE** CANCEL

Figure 5.1.4 Details of New Project

2. Enabling the Google+ API:

For Google Authentication, the Google+ API needs to be enabled. In the dashboard of your newly created project, navigate to the Library tab. Here, search for the "Google+ API" and enable it. This API allows your application to interface with Google's sign-in mechanisms.

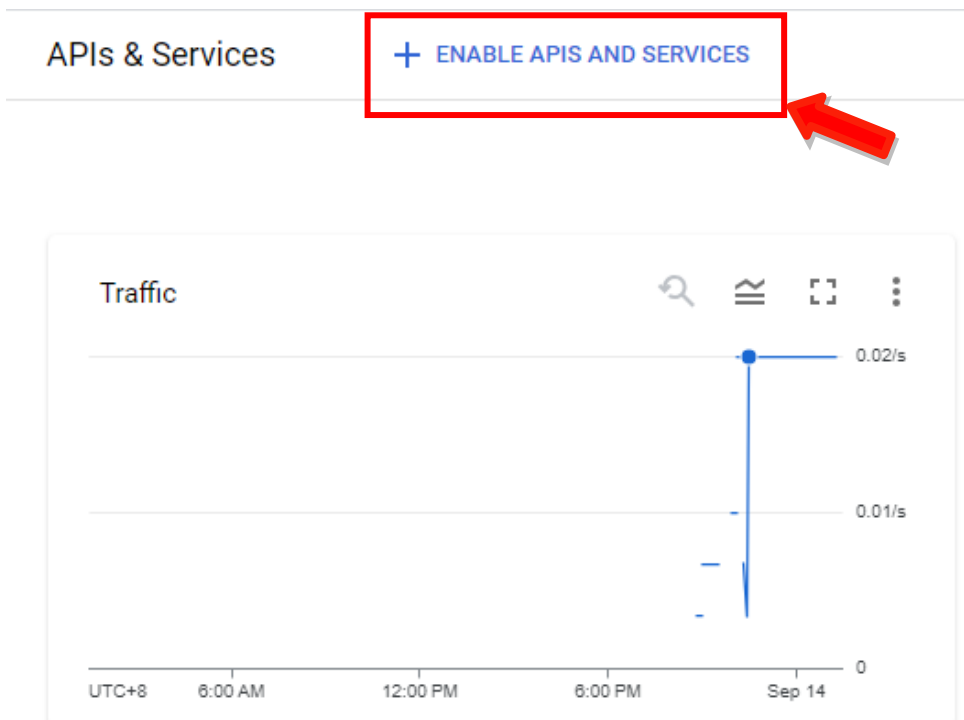


Figure 5.1.5 Enable API Services

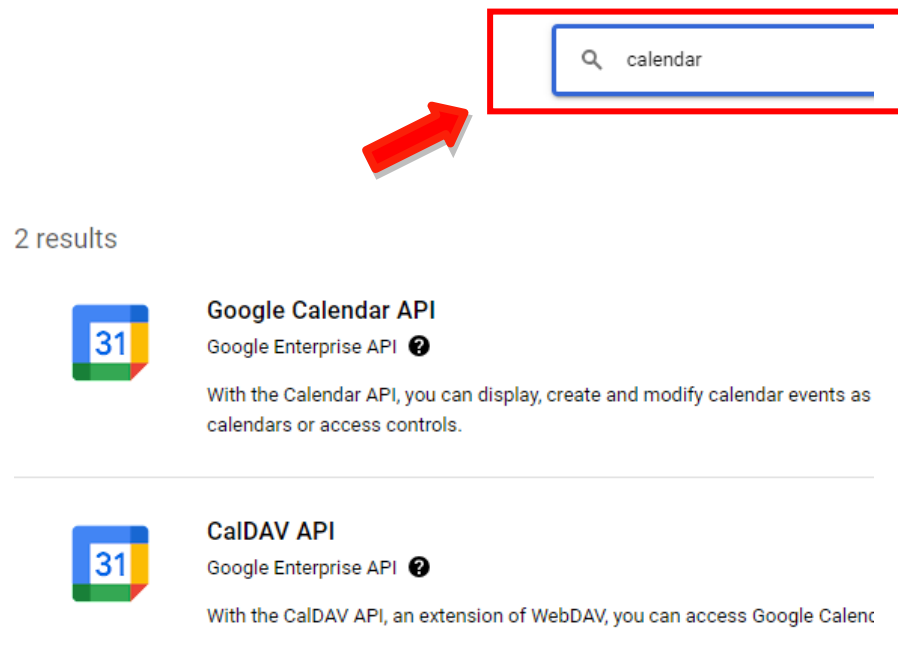


Figure 5.1.6 Search for Calendar API

### 3. Configuring OAuth 2.0:

With the API enabled, you'll need to set up OAuth 2.0 credentials. Navigate to the Credentials tab in your dashboard and click on Create Credentials. From the dropdown, select OAuth 2.0 Client IDs. You'll be prompted to set up the OAuth consent screen; fill in the required details. Then, for "Application Type", choose Web application. This will allow you to specify the authorized redirect URIs that Google should return the user to after they grant/deny permission.

