

**E-LEARNING PLATFORM FOR COLLABORATIVE CODING
ASSIGNMENTS**

CHONG MUM KHONG


**A project report submitted in partial fulfilment of the
requirements for the award of Bachelor of Science
(Hons.) Software Engineering**

**Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman**

May 2021

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 
Name : Chong Mum Khong
ID No. : 1803151
Date : 1 May 2021

APPROVAL FOR SUBMISSION

I certify that this project report entitled “**E-LEARNING PLATFORM FOR COLLABORATIVE CODING ASSIGNMENTS**” was prepared by **CHONG MUM KHONG** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : Too Chian Wen

Date : 15 April 2021

Signature : _____

Co-Supervisor : _____

Date : _____

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2021, Chong Mum Khong. All right reserved.

ACKNOWLEDGEMENTS

The completion of this project cannot be achieved without the moral supports I received from my families and friends had given to me. I hereby would like to take a moment to show my enormous appreciation to them, especially to my beloved parents Chong Sin Soon and Goh Suan Goat.

I would also like to express my gratification towards my supervisor, Dr. Too Chian Wen for providing guidance and assistance throughout the project. I would also like to thank my moderator Ms. Chean Swee Ling for taking the time to go through and assess each aspect of the project.

ABSTRACT

Ever since the pandemic rampages across the world, university students in Malaysia who partake in programming courses are having difficulty collaborating with each other in their group projects due to restriction rules imposed to combat the virus. Inefficient conventional social media, lack of features in the current platforms had added insult to injury on this matter. Background studies was conducted through literature review for the problems, solutions and similar applications on the market to justify the need for such system and possible technologies or features that can be incorporated into the system design. The platform comes with two different modules or interface for different users, namely students and lecturers. Methodology-wise, parallel model was adopted to implement the different modules concurrently. Figma was used to product wireframes for the screens of the system. The technologies used are Vue.js, Vuetify, Node.js, Express.js, TypeScript, Socket.io, Nodemailer, NodeSchedule, Git, GitHub, Firebase and other third-party open-source packages for implementation of the system. The system was developed to mitigate the dilemma that students faced, moreover improving the transparency of the students' progress by allowing lecturers to check on their work at any time. Students can collaborate in their coding assignments on the platform with rich functionalities provided such as real-time editor, chatting system, assignment management etc. whereas lecturers can freely check on any of their student groups to follow up their progress and even join their discussions to improve the assignment quality. With this platform, students and lecturers are able to communicate effectively on their work and provide efficient collaboration among group members for group coding assignments.

TABLE OF CONTENTS

DECLARATION	ii
APPROVAL FOR SUBMISSION	iii
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	xiii
LIST OF FIGURES	xvii
LIST OF SYMBOLS / ABBREVIATIONS	xxiii
LIST OF APPENDICES	xxiv

CHAPTER

1	INTRODUCTION	25
	1.1 Introduction	25
	1.2 Background of the Problem	25
	1.3 Problem Statement	26
	1.3.1 Inefficient Communication	26
	1.3.2 Closure of Campuses Nationwide	27
	1.3.3 Lack of Functionalities in Existing E-Learning Platform	27
	1.3.4 Difficulties in Keeping Track Student's Progress	28
	1.3.5 Difficulties to Find Group Members	28
	1.4 Objective	28
	1.5 Project Approach	29
	1.6 Proposed Approach	30
	1.7 Scope	31
	1.7.1 Targeted Users	31
	1.7.2 Features Covered	31

2	LITERATURE REVIEW	34
2.1	Introduction	34
2.2	E-Learning	34
2.2.1	E-Learning, Traditional Learning and Collaborative Learning	35
2.2.2	E-Learning in Higher Institutions	36
2.2.3	Necessity of E-Learning Platform in Higher Education in Malaysia	37
2.2.4	Microlearning	38
2.3	Collaborative Coding	38
2.3.1	Collaborative Coding in Student Context	39
2.4	Related Work	41
2.4.1	Real-time Collaborative Editing	41
2.4.2	Important Features	42
2.4.3	Transparent Adaptation	42
2.4.4	Implementation of CoVSCode	43
2.4.5	Implementation of CoRED	44
2.4.6	Implementation of CodeR	46
2.4.7	Findings	48
2.5	Software Development Methodology	49
2.5.1	Traditional & Classical Methodologies	49
2.5.2	Modern Approach: Agile Methodologies	55
2.5.3	Findings	57
2.6	Similar Systems	57
2.6.1	Web-based Text Editor	58
2.6.2	Communication Applications	66
2.6.3	Project Management	70
2.7	Project Initial Specification from Literature Review	72
2.8	Conclusion	72
3	METHODOLOGY AND WORK PLAN	74
3.1	Introduction	74
3.2	Parallel Development	74

3.3	Planning	75
	3.3.1 Requirement Gathering and Elicitation	76
	3.3.2 Project Scheduling	78
3.4	Analysis and Design	78
3.5	Implementation and Testing	79
3.6	Post-implementation Phase	80
3.7	Project Tools	80
	3.7.1 System Architecture	80
	3.7.2 Firebase	81
	3.7.3 Node.js	81
	3.7.4 Vue.js	82
	3.7.5 Vuetify	82
	3.7.6 TypeScript	82
	3.7.7 Git and GitHub	82
	3.7.8 Figma	83
	3.7.9 Jest	83
3.8	Work Plan	84
	3.8.1 Work Breakdown Structure	84
	3.8.2 Gantt Chart	87
3.9	Summary	92
4	PROJECT SPECIFICATION	93
4.1	Introduction	93
4.2	Questionnaire Evaluation	93
	4.2.1 Demographics	94
	4.2.2 Features/Functions Studied	95
	4.2.3 Usability Investigation	96
	4.2.4 Opinion on E-Learning	99
	4.2.5 Findings of Questionnaire	100
4.3	Interview Findings	100
4.4	Requirement Specification	101
	4.4.1 Functional Requirement	101
4.5	Non-functional Requirement	103

4.5.1	Availability Requirement	103
4.5.2	Adaptability Requirement	103
4.5.3	Capacity Requirement	103
4.5.4	Development Requirement	103
4.5.5	Performance Requirement	104
4.5.6	Security Requirement	104
4.5.7	Usability Requirement	104
4.6	Use Cases	104
4.6.1	Use Case Diagram	105
4.6.2	Use Case Description	106
5	SYSTEM DESIGN	126
5.1	Introduction	126
5.2	System Architecture Design	126
5.3	Database Design	128
5.3.1	Entity Relationship Diagram	128
5.3.2	Data Flow Diagram	135
5.4	Wireframe for User Interface Design	140
5.4.1	Shared View	140
5.4.2	Student View	143
5.4.3	Lecturer View	146
5.5	Page Navigation Flow (Site Map)	149
5.5.1	Student Flow	149
5.5.2	Lecturer Flow	150
6	SYSTEM IMPLEMENTATION	151
6.1	Introduction	151
6.2	Project Setup	151
6.3	Front-end Development	152
6.3.1	Application of UI/UX Theories	152
6.3.2	Login Page	152
6.3.3	Student Module	154
6.3.4	Lecturer Module	169

6.4	Back-end Development	177
6.4.1	Job Scheduling	178
6.4.2	Real-time Server	179
6.5	API Endpoints	180
6.6	File Structure	183
6.6.1	Front-end	183
6.6.2	Back-end	184
6.7	Version Control System	184
6.7.1	Remote Hosting with GitHub	185
6.8	Summary	186
7	SYSTEM TESTING	187
7.1	Introduction	187
7.2	Automated Testing	187
7.2.1	User Interface Test	187
7.2.2	Utility Functions Test	194
7.2.3	Automated Test Results	207
7.3	Manual Testing	209
7.3.1	Feature Testing	209
7.3.2	Screen Navigation Testing	218
7.3.3	API Testing	222
7.4	User Acceptance Test	225
7.5	Usability Test	227
7.5.1	Usability Test Template	228
7.5.2	User Satisfaction Survey Result	229
8	CONCLUSION AND RECOMMENDATION	230
8.1	Conclusion	230
8.2	Limitations and Future Work	230
	REFERENCES	233
	APPENDICES	237

LIST OF TABLES

Table 2.1: Text Editors and Its Feature	65
Table 2.2: Communication Application and Its Feature	69
Table 4.1: Use Case of Login	106
Table 4.2: Use Case of View Notifications	106
Table 4.3: Use Case of View Courses	107
Table 4.4: Use Case of View Assignments List	108
Table 4.5: Use Case of View Assignment Group Info	109
Table 4.6: Use Case of Create/Join Group	110
Table 4.7: Use Case of Edit Assignment Code	111
Table 4.8: Use Case of Send Text Messages	112
Table 4.9: Use Case of Use Gantt Chart	113
Table 4.10: Use Case of Add Discussion Booking	114
Table 4.11: Use Case of Arrange Discussion Time	115
Table 4.12: Use Case of Create New Assignment	115
Table 4.13: Use Case of Edit Assignment Details	116
Table 4.14: Use Case of Upload Assignment Question	117
Table 4.15: Use Case of Download Assignment Question	118
Table 4.16: Use Case of Submit Assignment	119
Table 4.17: Use Case of Download Student Group Assignments	120
Table 4.18: Use Case of View Statistics	121
Table 4.19: Use Case of Request Matchmake Group	121
Table 4.20: Use Case of Matchmake Assignment Groups	122
Table 4.21: Use Case of Send Email	123

Table 4.22: Use Case of Send Notification	124
Table 4.23: Use Case of View Group List	125
Table 5.1: Entities Description Table	129
Table 5.2: Data Dictionary for Credentials collection	130
Table 5.3: Data Dictionary for Subjects collection	130
Table 5.4: Data Dictionary for Assignments collection	130
Table 5.5: Data Dictionary for Lecturers collection	131
Table 5.6: Data Dictionary for Students collection	131
Table 5.7: Data Dictionary for Groups collection	132
Table 5.8: Data Dictionary for Gantts collection	132
Table 5.9: Data Dictionary for Notifications collection	133
Table 5.10: Data Dictionary for Bookings collection	133
Table 5.11: Data Dictionary for Messages collection	134
Table 6.1: Sample Cron expression and description	179
Table 6.2: API List for Assignment Entity	180
Table 6.3: API List for Group Entity	181
Table 6.4: API List for Lecturer Entity	181
Table 6.5: API List for Student Entity	182
Table 6.6: API List for Subject (Course) Entity	182
Table 6.7: API List for Other Uses	182
Table 6.8: Common Commit Prefixes (Tasevski, 2019)	185
Table 7.1: Test Case 1 - Render	188
Table 7.2: Test Case 2 - Props	189
Table 7.3: Test Case 3 - Data	190
Table 7.4: Test Case 4 - Lifecycle Hooks	191

Table 7.5: Test Case 5 – Interaction and Methods	192
Table 7.6: Test Case 6 – Emit	193
Table 7.7: Test Case 7 – Snippet	194
Table 7.8: Test Case 8 – Alias	195
Table 7.9: Test Case 9 – Python Range Function	196
Table 7.10: Test Case 10 – Random Colour	196
Table 7.11: Test Case 11 – Message Wrapper	197
Table 7.12: Test Case 12 – Date to Timestamp Method	198
Table 7.13: Test Case 13 – Timestamp to Date	199
Table 7.14: Test Case 14 – Timestamp to Date 2	199
Table 7.15: Test Case 15 – Get String Month	200
Table 7.16: Test Case 16 – Get AM PM	200
Table 7.17: Test Case 17 – Timestamp Get Time	201
Table 7.18: Test Case 18 – Is Object Empty	202
Table 7.19: Test Case 19 – Compute Time	203
Table 7.20: Test Case 20 – Booking Algorithm	204
Table 7.21: Test Case 21 – Random Choice	204
Table 7.22: Test Case 22 – Random Pop	205
Table 7.23: Test Case 23 – Login	209
Table 7.24: Test Case 24 – Join Assignment Group	210
Table 7.25: Test Case 25 – Create/Matchmake Group	211
Table 7.26: Test Case 26 – Send Text Messages	212
Table 7.27: Test Case 27 – Gantt Chart	213
Table 7.28: Test Case 28 – Add Discussion Time	214
Table 7.29: Test Case 29 – Submit Assignment	215

Table 7.30: Test Case 30 – Create Assignment	216
Table 7.31: Test Case 31 – Edit Assignment Details	217
Table 7.32: Test Case 32 – Student Navigation	218
Table 7.33: Test Case 33 – Lecturer Navigation	220
Table 7.22: Test Case 34 - 44	222
Table 7.35: Test Case 45	223
Table 7.36: Test Case 46 – 59	223
Table 7.37: Test Case 60 – 62	224
Table 7.38: Test Case 63	224
Table 7.39: Test Case 64 -66	224
Table 7.40: Test Case 67 – 68	225
Table 7.41: Test Case 69	225
Table 7.42: UAT Listing for Student	225
Table 7.43: UAT Listing for Lecturer	226
Table 7.44: User Satisfaction Survey (Brooke. J, 1986)	228
Table 7.45: Summary of User Satisfaction Survey	229

LIST OF FIGURES

Figure 1.1: System Architecture	29
Figure 1.2: Parallel Development (Scott, 2016)	30
Figure 2.1: Operational Transformation (Fan et al., 2019)	41
Figure 2.2: CoVSCode screenshot of 3 concurrent users.	43
Figure 2.3: System architecture of CoVSCode.	44
Figure 2.4: Screenshot of CoRED	45
Figure 2.5: System architecture of CoRED	45
Figure 2.6 Screenshots of CodeR	46
Figure 2.7: Algorithm for auto-saving code.	47
Figure 2.8: Algorithm for real-time collaboration.	47
Figure 2.9 Login Architecture of CodeR	48
Figure 2.10: Software Development Methodologies (Kumar and Bhatia, 2014)	49
Figure 2.11: Waterfall model (Kumar and Bhatia, 2014)	50
Figure 2.12: Parallel Development (Nugroho, Hadi and Hakim, 2017)	51
Figure 2.13: Spiral Model	52
Figure 2.14: V-Shaped Model	53
Figure 2.15: Rational Unified Process	54
Figure 2.16: Feature-driven development	55
Figure 2.17: Test-driven development (Kumar and Bhatia, 2014)	56
Figure 2.18: Screenshot of Codeshare	59
Figure 2.19: Codeanywhere	59
Figure 2.20: Screenshot of GitPod	60

Figure 2.21: Screenshot of StackBlitz	60
Figure 2.22: Screenshot of CodeSandbox	61
Figure 2.23: Screenshot of Expo	61
Figure 2.24: Screenshot of Collabedit	62
Figure 2.25: Screenshot of Jsfiddle	63
Figure 2.26: Screenshot of Playcode	63
Figure 2.27: Screenshot of Codepen	64
Figure 2.28: Screenshot of Jsbin	64
Figure 2.29: Screenshot of Slack	67
Figure 2.30: Screenshot of Discord	68
Figure 2.31: Screenshot of Microsoft Teams	68
Figure 2.32: Screenshot of Airtable	70
Figure 2.33: Screenshot of Asana	71
Figure 2.34: Screenshot of Trello	71
Figure 3.1: Lifecycle for e-Learning System	75
Figure 3.2: Proposed system architecture	80
Figure 3.3: Gantt Chart for Overall Project	87
Figure 3.4: Gantt Chart for 0 to 1.6.2	88
Figure 3.5: Gantt Chart for 1.7 to 1.8.4.3	89
Figure 3.6: Gantt Chart for 1.9 to 2.5	90
Figure 3.7: Gantt Chart for 2.6 to 3.1.3	90
Figure 3.8: Gantt Chart for 3.1.4 to 3.3.4	91
Figure 3.9: Gantt Chart for 3.4 to 4.4	92
Figure 4.1: Gender distribution of respondents	94
Figure 4.2: Distribution of year of study	94

Figure 4.3: E-Learning platform features	95
Figure 4.4: Real-time assignment platform	96
Figure 4.5: Functions that Respondents Actually Use	96
Figure 4.6: User-friendly Rating	97
Figure 4.7: Aesthetic Rating	98
Figure 4.8: Opinion on Helpfulness of University's e-Learning Platform	98
Figure 4.9: Opinion on Room of Improvement	99
Figure 4.10: Opinion on Helpfulness of E-Learning	99
Figure 4.11: Use Case Diagram for This Project	105
Figure 5.1: System Architecture	126
Figure 5.2: Physical Entity Relationship Diagram	128
Figure 5.3: Logical Entity Relationship Diagram	129
Figure 5.4: Context Diagram	135
Figure 5.5: A fairly complicated Level 0 Data Flow Diagram	136
Figure 5.6: Level 1 DFD for Login process	137
Figure 5.7: Level 1 DFD for View Pages process	137
Figure 5.8: Level 1 DFD for Manage Files process	138
Figure 5.9: Level 1 DFD for Workspace Interaction process	138
Figure 5.10: Level 1 DFD for Manage Assignment process	139
Figure 5.11: Level 1 DFD for Manage Groups process	139
Figure 5.12: Level 1 DFD for Manage Gantt process	140
Figure 5.13: Login Page	141
Figure 5.14: Homepage	141
Figure 5.15: Notification Page	142
Figure 5.16: Course Listing Page	142

Figure 5.17: Group Info Page	143
Figure 5.18: Assignments List Page	143
Figure 5.19: Resources Page	144
Figure 5.20: Group Workspace	144
Figure 5.21: Gantt Chart Page	145
Figure 5.22: Discussion Booking Page	145
Figure 5.23: Submit Assignment Page	146
Figure 5.24: Assignments List Page	146
Figure 5.25: Assignment Groups List Page	147
Figure 5.26: Analytics Page	147
Figure 5.27: Assignment Info Page	148
Figure 5.28: Site Map for Student Module	149
Figure 5.29: Site Map for Lecturer Module	150
Figure 6.1: NPM (Node Package Manager)	151
Figure 6.2: Main Login Page	152
Figure 6.3: Error message for unmatched credentials	153
Figure 6.4: Fallback Page	153
Figure 6.5: Not Authorized Page	153
Figure 6.6: Homepage	154
Figure 6.7: Navigation Bar	154
Figure 6.8: Miscellaneous Panel	155
Figure 6.9: User Panel	155
Figure 6.10: Notification Page	156
Figure 6.11: Courses Page	157
Figure 6.12: Resources Page	158

Figure 6.13: Assignments List Page for Students	159
Figure 6.14: Popup for New Assignment Group	160
Figure 6.15: Error Message for Empty Group ID	160
Figure 6.16: Error Message for Non-existing Group	160
Figure 6.17: Error Message for Wrong Assignment	161
Figure 6.18: Error Message for Unavailable Group	161
Figure 6.19: New Group Created	162
Figure 6.20: Snackbar for Matchmake Request Successful	162
Figure 6.21: Group and Assignment Info Page	163
Figure 6.22: Actions Shortcut	163
Figure 6.23: Group Workspace	164
Figure 6.24: Gantt Chart Page	165
Figure 6.25: Popup for adding a new activity	166
Figure 6.26: Discussion Booking Page	166
Figure 4.27: Results shown when all members placed their bookings	167
Figure 4.28: Popup for adding a new discussion time	167
Figure 4.29: Assignment Submission Page	168
Figure 4.30: Message for Successful Submission	168
Figure 4.31: Homepage	169
Figure 4.32: Navigation Bar	169
Figure 4.33: Message for Successful Submission	170
Figure 6.34: Courses Page	171
Figure 4.13: Assignments Page	171
Figure 6.36: Add Assignment Popup	172
Figure 6.37: Assignment Detail Page	173

Figure 6.38: Assignment Group List Page	174
Figure 6.39: Assignment Group List Page	174
Figure 6.40: Assignment Detail Page	175
Figure 6.41: Member Distribution Chart for UECS1234 A2	175
Figure 6.42: Member Distribution Chart after hiding Orphans	176
Figure 6.43: Chart with Tooltip Shown	176
Figure 6.44: Submission Status Chart for UECS1234 A2	177
Figure 6.45: Dependencies Excerpt from Project's package.json File.	177
Figure 6.46: Cron Expression	178
Figure 6.47: Cron Job in Server via node-schedule.	179
Figure 6.48: File structure of front-end repository	183
Figure 6.49: A div element using directives	183
Figure 6.50: File structure for back-end repository	184
Figure 6.51: Remote Repository for Front-end on GitHub	185
Figure 6.52: Remote Repository for Back-end on GitHub	186
Figure 7.1: Automated test results for UI	207
Figure 7.2: Automated test results for front-end utility	207
Figure 7.3: Automated test results for back-end utility	208

LIST OF SYMBOLS / ABBREVIATIONS

COVID-19	Coronavirus Disease 2019
CSS	Cascading Stylesheet
FDD	Feature Driven Development
HTML	Hypertext Markup Language
ICT	Information Communication Technology
IDE	Integrated Development Environment
MCO	Movement Control Order
MERS	Middle East Respiratory Syndrome
MOOC	Massive Open Online Course
OT	Operational Transformation
RAD	Rapid Application Development
RUP	Rational Unified Process
SARS	Severe Acute Respiratory Syndrome
SDLC	Software Development Life Cycle
TA	Transformation Adaptation
TDD	Test Driven Development
UAT	User Acceptance Test
UI	User Interface
VCS	Version Control System
WBLE	Web Based Learning Environment

LIST OF APPENDICES

APPENDIX A: Comparison Between Numerous Malaysian Universities' e-Learning Platform	237
APPENDIX B: Questionnaire	238
APPENDIX C: Interview Question and Summary	241
APPENDIX D: UAT Results	242
APPENDIX E: Usability Test Responses	256

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nowadays, assignment have become an essential aspect in college and university on evaluating a student's knowledge on a subject. Programming assignments for Computer Science students are especially vital for students to exhibit their technical knowledge and not just theoretical ones. Programming based assignments are team-oriented and thus requires a lot of communication by the teams.

It is also important for the lecturer to able to evaluate the student's coursework in an efficient and convenient way so that the core quality of the piece of work could be easily distinguished by the lecturer. It can also prevent mundane tasks that will act as a bottleneck in the whole process.

This project discusses the problems encountered by students on their assignment especially during the current pandemic lockdown so that the need of such system is justified. The necessary application that curbs the situation will be implemented onwards.

1.2 Background of the Problem

E-learning has been growing exponentially around the world as the electronic devices such as smart phones, tablets, laptops and computers that can access the internet are becoming more accessible by modern society (Aparico et al., 2016). According to The Economic Times (n.d.), e-learning is a learning system that is utilising electronic devices to provide formalised teaching pedagogy and is also defined as network enabled transferring of skills and knowledge, so that large amount of people can be benefited regardless of geographical and time constraints.

Higher education institutions in the modern era face prominent social, economic as well as technological changes that will inadvertently shift the educational experience of students (Neier and Zayer, 2015). Most university offers MOOC or e-learning platform like Coursera, Khan Academy, Udemy, Udacity and edX. However, it is not sufficient as it does not support collaborate learning, which is also a core part of e-learning. Since collaborative learning is not supported, computer

science students will face extra problems as they have coding, project-based group assignments to go through.

Without an online platform dedicated for students to work on their group assignment, they will have to seek for alternative ways to collaborate by using a combination of instant text messaging applications to communicate. As most of these applications are not specially designed for group assignment discussion, it will lead to some undesirable consequences such as communication issues, assignment version difference, confusion and worst, argument. Soon, these issues will become a bottleneck for the students to produce good quality assignment.

Students that are taking programming course will deal with extra troubles apart from doing regular assignments. Problems such as source code version difference, bugs and errors, bad programming practice etc. may frequently arise due to communication issues as the inability of the instant text messaging application to display source code in a human-readable way such as proper indentation and syntax highlighting as students may wrongly interpret the code.

To rectify these undesirable repercussions, there must be a software platform dedicated to foster communication between teammates as well as provide a place to conduct the assignment collaboratively. Concurrency will also be a key aspect that can maximize the collaboration experience between team members as it will demolish the need to send code snippets frequently through confusing instant messaging application when working together.

1.3 Problem Statement

1.3.1 Inefficient Communication

Doing group assignments in the college or university is an essential activity that no students can be exempted from. Besides discussing and working with the assignment face-to-face at an agreed date and time, most of the time students will work remotely from their home or hostel and communicate through the internet, primarily by using instant text messaging applications. However, most of them strive to provide the best experience and communication quality for students to work on their assignments collaboratively as they are not specifically made for that purpose.

For coding-based assignments, students will have a hard time on exchanging codes to the team members. They also will be confused on the nature of codes posted as some of them are the legit project codes, some of them are attempts, questions,

faulty code and some of them are outdated codes. Besides that, there is a great chance that students will miss some useful information such as announcement and updates as the conversation consists of chats, jokes, program files, links etc.

This will cause their quality of communication to drop drastically to a point where the quality of the assignment is afflicted. The statement is especially true for computer programming assignments where quality communication is crucial for the team to develop a fully functional, intricate program where a single miscommunication will yield disastrous consequences to the assignments.

1.3.2 Closure of Campuses Nationwide

The recent outbreak of Coronavirus Disease in 2019 that soon turned into a global pandemic had further complicate things for college and university students across the globe.

In Malaysia specifically, Movement Control Order (MCO) was promulgated by the government on 16 March 2020 that will be effective from 18 March in order to contain the virus from spreading further (Tang, 2020). The order entails prohibition of moment and mass assembly nationwide, and schools, campuses, entertainment whereabouts and other non-essential businesses were coerced to shut down.

Students were unable to meet face to face at all to discuss on the project as social-distancing measures and movement control are practiced across the nation. Students had to migrate their discussions fully to online via social medias or instant text messenger such as Whatsapp and Telegram which are already a popular tool for students to discuss assignments. Thus, the communication problems aforementioned will be prevalent in the current situation.

1.3.3 Lack of Functionalities in Existing E-Learning Platform

Most Malaysian university's online e-learning platform such as WBLE of UTAR, MMLS of MMU and Spectrum of UM does not provide collaborative platform for students to discuss their assignment. Students usually will just download the teaching materials uploaded by their lecturer with their platform. Only a small amount of university e-Learning platform actually provides extended functionalities such as forums and instant text messages. However, only a small portion of students actually

utilise those functionalities as they prefer communicating with their mobile phones either by calls or other social medias.