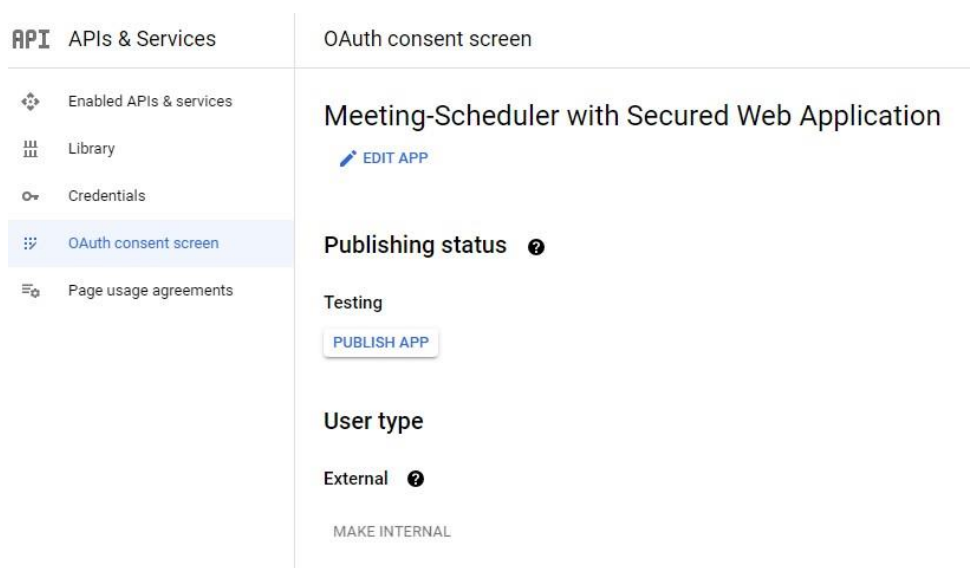


Figure 5.1.7 Configurations for OAuth Consent Screen

## CHAPTER 5

### 4. Storing Your Credentials Safely:

Upon completing the OAuth 2.0 setup, you'll be provided with a client ID and a client secret. These are vital to the authentication process and should be stored securely. They will be used in your application to initiate the Google Sign-In process and to handle the response from Google's servers.

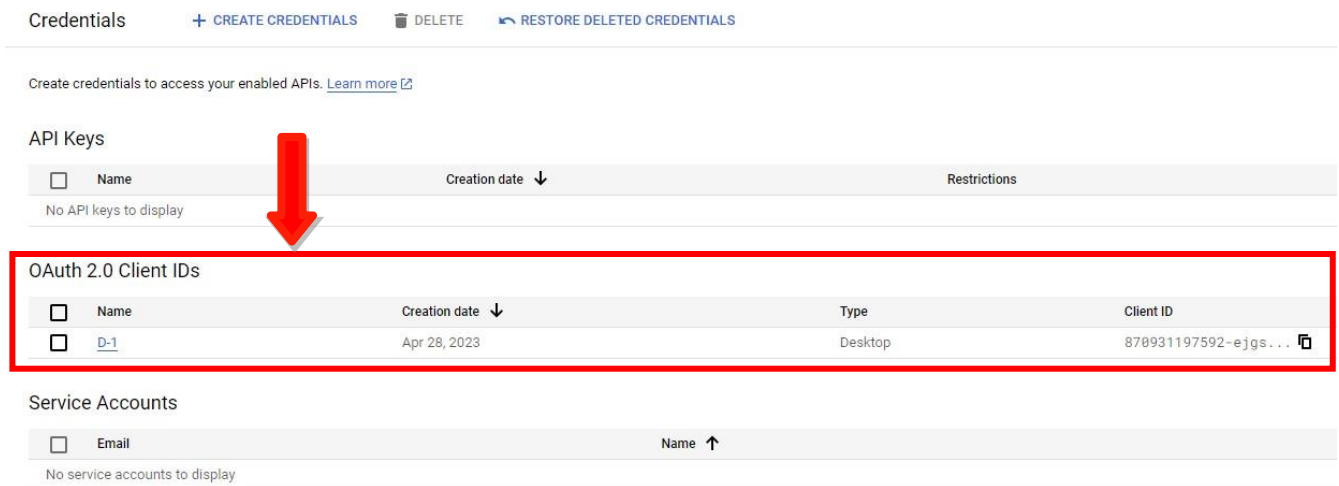


Figure 5.1.8 Generate credentials.json

### 5. Integrating Google Calendar into the Application:

Once you've secured your credentials, you can proceed to incorporate Google Calendar functionalities into your application. This involves utilizing the Google Calendar API to access, create, modify, or delete events based on the permissions provided by the user.

#### a. Adding Google Client Libraries:

Google provides client libraries in various languages like JavaScript, Python, Java, etc., which facilitate the integration. Depending on your application's platform, add the appropriate Google Client Library. For instance, if you're building a web application, you might use the JavaScript client library.

Available further steps, libraries, and examples:

```
[https://www.google.com/search?q=google+calendar+api+quickstart&rlz=1C1CHBF_enM  
Y718MY718&oq=google+calendar+api+&aqs=chrome.1.69i57j35i39i650j69i6013j69i6513.  
5062j0j7&sourceid=chrome&ie=UTF-8]
```

#### b. Initiating the Google Sign-In Process:

Within your application, prompt the user to sign in with their Google account. This process should request the specific permissions required for your needs, such as 'viewing calendar events' or



## CHAPTER 5

'editing calendar events'. This will usually appear as a pop-up or redirect to a Google sign-in page where the user can grant the necessary permissions.

### c. Accessing the Google Calendar API:

Post the sign-in process; your application will receive an access token. This token allows your application to make authenticated requests to the Google Calendar API. Using the token and the Google Client Library, you can then perform operations like fetching upcoming events, creating new events, or updating existing ones.

## 6. Testing and Troubleshooting:

After integrating Google Calendar, rigorous testing is indispensable to ensure its smooth operation.

- a. **Initiate Calendar Requests:** Begin by triggering calendar-related actions from your application. This might be fetching a list of upcoming events, creating a new event, or modifying an existing one.
- b. **Redirect Verification:** When prompting users to grant permissions, ensure they are correctly redirected to Google's permission page. Once permissions are granted, they should be seamlessly redirected back to your application without hitches.
- c. **Validate Access Tokens:** After the user grants permissions, your application will receive an access token. Verify that you're able to use this token to make authorized requests to the Google Calendar API successfully.
- d. **Check Data Retrieval and Actions:** For actions like fetching events, validate that the data received matches the user's actual calendar events. If you're allowing event creation or modification, check the Google Calendar user interface to confirm that these changes reflect accurately.
- e. **Handle Errors Gracefully:** Keep a vigilant eye out for any error messages or irregularities during these tests. If any errors arise, they should be handled gracefully in the application, providing users with clear messages or prompts on what went wrong and potential corrective measures.