1.3.4 Difficulties in Keeping Track Student's Progress

It is difficult for lecturer to keep track of the progress on assignment for each group. The lecturer will only realise the students' progress if they seek help from the lecturer. However, not much of students are willing to do so as it is difficult to find a suitable time and date which all the people involved are available. Hence, lecturer could not provide constructive feedback and advice to the group that does not seek help from them.

1.3.5 Difficulties to Find Group Members

Although most of the students can find a group for their assignments fairly easily, some of the students find that pretty difficult as they are not good at socializing with each other. This may pose a problem for them as they are unlikely to find a group for the assignments and fail eventually. Moreover, it become even more strenuous during the campus closure period as no physical classes were allowed for them to meet physically and form a group.

1.4 Objective

Main Objective

To promote e-Learning, communication and collaboration for students that are taking computer programming courses.

Sub Objectives

- To develop an online platform that provides exceptional communicative power to students so that collaborative e-learning can be fulfilled.
- To provide a conducive environment for students on coding-related group assignments and streamline the workflow.
- To facilitate the interaction between student and lecturer by improving the transparency of students' progress.

1.5 Project Approach

Based on the problem identified, it is more suitable to develop the designated e-learning platform with web technologies. The platform was mainly used for coding for the assignment so mobile development would be absurd. Compared to desktop application development, web application development does not have to concern about the operating system of the users and hence the application will be more unified and easier to implement.

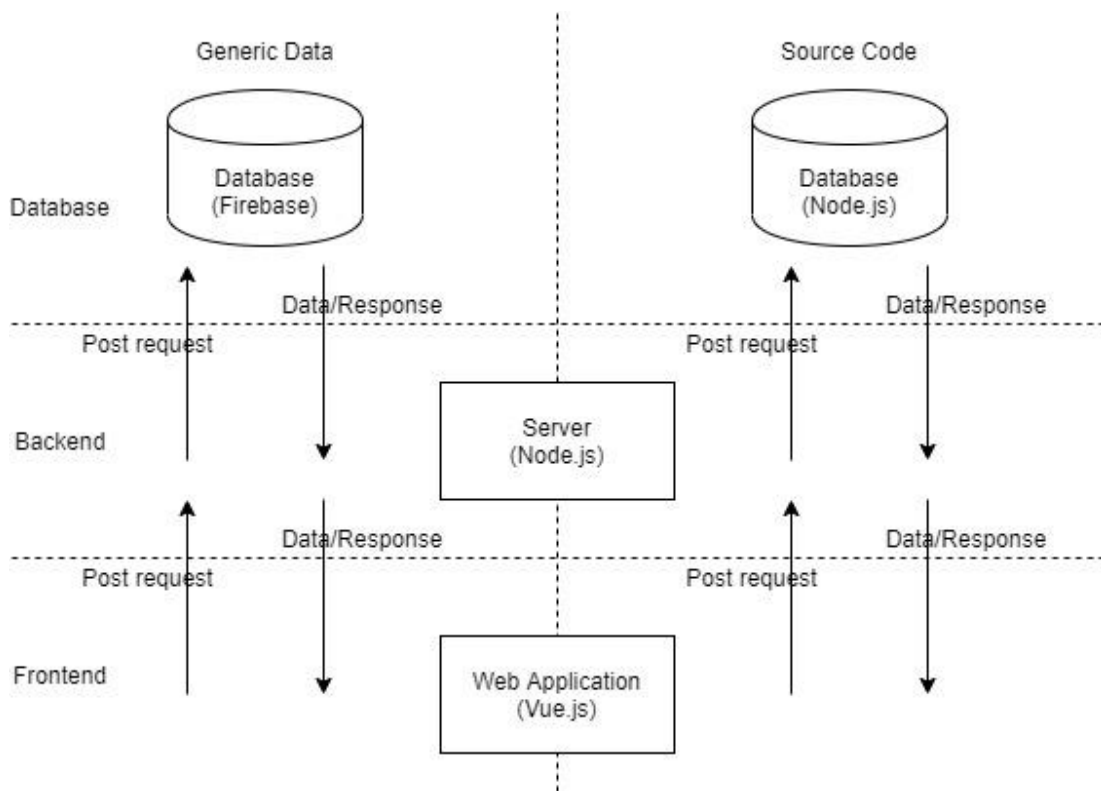


Figure 1.1: System Architecture

The architecture for this application is relatively simple but it is essential and act as foundation for most of the web application in the world. This tree-tier architecture separates presentation layer and database layer by a server layer which process all the necessary logic in between to provide security and flexibility to the application. For source codes, all the changes on the presentation layer will be saved immediately to the database without any checking needed. The changes will be reflected to the presentation layer immediately.

The frontend of the application will be using Vue.js to develop due to the following reasons. Vue is a JavaScript web framework that is highly structured and

reusable using the building block called component. Vue is chosen instead of React and Angular, the other web frameworks because Vue is lightweight, easy to understand and addresses the drawbacks of React and Angular (Kumar, 2019). Using a framework instead of starting from scratch can dramatically improve the efficiency and effectiveness of developing a web application or website as they will be taking care of all those mundane low-level tasks and implementation whilst providing a lot of modularity and reusability within the project.

1.6 Proposed Approach

Methodology is crucial in software development for optimising the time to deliver the product through a step-by-step framework and principles to follow while avoiding unnecessary troubles. Parallel development will be used to implement this project.

Parallel Development

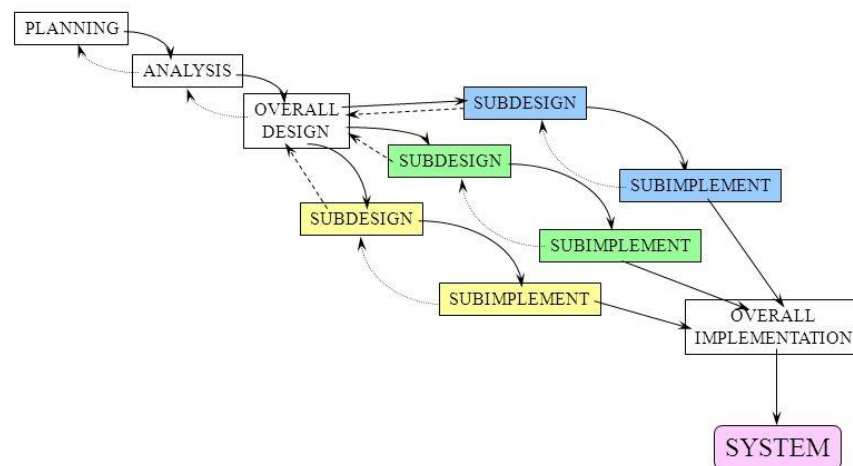


Figure 1.2: Parallel Development (Scott, 2016)

Parallel development is a structured, linear development life cycle similar to the waterfall pattern. The project is broken down into smaller, distinguishable projects that are used to develop the subsystems identified simultaneously to reduce the time needed to deliver the whole software. The subsystems are integrated into the final version of the application eventually.

In this project, the system will be broken down into 2 main subsystems or interfaces which is student interface and lecturer. The student interface will be focusing on code workspace, assignment groups and communication whereas on the lecturer side, it will be focusing on assignment management, group management and performance monitoring. The implemented features will be refined from time to time until the requirement is fulfilled and robust.

1.7 Scope

1.7.1 Targeted Users

The targeted users for this system are the university students or any other students that are having programming projects/assignments as a team. Besides that, the second target user is lecturer that are in charge of the subjects and assignments.

1.7.2 Features Covered

1.7.2.1 Concurrent Text Editing Interface

The user interface will be referring to Visual Studio Code, a free but impactful text editor to date.

The text editor will support syntax highlighting for a multitude of popular programming languages such as C++, Java, C#, Python, JavaScript etc. The proposed text editor will also allow auto indentation when certain keyword was found, just like an IDE that will auto-indent the cursor to one tab deeper when keywords such as curly braces, and so on. The text editor can also provide code snippet and auto completion to improve the accessibility of user.

However, this text editing interface does not compile or run the codes on the go as there will be a plethora of different programming languages that will need to be taken care of, which any one of the fully supported language for the editor can be a project of its own. Users will need to copy and paste the codes from the platform to their native environment in order to debug or get the output. Hence, this editor only serves as a collaborative syntax highlighting web editor for the assignments.

1.7.2.2 Chat Rooms

The chat rooms will act as the catalyst between group members for engaging in constructive discussions while they are working on the assignment together. Each user can have multiple assignment group, and hence multiple chat rooms. Users can

discuss among others in the relevant channel and the chat history will be available to users all the time.

1.7.2.3 Project Submission

After each group has completed their assignment, group leader can submit the assignment file in the form of zip format for multiple files to the server. Other groupmates as well as lecturer will be notified for the submission.

1.7.2.4 Project Management

The system allows students to create a Gantt Chart that outline all the activities and the milestones for the project with a clear duration allocated for the activity to allow students to plan properly without doing all of the things in the eleventh hour. This will balance their time and allow them to work more efficiently towards to goal. Each activity may be assigned to one of the team members by anyone inside the team to show that who is in charge of the tasks planned.

1.7.2.5 Assignment Grouping Mechanism

Students can feel free to create an assignment group of their own and invite others to join. The student that creates the assignment group will automatically be the leader for the group.

Besides creating their own group or joining others, the students can choose to matchmake through the system for the assignment that they are taking. Once the queue has reached the maximum number of allowed students, they will be assigned a new group and a leader is randomly chosen. All the joining events will trigger email and notification sending.

1.7.2.6 Arrange Discussion Time

Students within a group can drop down all their available times within a period and once all of the groupmate had booked the discussion time, the system will process and come out with a time that all of the team members are available.

1.7.2.7 Display Statistics

The statistics is used to get a quick glance of the formation of groups of an assignment such as how many groups formed that consist of 3 members and how

many students has yet to find a group. It will also show the submission status for each group of the assignment as well.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews the topics which is highly relevant to the project title to provide a better understanding on the rationale of doing such research and implementation. The literature review is categorized into the review of e-Learning where the background and scholar adaptations will be outlined. Next, collaborative coding will be reviewed to provide the background as well and what it had to do with students. The last field that will be reviewed is the methodology of implementing a software. This is to provide a clear understanding on what are is pros and cons of a Software Development Life Cycle (SDLC) model and which one is suitable for implementing the project.

2.2 E-Learning

Information Communication Technology has been revamped perpetually and have reached the level of maturity in the 21st century. This has caused the inevitable rise of e-Learning where peoples around the world is utilising the power of World Wide Web invented by Tim Bernes Lee back in 1989 to share researches, knowledge and information in other formats as well. E-Learning has become a trending topic in today's world.

Clark and Mayer (2016) had defined e-Learning as follows: Instruction delivered on a digital or electronic device such as laptop and smart phones that utilises the internet to support learning (Aldiab et al., 2017). The evolution of e-learning over time that caused a paradigm shift on implementations and the difference of setting had gave birth to a variety of definition to e-Learning (Alhabeeb and Rowley, 2017). Although there are a multitude of definitions available for e-Learning in the literature, Tavangarian et al. (2004) suggested that the adoption of electronic media in the learning process is already adequate to define e-Learning. However, Alhabeeb and Rowley (2017) had mentioned a more complex, but widely accepted definition of e-Learning was offered by Sangrà, Vlachopoulos, and Cabrera back in 2012.

The e-Learning became a huge success due to the following factors. Digital device that serves as one of the important mediums to e-Learning can be seen ubiquitously. It allows user to learn anytime, anywhere with little restrictions. The development of multimedia also plays an important role on shaping e-Learning.

E-learning has been experiencing a surge in the global usage during the recent outbreak of the new coronavirus disease. However, even before the pandemic, e-learning is already highly adopted and used in the education technology, reaching USD18.66 billion of investment in 2019 and is forecasted that the amount will grow exponentially to USD350.00 billion by 2025 (Li and Lalani, 2020).

2.2.1 E-Learning, Traditional Learning and Collaborative Learning

Students from the olden days learn by attending a lecture inside the claustrophobic classroom. This is referred to as traditional learning now. This method of teaching and learning have more emphasis on the teacher instead of the learners or students. The process is also imperative, means that students cannot voice out their opinion or object the order. All they need to do is to defer to the orders provided, which makes the learning experience rigid and inflexible. According to Bloom's Taxonomy, traditional teaching method only covers the bottom layers of the hierarchy that is extremely unfavourable in the learning process.

E-Learning, also referred to modern learning because the learning is carried out by applying modern methods such as technology. E-Learning exist to keep up with the evolution of the conventional learning process because of the widespread application of ICT in the classroom setting (Yakubu and Dasuki, 2018). Bouroumi and Fajr (2014) noted that there is little, or no group interaction in the traditional learning approach whereas modern e-learning emphasize on interactions in group of students. This is tell-tale sign that sign of collaboration was not included as part of learning process in the traditional learning.

Lauron (2008) mentioned that collaboration and conversation between students and instructors are the utmost important factor to effective learning. He explains the reasons that collaboration is essential in learning because humans are social beings that gain quality of life and have boosted self-esteem by engaging in collaborative effort as they feel the dependence on each other. Lawson (2004) provided a comprehensive list of benefits that can be achieved through collaboration in learning which encompasses *effectiveness gains* such as improved results or

enhanced problem-solving capability, *efficiency gains* that is eliminating redundant issues, *resource gains* such as expanded funding, *capacity gains* such that weaknesses are covered and the workforce retention improves, *legitimacy gains* such that power and authority are enhanced as well as *social development benefits*.

Collaboration learning is tightly linked to e-Learning as e-Learning provides a wide assortment of tools that makes collaboration convenient and effortless. Moreover, it is cheap to gain access to these tools as most of them provides the core functionality for free. All it takes is just an internet connection to get going. Tools such as instant messaging application, e-mail, video conferencing provider, real-time voice messaging etc. are the essentials to collaborate.

2.2.2 E-Learning in Higher Institutions

The improved accessibility of the internet by the students has caused the educationist change the traditional way to teach and learn (Yakubu and Dasuki, 2018). As the importance of e-Learning has been realised, higher education institutions such as universities and colleges had begun to incorporate e-learning into the university's learning ecosystem by launching their own dedicated e-learning platform. Pei-Chen Sun et al. (2008) believed that this is a crucial move to ensure that limitations of time and space of learners and instructors can be ameliorated through synchronous or asynchronous learning network model (Goyal, 2012). Not to mention that higher education students are one of the most highly networked and connected population that a survey shows that university students bring an average of 3-4 electronic mobile devices to the campus (Perry, 2018) allows e-learning to become omnipresence among the students.

Institutions uses multimedia as an aid upon delivering the classes. In order to foster collaboration among the students, courses of the institution often offer group assignment to harvest the benefits mentioned earlier. Group or individual presentation which requires students to deliver core contents of their work using computerized presentation tools in front of others encourages two-way communication.

Numerous studies had shown the success of e-learning in the context of higher education institution. Studies done in Saudi Arabia (Aldiab et al., 2017; Alhabeeb and Rowley, 2017), Morocco (Bouroumi and Fajr, 2014), Nigeria (Yakubu and Dasuki, 2018) had showed a major success of e-learning in the local university.

2.2.3 Necessity of E-Learning Platform in Higher Education in Malaysia

An online learning platform comprises of numerous interactive online services that provide educators, learners and any men on street to be able to involve in education with the information, tools and resources provided to advocate and enhance education quality. It has been a few decades since the adaptation of e-learning become widespread. However, that is not the *raison d'être* of the e-learning platform. Since March of 2020, there is a major lockdown around the world due to the recent outbreak of Coronavirus 2 (Covid-19) in an attempt to contain the virus from spreading further, causing fatalities.

At the time of reporting by Li and Lalani (2020) on 29th of April, over 1.2 billion students were affected and unable to attend classes as schools were shut globally. This has resulted the distinctive rise on e-learning, where teaching was done remotely via digital platforms. Microsoft Teams, Google Classroom and Zoom is the leading technology offered by well-established company that are being used by many institutions to deliver classes online. Li and Lalani (2020) also reports that there is an increased of retention of information from research and hence the booming e-learning trend are more likely to stay, perhaps not fully, but adapt accordingly.

The first ever case of Covid-19 was diagnosed in Malaysia in 25 of January 2020 and it wasn't until 16 of March when Malaysian government had imposed Movement Control Order (MCO) that will be effective since 18th March as there was an upsurge of infection cases that was allegedly linked to a mass assembly of religious event at the Sri Petaling mosque in Kuala Lumpur (Tang, 2020). All educational institutions were ordered to shut down. The situation lasts for almost half a year until restrictions have been slowly lifted.

For students that are having group assignments amidst this hardship, they are unable to physically meet each other to discuss their project. This is especially true to students that are having computing programming assignments. The code management, communication and workflow management are necessary to progress and achieve milestones on their assignment. Hence, they will need to rely on online collaborative tools to accomplish their task. This may get their task done but however, it is less efficient as the tools are not specifically created to solve the problem. A specially designed e-learning platform is required to empower the

students in doing programming assignments would be a pragmatic approach to resolve the problem.

The H1N1 pandemic in 1918, H2N2 in 1957 to the recent SARS and MERS outbreak shows that air-borne pandemic can happen at any time. Not to mention that another pandemic-potential Swine Flu has been recently identified in China (Cohen, 2020) as researchers say that COVID-19 won't be the last pandemic (Gill, 2020). This might cause another wave of lockdown which students will have to stay at home and will be unable to meet each other and discuss the assignments.

To conclude, these occasions serve as the supporting proof that consolidates the needs of needing such systems for the well-being of student in the future.

2.2.4 Microlearning

Microlearning is a trend of e-learning. Microlearning is the information delivery through small and manageable content such as infographic, tables, visualisations and short video. Studies had shown that students have increased information retention by 20% through microlearning (Giurgiu, 2017). Giurgiu (2017) suggested that inundating learner with large amount of content at once contributes to little interaction between the information and the learner.

2.3 Collaborative Coding

Software projects are meant to deliver complex functionalities that comply with business needs and requirements so that the software can be beneficial to help to solve some of the fundamental problems faced by the community. Hence, they are not an easy task that can be done by one people and it is not viable to write in a monolithic fashion. Hence, multiple programmers will need to work on the program as a team to properly organise the workflow and make it efficient for the delivery of the program. However, it is not easy to collaborate between a team of programmers.

There are two general kind of collaborative programming approach or technique, that is *real-time* and *non-real-time* collaboration (Fan et al., 2019). In non-real-time collaboration, programmers will have to either sending the source code to-and-fro with one another if they are not sharing a single computer or use some version control system such as Git to manage the project and keep them up to date. However, neither of these are helpful for teams that works together in a same file as the supports provided are insufficient for close synchronous collaboration (Goldman,

Little and Miller, 2011b). By using service control system also manifest a problem, which is code transparency. The other collaborators will not know how the assigned programmer is doing and how he is dealing with the problem as the copy of the program he is working with is kept private and it would not be revealed until the code is pushed and merged with the master branch of the program. This may lead to conflicts and inconsistency of software architecture that will need extra time to resolve.

To address the issue, programmers will have to use an online coding platform that rely on cloud service such as real-time IDE to collaborate simultaneously to make their effort real-time and consequently contribute to better coding performance and quality. It allows transparency between the development teams in which misunderstandings and ambiguities can be avoided effectively. If it is made delicately, it may become an inseparable platform for programmers' daily routine. The collaborative platform can provide development environments, code hosting, content storage and more importantly, connection with stakeholders that are unable to meet that frequent. With that being said, the collaborative coding platform should have functionalities not just limited to concurrent coding platform, but also including communication functionalities as well such as text chats, discussion boards, video/voice calls and project management tools are invaluable to foster quality communication among teams especially those who are geographically distributed (Göger, 2019).

According to Göger (2019), the approach of organizing the workspace to put everything regarding the project in one place allow programmers to access important files at anytime and anywhere as long as they have access to the internet. Göger (2019) also mentioned that programmers that are accustomed to the private messaging apps such as Whatsapp and Telegram may find the approach uncomfortable initially because their conversation are visible to everyone in the team. Furthermore, it ensures the perpetual collaboration. If a developer resigned out of the blue on the collaboration work, new people are able to overtake his job with ease.

2.3.1 Collaborative Coding in Student Context

Collaborative coding is also relevant in the student context. Fan et al. (2019) had identified that collaborative coding is beneficial to students who enrolled on the programming courses. In academia, students that are having any programming

courses will have to participate in practical group assignments that are supposed to solve a problem as formulated by their instructor in order to access their practical skills altogether. However, they may find it extremely difficult to collaborate because each member of the group may have different schedules and obligations to adhere to, which makes their time inflexible thus unable to have frequent group meetings and will inevitably slows down their progress.

As most of the students or freshmen are ignorant about or does not know how to use the complicated Version Control Systems to manage their group assignment project, student who work on their team project will have to rely on instant text messaging application to pass around their code to each other. However, those application are not specifically designed to perform team project. The students sometimes will find the chat contents overwhelming as it includes chats, announcement, images, jokes and source codes all mixed together and this will destruct their performance quality at times.

Collaborative, concurrent coding can solve the identified problems, whilst reaping the benefits of collaborative learning as mentioned on the previous section of this paper. With this, there is no need to hassle about sending source code to-and-fro to get the latest version of it. This will inherently solve most of the issues and allows them to have a better collaboration experience. When students collaborate and learn together, they are more likely to retain information and therefore constitute to effective learning. When the students are coding together, they might find be confused or surprised with something unfamiliar that is being used by their peers and this inspires them to communicate internally more actively and learn from the process.

Goldman, Little and Miller (2011a) noted that instructors and students that are having programming courses collaborate with each other to achieve educational milestones but their role are extremely asymmetric as one is structuring the learning process and one is experiencing it and this discrepancies could be mitigated by a programming platform specifically designed for computer science students as their bridge of communication. According to Roddan (2002), a customized programming environment for computer science students is effective on revamping students' learning engagement towards the subject (Lu et al., 2017). With dedicated web-based platform for students to collaborate in coding assignments, instructors are also able to access the students' assignment project files, since all their files are stored in a

centralized cloud server. Instructors can get to know the process of the students in a summary form (Goldman, Little and Miller, 2011a) or directly view their assignment files to make sure that students are having progress and not facing major problems. Instructors must intervene after identified students that are at risk in a timely manner (Lu et al., 2017) to provide appropriate assistance so that they all can catch up with the progress.

2.4 Related Work

2.4.1 Real-time Collaborative Editing

Synchronisation and consistency are important when it comes to providing a real-time reactive coding interface in the web without constantly reload or refreshing it. Multiple collaborators can type at the same document at the same time and all the changes that happens must be reflected to all their screens immediately without errors and delay. Hence, the high local responsiveness must be assured by designing it with *replicated architecture* as most others collaborative application does (Sun and Sun, 2009 as cited in Fan et al., 2019). Fan et al. (2019) had illustrated the consistency maintenance problem that can be summarised on the figure below.

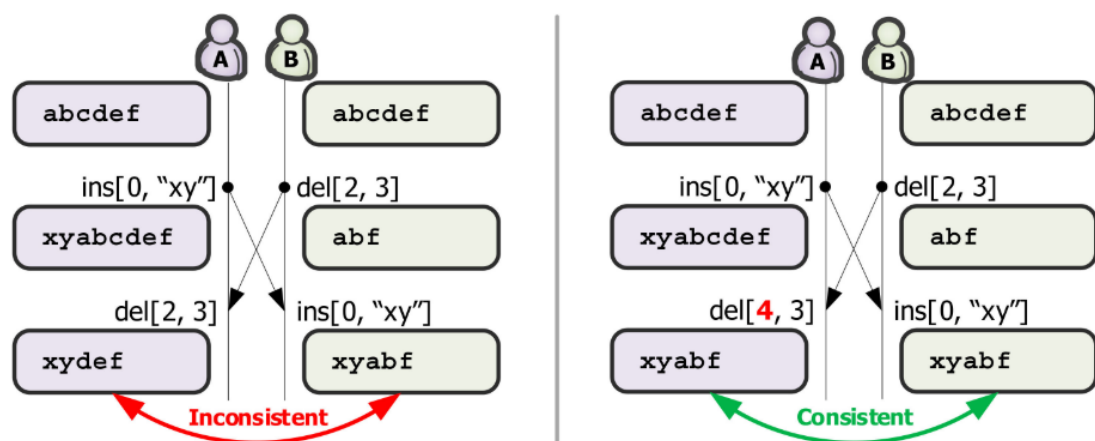


Figure 2.1: Operational Transformation (Fan et al., 2019)

The synchronisation of document can be accomplished by the *operational transformation (OT)* technique (Fan et al., 2019; Rantala, 2015; Kurniawan, Soesanto and Wijaya, 2015). If an edit for a user did not update immediately or failed to do so will result in a forked document that will render the documents unreadable

since any further edits are placed in the wrong position that will cause avalanche effects as the synchronisation gets further disrupted (Rantala, 2015).

However, it is not viable to implement the OT algorithm from scratch as it is extremely arduous and time consuming to implement it (Gentle, n.d.). Testimonial from an ex-Google Wave Engineer Joseph Gentle mentioned that they have used 2 years to develop Google Wave that uses OT algorithm and it will take as long to develop today (Gentle, n.d.). Unfortunately, Google Wave had failed due to lack of promotion.

2.4.2 Important Features

According to Nieminen and Lautamäki (2013), the core features that an online collaborative code editing application must have the following characteristics:

1. Real-time visibility
2. Deploying with underlying errors
3. Project management tools
4. Single click deployment
5. High awareness level between the users
6. Undo/redo function
7. Communication feature
8. Utility for testing and debugging
9. Version control system

The features mentioned will act as a guidance on this project in the requirements gathering phase due to the close resemblance on the nature of the project.

2.4.3 Transparent Adaptation

Transparent Adaptation (TA) is a typical approach for implementing a multi-user collaborative applications by converting from the existing single-user application without tampering or altering the source code of the original system (Fan et al., 2019). TA-based system can be deemed as a single-user application that is equipped with an additional collaborative adaptor that turns it into a multi-user collaborative application. Users are able to use the application both in single and multi-user mode without learning extra functionalities as both operates the same.

Fan et al. (2019) had reviewed a few TA-based applications such as OpenOffice Writer, CoWord, CoMaya, Co-AutoCAD, CoVim, CoWebDraw and Codox Wave in their paper and used the approach to build their collaborative editing addon which will be discussed below.

2.4.4 Implementation of CoVSCode

CoVSCode is a plugin for the official Visual Studio Code (hereafter referred to as VS Code) text editor made by Fan et al. (2019) to provide productive and flawless experience for multi-user collaborative coding. The rationale of doing this project is that the existing real-time collaboration plugins such as Visual Studio Live Share for VS Code and Teletype for Atom are limited to host-participator pattern where one programmer needs to be the host for the session and the files will be stored in that programmer's local machine instead of hosting on cloud that may possibly causes a lot of inconvenience to the collaborators internally.

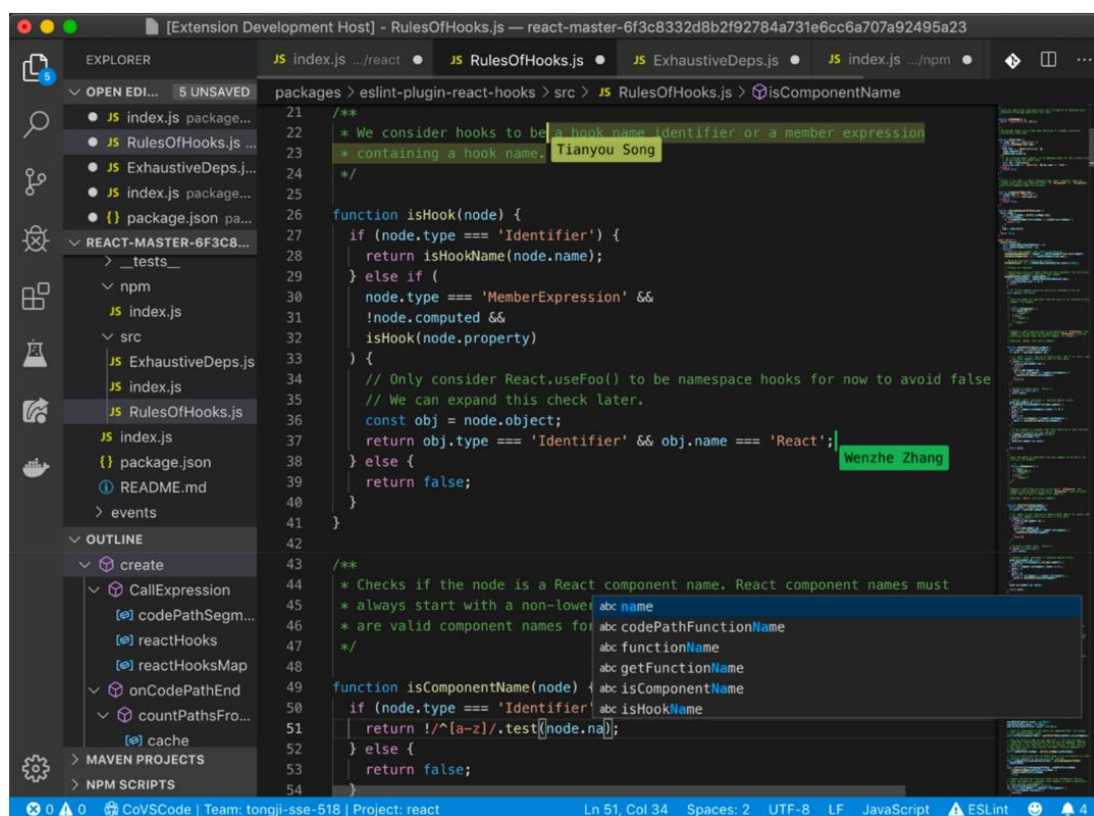


Figure 2.2: CoVSCode screenshot of 3 concurrent users.

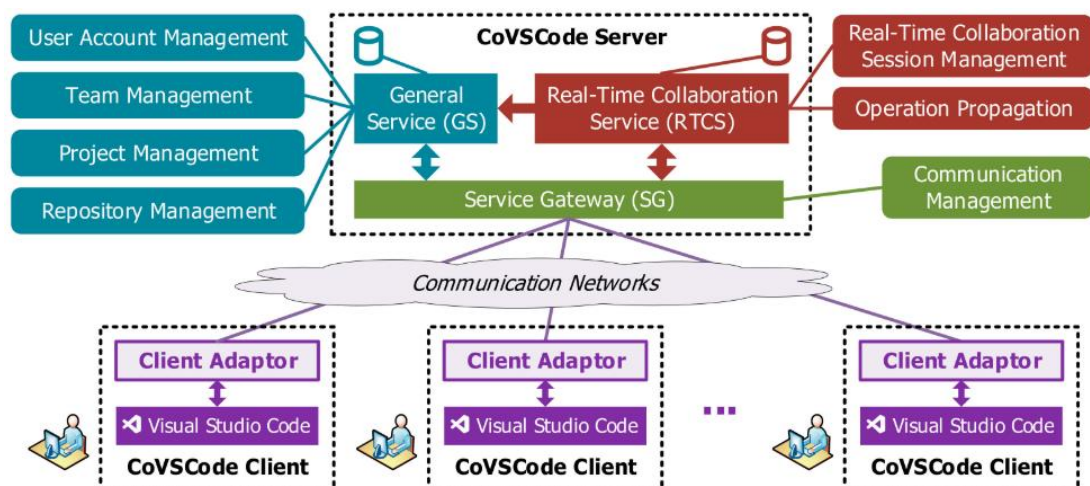


Figure 2.3: System architecture of CoVSCoDe.

Fan et al. (2019) had outlined three core design objectives that the software must achieve and designed the system architecture on above. The CoVSCoDe plugin contains a server and several clients that are connected by communication networks and each client will be occupied by one collaborator. The server can be deployed internally within an organization or hosted via a cloud service.

2.4.5 Implementation of CoRED

CoRED was developed by Nieminen and Lautamäki (2013) which is also a web-based collaborative code editor which get rids of the tedious process of installing, configuring and updating the environment that supports developing, testing, debugging and executing like any other native IDE will do. CoRED can be used like a social media as it provides a chat dialog for the collaborators to communicate through the text messages.

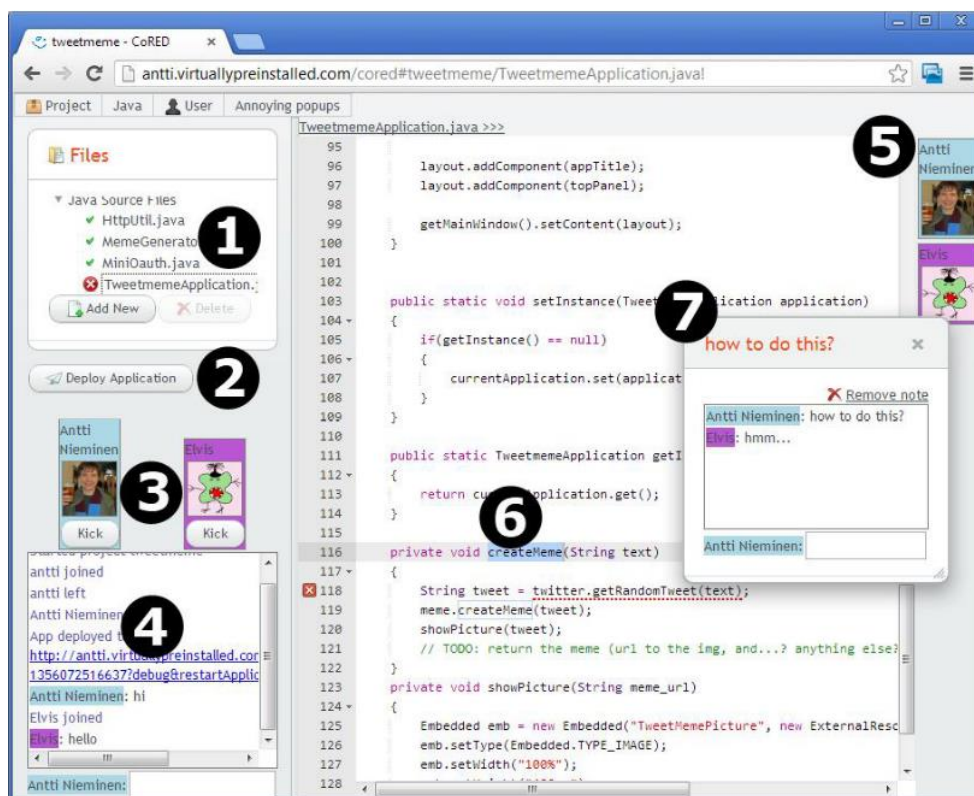


Figure 2.4: Screenshot of CoRED

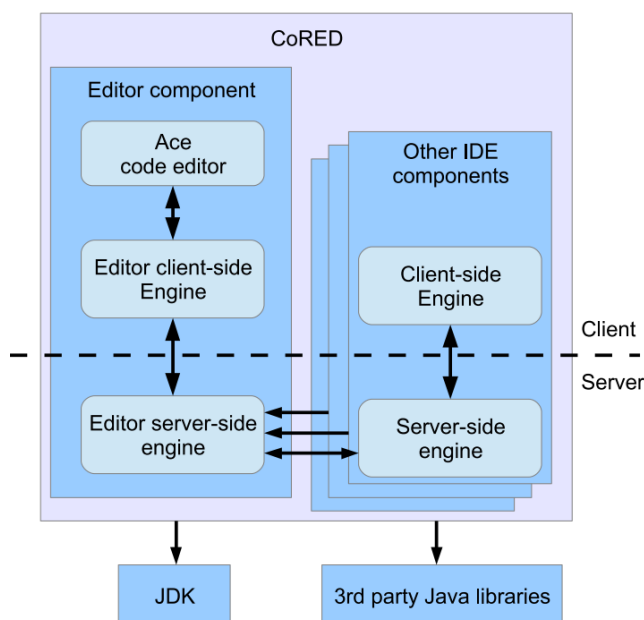


Figure 2.5: System architecture of CoRED

The client-side will handle user interactions of the application and the server-side will handle most of the business logic of the system. CoRED was developed by using Vaadin Framework, a Java framework that provides UI components for a full-

stack web application. The code editor used in this project is Ace Editor, an open-sourced JavaScript code editor.

2.4.6 Implementation of CodeR

CodeR is another web-based collaborative code editing platform that was designed and implemented by Kurniawan, Soesanto and Wijaya (2015). It supports both desktop view and tablet view.

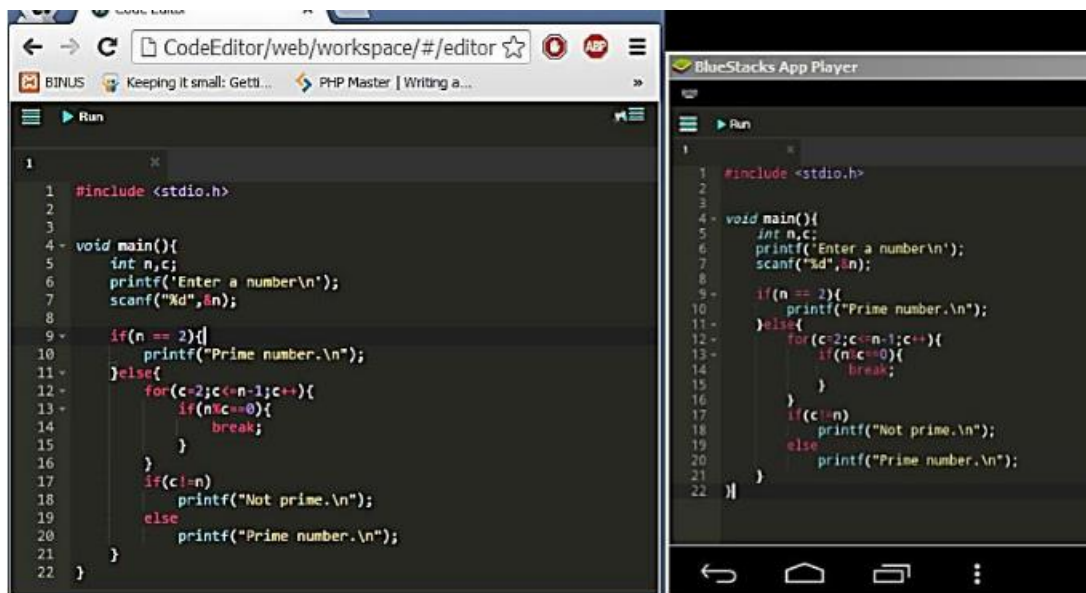


Figure 2.6 Screenshots of CodeR

CodeR uses Facebook plugin to for users to login more easily and provides an additional connection to social media that can boosts collaboration performance. CodeR does not support a variety of programming languages as it offers only C, C++ and Java developments. However, users can compile and run the codes on the cloud and get the output of the program.

```

1: //javascript
2: initialisation flag ← false
3: initialisation flagevent
4: if keyup() then // Run when button was not pressed
5:     if flagevent ≠ undefined then
6:         clearTimeout flagevent // Clear timeout on
           flagevent
7:     end if
8:     flagevent ← function setTimeout 60 // Set the time
           of timeout along 60 seconds
9:     if flag = true then
10:    code ← REQUEST['code']
11:    fileId ← REQUEST['fileId']
12:    send code and fileId with ajax
13:    end if
14:    flag ← true
15:    end function
16: end if
17: if keydown() then // Run when button was pressed
18:     flag ← false
19: end if
20: //php
21: function updateFile(fileId, code) // function updateFile which
           on the service with fileId, and code parameter
22: update code from file database with fileId based on
           parameter
23: end function

```

Figure 2.7: Algorithm for auto-saving code.

```

1: //javascript
2: friendId ← REQUEST['friendId']
3: userId ← REQUEST['userId']
4: accessRestriction ← REQUEST['accessRestriction']
5: send accessRestriction, friendId, and userId with ajax
6: //php
7: function insertCollaborate(friendId, userId, accessRestriction)
           // function insertCollaborate on the service with friendId,
           userId, accessRestriction parameter
8: get in all parameters to file database
9: end function

```

Figure 2.8: Algorithm for real-time collaboration.

Figure 2.7 shows the algorithm for the auto-saving feature of the real-time editor represented in pseudocode whereas Figure 2.8 shows the pseudocode algorithm for the real-time collaboration between multiple different users in one session that are used in the implementation of CodeR.

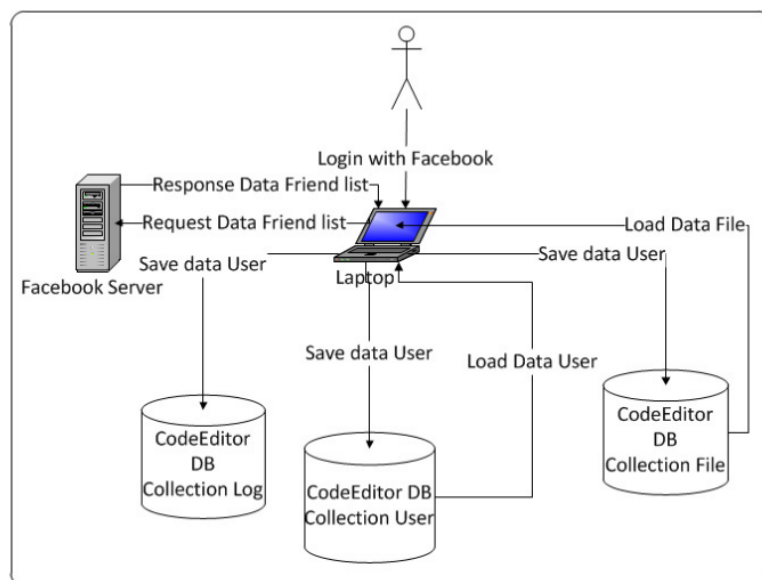


Figure 2.9 Login Architecture of CodeR

The Facebook user can invite other users that are on their friend list to collaborate on the same project. Users are able to communicate with each other as soon as receiving an invitation to join the same project with chat. For non-Facebook user, they will be logged in as anonymous and will automatically connect with other users who is also using anonymous account.

2.4.7 Findings

After reviewing the related works in the past, it was found that operational transformation is a crucial technique to develop the real-time editing system to keep the views consistent. However, it is not viable to code the algorithm from scratch. Hence, the project will adapt existing OT library by the community such as Share.js from Gentle (n.d.) and a more reliable Together.js by Mozilla Foundation (2011).

Transparent adaptation that was used in the implementation of CoVSCode is important as well but it is based on a single user application that is being modified into a multi-user application using an adapter. However, this skill is not likely to be applied in this project unless the project started off by developing a single-user application. More details will be discussed in Chapter 3 and Chapter 4. The nine important features suggested by Nieminen and Lautamäki (2013) is extremely relevant to the project and will acts as the foundation for text editor for this project. The architecture and algorithms used in CoRED and CodeR will be taken into reference for this project.

2.5 Software Development Methodology

A software development life cycle (SDLC) model is also referred as software development methodology is a conceptual framework describing all activities and principles to follow in a software development project begin from planning down to the maintenance. The process can be undergone a myriad of development methodologies, where each of the methodologies has its own unique lifecycle. This term is also known as software development process model. However, choosing them would not be an easy or arbitrary task as it needs to take a lot of factors into consideration before a decision can be made. Alshamrani and Bahattab (2015) highlight the importance of considering their properties and their strengths and weaknesses before choosing the SDLC model, and when to apply the suitable model. According to Kumar and Bhatia (2014), software engineering methodology is classified into traditional & classical methodologies which emerges from the early days and modern & popular methodologies two categories as shown in the figure below.

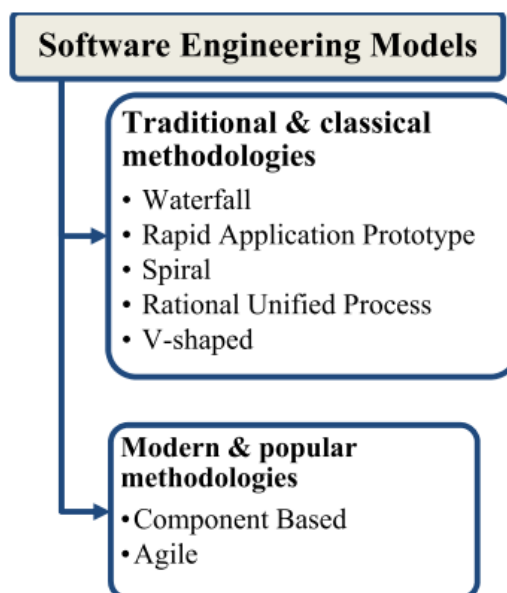


Figure 2.10: Software Development Methodologies (Kumar and Bhatia, 2014)

2.5.1 Traditional & Classical Methodologies

2.5.1.1 Waterfall Model

Waterfall model is the oldest SDLC model existing to date, that was attributed to Walter Royce back in 1970 (Petersen, Wohlin and Baca, 2009). However, incorporating waterfall model in the software development process has several

problems. Due to its intensive documentation and planning, this model is suitable for projects which quality control is a major concern (Alshamrani and Bahattab, 2015). This model employs a linear and sequential method of delivering a software within its process whole lifecycle of the project. As the model is not change tolerant, frequent rework will leads to unpredictable software quality (Petersen, Wohlin and Baca, 2009)

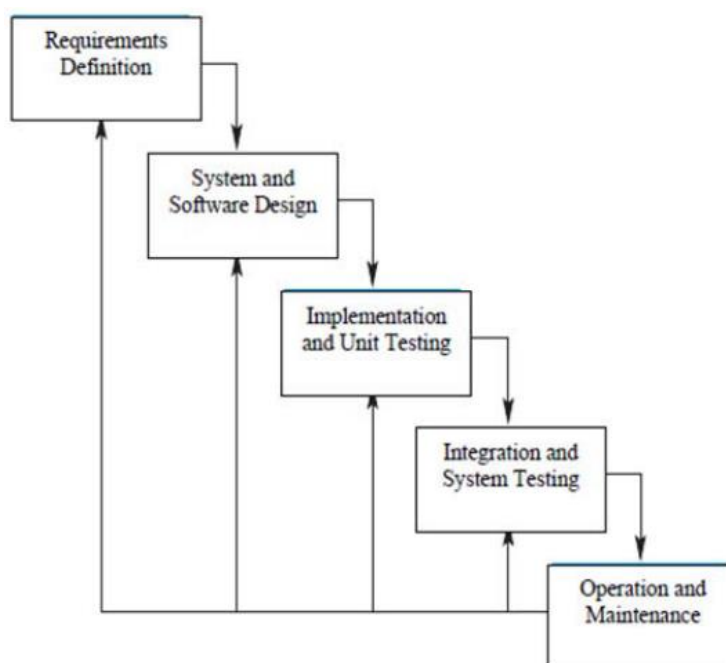


Figure 2.11: Waterfall model (Kumar and Bhatia, 2014)

Waterfall model is typically divided into 6 different phases within the whole life cycle. The first phase is the requirement gathering stage that aims to obtain a detailed, through key requirements and scope that is zero ambiguity from the assigning clients through intensive requirement gathering methods such as interview sessions and observation. It is also crucial for the clients to know what they want. Waterfall requires rigorous and extreme documentation in the initial phase.

Waterfall is not flexible. If changes are needed, the whole process will at worst, restart anew. The documentation is very detailed and elaborate as it is done during each phase of the process before transitioning to the next phase. However, documentation process is extremely time consuming. The Waterfall model is suitable for projects that the requirements are more likely to be fixed and unwavering. It is also suitable for project is short and simple, providing the requirement is crystal clear. The environment and technology tools used are stable.

2.5.1.2 Parallel Development

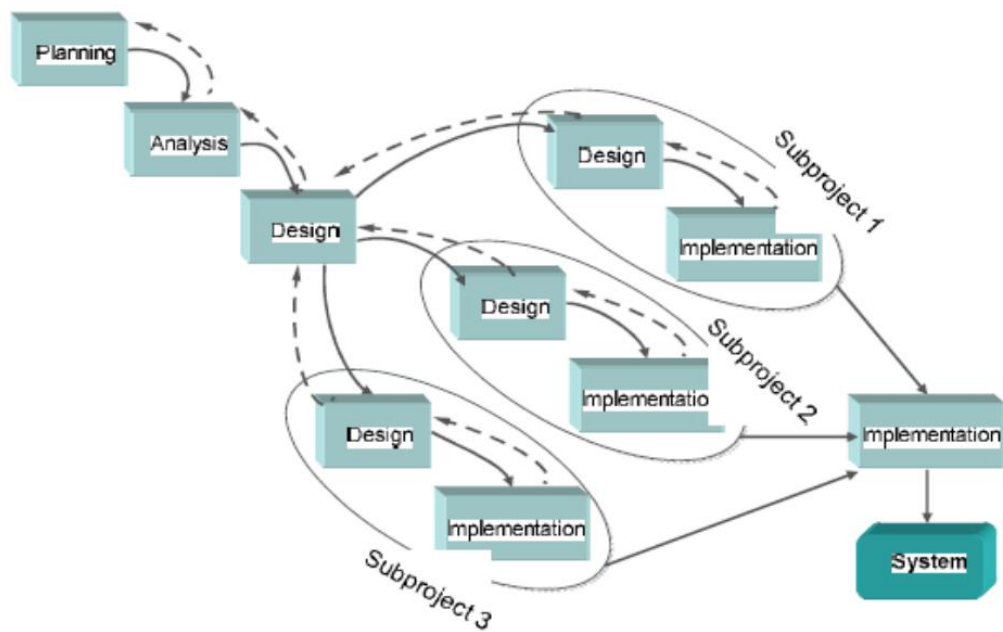


Figure 2.12: Parallel Development (Nugroho, Hadi and Hakim, 2017)

Parallel development on the other hand is a variant of the Waterfall model. Parallel development is needed when a project requires separate for faster delivery. Parallel development attempts to mediate the long gap between analysis and the delivery stage. Instead of doing the design and implementation in sequence, it separates the design and implementation part into several distinct subprojects that can be implemented simultaneously. The final project will be integrated from the several subprojects together ultimately.

Parallel development has its downfalls. It could not be exempted by the problems that causes long delivery time that was suffering by Waterfall Model and sometimes the subprojects are dependent on each other as the decision made on this subsystem may affect the subsystem that had done previously and causes a need to change (Nugroho, Hadi and Hakim, 2017).

2.5.1.3 Rapid Application Development (RAD)

Rapid application development is a category of a newly emerged methodology trend or category compared to the structured approach and begun to make its debut back in the 1990s. Most RAD-based methodologies advocate the development team to speed up analysis, design and implementation by using special techniques and computer tools, such as computer-aided software engineering (CASE) tools and joint

application design (JAD) sessions. Visual programming languages such as Visual Basic, C# or Swift can significantly simplify and speed up the programming process. The three notable methodologies for RAD are *phased development*, *prototyping* and *throwaway-prototyping*. However, not all projects are suitable for using RAD model. Only the system that can be split into modules is applicable for the model. Besides, it requires skilful developers and designers because the modelling skill is crucial for this methodology work out (Kumar and Rashid, 2018).

2.5.1.4 Iterative Model

Iterative model or incremental framework emphasis on initial and simplified implementation which is then improved iteratively to adding in more complexities and features until a final version of the desired software is completed (Powell-Morse, 2016). This model is basically a waterfall model in an iterative fashion so that each iteration produces a software deliverable that is incremented into the software application to produce increased functionality (Alshamrani and Bahattab, 2015). Prototyping, Rational Unified Process (RUP), agile development, Rapid Application development and Spiral model are the examples of iterative model.