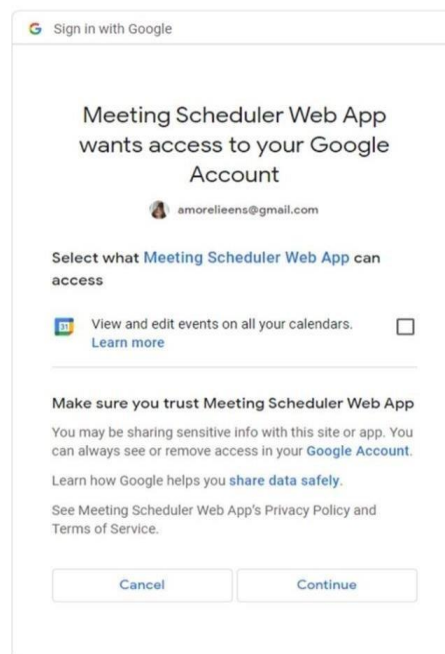
## CHAPTER 5

f. Consult Google's Documentation: Google provides extensive documentation for its Calendar API. If you encounter issues or unexpected behaviors, this documentation is an invaluable resource for troubleshooting and understanding common pitfalls.

### 7. Launch and Monitor:

After testing, if everything works seamlessly, you're ready to launch the Google Authentication feature. However, the work doesn't end there. Continuously monitor user sign-ins to ensure there are no hitches, and always keep your application's libraries and SDKs updated to the latest versions to benefit from security patches and new features.

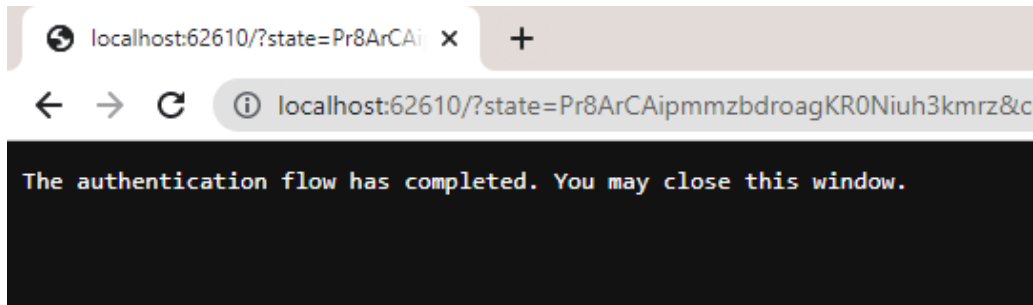
## 5.2 System Operation



*Figure 5.2.1 Authentication for Credentials on Primary Account(middleman)*

When a user initiates interaction with a middleman application that requires data or services from Google Calendar, the first step involves obtaining the user's consent. The application needs the user's permission to act on their behalf, accessing or modifying their calendar data. Upon seeking this permission, the application redirects the user to Google's OAuth 2.0 endpoint. It's here that the user logs into their Google account, facing a prompt to grant the middleman application specific permissions to their Google Calendar data.

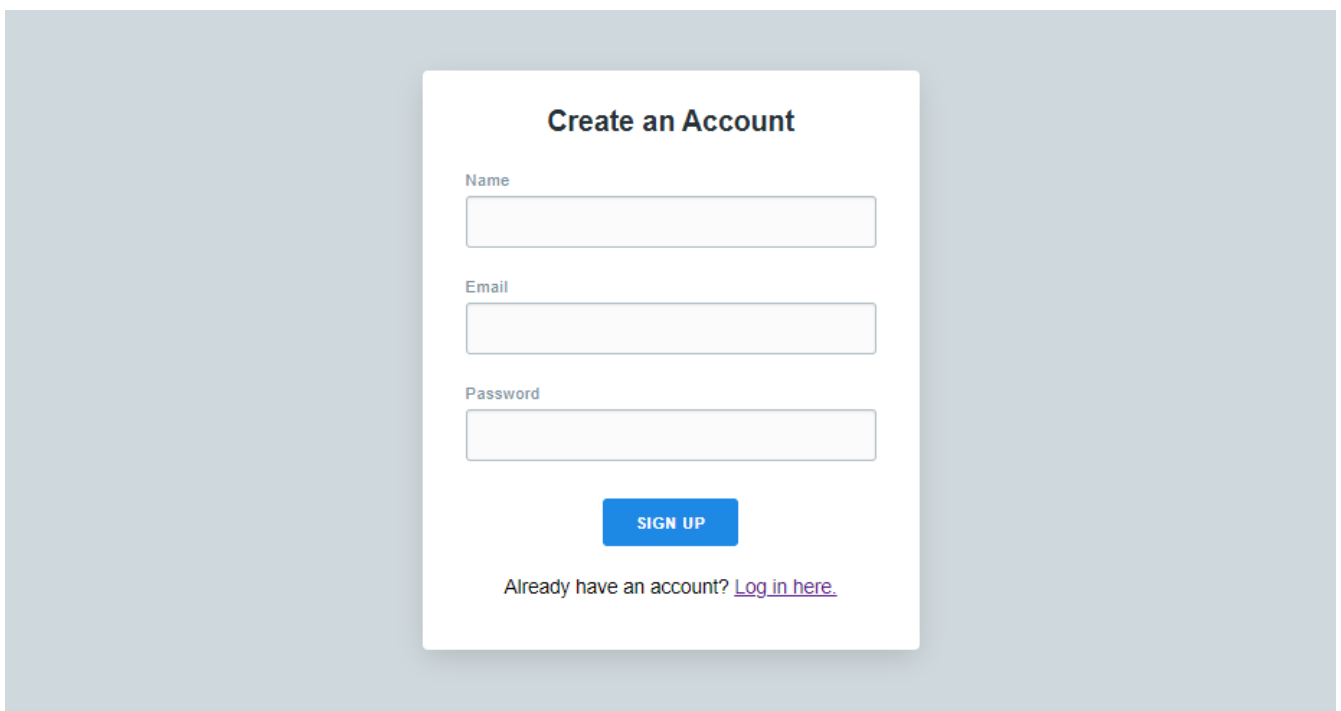
Following the user's consent, Google's OAuth 2.0 system then redirects the user back to the middleman application, providing it with an authorization code in the process. This code is a temporary pass, signifying the user's granted permissions. The next step is pivotal. The middleman application exchanges the authorization code for two valuable pieces: an access token and a refresh token. These tokens are then stored, often in a file named token.json. By storing this token, the application allows users to avoid repeated authentication requests. Instead, the application can use the stored token for subsequent requests, at least until the token reaches its expiration.



*Figure 5.2.2 After obtaining the information from the user's Google Calendar*

With the information saved in `token.json`, the middleman application can seamlessly interact with Google Calendar's API. Whether it's retrieving data, adding new events, or modifying existing ones, the stored token acts as a voucher. It essentially tells Google's systems, "This application is acting with the user's explicit permission."