2.5.1.5 Spiral Model

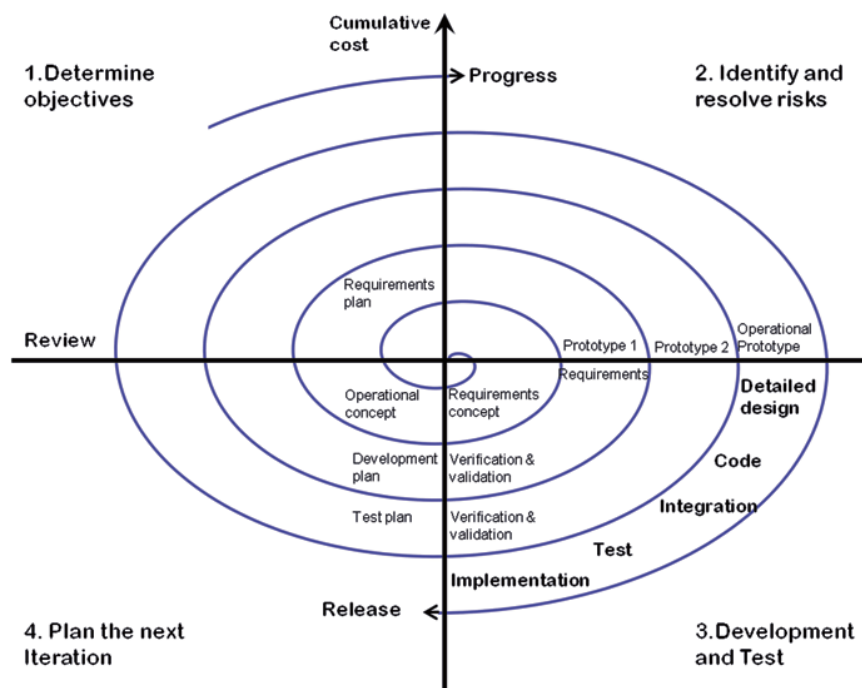


Figure 2.13: Spiral Model

Spiral model combines elements of design and prototyping together to harvest the benefits of both top-down and bottom-up approach (Alshamrani and Bahattab, 2015). Spiral model can help to minimize risk by breaking the project into smaller segment and building high-risk part of the project first during the inception. The versions of software produced by increments is called a prototype until a fully satisfied production version is developed. The typical iterations or phases for spiral model is planning, risk analysis, development/engineering and evaluation phase. After each phase, if the project is not completed yet, the next iteration begins.

According to Kumar and Rashid (2018), Spiral model is very costly to use as it requires highly specific expertise for risk analysis and it is not suitable for small projects because they have lower risk as the key to success with this model is the risk analysis.

2.5.1.6 V-Shaped Model

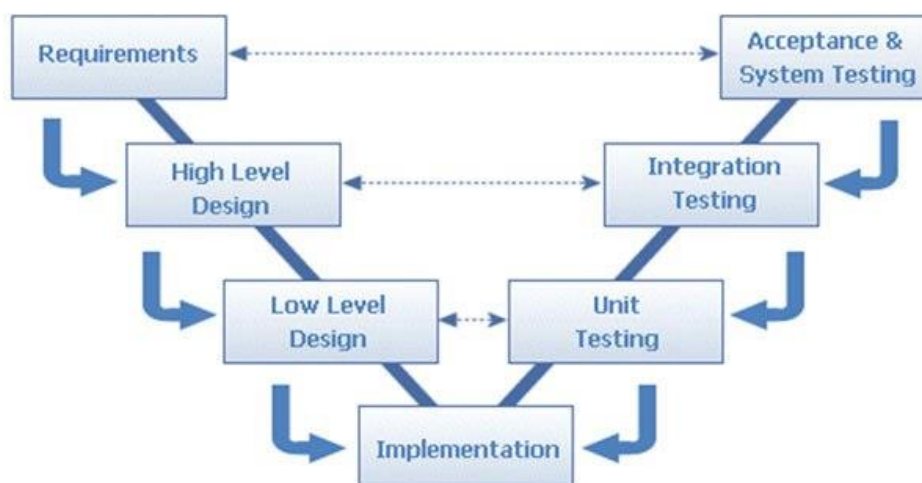


Figure 2.14: V-Shaped Model

V-shaped model is similar to waterfall model as it involves the step-by-step approach in the process. It is an extension to the waterfall model in which the phases after the coding or implementation phase were bent upwards forming a V-like flow in which testing or integration phase for associated phase were done after the main phase was completed (Kumar and Bhatia, 2014). An example of the pairs will be requirements gathering and user acceptance test. The early involvement of testing will allow the developers to identify any underlying problem at an early stage of development so that they can handle and fix it to avoid expensive defunct that waterfall model often

produces. However, it is still a rigid methodology that intended changes to the software may cost an exorbitant amount of money (Nugroho, Hadi and Hakim, 2017).

The other disadvantages is highlighted by Kumar and Rashid (2018) that the software will be only developed during the implementation phase and thus there is no tangible product or prototype during the earlier phase of development. Besides that, if any changes happen in the middle of the process, the test documents along with the requirement documents needs to be updated.

2.5.1.7 Rational Unified Process (RUP)

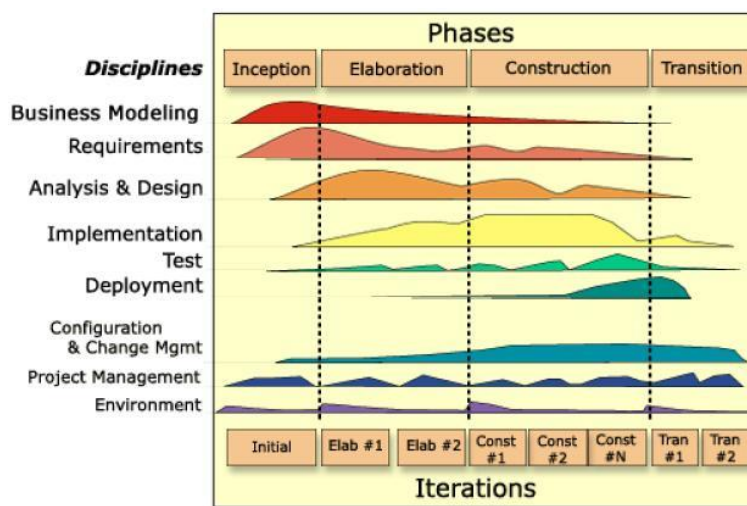


Figure 2.15: Rational Unified Process

Rational Unified Process, also called as Unified Process is an agile framework that separates the whole development life cycle into a series of process and workflows. RUP is not a solid SDLC methodology but instead it offers the rules and principles for the development team to follow and adapting to other methodologies that is suitable for the process/program (Powell-Morse, 2017). RUP is an iterative approach for object-oriented applications (Kumar and Bhatia, 2014). Workflows defines the logical process that a program can have, such as business modelling whereas the phases for all workflows is the same, which is inception, elaboration, construction and followed by transition. Different project may have different workflow but same phases for each activity.

2.5.2 Modern Approach: Agile Methodologies

Agile methodologies are the processes that adhere the agile philosophy such as agile values and principles in which each agile methods consists of different practises that defines what are the things that a software developer should do in a daily basis (Alsaqqa, Sawalha and Abdel-nabi, 2020). The sprints used in Agile help teams to respond to the uncertainty of building software through incremental or iterative work strategy (Kumar and Bhatia, 2014).

2.5.2.1 Feature-driven Development

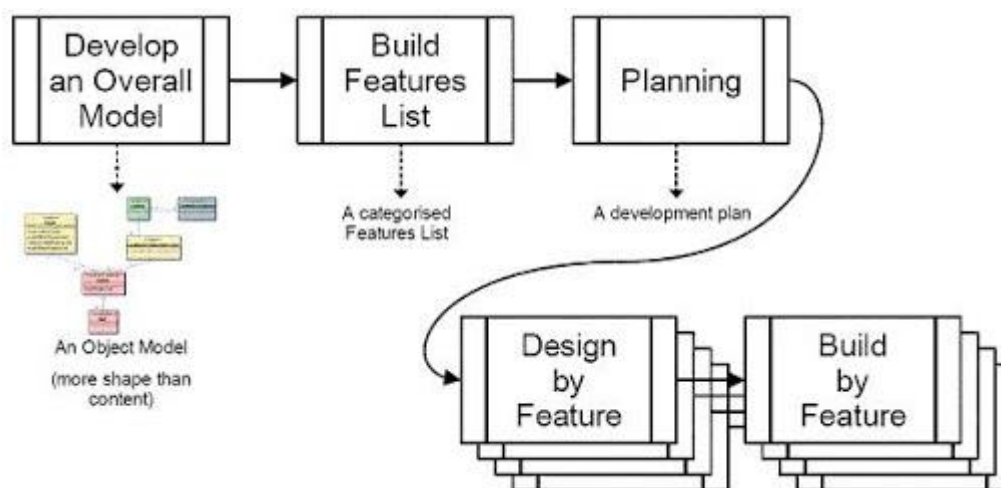


Figure 2.16: Feature-driven development

FDD is a pragmatic software process that involves the focus of client and architecture, and frequent delivery of working software and it is good for smaller development teams (Sajewski, 2019). FDD works by deconstructing the program into a set of features that are to be delivered frequently to the client. The acceptable delivery time is between 1 to 4 weeks but is usually 2 weeks where features must be delivered every fortnight and if the feature has the potential to take any time longer than that, it must be further broken down into several more features to comply with the rule. The five main activities of FDD are developing an overall model, creating a feature list, plan by feature, design by feature and build by feature that are performed iteratively. FDD is often used as it is highly adaptive and can accept late changes by the client (Alsaqqa, Sawalha and Abdel-nabi, 2020).

2.5.2.2 Scrum

Scrum is a solid implementation of the agile framework that aims to deliver highest value/quality in the shortest amount of time (Alsaqqa, Sawalha and Abdel-nabi, 2020). Scrum is ideal for small teams that are having less than 10 engineers and will be more effective if the workers are experienced and disciplined as Scrum relies heavily on self-commitment, self-organization, and emergence rather than command-and-follow manner (Kumar and Bhatia, 2014). Scrum process has 3 phases: *pre-game*, *development* and *post-game*. Scrum is a most used agile methods because of the simplicity and focus on software development issues that is applicable to many domain of development (Alsaqqa, Sawalha and Abdel-nabi, 2020).

2.5.2.3 Test-driven Development

Short development cycle that is repeated a number of times are the core aspect of Test-driven development. The test case was first produced and followed by the codes necessary to pass the test. Although Kent Beck came out with the idea of TDD in 2003, it was already used by NASA in 1950 when working with the Mercury project (Alsaqqa, Sawalha and Abdel-nabi, 2020)

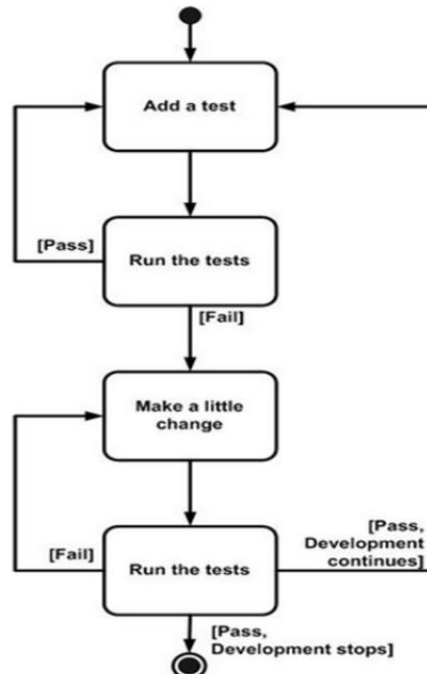


Figure 2.17: Test-driven development (Kumar and Bhatia, 2014)

Test-driven development does not require debugging as all the codes are being tested earlier in the development process to ensure that they work as expected.

This method is also making the development process easier as the software has been broken down into small parts and the developers are focusing on one problem only for each iteration. However, Kumar and Bhatia (2014) suggest that TDD may lead to over testing and re-write of test case when requirement changes that will waste a lot of time. Alsaqqa, Sawalha and Abdel-nabi (2020) suggest that the over-reliance of test case will lead to poor documentation and programmers are taking extra job as testing is usually done by software testers, which is another field of expertise. Doing both at once may lead to extreme confusion.

2.5.3 Findings

It is undeniably that each methodology has its own strength in handling a particular workflow. However, there is no perfect methodology as they come with their trade-offs as well. So, it is crucial to select a proper methodology that fits the nature of the project.

Since this project is a solo project, any methodologies that is heavily oriented to teams will be disqualified, such as agile methodologies. Besides, there is no risk associated to this project and therefore spiral model is not suitable as well. Rational Unified Process can be excluded as well as the working environment for this project does not involve business context. From Waterfall, Parallel and V-Shape, waterfall may be excluded as it involves long time delivery for a project whereas V-Shape is unnecessary because if any changes happens in the middle, the test documents and requirement documents will need to be changed (Kumar and Rashid, 2018). Hence, parallel development is chosen for this project. Parallel was chosen because the project also involves several subsystems which is online text editor, communication platform and miscellaneous system that can be developed in parallel. Also, Parallel Model is the appropriate choice because the subsystems implemented are independent of each other, which negates the worries of inter-dependency issues mentioned by Nugroho, Hadi and Hakim (2017).

2.6 Similar Systems

In this section, application that have similar features that this project is going to deliver will be reviewed. The reviewed applications are grouped into 3 main category which are *text editor*, *communication application* and *project management*.

2.6.1 Web-based Text Editor

Text editor is essential for any form of coding. Without a proper, delicate text editor, the coding process will be extremely tedious and un motivating. Besides, there might be fatal errors that students might not notice because there is no syntax highlighting to provide guidance and tips to them. Hence, a well-made text editor is important as it empowers programmers to become productive and saves a lot of time. Besides text editors chosen by myself, some of the real-time collaborative text editor reviewed by Rantala (2015) will be selected and reviewed once again in the following section.

The online text editors selected are as below:

1. Codassium (<https://codassium.com>)
2. CodePen (<https://codepen.io>)
3. Codeanywhere (<https://codeanywhere.com>)
4. Code Sandbox (<https://codesandbox.io>)
5. Code Share (<https://codeshare.io>)
6. Collabedit (<http://collabedit.com>)
7. Expo (<https://expo.io>)
8. GitPod (<https://gitpod.io>)
9. JsFiddle (<https://jsfiddle.net>)
10. JS Bin (<https://jsbin.com>)
11. Play Code (<https://playcode.io>)
12. StackBlitz (<https://stackblitz.com>)

The full comparison of the text editors can be found at the end of the section where analysis was made. The review for this field will be sub-sectioned for cleanliness and ease of read.

2.6.1.1 Codeshare

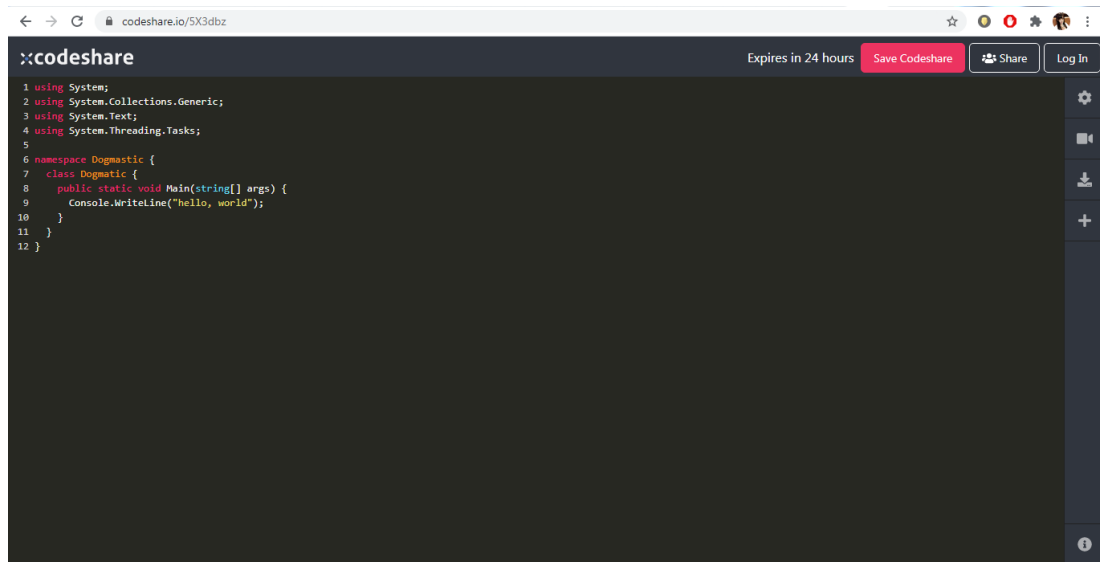


Figure 2.18: Screenshot of Codeshare

Code Share is an online text editor that is made to collaborate with others. It is marketed as a tool that allows programmer to code as a team, coding interview and teaching programming as it has a built-in video call feature. There is a plethora of programming languages supported including esoteric languages such as Brainfuck which may be very handy for users that code in uncommon languages.

2.6.1.2 Codeanywhere

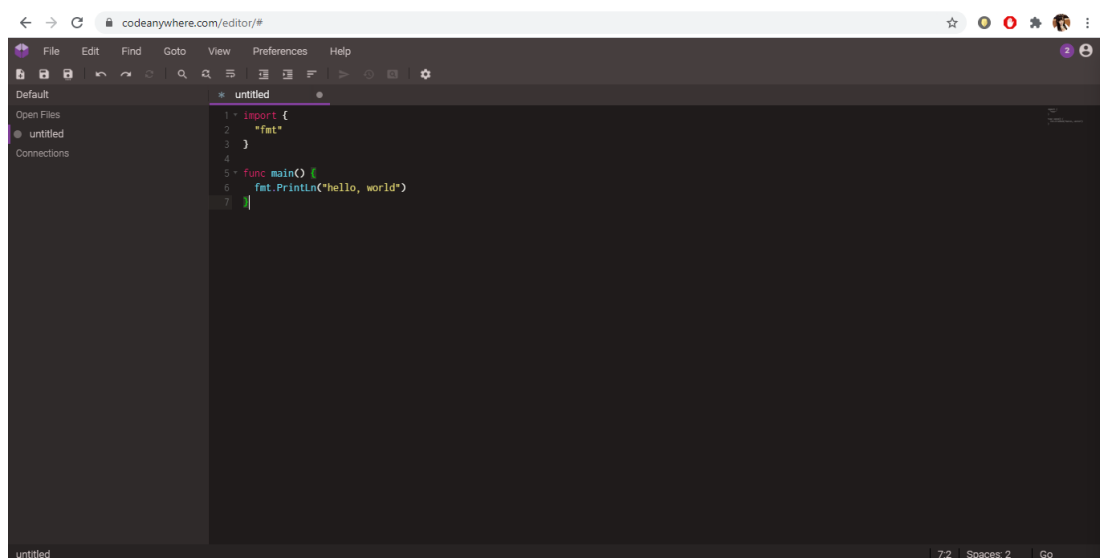


Figure 2.19: Codeanywhere

Codeanywhere is a cloud IDE that enables fast deployment from a development environment, all done just inside the browser. It is integrated with numerous famous file repository service out there such as GitHub and Bitbucket where user can perform the development of any repositories in their account if they are logged in with the specific platforms as mentioned.

2.6.1.3 GitPod, StackBlitz and CodeSandbox

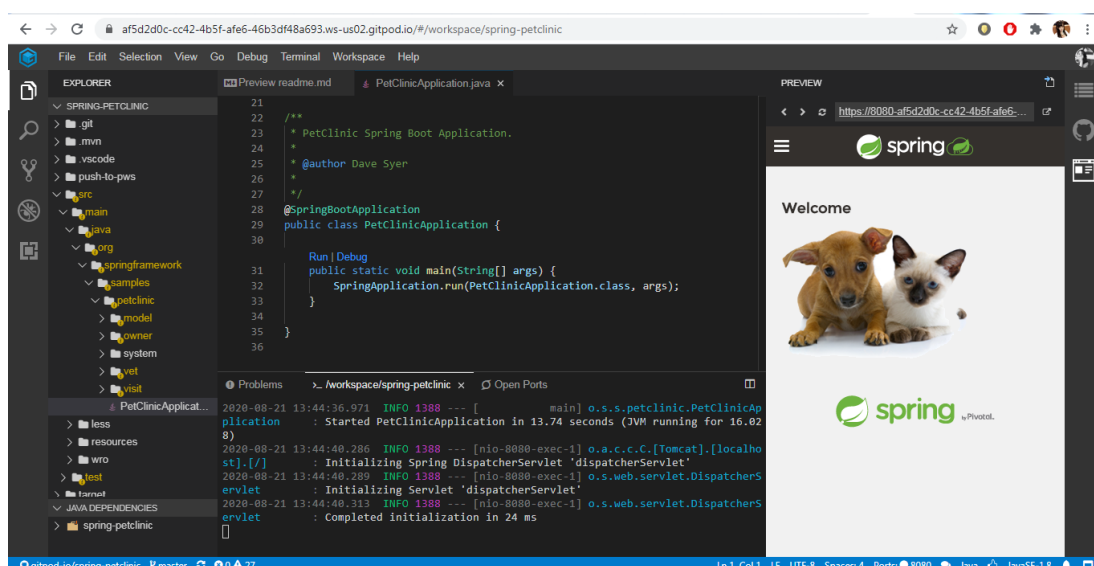


Figure 2.20: Screenshot of GitPod

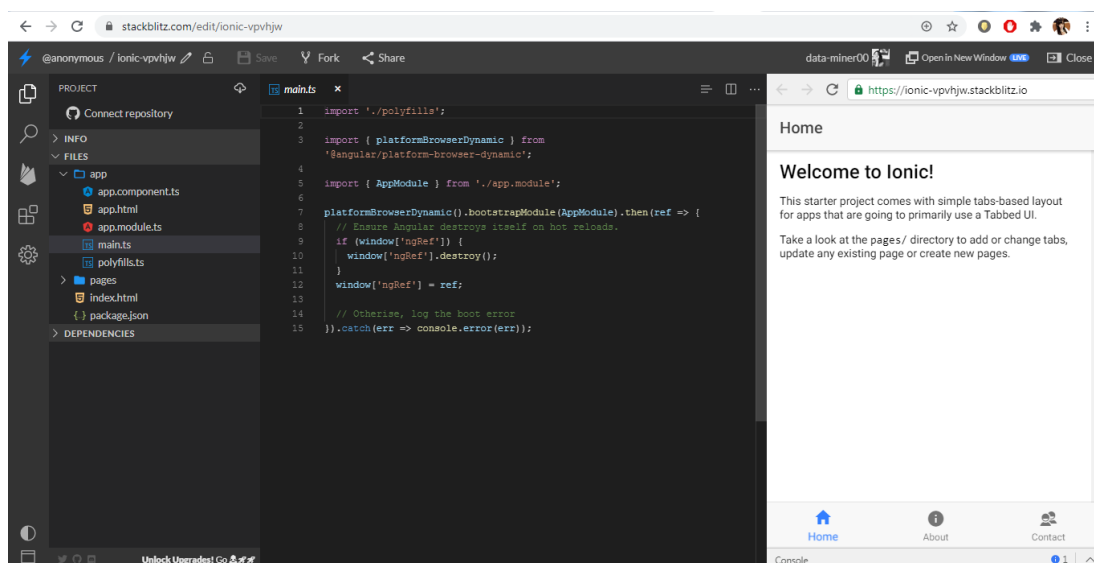


Figure 2.21: Screenshot of StackBlitz

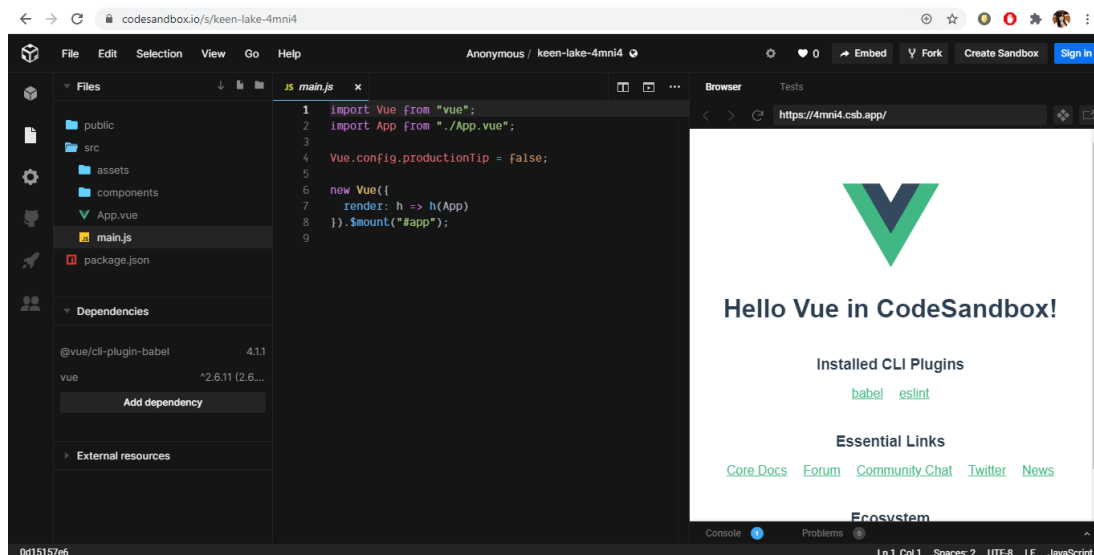


Figure 2.22: Screenshot of CodeSandbox

GitPod is an online cloud IDE that is built for GitHub and GitLab users that allows them to execute their program in one click. It has the exact same user interface as VS Code's as well as a built-in terminal as shown on the figure above. However, Gitpod is not free, it costs \$8 nett per month. Similar to GitPod, StackBlitz and CodeSandbox are the online IDE that are having the same user interface as VS Code's and they offer support to React, Angular, JavaScript, Svelte, TypeScript, Ignite, Ionic, RxJS and Vue. StackBlitz also offers integration with GitHub.

2.6.1.4 Expo

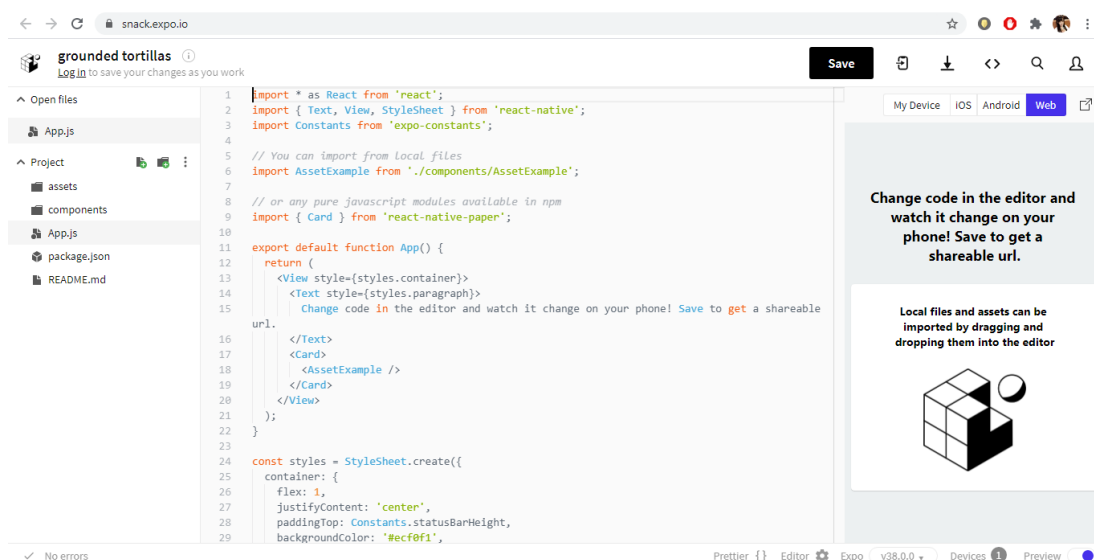


Figure 2.23: Screenshot of Expo

Expo is an online IDE for cross-platform mobile development with React Native. Expo offers a clean user interface with easy accessibility to the files and folders and the output will be shown in a mobile-sized display on the right of the page.

2.6.1.5 Codassium

Codassium on the other hand is specially made for real-time coding interview session by an interviewer and joined by a candidate with video call feature. It is also priced and no free access is available.

2.6.1.6 Collabedit

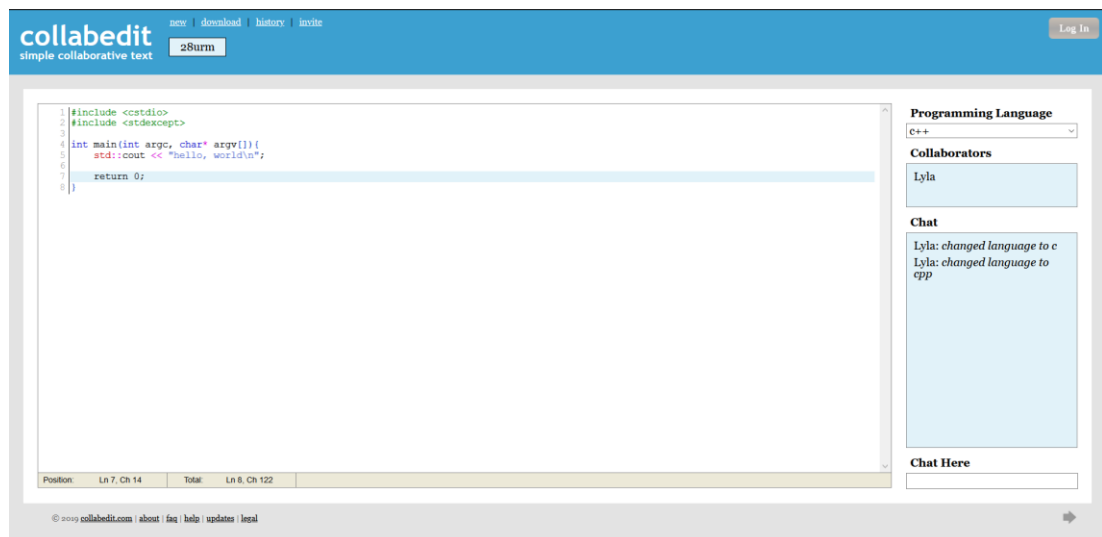


Figure 2.24: Screenshot of Collabedit

Collabedit is the simplest, yet very similar to the requirements for this project. It allows synchronous editing to peers in programming languages in most of the popular languages and provides syntax highlighting to them respectively. Collaborators will have to just access the URL link for the document and able to edit right away. However, it only allows one file for editing only. There is no workspace or repository like a typical project would have and this is inconvenient when working with a project.

2.6.1.7 JsFiddle, Playcode, Codepen, Jsbin

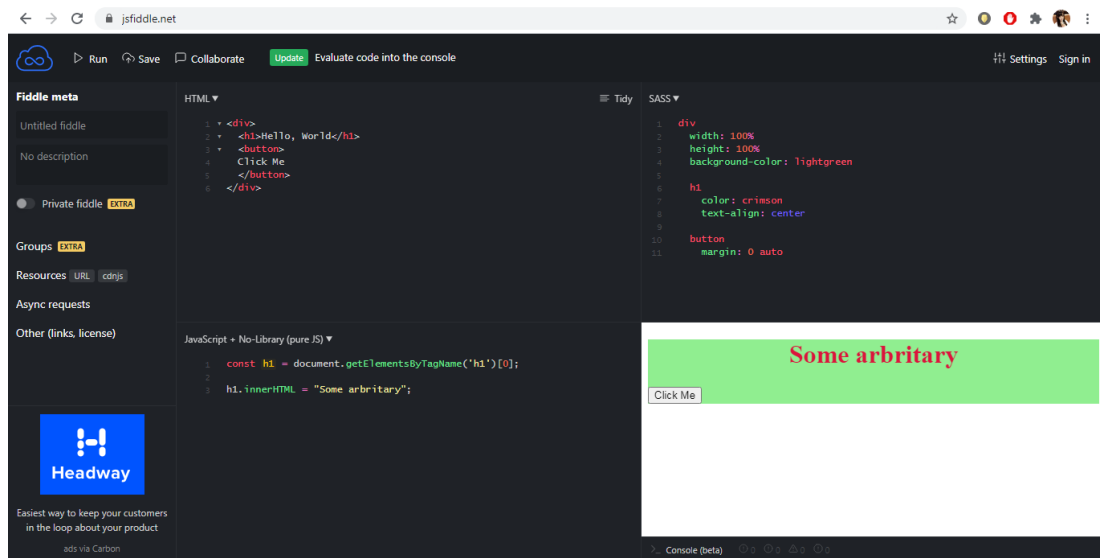


Figure 2.25: Screenshot of JsFiddle

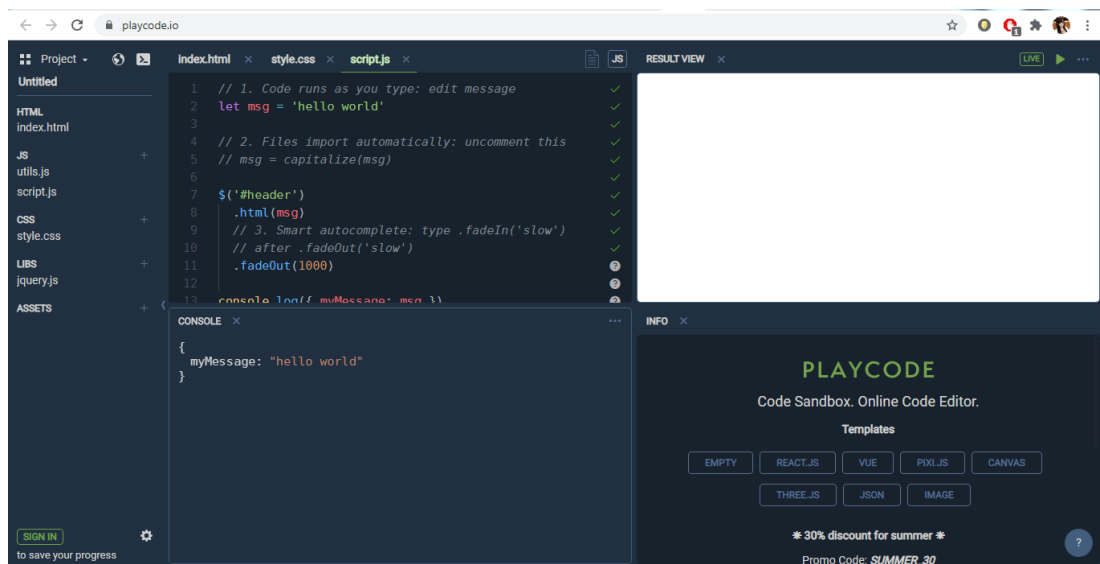


Figure 2.26: Screenshot of Playcode

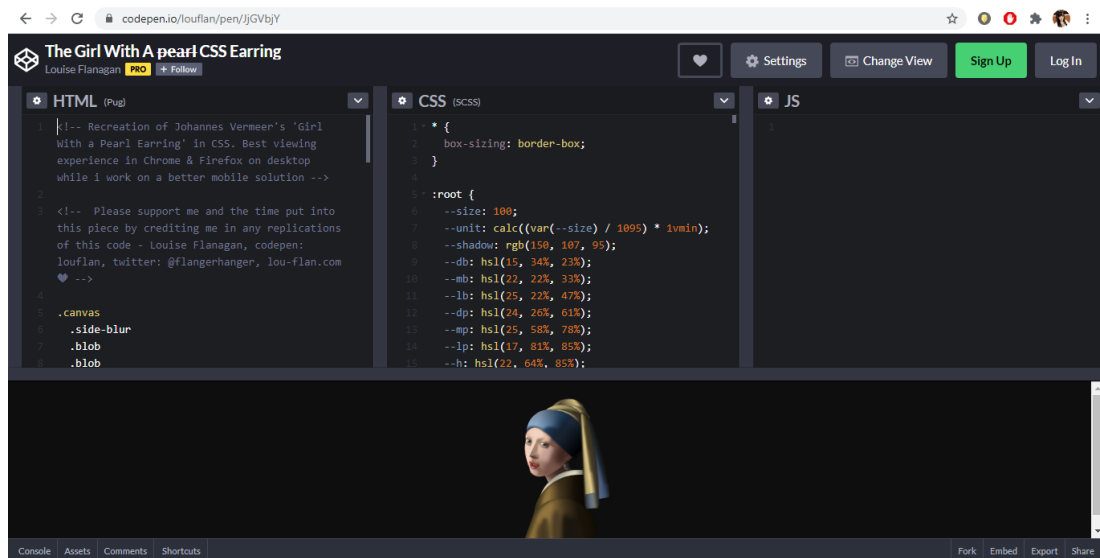


Figure 2.27: Screenshot of Codepen

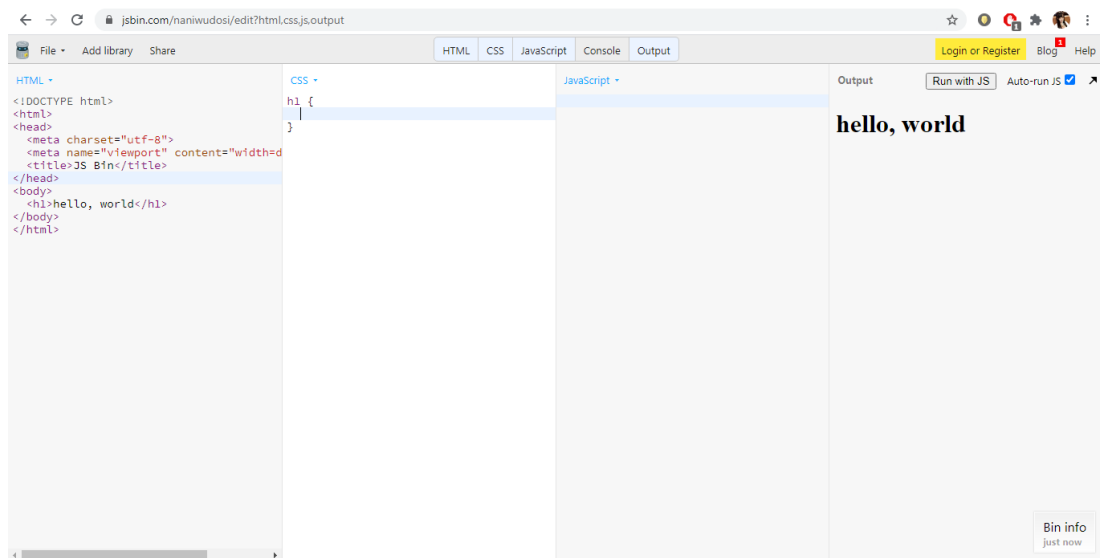


Figure 2.28: Screenshot of Jsbin

Jsfiddle, Codepen, Playcode and Jsbin on the other hand is the pastebin for web development that utilises HTML, CSS and JavaScript and their respective framework such as Pug, SASS and React that the result will be rendered inside a separated window within the browser. The real-time collaboration for Jsfiddle is free but is paid for Codepen whereas Playcode and Jsbin does not support real-time collaboration. They are meant to produce widgets or showcase ideas that uses web technologies and are not suitable for generic programming.

2.6.1.8 Summary and Comparison of Web Text Editors

Table 2.1: Text Editors and Its Feature

	Text Editor	Pastebin	IDE	Collaborative	Text chat	Voice chat	Video call	UI
Codassium	✓	✗	✗	✓	✗	✗	✓	No Data
Codepen	✓	✓	✗	✓ (Paid)	✓	✗	✗	Impressive
Codeanywhere	✓	✗	✓	✗	✗	✗	✗	Clean
Code Sandbox	✓	✗	✓	✓	✗	✗	✗	Impressive
Code Share	✓	✗	✗	✓	✗	✗	✓	Clean
Collabedit	✓	✗	✗	✓	✓	✗	✗	Basic
Expo	✓	✗	✓	✗	✗	✗	✗	Impressive
GitPod	✓	✗	✓	✗	✗	✗	✗	Impressive
Jsfiddle	✓	✓	✗	✓	✓	✓	✗	Clean
Jsbin	✓	✓	✗	✗	✗	✗	✗	Normal
Play Code	✓	✓	✗	✗	✗	✗	✗	Clean
StackBlitz	✓	✗	✓	✗	✗	✗	✗	Impressive

Based on the code editors reviewed, the pastebin code editor's unique multiwindow feature can be taken into reference as the students sitting for the usability test conducted by Nieminen and Lautamäki (2013) on their custom collaborative IDE called CoRED had complained that the lack of multiwindow is hard to work. Instead of 4 windows offered by the pastebin editor, it is more suitable to implement 2 windows only as 4 windows will get the screen too clogged for the programming purpose as pastebin are based on web technologies such as HTML, CSS and Javascript and is not for general purpose programming.

Besides, the online IDE such as Expo and GitPod will be taken into reference on how they store and manage the file directory on the server for easy retrieval. StackBlitz, GitPod and CodeSandbox will be taken into consideration on how they design the UI layout that provides intuitive navigation so that user would not be confused and upset about the platform due to frustration on usability issues.

Collabedit on the other hand offers a bland UI. However, it includes all the essential component that this project needs – a simple real-time editor, a single-channel chat box and a programming language selection. It will be taken into reference on how to construct the simple editor and the chat systems in the early development of the project.

2.6.2 Communication Applications

Communication is deemed an important factor that leads to success in every realm. Since the communication online had shown its effectiveness, people often use applications such as social media, forums, discussion boards and instant messaging application as the means to relaying their message to the targeted parties. Instant text messaging application will be the primary focus of this section. The application chosen are listed below:

1. Whatsapp (<https://whatsapp.com>)
2. Telegram (<https://telegram.org>)
3. Discord (<https://discord.com>)
4. Slack (<https://slack.com>)
5. Microsoft Team (<https://www.microsoft.com/en-my/microsoft-365/microsoft-teams/group-chat-software>)

2.6.2.1 Whatsapp and Telegram

Whatsapp is the most used instant messaging application that top 2 billion users worldwide as of mid-February this year (Butler, 2020). It is not without reason as Whatsapp offers a lot of essential functionalities that are very useful in communication such as instant text messaging, voice messaging, video or voice call, location sharing, file sharing that virtually has no size limit and it is mobile based. Telegram is another mobile-based communication application that have similar functionalities with Whatsapp. Both Whatsapp and Telegram allows the creation of group chat to specifically discuss about one topic. However, it is extremely difficult to discuss about an assignment or project as the group chats are not dedicated for such discussions. Problems identified earlier will arise can causes the communication quality to drop immensely.

2.6.2.2 Discord, Slack and Microsoft Teams

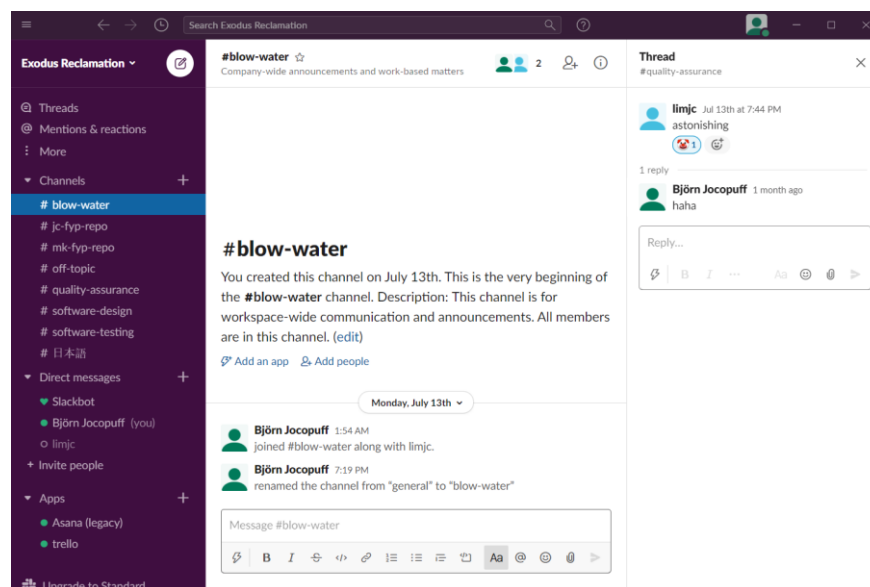


Figure 2.29: Screenshot of Slack

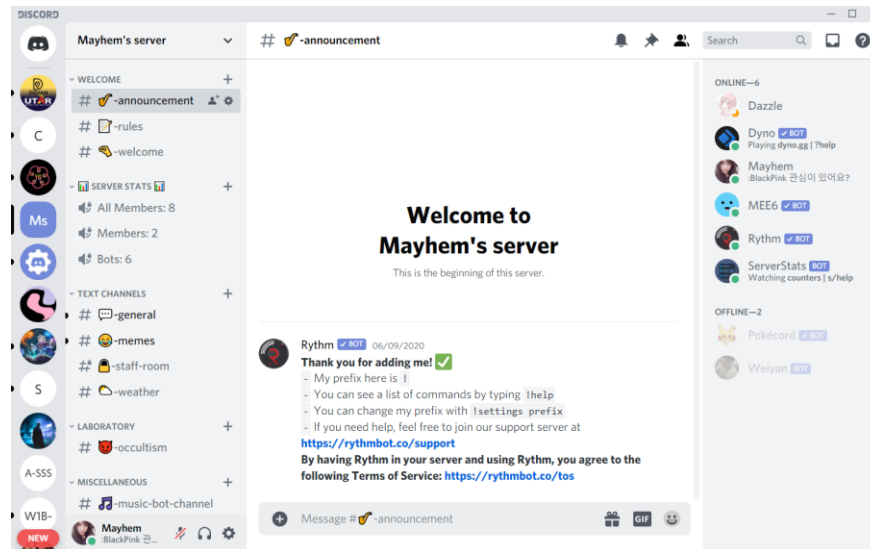


Figure 2.30: Screenshot of Discord

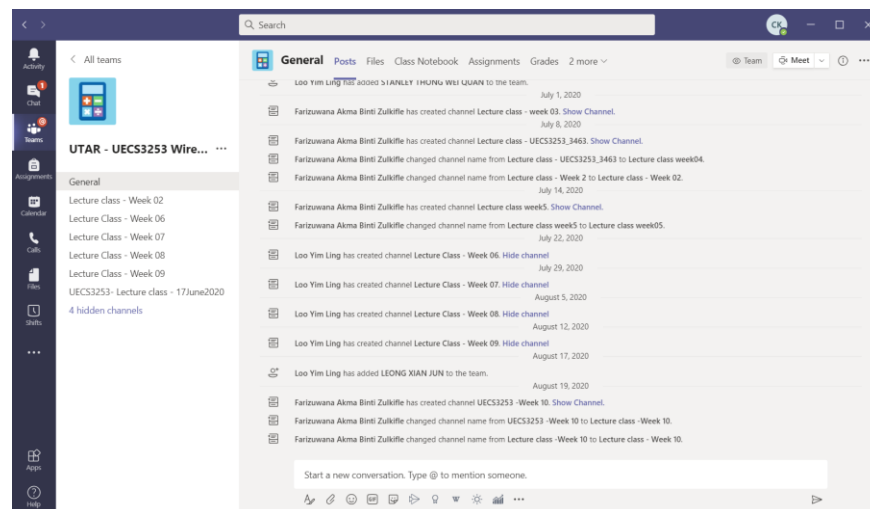


Figure 2.31: Screenshot of Microsoft Teams

Discord, Slack and Microsoft Teams is better in terms of content organisation as it allows multiple text channel for a topic that can discuss more details about them. They may also utilise some text channel to post about announcement or important tips so that they can be viewed by all the team members at all times without worrying the content is overflowed by other chat messages. Discord allows admin to be able to pin important messages as another way of revisiting them too. Slack and Microsoft Teams allows threaded discussion that will further organise the conversation. Another perk is that all of them allows code snippets to be discussed in the chat with rich syntax highlighting that supports a multitude of languages. This way, code snippet discussed will be presented in a more human readable way.

2.6.2.3 Summary and Comparison of Communication Application

Table 2.2: Communication Application and Its Feature

	Whatsapp	Telegram	Discord	Slack	Microsoft Team
File sharing	✓	✓	✓	✓	✓
Threaded chat	✗	✗	✗	✓	✓
Groups	✓	✓	✓	✓	✓
Channels	✗	✗	✓	✓	✓
Direct Message	✓	✓	✓	✓	✓
Video call	✓	✓	✓	✓	✓
Voice call	✓	✓	✓	✓	✓
End-to-end Encryption	✓	✓	✗	✗	✓
Search/Filter	✓	✓	✓	✓	✓
Calendar	✗	✗	✗	✓	✓
Based-on	Phone	Phone	Desktop/We eb	Desktop	Desktop/We b

Direct message is the fundamental if not the core feature of all communication applications out there. Direct message is important is important when the communication is end-to-end only that does not involve any third party along the process. This will enhance the effectiveness of communication. Most group forming communication application such as in the table will still provide a place for users to converse privately. This is an important idea for this project as this project is team collaboration oriented and probably plan to include direct message in the system as well.

All the compared communication application offers file sharing, video call and voice calls that are indispensable for a typical system. It is noted that Whatsapp and Telegram have higher security and Discord and Slack does not, whereas Microsoft Teams have the best of both world for having all of the features in the table that some others do not. However, the UI for Microsoft Teams is mediocre but impressive for Discord that will be referred in this project.

2.6.3 Project Management

Project management is an important topic that not just for programming projects, but all other kinds of projects as well. Every project can be broken down into several smaller parts that can be assigned to different people to perform the task. However, their progress must be made transparent so that the overall progress of the project can be traced and quick decisions can be made in case any accidental occasion happens. Each project must be completed before the stipulated deadline or else it can be detrimental to the project or the company. In the context of students, assignments are also very similar to the real-life projects and project management tools may help them to manage their assignments effectively.

There are abundant of project management tools out in the market that simplifies the workflow of the project and the most notable are:

1. Asana (<https://asana.com>)
2. Trello (<https://trello.com>)
3. Airtable (<https://airtable.com>)

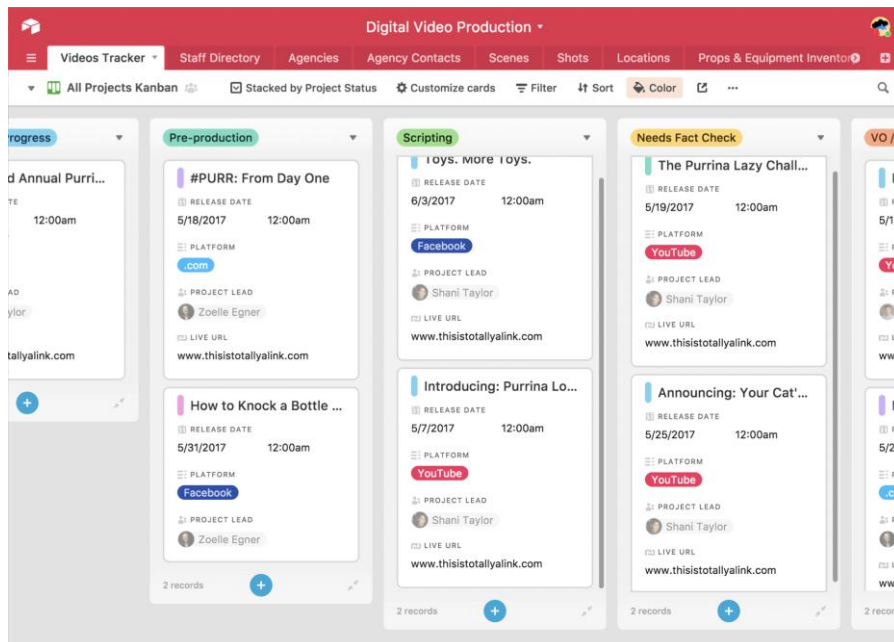


Figure 2.32: Screenshot of Airtable

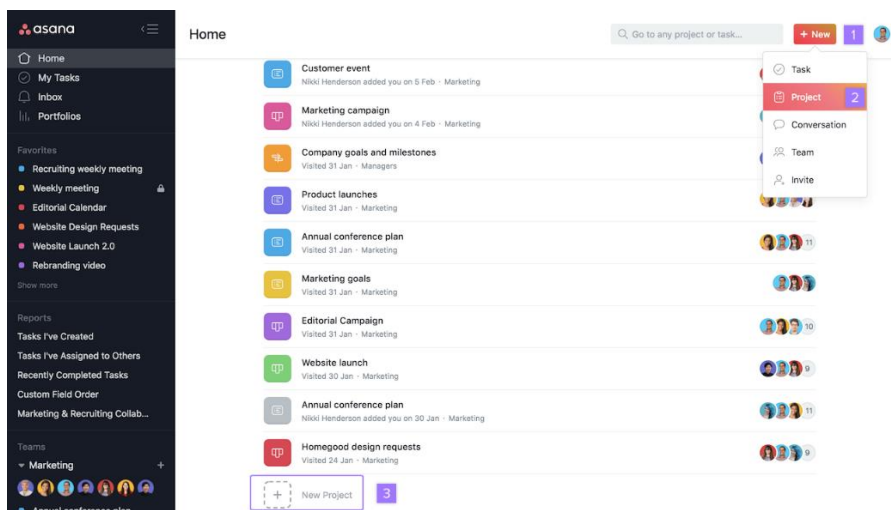


Figure 2.33: Screenshot of Asana



Figure 2.34: Screenshot of Trello

All of them have a work calendar which shows the things needed to be done before a stipulated date. Gantt chart is the core foundation for these systems and the reason that they stand out is their ability to customise, to extend the chart into a more interactive object where multiple users can edit their tasks, comment etc. so that the chart is not just a boring picture on the notebook. Asana provides the ability to switch a to-do list into board, timeline or even calendar. Asana also able to assign a task for a group member to perform.