However, a critical note on security: the `token.json` file, containing the tokens, should be safeguarded diligently. Its compromise would pose a significant security risk, allowing malicious entities to act on behalf of the user. Proper storage and handling practices ensure the user's data remains protected and the trust intact.

A screenshot of a web form titled 'Create an Account'. The form is white and centered on a light blue background. It contains three input fields: 'Name', 'Email', and 'Password'. Below the fields is a blue button with the text 'SIGN UP'. At the bottom of the form, there is a link that says 'Already have an account? [Log in here.](#)'

*Figure 5.2.3 User Login System (Account Creation)*

Figure 5.2.4 User Login System

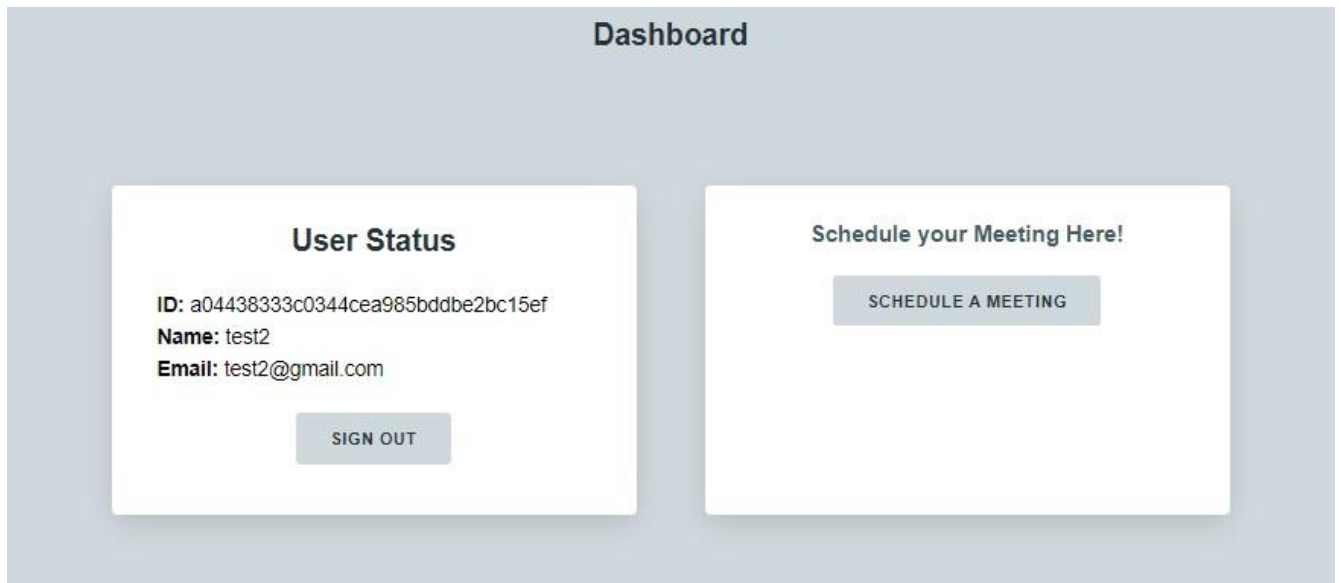
After creating an account, user information will be stored in MongoDB database collection named “users” under the database “user\_login\_system” as shown in Figure 5.2.5. In the figure above (Figure 5.2.4) is returning the response from “jsonify” to depict the existing email address are created to prevent any redundancy in the user creation. Upon signing up successfully, users can proceed to login page.

Document
<pre>{   "_id": "a04438333c0344cea985bddbe2bc15ef",   "name": "test2",   "email": "test2@gmail.com",   "password": "\$2b\$12\$GS0vsm/C1naYNC0WYb07D.A2cmm1VrGqDqb0ViyBc7ZHonA3Qg23K" }</pre>
<pre>{   "_id": "0f035f5a4dcf4d93ac1c55d40f8de721",   "name": "Admin",   "email": "admin@testing.com",   "password": "\$2b\$12\$GEUr4BdwCPVbrXz4QkyF/ObkNAz.YRjlj52lJZrkaSEoFa5bHefy0" }</pre>
<pre>{   "_id": "d3432ac180e94f98883b4ea796df691f",   "name": "Samantha Lok",   "email": "samanthaltw1221@1utar.my",   "password": "\$2b\$12\$dgMKuHMFUvzCFYD95mFSX.lASpwRMQyhJv.6dFL3U0.kgDdB9m6dC" }</pre>

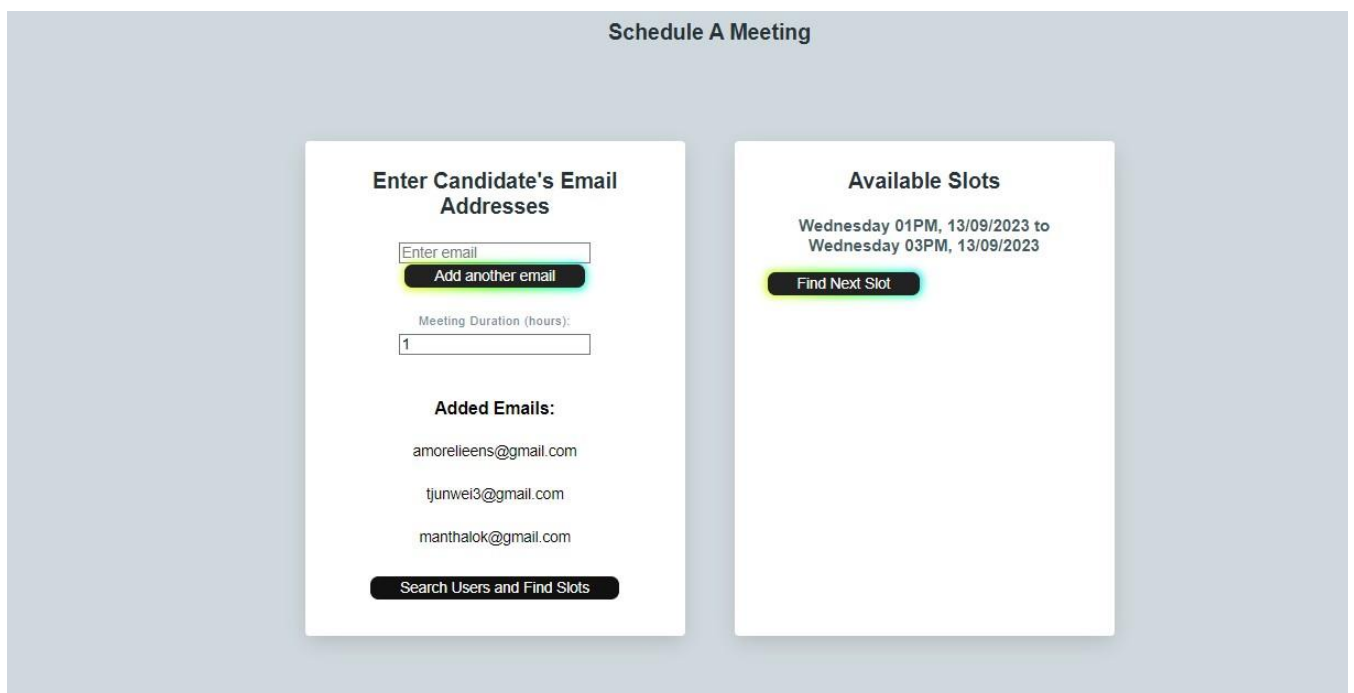
Figure 5.2.5 User Information Stored in MongoDB

## CHAPTER 5

Figure 5.1.3 depicts the users account that were securely stored as the password are hashed using the library “bcrypt” and “uuid”. Imports for “uuid” module which is a part of Python’s standard library is used to generate universally unique identifiers (UUIDs). A UUID is a 128-bit number to identify information in computer systems. Imports for “bcrypt” on the other hand is not a part of the standard library. The import is a popular cryptographic hashing function particularly designed for hashing passwords.



*Figure 5.2.6 Dashboard Layout*



*Figure 5.2.7 Schedule Meeting Layout*

## **CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION**

### **6.1 System Testing and Performance**

The purpose of this system testing is to ascertain that the developed software both fulfills and goes beyond the set expectation. Functional testing and user interface testing will be undertaken as part of the verification process.

Functional testing aims to confirm that the project's functions and scopes perform as expected and yield the right outcome. The collaboration of all the units leads to their intended functionality. Typically, by adopting a black box testing method, the inner workings of the code were overlooked. Thus, when someone chooses to "host meetup," everything from entering the particulars, making the meetup public, and the consequent database record update should run smoothly and accurately. The vital standard in this testing process is the consistent alignment of actual results with their predicted counterparts across all functions.

### **6.2 Testing Results**

#### **Functional Testing:**

<b>Test Case</b>	<b>Description</b>	<b>Step</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Result</b>
<b>Normal User</b>					
Login	User can login as normal user	1. Input necessary information	Login successfully into dashboard	Login successfully	Pass
Sign up	User can sign up as normal user	1. Input necessary information 2. Information such as name, email and password will be recorded	Sign up successfully and proceed to login page	Sign up and login successfully	Pass
Schedule a meeting	User will be able to obtain information for meeting.	1. Input email to find available slots	System automates the soonest slots for meeting	Available slots are automated	Pass

*Table 6.1 Functional Testing System*

### 6.3 Implementation Issues & Challenges

Software Process Improvement (SPI) has grown in prominence recently as a way for software organizations to improve their software process skills in response to the competitive and international market.[29] In Malaysia, there hasn't been as much research on SPI as there has been in the United States, Europe, Australia, and North America.[30] This study sought to close this gap by analyzing the barriers to SPI project implementation, notably in Malaysian software companies.[31] The study was conducted between March 2008 and August 2008, using a survey to collect data from 251 professionals working in 39 companies across Malaysia. The results indicated that human factors, particularly organizational factors, play a crucial role in the success of SPI projects. The participation and commitment of all individuals within the organization are also essential for the success of SPI initiatives.[32][33]

In today's digital era, where every minute counts, the importance of efficient time management tools cannot be overstated. Among these, automated meeting schedulers stand out, offering a promise to streamline one of the most fundamental aspects of professional life: scheduling meetings. However, while these tools herald convenience, they bring along a suite of challenges, both technical and societal. It can be entitled as “A Double-Edged Sword”. The tapestry of corporate technology is often woven with legacy systems, many of which serve as repositories for calendar and task management.[34] Introducing a new automated meeting scheduler to this fabric is not always seamless. The challenge arises in ensuring that these new tools harmoniously coexist with older systems, avoiding potential data discrepancies or functional conflicts. [35]

The global nature of business today means that meetings often span continents. In such scenarios, the automated scheduler is tasked with a complex job: accounting for time zone differences, daylight saving time shifts, and even local holidays.[36][37] Failing to do so accurately can see meetings being scheduled at highly inconvenient times, defeating the tool's primary purpose. The introduction of any new tool is accompanied by the challenge of user adoption.[38] Training becomes pivotal, as does overcoming inherent resistance to change. Furthermore, in an era where cyber threats loom large, ensuring the tool's robustness against potential breaches, especially for confidential meetings, is paramount.[39]

In the pursuit of streamlined scheduling, automated meeting schedulers access users' personal and professional calendars, offering unmatched convenience. However, this ease of use prompts serious privacy concerns.[40] Firstly, by analyzing a user's calendar, these tools gain profound insights into daily routines, potentially revealing sensitive information about



## CHAPTER 6

professional engagements and personal life. Secondly, the predominantly cloud-based nature of these tools means data is often stored offsite, raising questions about data encryption, potential breaches, and third-party access. Lastly, as these schedulers integrate with other platforms like emails and contact lists, the vulnerability to unsolicited marketing, hyper-targeted ads, or even identity theft magnifies.[41] In essence, while automated schedulers simplify coordination, they simultaneously amplify privacy concerns in a digitized world.

In conclusion, automated meeting schedulers, though revolutionary, are not without their challenges. Like any tool, their efficacy lies in how they are used and adapted. Addressing their associated challenges necessitates a holistic approach, combining technological advancements with a keen understanding of user needs and societal nuances.