Trello on the other hand is a card-based project management system which consists of lists that make out of a workflow. It is based on the Japanese Agile method called Kanban. The item in a list is called cards which is an activity that needed to be done at some period of time. Typically, the workflows are categorised

as To-do, Doing and Done. All the activities start out from To-do and are dragged into the categories that their status is as time goes.

2.7 Project Initial Specification from Literature Review

1. Learning through peers in the student context is important. Hence, in an online e-learning platform, chat feature must be available.
2. Collaboration is important, but student do not have time to arrange meeting to discuss at all times, arrange discussion time feature must be available.
3. Quality collaboration enhances effectiveness and efficiency, but most undergraduate students do not know how to use VCS, real-time editing interface must be available.
4. Interaction between instructor and student is important, instructor must intervene students when their problem is recognized to help them from agony. Instructors must be able to access or edit student's work on the platform.
5. Information management is important to convey clear message to others, threaded chat and chat channel must be available similar to Discord's and Microsoft Teams.
6. Proper project management is important to complete a task on time, Gantt Chart, calendar management and Kanban style to-do list that is similar to Asana, Trello and Airtable should be available.

2.8 Conclusion

To conclude, this literature review had a thorough study on 5 different fields that are relevant to the project and the technical findings for implementation has been made. Suitable conclusion or summary has been made. E-Learning being the first field of the study has shown that e-learning is extremely popular already but due to the pandemic e-learning is becoming more demanding, in a way that it becomes the necessity to the community which are instructed to stay home. This has caused the students that takes programming course facing hardship for collaboration in assignment.

A dedicated e-learning platform that supports coding is important to foster close collaboration and deep conversation between students that will contribute to educational goals. The implementation of the project requires proper selection of methodology so that the project will be implemented with minimal hassle. Suitable

features that other application have may be incorporate in the implementation to make the application more useful.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

In this chapter, the software development methodology selected to develop the system, tools used, and project schedule will be explained in detail. Moreover, the work breakdown structure and Gantt chart of the project will also be proposed at the end of this chapter. The system development methodology selected for this project is parallel development and further details will be discussed in the next section.

3.2 Parallel Development

To develop the application of this project, parallel model is selected from the pool of methodologies reviewed for several reasons. The predecessor for parallel model is the waterfall model, which is known to be very strict and rigid in the documentation phase. The methodology selected is suitable because the requirements for this project is well understood and are not likely to change in the foreseeable future.

The time allocated for implementing this project is relatively limited, for around 14 weeks, hence it is very important to ensure that the application will be delivered in a shorter time while not losing the quality that it supposed to have and parallel model solves the issues of long delay of its ancestry model, waterfall model whilst retaining the quality that the waterfall model provides.

Parallel development is essentially breaking up the bigger projects into smaller projects so that the milestones are more achievable and allowing them to be developed simultaneously, without waiting for the full delays of transitioning between the tasks.

By using parallel development in a solo project will also means that I will have to implement all the subsystems by myself. However, the critical functionalities of the system will be granted priority first before doing other miscellaneous functions. This is to ensure that the system will still be able to deliver important functionalities in case the time allotted was unfulfillable. Figure 3.1 illustrates the development cycle of Parallel development methodology in the e-Learning system.

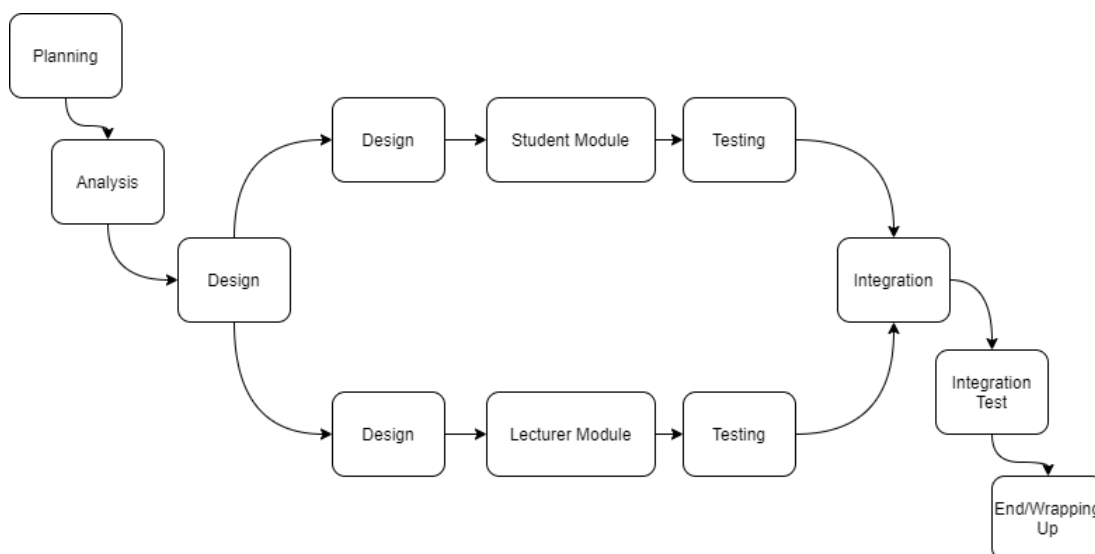


Figure 3.1: Lifecycle for e-Learning System

3.3 Planning

Literature review was first conducted to study the background of e-Learning, e-Learning platforms and the current situation faced by Malaysian tertiary education students. The need for a dedicated e-learning platform was then justified, especially for programming course students.

To begin the planning phase of the project, it requires a thorough understanding of the problems faced by students and lecturers with regards to programming assignments issues in this trying period. After an in-depth study on the situation that they are facing right now, the main issues discovered are listed below:

1. Students are unable to communicate effectively when working together as a team.
2. Students are having a hard time managing groups and assignments.
3. Lecturers are unable to keep track of the progress of assignments for each group of students.

The issues discovered acts as the foundation for the objective formulation of this project. The determined objectives serve as the direction of research and requirement elicitation for this project as well. The objectives formulated are listed as follows:

Main Objective

To promote e-Learning, communication and collaboration for students that are taking computer programming courses.

Sub Objectives

- To develop an online platform that provides exceptional communicative power to students so that collaborative e-learning can be fulfilled.
- To provide a conducive environment for students on coding-related group assignments and streamline the workflow.
- To facilitate the interaction between student and lecturer by improving the transparency of students' progress.

After discovering the problems and its associated objectives, the project proceeds on examining the most appropriate solution as well as methodology for this project. The selected methodology for this project is the parallel model. The proposed solution and proposed approach as well as the project scopes were investigated thoroughly to avoid erroneous decision that will result in undesirable consequences such as extension of project lifecycle. The adoption of software development methodology within this project instead of using the ad hoc approach because it will structure and organise the workflow of the project so that each and every deliverables and time restrictions can be made clear and adhere to the plan for timely delivery of the application.

The proposed solution on the other hand outlines the system architecture and tools that is needed to develop the program and provide visual comprehension of how the overall development is going to be. The project scopes had defined the necessary features that must be fulfilled in the final deliverables for the project.

3.3.1 Requirement Gathering and Elicitation

In the planning phase, requirements gathering and elicitation is an important activity that will has a huge impact on the project. The studies were conducted by literature reviewing, interviewing, distributing questionnaire paper for students from different universities in Malaysia and reviewing the applications with similar and or desired features that can be implemented along with it. A complete list of features was finalized based on the results obtained from the study.

3.3.1.1 Literature Review

When conducting the literature review for the background, the importance of e-learning, collaboration and communication between students and lecturers was first identified. However, there are several issues found that hinders collaboration

between students that ultimately causes poor assignment quality. This serves as the scope definition for this project to develop an e-learning platform that will solve most of the issues identified.

Literature review of the related work had also helped on gathering some essential requirement that should be offered by a collaborative editing platform that will be integrated into the e-learning platform as well.

3.3.1.2 Reviewing Similar Applications

There are 20 total reviewed applications that are having the similar or desired are grouped into three main categories as mentioned earlier. The full summary of comparison for each category can be found in the previous chapter. The reviewed applications are based on real-time collaboration, communication and project management that will foster close collaboration between students and deliver quality assignment on time.

3.3.1.3 Questionnaire

This questionnaire is short, but it was prepared by bearing in mind the simplicity and effectiveness of the question asked so that quality insights can be extracted from the respondents without letting them feel burdened along the process of answering.

The purpose for this questionnaire is to analyse e-Learning platforms used in different universities or higher education institute in Malaysia so that the need for a dedicated e-learning platform could be justified as well as study the usefulness of the features provided.

Instead of asking directly what are the features that they would like to have on the e-learning platform, the students were asked on what are the functions that they actually use based on what their university's e-Learning platform provide. This is because some of the students might not know what they want from this platform and they are indifferent to it and thus they will just simply answer the question. Also, their actual usage of the feature is sufficient to provide useful information on which feature can be included in this project and which less recommendable.

The full analysis and conclusion of the questionnaire can be found on Chapter 4 where the question is discussed one by one.

3.3.1.4 Interview

Since this is an e-Learning platform for university, an interview was conducted to gather the requirements from lecturer's perspective. This will ensure that the e-Learning platform developed will have full-coverage that will benefit both students and lecturers towards the educational goals.

3.3.2 Project Scheduling

After much research were conducted through the distribution of questionnaire and reviewing the literature, the scope of the project is also formulated and followed by drafting a detailed work breakdown structure.

After the WBS has been successfully drafted, numerous Gantt Charts were constructed for the whole software development lifecycle. The Gantt Charts will be crucial to visualise the timeline of the whole project that provides a goal or milestone that needed to be achieved within the stipulated time. This is to provide a guideline for the project development process and keep track of the overall lifespan of the project to keep away from unplanned extension.

3.4 Analysis and Design

By using the requirements gathered and the project scope, analysis and design was done to provide a glimpse of how the workflow and the architecture for the project is going to look like. Use case diagrams were constructed and the use case descriptions were available as well to provide detailed information on each of the designated use case.

Other than that, prototypes for different components of the project were constructed to provide a tangible representation of the full product to the stakeholders involved so that critique, recommendation and verification can be obtained as constructive feedback to improve the system implementation. A combination of high-fidelity prototype and low-fidelity prototype were produced as some of the interactive functionalities will only be available at the end-product as the implementation for the prototype is complicated.

The low-fidelity prototype is the wireframe for the different users' view (students and lecturers) and interfaces that shows each and every possible combination for the activities that they can perform. This prototype will be analysed

in terms of user experience as well. The high-fidelity prototype will also be produced if time allow.

For the design for data model diagrams data flow diagrams and class diagram, as well as other important design documents that were not yet identified, will not be available at the initial phase for this stage and will be produced around the end of December this year to mid of January next year. A more detailed, robust and complete design documentation will be expected by then.

3.5 Implementation and Testing

The implementation for the project will be adapting the parallel pattern in the workflow that splits the projects into distinct sub-projects. All the subprojects will commence at the same time. After setting up the environments required for front-end application and the server, Firebase setup will follow next.

The first module that will be developed is students' module. It is the part of the system that is only intended for student use only. The feature workspace and chat messages will be experimented and developed along with the Socket.io library for real-time data transfer across clients.

The second module that will be implemented is the lecturers' module. It is the part of system that intends to let lecturer have a better management power on the assignments and groups under the programming course that they are conducting. Students does not have access to such interfaces. The lecturers' module will be connecting to part of the students' module on the workspace part so that lecturer can view the group's progress from time to time or even join their discussion.

The functions that do not related directly to both of the modules will be implemented from time to time as well. Feature such as login function will be the first to build, followed by quick links, fallback pages and open and close group function will be implemented subsequently.

After all the modules are done, unit testing will be performed first to check if any of the components have anomaly behaviours that could possibly cause havoc to the final system if neglected. After all the unit testing passed, the system will be integrated into the final product followed by an integration testing to ensure all the subsystems are working together in harmony.

3.6 Post-implementation Phase

Before delivering the system, volunteer was gathered to participate in the user acceptance test for the project to verify the features offered are deemed helpful to them. The full documentation of the project was completed as well, starting from the proposal to the actual functioning system.

Besides, the presentation slide was prepared to explain the development process of the entire project. This slide included the content from the planning stage of this project until the project closure. At the same time, a poster was created to provide a brief insight into the entire project.

3.7 Project Tools

3.7.1 System Architecture

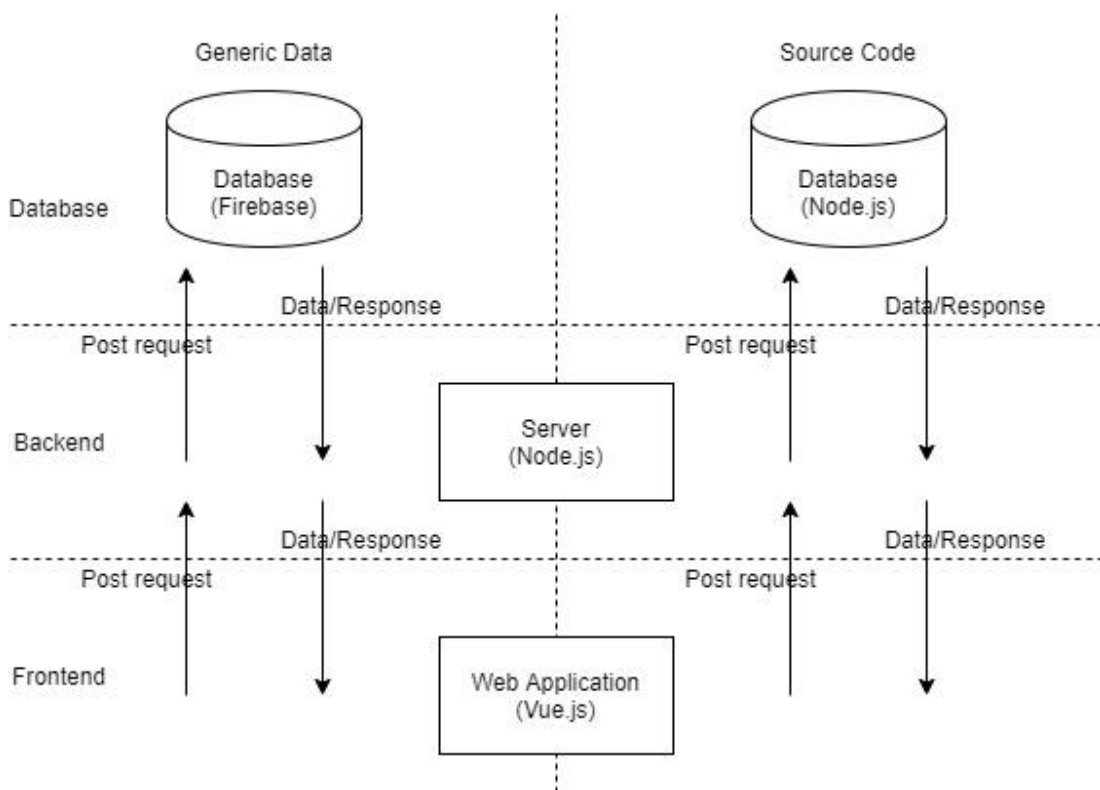


Figure 3.2: Proposed system architecture

This project was implemented with the 3-tier architecture that most of the generic client-server web applications uses. 3-tier architecture is beneficial to the production and development environments as it modularizes the user interface, business logic and data storage layers to produce a more flexible environment for maintenance. The

application also will be more secured as the database is separated by the server from the user interface to prevent user to directly tampering with it.

The application tier is the front-end layer that displays the user interface as well as its data that made up the interactivity of the whole application. The backend later is used to fetch the data from the database for processing. After processing with the logic embedded, the data will be forwarded to the application layer to display it to the corresponding user.

There are two databases that reside on the database layer, the first one being a NoSQL database that stores data such as user information etc. and the other database which is essentially the Node.js server itself but will also use to store source codes and project folders.

3.7.2 Firebase

Firebase is an online service featuring NoSQL by Google that can be used in both mobile and web applications. Firebase includes a number of built-in features that is grouped into 3 categories – building better apps, improve app quality and grow the business. Since the database is using a NoSQL pattern of storing the information, it allows flexible datatypes that does not limited to a blueprint that was predetermined but allows the data to polymorphs to whatever suites it.

Firebase also provide analytics that can used to analyse the user behaviour of the application that will provide useful insights to the developer to improve the application. Besides that, Firebase also provides services such as cloud messaging, crash reporting and Android test lab to the developer, bearing in mind that some of them are paid features (Khawas and Shah, 2018).

3.7.3 Node.js

Node.js (OpenJS Foundation, 2009) is the runtime instance of JavaScript that can be used to run as a server for web application instead of using servers with PHP script such as Wamp server or Xampp. Node.js offers many third-party packages that have specific function or purpose that is written by the community. The V8 Chrome Engine that used to run client-side JavaScript code is used in Node.js as well that compiles into fast machine codes. Node.js is also recognized by its non-blocking I/O feature. This property of Node.js obliterate the need for multi-threading since the server can handle multiple requests at the same time (Patel, 2018). All and all,

Node.js is very memory efficient as it runs single-threaded, non-blocking asynchronous programming.

3.7.4 Vue.js

Vue.js is an open-source progressive framework that builds user interface. It is extremely versatile as it is an incrementally adoptable ecosystem that scales between a library and a full-featured framework. It is also very performant and lightweight as the framework can even be used immediately after appending the source script tag in the header of the HTML document on the go unlike React and Angular which usually runs optimally only with Node.js server.

3.7.5 Vuetify

Vuetify is the Material UI framework built specifically for Vue.js applications. Vuetify offers a lot of useful utilities such as screen break points, pre-written CSS classes similar to Bootstrap UI Framework, material icons, and many useful on-the-go components such as button and text field that uses material design principle for their appearance. It is always plus for installing Vuetify into any Vue.js projects as some tedious tasks such as writing a popup can be avoided as Vuetify enable the use of popup with just a single line of code.

3.7.6 TypeScript

TypeScript is the superset of JavaScript. Any valid JavaScript is valid TypeScript. TypeScript extends JavaScript by adding type into the language as JavaScript is a runtime language that determines the type of the variables when it is executing. JavaScript also does not have the concept of Classes that are commonly used in any other Object-Oriented Languages such as C# and Java. TypeScript adds in the class and type concept into the language so that the type-safety feature can also help to debug at compile time instead of crashing incessantly at runtime. TypeScript codes is transpiled into native JavaScript code by dedicated Transpilers.

3.7.7 Git and GitHub

Git is a Version Control System that keep tracks of the changes made on the source codes over time and allows possible revert to the previous state if the latest version has turned buggy and erroneous. GitHub on the other hand is a file repository hosting

platform that is well integrated with Git. All the commits from Git on the local machine can be pushed into GitHub just by typing commands. Git and GitHub will be used in this project and the final repository links for this project will be made accessible at the end of this document.

3.7.8 Figma

Figma is a vector graphics editor and prototyping software that is used in designing the user interfaces and interaction for webpages and applications. It is a popular tool used by designers across the world. It is totally free for user that does not use its real-time collaborative feature, so it is ideal for this project as it is a solo project. It offers a wide variety of prototyping necessities such as drag and drop interface, resizing etc. that enables user to create a prototype that is highly resilient and resemble the final product, all without a single line of code. Figma will be used to create wireframe prototype as well as screens mock-up for this project.

3.7.9 Jest

Jest is a JavaScript unit testing framework that is dedicated by a team of open-source contributors, while being maintained by Facebook. With that in mind, it is pretty popular among JavaScript developers around the world as it officially supported by one of the famous and infamous tech-giants, as their quality of work is being assured.

The clean and simple syntax as well as extremely useful features ranging from assertion, mocks, stubs, async support, spy and plethora of matchers is loved by the community. Hence, supports for Babel, TypeScript, React, Vue, Svelte, Node.js and Angular had been made publicly available.

3.8 Work Plan

3.8.1 Work Breakdown Structure

1.0 Planning

1.1 Study Background of the Problem

1.2 Determine Problem Statements

1.3 Define Objectives

1.4 Propose Project Solution

1.4.1 Design System Architecture

1.5 Propose Project Approach

1.5.1 Research Available Approach

1.5.2 Choose Development Approach

1.6 Define Scope

1.6.1 Identify Target Users

1.6.2 Identify Covered Features

1.7 Literature Review

1.7.1 Further Studies on Background

1.7.1.1 What is e-Learning

1.7.1.2 Necessity of e-Learning

1.7.2 Extract Insights from Related Works

1.7.3 Determine Best Methodology

1.8 Requirement Gathering and Elicitation

1.8.1 Review Related Work

1.8.2 Review Similar Systems

1.8.2.1 Review Web-based Text Editor

1.8.2.2 Review Communication Application

1.8.2.3 Review Project Management Application

1.8.3 Distributing Questionnaire

1.8.3.1 Prepare Questionnaire

1.8.3.2 Distributing Questionnaire

1.8.3.3 Evaluating and Analysis the Result

1.8.4 Conducting Interview

1.8.4.1 Prepare Interview Questions

1.8.4.2 Schedule for Interview

1.8.4.3 Conducting Interview Session

1.9 Project Scheduling

1.9.1 Construct WBS

1.9.2 Construct Gantt Chart

2.0 Analysis and Design

2.1 Design Use Case Diagram

2.2 Write Use Case Description

2.3 Design Entity Relationship Diagram

2.4 Produce Data Dictionary

2.5 Design Data Flow Diagram

2.6 Design Prototype/Wireframe

2.7 Page Navigation Flow

2.7.1 Construct Student Page Navigation Flow

2.7.2 Construct Lecturer Page Navigation Flow

3.0 Implementation and Testing

3.1 Implementation of Student Module

3.1.1 Build Necessary Pages

3.1.2 Create Group Management Feature

3.1.2.1 Add Join Group Function

3.1.2.2 Add Create Group Function

3.1.2.3 Add Matchmake Group Function

3.1.3 Build Group Workspace

3.1.4 Implement Gantt Charts

3.1.5 Build Discussion Algorithm

3.2 Implementation of Lecturer Module

3.2.1 Implement Necessary Pages

3.2.2 Implement Assignment Management Function

3.2.2.1 Build Add Assignment Function

3.2.2.2 Enable Assignment Editing

3.2.2.3 Build Upload Function

3.2.3 Connect Workspace

3.3 Implementation of Other Functions

3.3.1 Construct Login

3.3.2 Add Fallback Pages

3.3.3 Add Quick Links Function

3.3.4 Add Open/Close Group

3.4 Unit Testing

3.4.1 Conduct Automated Test

3.4.1.1 Test User Interface

3.4.1.2 Test Utilities Functions

3.4.2 Conduct Manual Test

3.4.2.1 Feature Test

3.4.2.2 API Test

3.5 Integration of Subsystems

3.6 System testing

3.7 Conduct User Acceptance Test

3.7.1 Prepare UAT Test Cases

3.7.2 Gather UAT Volunteers

3.7.3 Perform UAT

4.0 Post-implementation Phase

4.1 Finalize the Documentation for the Project

4.2 Prepare a Presentation Slide

4.3 Prepare a Poster for the System

3.8.2 Gantt Chart

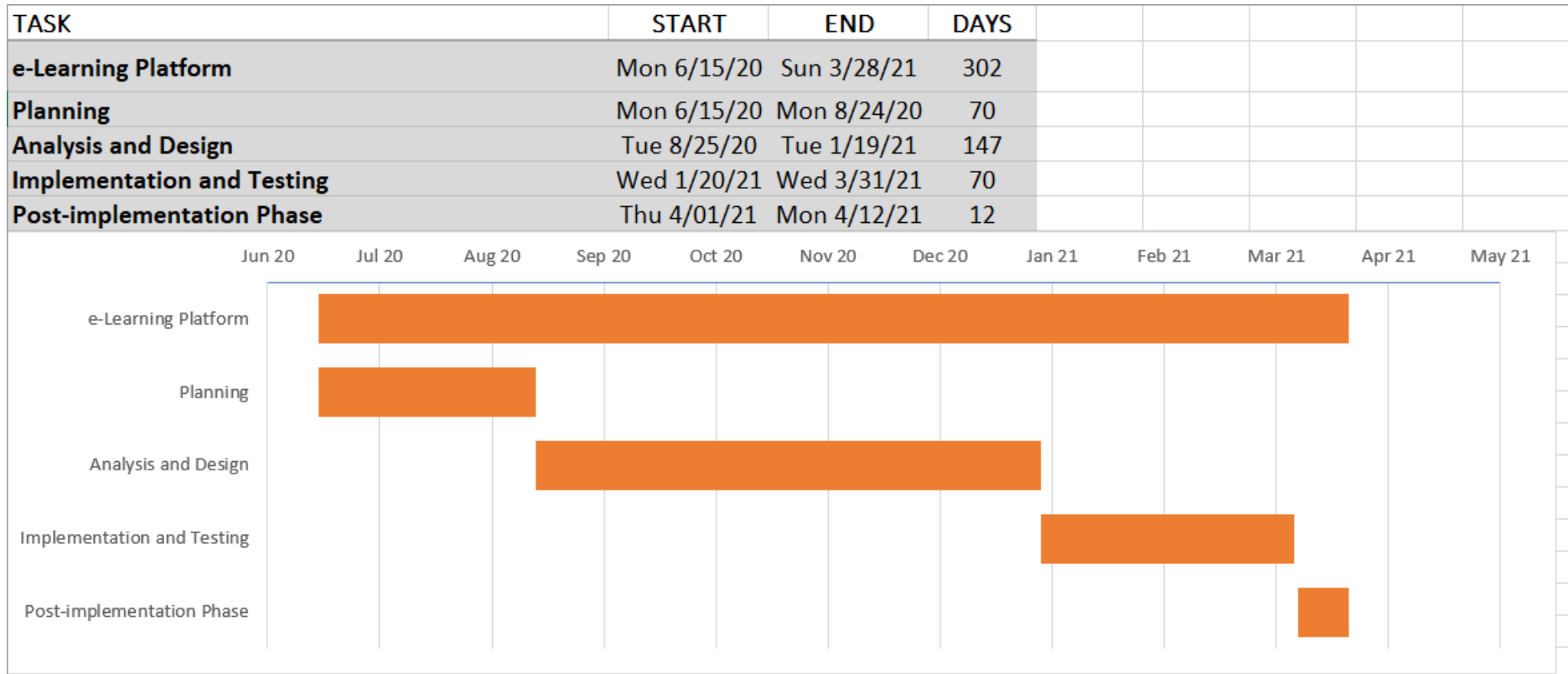


Figure 3.3: Gantt Chart for Overall Project

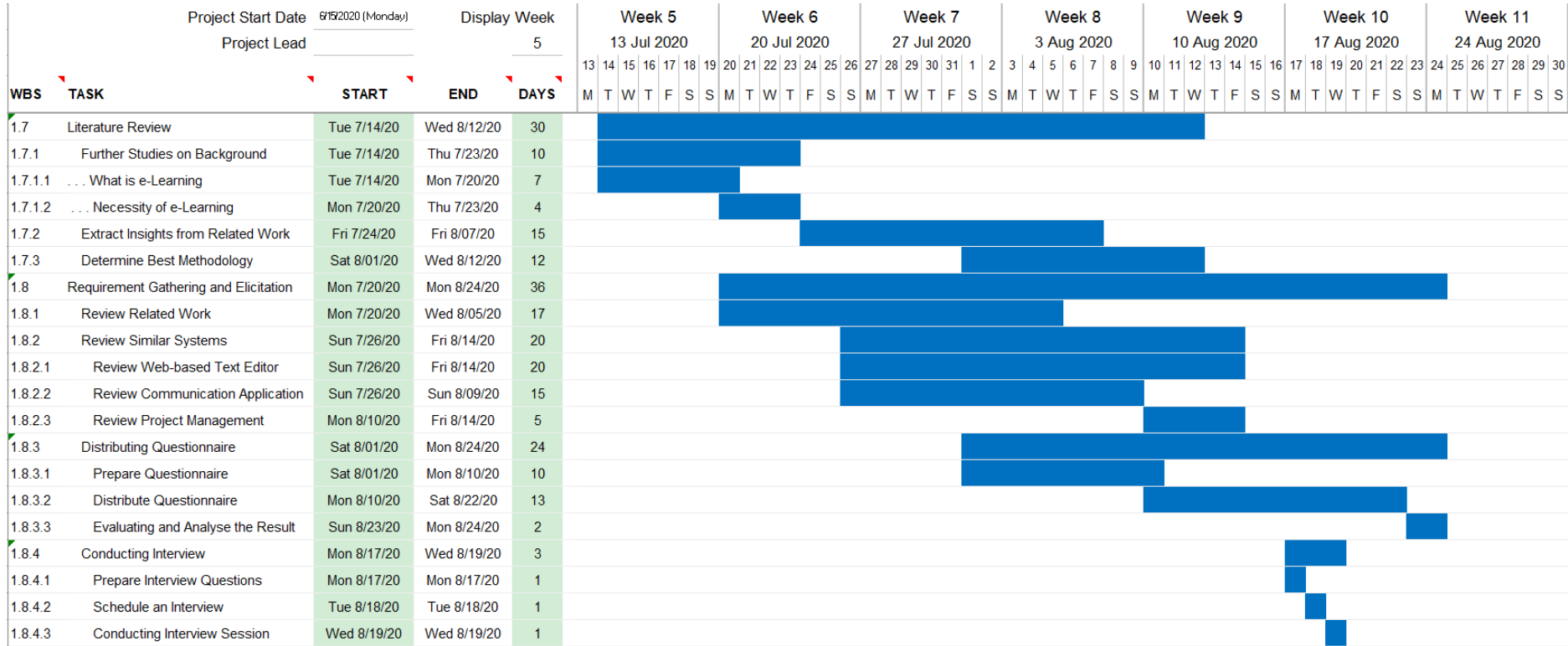


Figure 3.5: Gantt Chart for 1.7 to 1.8.4.3

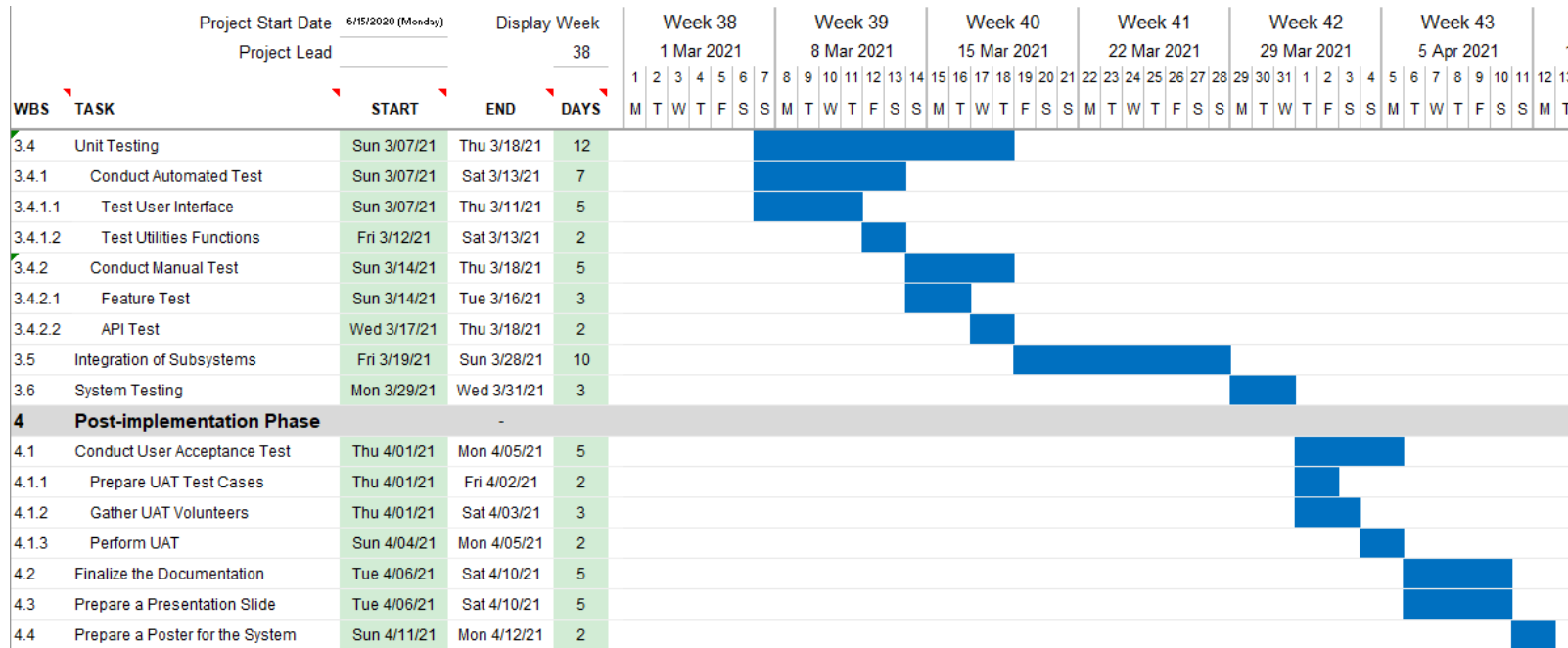


Figure 3.9: Gantt Chart for 3.4 to 4.4

3.9 Summary

This project to develop an e-Learning platform to enhance collaboration between students of programming subjects will be done in 4 phase which is planning, analysis and design, implementation and testing as well as post-implementation phase that will require 302 days including a hiatus in the middle of design phase due to the internship.

CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction

This chapter will outline all the proposed and gathered requirement as well as questionnaire analysis to make proper specification. With the initial specification, use case diagrams, wireframes and initial system designs were constructed to visualize the designs and functionalities of the system.

4.2 Questionnaire Evaluation

The incentive of the questionnaire is to study the different e-Learning platform that Malaysian Universities are using and make an analysis on the features that they provide. The analysis goes by judging the e-Learning platforms based on the presence of the critical features that have identified previously in the report that will boost the collaboration and quality of group assignments. From the questionnaire, 16 universities were identified, namely:

1. Universiti Tunku Abdul Rahman, UTAR (<http://www.utar.edu.my>)
2. UCSI University (<https://www.ucsiuniversity.edu.my/>)
3. Universiti Malaya, UM (<https://www.um.edu.my>)
4. Tunku Abdul Rahman University College, TARUC (<https://www.tarc.edu.my/>)
5. Wawasan Open University, WOU (<http://www.wou.edu.my/>)
6. Multimedia University, MMU (<https://www.mmu.edu.my/>)
7. Universiti Teknikal Meleka, UTeM (<https://www.utm.edu.my/>)
8. Universiti Sains Malaysia, USM (<https://www.usm.my/>)
9. National University of Malaysia, UKM (<https://www.ukm.my/portal/>)
10. Universiti Tun Hussein Onn, UTHM (<https://www.uthm.edu.my/en/>)
11. Universiti Pendidikan Sultan Idris, UPSI (<https://www.upsi.edu.my/>)
12. Universiti Malaysia Terengganu, UMT (<https://www.umat.edu.my/>)
13. Universiti Malaysia Pahang, UMP (<http://www.ump.edu.my/>)
14. Universiti Teknologi Malaysia, UTM (<https://www.utm.my/>)
15. Universiti Utara Malaysia, UUM (<http://www.uum.edu.my/>)
16. SEGI University (<https://www.segi.edu.my/>)

- 17. IACT College (<https://iact.edu.my/>)
- 18. Sunway University (<https://university.sunway.edu.my/>)
- 19. *University of Melbourne (<https://www.unimelb.edu.au/>)

The full comparison chart for the e-learning systems for each of the identified university can be found on Appendix A.

4.2.1 Demographics

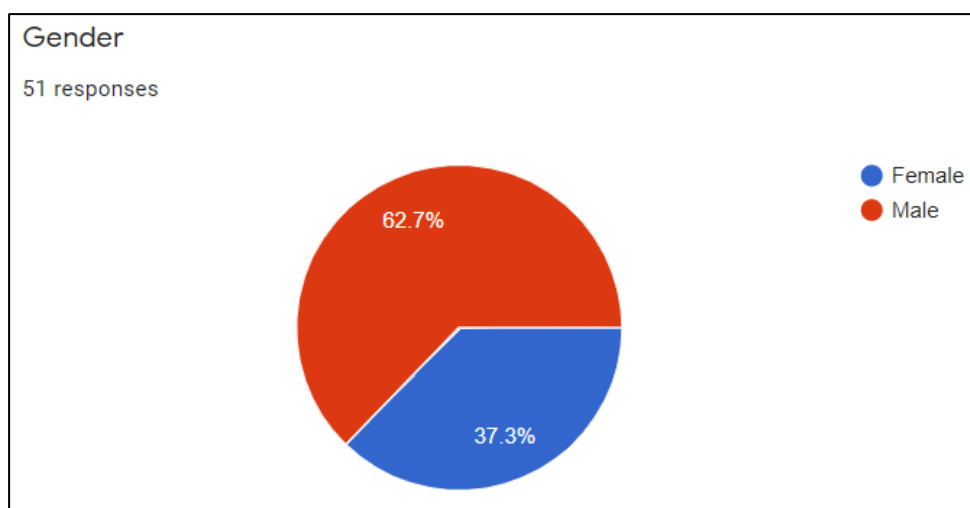


Figure 4.1: Gender distribution of respondents

The questionnaire managed to obtain 51 responses from tertiary education students across Malaysia and 32 (62.7%) of them are male whereas 19 of them are female (37.3%) as shown on Figure 4.1.

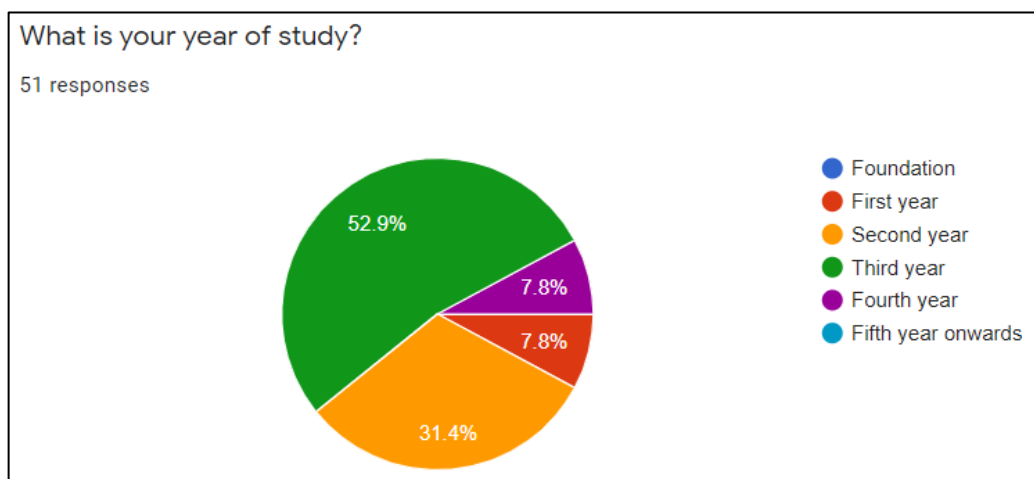


Figure 4.2: Distribution of year of study

Most of the respondents (52.9%) are third year students, followed by 16 (31.4%) second year students, 4 (7.8%) fourth year students and 4 (7.8%) student are studying at first year as shown on Figure 4.2.

4.2.2 Features/Functions Studied

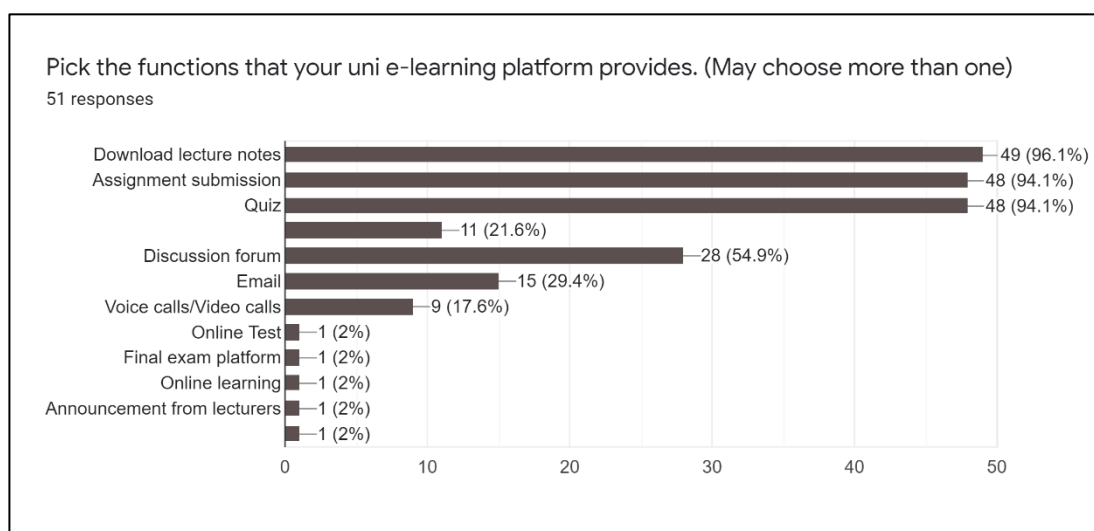


Figure 4.3: E-Learning platform features

Based on the bar chart above, most of the respondents reported that downloading lecture notes, assignment submission and quiz are the features that their e-learning platform provides, towering at 49 (96.1%), 48 (94.1%) and 48 (94.1%) respectively. Almost half of the respondents (54.9%) report that their universities' e-learning platform are having discussion forums for students to ask questions or discuss among each other.

Next, 15 of them (29.4%) reports email being the innate feature for the platform that they are using. However, only 11 of them (21.6%) reports that their e-learning platform are having instant text messaging features and 9 of them (17.6%) reports of having voice or video call functionality. This shows that the important features that will enhance collaboration among students as identified have limited presence in the e-learning platforms. Other reports of their e-learning platform are also used as a final exam platform or online test as well as receiving announcement from lecturers.

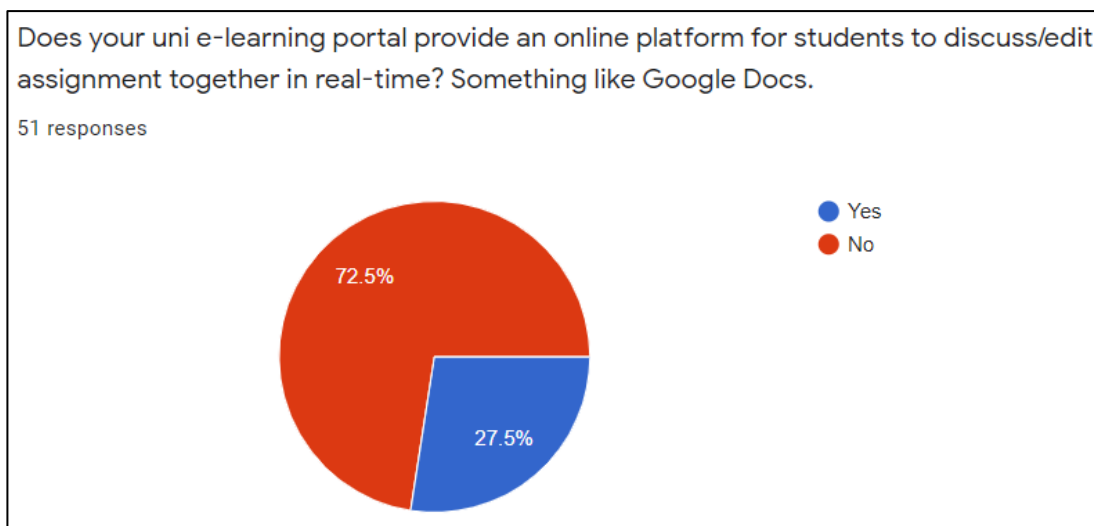


Figure 4.4: Real-time assignment platform

Based on the response above, 37 respondents (72.5%) disagree that the e-learning platform they use have the feature that allow students to collaborate on assignments with real-time editing and 14 of them are positive for that. However, this information may be subjected to bias and misinterpretation of the question itself. So, the university was taken instead of the individual students to analyse this problem. Out of 18 Malaysian universities, only 6 of them (33.3%) offer this feature in their e-Learning platform. This shows that most students will still need to rely on external communication application or do the assignments in a traditional way. Either way will affect the student as well as the assignment quality as collaboration between students are progressively disrupted.

4.2.3 Usability Investigation

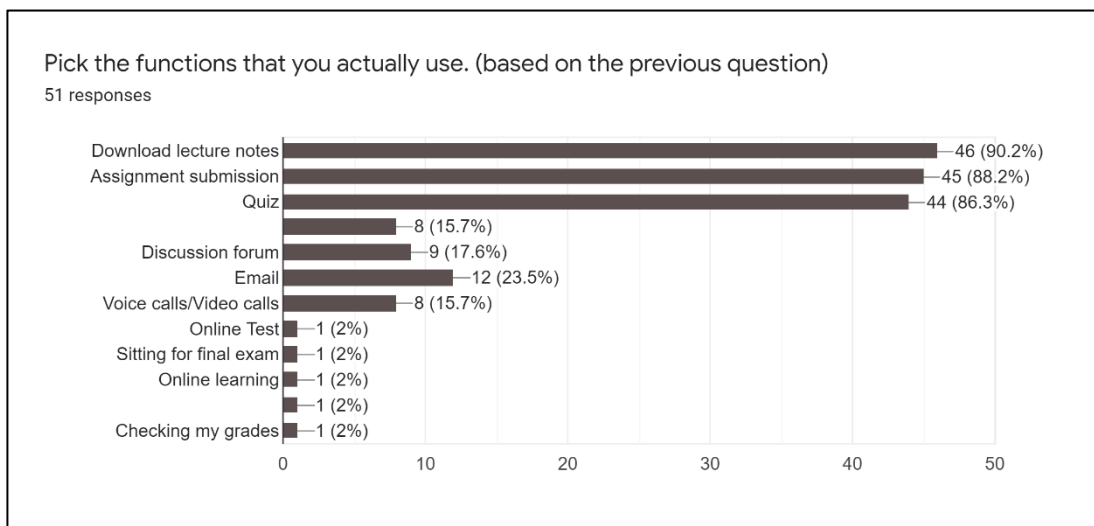


Figure 4.5: Functions that Respondents Actually Use

This question is asking respondents which functionalities offered by their university's e-learning platform as identified by them in the previous question (Figure) that they actually utilise it within their academic journey in their respective university.

As expected, download lecture notes, assignment submission and trying quiz is the feature that mostly used by the students in the e-learning platform where the usage are 46/49 (93.9%), 45/48 (93.8%) and 44/48 (91.7%) respectively. For instant text messaging feature, 8 out of 11 (72.7%) of the reported student actually use the function. However, for discussion form, only 9 out of 28 students (32.1%) actually use it. This may be due to the slow delivery of information and inconvenient to use. For the email feature, 12 out of 15 respondents (80.0%) reports of using it whereas 8 out of 9 students (88.9%) uses the voice or video call function.

The high percentage usage of instant text messaging, email and voice or video call shows that the demand for fast information transmission technology in the university context but unfortunately, only a few universities offered such useful features.

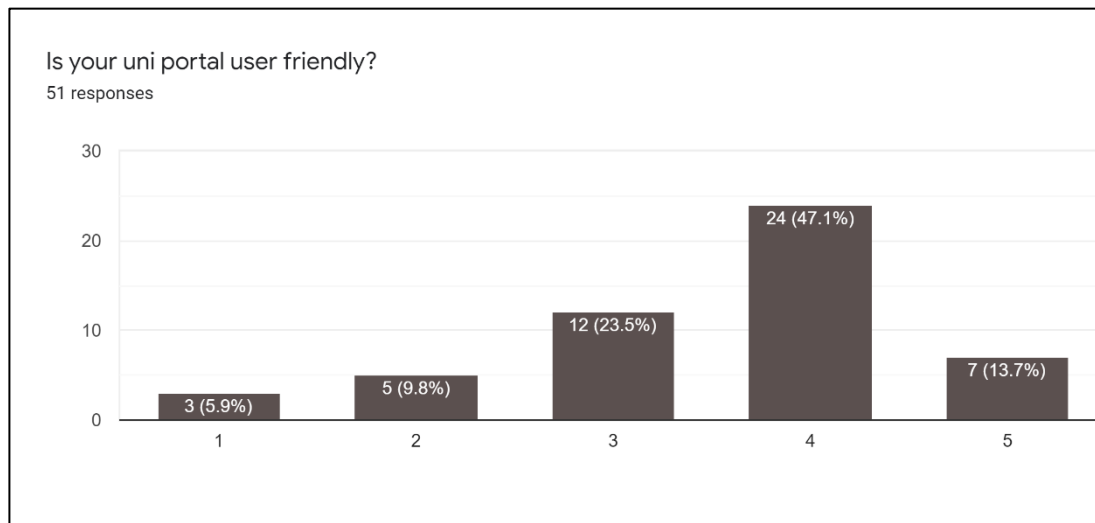


Figure 4.6: User-friendly Rating

The figure above shows the rating of user friendliness of the e-learning platform that each of the student gave for their university and it is found that most of the students (for rating 4 and 5) feels that their e-learning platform are easy to navigate, intuitive and non-confusing.

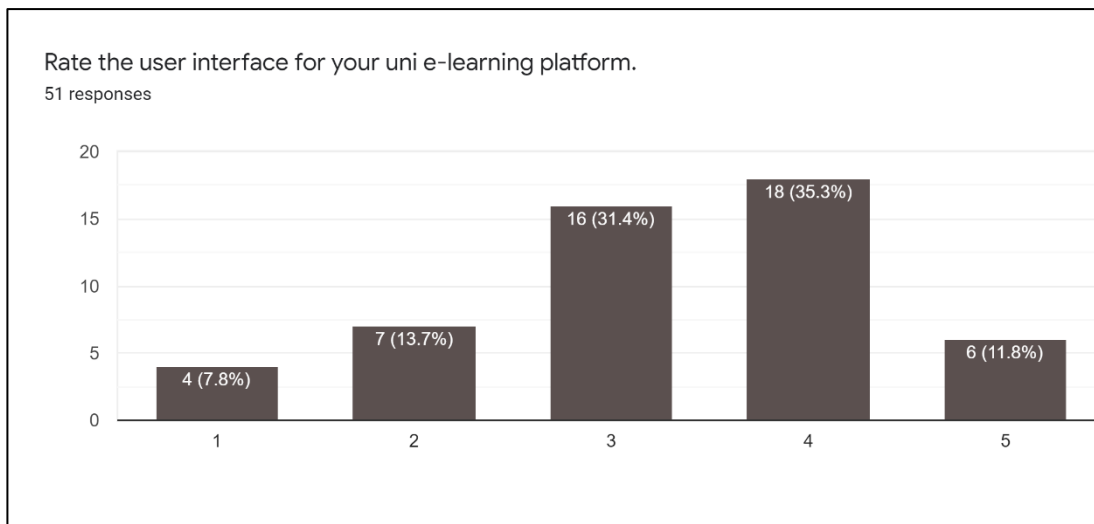


Figure 4.7: Aesthetic Rating

Compared to the rating of user-friendly, the chart for aesthetic rating is more equally distributed as the chart for rating can be seen obviously skewed to the left.