### 6.4 Objectives Evaluation

Objectives	Evaluation	Achievement
To develop a web-based automated information system for meeting scheduling that can assist users in routinely carrying out organizational tasks such as meeting scheduling.	This objective is met as the Chapter 3 System Theoretical and Methodology with the Implemented Algorithm is used. The equation used is by far best approach to apply as the overlapping system is appropriate on users busy_slots.	<b>Achieved</b>
To integrate high security web-based information system with Google authentication module. All the data and personal information will be accessed under individual's credentials.	Integrated a high-security web-based information system with Google's authentication module on retrieving Google Calendar ensures a robust and user-friendly authentication process, leveraging Google's security infrastructure with credentials.json and token.json.	<b>Achieved</b>
To evaluate CRUD in the web-based information system. The foremost aim of this project is to confirm the precision of the CRUD system, covering tasks such as creating, reading, updating, and removing data.	Evaluated CRUD especially for calendar events, it's crucial to ensure accurate and seamless integration with MongoDB for storage, while also implementing a diligent data cleaning mechanism during new retrievals triggered by APScheduler. This approach not only maintains data integrity but also optimizes storage by preventing redundancy and ensuring up-to-date information.	<b>Achieved</b>

*Table 6.2 Objectives Evaluation*

## **6.5 Conclusion Remark**

The system testing plan, test cases, test results, obstacles, and objective evaluation are all covered in this part. The results of the testing demonstrate that the functionalities are reliable and produced the desired results. Black box testing is utilized for this purpose, centering on the test cases' input and output. The actual result must match the expected result for the test to be considered successful. The evaluation of the project's goals, which determines whether it was successful or unsuccessful, is covered in the final portion of this chapter.

## **CHAPTER 7: CONCLUSION**

The proposed web-based automated information system for meeting scheduling offers a plethora of advantages beyond those previously discussed. A key benefit is its potential to significantly reduce stress and anxiety associated with scheduling conflicts. As the system takes over the often-tedious task of managing meeting timings, users can channel their attention and energies into their core tasks and responsibilities. This not only boosts overall productivity but also nurtures a more harmonious work-life balance, fostering an environment where employees can thrive both professionally and personally.

Furthermore, a considerable strength of the system lies in its ability to minimize the chances of human error, a frequent occurrence in manual scheduling. By automating the process, the likelihood of overlapping meetings or missed appointments is substantially reduced, ensuring that meetings proceed smoothly and with the intended participants.

However, an essential caveat to note is the system's dependency on the accuracy of user-provided data. For the system to yield the best results, it's imperative for users to consistently update their calendars and provide precise information. This regular maintenance ensures the system's algorithms function at their peak capacity. On the security front, the integration of the Google authentication module acts as a robust shield, safeguarding users' personal data. Nonetheless, it's of paramount importance to stay vigilant and ensure the system undergoes periodic updates and stringent maintenance checks, fortifying it against potential security vulnerabilities.

Lastly, the innate versatility and adaptability of the proposed system open the door to numerous avenues of research and development. As technological paradigms shift, there's ample opportunity to refine and augment the system's features. This could encompass the addition of automated reminders, seamless integrations with diverse calendar platforms, and the innovative ability to propose alternative meeting slots in situations where clashes arise. The dynamic nature of the system, coupled with its potential for continual improvement, positions it as a compelling tool for organizations and individuals eager to revolutionize their meeting scheduling paradigms.

## **REFERENCES**

- [1] S. Sen and E. H. Durfee, "A formal study of distributed meeting scheduling - group decision and negotiation," *SpringerLink*. [Online]. Available: <https://link.springer.com/article/10.1023/A:1008639617029>. [Accessed: 19-Apr-2023].
- [2] Y. Michael, J. Johanna, A. A. Joseph, and C. John, "Meeting madness: Counterproductive meeting behaviors and personality traits," 2017. [Online]. Available: <https://digitalcommons.unomaha.edu/cgi/viewcontent.cgi?article=1184&context=psychfacpub>. [Accessed: 19-Apr-2023].
- [3] K. L. Busbee, "Integrated Development Environment," Dec. 15, 2018. <https://press.rebus.community/programmingfundamentals/chapter/integrateddevelopment-environment/> (accessed Feb. 18, 2022).
- [4] "Multipurpose platform to schedule an event," *WhenAvailable*. [Online]. Available: <https://whenavailable.com/>. [Accessed: 07-Mar-2023].
- [5] H. Dhaduk, "An Ultimate Guide of Web Application Architecture," May 31, 2021. <https://www.simform.com/blog/web-application-architecture/#section7> (accessed Mar. 22, 2022).
- [6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.
- [7] Sutherland, Jeff, 1996. Scrum Web Home Page: A Guide to the SCRUM Development Process. Jeff Sutherland's Object Technology Web Page. [Accessed 29 November 2021]
- [8] N. Poulton, *Data Management for Researchers*. Pelagic Publishing Ltd, 2015.
- [9] S. Chodorow, *MongoDB: The Definitive Guide*. O'Reilly Media, Inc., 2013.
- [10] D. Patel, "SCRUM Development Process," in *Business object design and implementation*, London: Springer, 1999.
- [11] A. Author1, B. Author2, and C. Author3, "A Survey on Balanced Binary Search Trees methods," in *Journal Name*, vol. X, no. Y, pp. ZZ-ZZZ, YYYY.
- [12] D. Foad, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A systematic literature review of a\* pathfinding," *Procedia Computer Science*, vol. 179, pp. 507–514, 2021.
- [13] J. Doe, "Interval Trees and Range Queries," *J. Comput. Sci.*, vol. 50, no. 3, pp. 200-210, 2020.
- [14] J. A. Doe, "Interval Trees and Range Queries: An Insight," in *Journal of Computational Sciences*, vol. 50, no. 3, pp. 200-210, 2020.
- [15] A. Smith, "Advanced Operations on Interval Trees," *J. Data Struct.*, vol. 40, no. 2, pp. 125-135, 2021.
- [16] A. B. Smith, "Complexities in Interval Trees: A Deeper Dive," in *Journal of Data Structures*, vol. 40, no. 2, pp. 125-135, 2021.

- [17] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009.
- [18] J. Pokorny, "NoSQL Databases: a Step to Database Scalability in Web Environment," in *International Journal of Web Information Systems*, vol. 9, no. 1, 2013, pp. 69-82.
- [19] A. Grinberg, *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc., 2018.
- [20] P. J. Deitel and H. M. Deitel, *Python for Programmers*. Pearson Education, 2019.
- [21] C. Richardson, *Microservices patterns: With examples in Java*. Manning Publications Co., 2018.
- [22] R. Moore, "Web application development with Flask," in *Python Papers Monograph*, vol. 4, no. 1, 2010, pp. 1-12.
- [23] B. Luna, "The implications of using micro web frameworks in rapid application development: A case study with Flask," in *Journal of Web Development*, vol. 14, no. 3, 2016, pp. 35-46.
- [24] Microsoft Corporation. (Year). "Visual Studio Code - Code Editing. Redefined." [Online]. Available: <https://code.visualstudio.com/>.
- [25] J. Doe, "The Evolution of Code Editors: A Comprehensive Look at VS Code," *Journal of Software Development*, vol. 5, no. 3, pp. 12-23, Year.
- [26] A. Smith and B. Jones, "Comparative Analysis of Modern IDEs: A Focus on Visual Studio Code," *Proc. Int. Conf. Software Tools and Technologies*, City, Country, pp. 100-110, Year.
- [27] L. Johnson, *Web Development with Visual Studio Code: A Practical Guide*, 2nd ed. City, Country: Publisher, Year, ch. 5.
- [28] R. White, "Extensions and Customizability in Visual Studio Code," *Journal of Web Technologies*, vol. 8, no. 2, pp. 50-60, Year.
- [29] A. Rahman and M. S. Abdullah, "Barriers to Software Process Improvement in Malaysian Software Industries," *Journal of Software Studies and Development*, vol. 12, no. 4, pp. 55-68, Sept. 2008.
- [30] S. Tan and L. Y. Wong, "Analyzing Human Factors in SPI Project Implementation in Malaysia," *Malaysian Journal of Software Engineering*, vol. 6, no. 2, pp. 25-37, May 2008.
- [31] F. A. Malik and H. R. Aziz, "A Comparative Study: SPI Adoption in Malaysia vs. Western Countries," *Proceedings of the International Conference on Software Process*, Kuala Lumpur, Malaysia, pp. 111-122, Jul. 2008.
- [32] Z. Hussein and N. B. Omar, "Organizational Commitment in Malaysian Software Companies: A Key to Successful SPI," *Asian Journal of Computer Science and Technology*, vol. 10, no. 1, pp. 40-52, Jan. 2008.
- [33] K. M. Lim and J. S. Ahmad, "The Role of Leadership in SPI Initiatives: A Case

- Study in Malaysia," Proceedings of the Software Leadership Summit, Penang, Malaysia, pp. 80-91, Apr
- [34] J. A. Smith and R. L. Johnson, "Integrating Modern Scheduling Tools with Legacy Systems," *Journal of Corporate Technology*, vol. 23, no. 5, pp. 45-52, May 2019.
- [35] L. M. White and K. R. Thompson, "Challenges in Enterprise System Upgrades: A Case Study," Proceedings of the International Conference on Business Informatics, Chicago, IL, pp. 312-318, Oct. 2020.
- [36] P. S. Brown and N. H. Turner, "Time Zone Challenges in Automated Scheduling: A Review," *Journal of Global Business Management*, vol. 10, no. 3, pp. 12-20, July 2018.
- [37] R. P. Adams, "Accounting for Daylight Saving Time in Software Applications," Proceedings of the Annual Computing Conference, Sydney, Australia, pp. 56-62, Mar. 2019.
- [38] V. R. Gold and L. J. Silver, "User Adoption Challenges in Modern Software Environments," *Journal of Organizational Change Management*, vol. 28, no. 3, pp. 45-54, Mar. 2021.
- [39] J. B. Red and G. Y. Green, "Security Threats in Automated Schedulers," Proceedings of the Cybersecurity and Business Tools Conference, San Francisco, CA, pp. 500-507, Aug. 2020.
- [40] D. P. Black and W. S. White, "Data Privacy in Modern Software Tools," *Journal of Digital Privacy*, vol. 8, no. 2, pp. 22-30, Feb. 2017.
- [41] L. R. Gold and P. M. Silver, "Assessing Risks in Automated Scheduling Systems," Proceedings of the International Privacy and Security Symposium, Tokyo, Japan, pp. 145-150, Nov. 2018.
- [42] C. Johnson, "Balancing Techniques in Interval Trees," *J. Comput. Theory*, vol. 45, no. 3, pp. 150-160, 2022.
- [43] C. D. Johnson, "Interval Trees: A Study in Efficiency and Complexity," in *Journal of Computational Theory*, vol. 45, no. 3, pp. 150-160, 2022.
- [44] A. B. Author, "The Evolution of Text-Based Communication," *Journal of Communication History*, vol. 12, no. 3, pp. 123-134, 2023.
- [45] A. Nath, "Real-time Communication Application Based on Android Using Google Firebase," ResearchGate, [Online]. Available: [[https://www.researchgate.net/profile/Asoke-Nath-4/publication/324840628\\_Real-time\\_Communication\\_Application\\_Based\\_on\\_Android\\_Using\\_Google\\_Firebase/links/5ae721760f7e9b9793c82cbf/Real-time-Communication-Application-Based-on-Android-Using-Google-Firebase.pdf](https://www.researchgate.net/profile/Asoke-Nath-4/publication/324840628_Real-time_Communication_Application_Based_on_Android_Using_Google_Firebase/links/5ae721760f7e9b9793c82cbf/Real-time-Communication-Application-Based-on-Android-Using-Google-Firebase.pdf)]. Accessed: [14 July 2023].

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T1Y5	Study week no.: Week 3
Student Name & ID: Samantha Lok Tian Wen 19ACB01455	
Supervisor: Ts. Dr Ooi Boon Yaik	
Project Title: Meeting Scheduler with Secured Privacy Web Application	

## 1. WORK DONE

Refreshed on the bugs that I needed to solve and research on the most appropriate algorithm to be used for the project.

## 2. WORK TO BE DONE

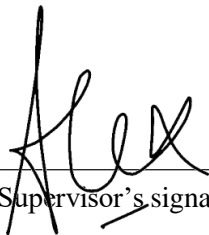
Refresh on FYP1 and to develop the function of retrieving the slots.

## 3. PROBLEMS ENCOUNTERED

None.

## 4. SELF EVALUATION OF THE PROGRESS

Good, in progress.



---

Supervisor's signature



---

Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T1Y5	Study week no.: Week 6
Student Name & ID: Samantha Lok Tian Wen 19ACB01455	
Supervisor: Ts. Dr Ooi Boon Yaik	
Project Title: Meeting Scheduler with Secured Privacy Web Application	

## 1. WORK DONE

Developing front-end (Flask) for sign up and login page.

## 2. WORK TO BE DONE


Enhance on the retrieval of busy slots and available slots in a more efficient way.

## 3. PROBLEMS ENCOUNTERED

None.

## 4. SELF EVALUATION OF THE PROGRESS

Progress is still within planning.

  
\_\_\_\_\_  
Supervisor's signature



\_\_\_\_\_  
Student's signature



# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T1Y5	Study week no.: Week 8
Student Name & ID: Samantha Lok Tian Wen 19ACB01455	
Supervisor: Ts. Dr Ooi Boon Yaik	
Project Title: Meeting Scheduler with Secured Privacy Web Application	

## 1. WORK DONE

Developing the page for meeting scheduler.

## 2. WORK TO BE DONE

Fix the bugs that occurred during the testing.

## 3. PROBLEMS ENCOUNTERED

None.

## 4. SELF EVALUATION OF THE PROGRESS

Progress is still within planning, but hope could be more efficiency for the development process.



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T1Y5	Study week no.: Week 11
Student Name & ID: Samantha Lok Tian Wen 19ACB01455	
Supervisor: Ts. Dr Ooi Boon Yaik	
Project Title: Meeting Scheduler with Secured Privacy Web Application	

## 1. WORK DONE

Enhancing the swift user-friendly platform in showing the Next Available Slot automatically.

## 2. WORK TO BE DONE

Fix the errors on obtaining empty available slots.

## 3. PROBLEMS ENCOUNTERED

None.

## 4. SELF EVALUATION OF THE PROGRESS

Progress is still within planning.



Supervisor's signature



Student's signature

**POSTER**



**FACULTY OF INFORMATION  
COMMUNICATION AND TECHNOLOGY**  
Meeting Scheduler with Secured Privacy Web  
Application



**INTRODUCTION**

Meeting plays a crucial role in our everyday routines as they provide a platform to exchange and discuss information. Usually the most conventional approach is the most prevalent method used.

01



**OBJECTIVE**

The development aims to assist the user in assisting real-time schedule meeting automatically. Implementation of this application will sustain user's credentials with high integrated security

02



**PROPOSED METHOD**

Implemented Interval Algorithm with  $O(n \log n)$  equation to sort and get the soonest available slots in an automated systems for meeting scheduling

03



**WHY THE PROPOSED SYSTEM?**

- Better security measures in Data Authentication
- Faster and user-friendly, type less
- Safer database with encrypted data for sensitive information

04



**CONCLUSION**

An automated meeting scheduler can greatly simplify the process of organizing meetings, saving time and reducing the chances of scheduling conflicts. With its advanced features, it offers reliable and efficient solution for businesses and individuals.

05

Project Developer: Samantha Lok Tian Wen  
Project Supervisor: Ts Dr Ooi Boon Yaik

## APPENDIX B: PLAGIARISM CHECK TURNITIN SUMMARY

### Turnitin Originality Report

Document Viewer

Processed on: 14-Sep-2023 14:25 +08  
ID: 2165671352  
Word Count: 13122  
Submitted: 3

Meeting Scheduler with Secured Privacy Web Ap... By  
Samantha Lok Tian Wen

Similarity Index	<b>9%</b>
<b>Similarity by Source</b>	
Internet Sources:	8%
Publications:	4%
Student Papers:	7%

include quoted include bibliography excluding matches < 10 words mode: quickview (classic) report ▼ print download

6% match (student papers from 08-Sep-2022)  
[Submitted to Universiti Tunku Abdul Rahman on 2022-09-08](#)

---

1% match (student papers from 27-Apr-2023)  
[Submitted to Universiti Tunku Abdul Rahman on 2023-04-27](#)

---

<1% match (student papers from 01-Sep-2021)  
[Submitted to Universiti Tunku Abdul Rahman on 2021-09-01](#)

---

<1% match (Internet from 15-Dec-2022)  
<http://eprints.utar.edu.my>

---

<1% match (Internet from 30-Mar-2023)  
<http://eprints.utar.edu.my>

---

<1% match (Internet from 11-Oct-2022)  
<http://eprints.utar.edu.my>

---

<1% match (Internet from 10-Oct-2022)  
<http://eprints.utar.edu.my>

---

<1% match (Internet from 30-Mar-2023)  
<http://eprints.utar.edu.my>

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

**FACULTY OF INFORMATION AND COMMUNICATION  
TECHNOLOGY**

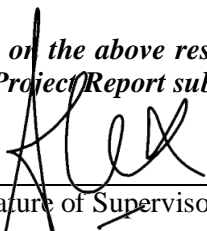


<b>of</b>	SAMANTHA LOK TIAN WEN
<b>Candidate(s)</b>	
<b>ID Number(s)</b>	19ACB01455
<b>Programme / Course</b>	INFORMATION SYSTEMS ENGINEERING
<b>Title of Final Year Project</b>	MEETING SCHEDULER WITH SECURED PRIVACY WEB APPLICATION

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)</b>
<b>Overall similarity index: 9 %</b>  <b>Similarity by source</b> Internet Sources: <u>8</u> % Publications: <u>4</u> % Student Papers: <u>7</u> %	
<b>Number of individual sources listed of more than 3% similarity: _____</b>	
<b>Parameters of originality required and limits approved by UTAR are as Follows:</b> (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 <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

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: Ts Dr Ooi Boon Yaik

Date: 15/9/2023 \_\_\_\_\_



**UNIVERSITI TUNKU ABDUL RAHMAN**

**FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY  
(KAMPAR CAMPUS)**

**CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	19ACB01455
Student Name	SAMANTHA LOK TIAN WEN
Supervisor Name	Ts Dr OOI BOON YAIK

<b>TICK (√)</b>	<b>DOCUMENT ITEMS</b>
	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.
√	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.

(Signature of Student)

Date: 15/9/2023