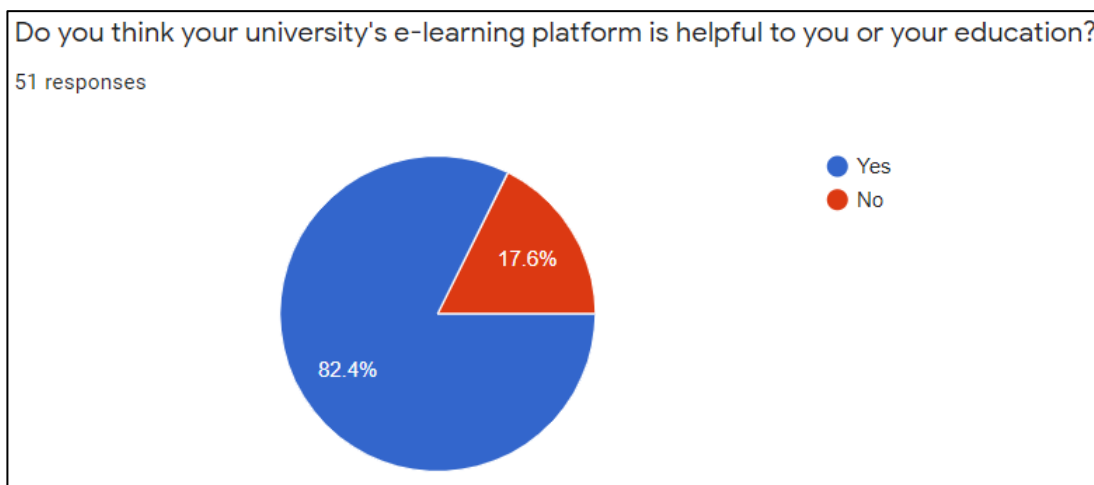


Figure 4.8: Opinion on Helpfulness of University’s e-Learning Platform

Most of the respondents (82.4%) think that the e-learning platform has helped on improving their educational skills. This implies that although the interface of the university e-learning platforms are not that impressive, as long as they are user-friendly, that is simple and not complicated, they will be helpful to the students in their academic.

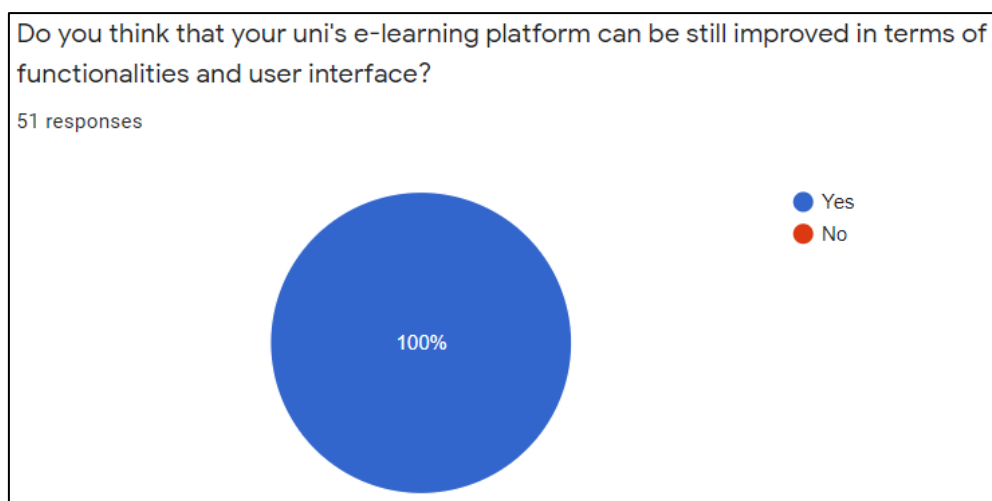


Figure 4.9: Opinion on Room of Improvement

The above figure shows that all of the respondents feel that their university's e-learning platform still have room for improvement for more advanced functionalities that may aids them in their education as well as the user interface to become more visually appealing. If this insight is adapted by the universities, the usage of their e-Learning platform will increase exponentially.

4.2.4 Opinion on E-Learning

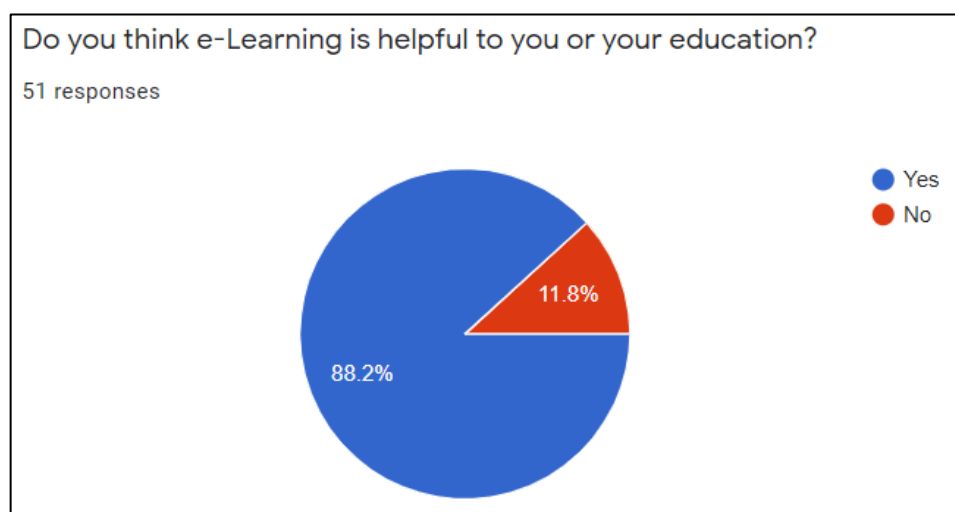


Figure 4.10: Opinion on Helpfulness of E-Learning

Another question shows that 88.2% of the respondents found that e-Learning itself such as learning through the internet, multimedia infographics and videos are helpful to their education as a whole as the e-Learning is a broad and flexible domain

that are not bounded by just a platform that might have some important features absent.

4.2.5 Findings of Questionnaire

From the questionnaire, it was found that most of the important collaborative features identified in Chapter 2 are absent from the existing e-Learning platform. This shows the limited capability of the existing platform to enhance the experience of working together as a team to meet the assignments' goal.

For the requirements, assignments submission and instant text messaging features will be implemented for this project because of the high usage percentage as exhibited by the respondents. Attempts for voice or video calls will be done as well. However, discussion forum will not be implemented for this project as the usage among students are extremely trivial where only a third of the respondents utilise the function.

Although, downloading lecture notes, quiz and email are very popular among the students as well, they will not be implemented because they are out of the scope and are not appropriate for this project.

4.3 Interview Findings

From the short interview with Dr. Too Chian Wen, an academic personnel in UTAR that specialises in teaching Object-Oriented Programming Course, it is found that it is difficult for the instructors to keep track on all the assignment progress of each team unless the team directly seek help or report to the instructor. It will be delightful and ideal for them to be able to get access to any of the student's assignment progress at any time to evaluate their performance and identify if there is any problem faced by them so that timely intervention as mentioned in the previous chapter could be done to help the students.

Students will sometimes submit the assignments late because they need to organise the files and zip it accordingly, which takes quite some time. It will be more helpful for the assignments to submit directly in the platform without the need to configure anything manually.

Besides that, she mentioned that there are many students are unable to find assignment groups when in the physical class as some students are not very good at socializing. Some of them does not even try to find a group even though the assignment due date is just around the corner, which is quite frustrating. To further

aggravate things, the MCO due to the pandemic are inhibiting students to find assignment teammates face to face which results in increment of ‘orphans’ in the class where they do not have a group and so it would be helpful if a group matchmaking system that aids on the process of group formation.

It will be nice if the instructor can get a bird’s eye view on the overall performance of the students in a visualisation chart dashboard as they would need to use excel in type in the details for each group one by one, which is quite a mundane task to do.

4.4 Requirement Specification

The requirements specification section discusses the functional requirements and the non-functional requirement of the system for both the server and the front-end of the application. The non-functional requirements of this system include availability requirements, development requirements, operational requirements, performance requirements, security requirements and usability requirements.

4.4.1 Functional Requirement

Functional requirement outlines the service that a software must offer. The functional requirement of the project is separated into two parts, which is functional requirement for front-end interface and for back-end server respectively. The functional requirement was identified through literature review, reviewing similar systems, questionnaire and interview.

4.4.1.1 Backend Server

1. The backend server must be able to store User Details, Assignment Details, Group Details and their dependencies into the database
2. The backend server must be able to store and update the source codes using file I/O.
3. The backend server must be able to allocate sessions to users, students and lecturer as well that are on the same project workspace.
4. The backend server must be able to calculate the discussions session based on the student’s booking input within a group and announce when it is confirmed.

5. The backend server must be able to handle group matchmaking request sent by students who does not have a group but wish the systems to matchmake for them.
6. The backend server must notify users on the happenings based on their roles:
 - a. Student:
 - i. When lecturer upload assignment question file.
 - ii. When group leader submitted the assignment.
 - iii. When group is joined by new member.
 - iv. When matchmaking request is realised.
 - v. When discussion session has been confirmed.
 - b. Lecturer:
 - i. When students submit their assignments.

4.4.1.2 Web Application

1. The application must be able to authenticate user credentials upon logging in.
2. User shall be able to logout from the site.
3. The application must allow students to form group with other students.
4. The application must be able to allow text chat within a group assignment for discussion.
5. The application must allow students to invite other students to form a group.
6. The system should notify the students affected by the matchmaking system through email notifications.
7. The application must automatically display the up-to-date view of the source code at any given time to all the users.
8. The application should enable students to propose desired time and date for discussion.
9. The application should make up the best time for the students to discuss assignment based on the input it received.
10. The application must allow students to upload any related file along with it to turn in.
11. The application must allow lecturers to view the students' progress.

4.5 Non-functional Requirement

Non-functional requirements specify the properties of the product, instead of the features that they offered. There are a large number of different non-functional requirement that a project can have. However, they are optional and can be ignored unless specified.

4.5.1 Availability Requirement

1. The program should be available for access every day for 24 hours with a maximum of 10 minutes downtime.
2. The program shall allow users to use it regardless of physical geographical location that they are in.

4.5.2 Adaptability Requirement

1. The application should have cross-browser support for Chrome, Safari, Edge and Firefox browsers.
2. The application shall have different responsive views for different window resolution.
3. The application shall NOT support mobile or tablet views.

4.5.3 Capacity Requirement

1. The source codes of a project must not exceed 5MB of physical space on the server.
2. The assignment question size that uploaded by lecturer must not exceed 5MB.
3. The assignment file submitted by student must not exceed 30MB.

4.5.4 Development Requirement

1. The system must be developed as a web-based application.
2. Parallel development methodology must be practised in the development.
3. Vue.js must be used as front-end development framework.
4. Node.js must be used as the back-end server that supports the web user interface.
5. Express.js shall be used as the back-end framework for the Node.js server.
6. JavaScript must be used for front-end development.
7. TypeScript must be used for back-end development.

8. The data should be recorded in an online cloud Firebase database.

4.5.5 Performance Requirement

1. The system should catch 95% of errors during runtime and display the messages accordingly.
2. The time required to make a request to the server and obtain the response back from the server shall not exceed 3 seconds.
3. The web user interface should not take more than 3 seconds to render completely.
4. The system must handle different concurrent request without crashing.

4.5.6 Security Requirement

1. The system must allow authorized users to use the application only.
2. The system must employ MD5 algorithm to store the password of the users.
3. The system must prohibit students from accessing the lecturer's interface.

4.5.7 Usability Requirement

1. The UI for the application must be simple and intuitive for users to navigate across the interface.
2. The UI for the application must have achieved a certain level of aesthetic value.
3. The application must display a constructive error message if anything wrong occurred.

4.6 Use Cases

This section demonstrates the activities that can be performed by the different actors of this system. The actors identified are student, lecturer and system. In total, 23 use cases were identified and described in this section. Each use case description consists of the primary actor of the use case, the relationship with other use cases, the event flow as well as the exceptional flow of the event.

4.6.1 Use Case Diagram

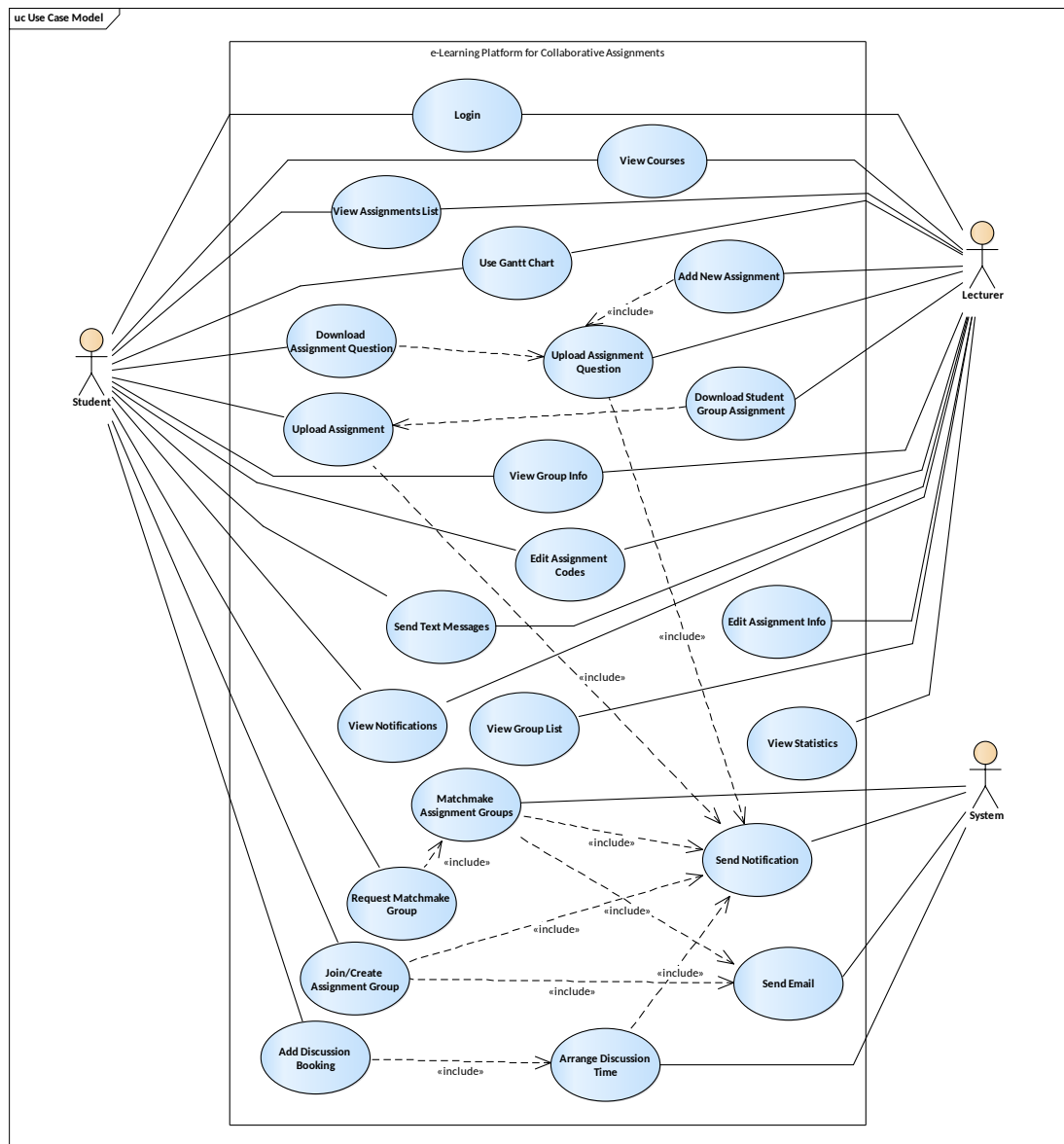


Figure 4.11: Use Case Diagram for This Project

4.6.2 Use Case Description

Table 4.1: Use Case of Login

Name: Login	ID: 1	Priority: High
Actor: Student, Lecturer	Type: Detail, Essential	
Stakeholder Information: Student → User that needs to access the platform but already has the account. Lecturer → User that keep track of students' progress and collect assignment from the platform.		
Summary: This use case shows the steps when students and lecturer who wish to log on the platform for respective purpose.		
Trigger: The user who wishes to use the platform for their own purpose and they already have a valid account.		
Relationship: <ul style="list-style-type: none"> • Association: Student, Lecturer 		
Main Flow: <ol style="list-style-type: none"> 1. The user types in their respective id in the id field. 2. The user inserts their private password in the password field. 3. User clicks on the login button. If credential was incorrect, perform sub-flow 3.1. 4. Upon successful login, users will be directed to their respective homepage of the platform. 		
Sub Flow: <ol style="list-style-type: none"> 3.1. User re-enter the correct credentials in the respective text field provided. 		
Alternate Flow: -		

Table 4.2: Use Case of View Notifications

Name: View Notifications	ID: 2	Priority: High
Actor: Student, Lecturer	Type: Detail, Essential	
Stakeholder Information: Student → User that uses the platform for doing group assignments.		

Lecturer → User conducted programming course and using the platform for collecting assignments.
Summary: This use case illustrates the steps taken for user to view his/her own personalized notifications.
Trigger: The user wishes to check for what is happening recently on the platform.
Relationship: <ul style="list-style-type: none"> • Association: Student, Lecturer • Include: -
Main Flow: <ol style="list-style-type: none"> 1. User clicks on the Notifications link on the sidebar. 2. The relevant notifications up to 25 of them will be shown.
Sub Flow: -
Alternate Flow: -

Table 4.3: Use Case of View Courses

Name: View Courses	ID: 3	Priority: Low
Actor: Student, Lecturer	Type: Detail, Essential	
Stakeholder Information: Student → User that uses the platform for doing group assignments. Lecturer → User conducted programming course and using the platform for collecting assignments.		
Summary: This use case illustrates the steps taken to view user's courses.		
Trigger: The user wishes to get more information of the course that they enrolled/in charge for the semester.		

<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Student, Lecturer • Include: -
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. User clicks on the ‘Courses’ link on the navigation sidebar. 2. The associated courses will be shown upon successful loading.
<p>Sub Flow:</p> <p>-</p>
<p>Alternate Flow:</p> <p>-</p>

Table 4.4: Use Case of View Assignments List

Name: View Assignment List	ID: 4	Priority: High
Actor: Student, Lecturer	Type: Detail, Essential	
<p>Stakeholder Information:</p> <p>Student → User that uses the platform for doing group assignments.</p> <p>Lecturer → User conducted programming course and using the platform for collecting assignments.</p>		
<p>Summary:</p> <p>This use case illustrates the steps taken to view user’s assignments list.</p>		
<p>Trigger:</p> <p><u>General</u></p> <ol style="list-style-type: none"> 1. The user wishes to get a general overview of the total assignments that they partake for the semester. <p><u>Student</u></p> <ol style="list-style-type: none"> 1. Student wishes to join/create/matchmake an assignment group. 2. Student wishes to enter the assignment group workspace. 3. Student wishes to view the assignment group’s details. <p><u>Lecturer</u></p> <ol style="list-style-type: none"> 1. Lecturer wishes to see the list of assignment groups. 2. Lecturer wishes to view/edit the assignment details including upload assignment question. 3. Lecturer wishes to add a new assignment for a course. 		
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Student, Lecturer • Include: - 		

<p>Main Flow:</p> <ol style="list-style-type: none"> 1. User clicks on the Assignments link on the navigation sidebar. 2. The associated assignments will be loaded on the screen.
<p>Sub Flow:</p> <p>-</p>
<p>Alternate Flow:</p> <p>-</p>

Table 4.5: Use Case of View Assignment Group Info

Name: View Assignment Group Info	ID: 5	Priority: High
Actor: Student, Lecturer	Type: Detail, Essential	
<p>Stakeholder Information:</p> <p>Student → User that uses the platform for doing group assignments.</p> <p>Lecturer → User conducted programming course and using the platform for collecting assignments.</p>		
<p>Summary:</p> <p>This use case illustrates the steps taken to view assignment group details.</p>		
<p>Trigger:</p> <p><u>Student</u></p> <ol style="list-style-type: none"> 1. Student wishes to view the assignment group's details. <p><u>Lecturer</u></p> <ol style="list-style-type: none"> 1. Lecturer wishes to check on one of the group's details. 		
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Student, Lecturer • Include: - 		
<p>Main Flow:</p> <p><u>Student</u></p> <ol style="list-style-type: none"> 1. Student clicks on the Assignments link on the navigation sidebar. 2. Student clicks on the Information icon of the group. 3. The group info page will be shown. <p><u>Lecturer</u></p> <ol style="list-style-type: none"> 1. Lecturer clicks on the Assignments link on the navigation sidebar. 2. Lecturer clicks on the Group List link on the desired assignments. 3. After selecting the intended group of students, lecturer clicks on the Information icon of the group. 4. The lecturer will be redirected to the group's information page. 		

Sub Flow: -
Alternate Flow: -

Table 4.6: Use Case of Create/Join Group

Name: Create/Join Group	ID: 6	Priority: High
Actor: Student	Type: Detail, Essential	
Stakeholder Information: Student → User that participate in an assignment group to do the tasks as a team.		
Summary: This use case demonstrates the steps required by a student to create or join an assignment group based on the programming course that they have joined.		
Trigger: Student that are new to a programming course that still does not have found an assignment group yet that wishes to join or create an assignment group.		
Relationship: <ul style="list-style-type: none"> • Association: Student • Include: Send Email (Use Case ID: 21), Send Notification (Use Case ID: 22) 		
Main Flow: <u>Create Group</u> <ol style="list-style-type: none"> 1. Student navigates to the assignments page. 2. Student clicks on the create group button. 3. Student is required to create a group name. 4. Student confirms the name. 5. Upon successful creation, student will be redirected to the page where the group information is shown. <u>Join Group</u> <ol style="list-style-type: none"> 1. Student navigates to the assignments page. 2. Student clicks on the join group button. 3. Student are required to insert the group id. 4. When the system found the group, display the group name and details for confirmation. 5. Student click the confirm button. 6. Student will be redirected to the homepage of the assignment group. 		
Sub Flow:		

<p><u>Create Group</u></p> <p>5.1. Student can invite other student to join the group by sharing the group id in the details page.</p>
<p>Alternate Flow:</p> <p><u>Join Group</u></p> <p>4.1. If no group with such id was found, display an error message indicating no such group.</p> <p>5.1. Students click cancel if the details of the group do not match.</p>

Table 4.7: Use Case of Edit Assignment Code

Name: Edit Assignment Code	ID: 7	Priority: High
Actor: Student, Lecturer	Type: Detail, Essential	
<p>Stakeholder Information:</p> <p>Student → Users that are having programming assignment that wishes to work on the assignment.</p> <p>Lecturer → User that are conducting a programming course and had released the group assignment.</p>		
<p>Summary:</p> <p>This use case shows the steps needed to be taken for both student and lecturer to edit an assignment (coding).</p>		
<p>Trigger:</p> <ol style="list-style-type: none"> 1. The student needs to work on the group assignment given that they already have a group. 2. Lecturer wishes to check on the different group of students under the course that is conducting. 		
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Student, Lecturer • Include: - 		
<p>Main Flow:</p> <p><u>Student</u></p> <ol style="list-style-type: none"> 1. The student clicks on the workspace icon on the assignments page. 2. Student now can edit the file after loaded. <p><u>Lecturer</u></p> <ol style="list-style-type: none"> 1. Lecturer navigates to the assignment list page. 2. Lecturer selects the assignment that he/she wishes to improvise. 3. Lecturer clicks on the workspace icon of the desired assignment group. 4. After loaded the group's project, lecturer can edit their work. 		
<p>Sub Flow:</p> <p>-</p>		

Alternate Flow: -

Table 4.8: Use Case of Send Text Messages

Name: Send Text Messages	ID: 8	Priority: High
Actor: Student, Lecturer	Type: Detail, Essential	
Stakeholder Information: Student → User that are concurrently editing the assignment with another user. Lecturer → User that supervise the course and assignment.		
Summary: This use case shows the steps required in order to use the chat.		
Trigger: Students that are working together in the assignment workspace on the platform that wishes to communicate and discuss ideas through each other. Lecturer that wishes to help or communicate with a group of students regarding the assignment in the workspace.		
Relationship: <ul style="list-style-type: none"> • Association: Student, Lecturer • Include: - 		
Main Flow: <ol style="list-style-type: none"> 1. User navigates into the assignment workspace. 2. User type messages in the chat box. 3. User hits enter to send the message. 		
Sub Flow: -		
Alternate Flow: -		

Table 4.9: Use Case of Use Gantt Chart

Name: Use Gantt Chart	ID: 9	Priority: High
Actor: Student, Lecturer	Type: Detail, Essential	
<p>Stakeholder Information: Student → User that are a member in an assignment group in a programming course. Lecturer → User that are in charge of the assignment where the groups are based.</p>		
<p>Summary: This use case shows the steps required in order to use the Gantt Chart.</p>		
<p>Trigger: Students that are working together in the assignment workspace on the platform that wishes to schedule and assign task. Lecturer that spotted the inactive group and wishes to assign the students tasks by him/herself.</p>		
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Student, Lecturer • Include: - 		
<p>Main Flow:</p> <p><u>Create Gantt Activity</u></p> <ol style="list-style-type: none"> 1. User navigates into the assignment workspace. 2. User clicks on the Gantt Chart icon on the left side. 3. User clicks on the create task button. 4. User enters the task name and assignee for the task. 5. User inserts the planned duration and expected deadline for the task. 6. User clicks Confirm. If user is hesitant about the creation, perform Sub Flow 1. 7. Upon successful insertion, a Gantt Chart with the new activity will appear at bottom. <p><u>Edit Gantt Activity</u></p> <ol style="list-style-type: none"> 1. User clicks on the edit icon on the right side of the desired task. 2. User makes changes on the popup shown. 3. User clicks Confirm. If user is hesitant about the edit, perform Sub Flow 1. 4. The updated details will be shown in the Gantt Chart. <p><u>Delete Gantt Activity</u></p> <ol style="list-style-type: none"> 1. User clicks on the delete icon on the right of the desired task. 2. User clicks Confirm on the popup prompt. If user is hesitant about the deletion, perform Sub Flow 1. 3. Successful deletion will be reflected on the Gantt Chart. 		

Sub Flow: 1. User clicks cancel button OR clicks away the modal to exit.
Alternate Flow: -

Table 4.10: Use Case of Add Discussion Booking

Name: Add Discussion Booking	ID: 10	Priority: Moderate
Actor: Student	Type: Detail, Essential	
Stakeholder Information: Student → User that is obligated to discuss assignments among teammates.		
Summary: This use case illustrates how the students can book a booking time.		
Trigger: Student who wishes to announce their available time for discussion.		
Relationship: <ul style="list-style-type: none"> • Association: Student • Include: Arrange Discussion Time (Use Case ID: 11) 		
Main Flow: <ol style="list-style-type: none"> 1. Student navigates to the assignment list page. 2. Student clicks on the workspace link button. If the workspace link icon is outlined, they do not have a group yet, required to perform UCID: 3. Student clicks on the discussion booking link on the sidebar. 4. Student clicks on the add discussion time button. 5. Student chooses the preferred day of the week. 6. Student fills in the starting time of the preferred day. 7. Student fills in the ending time of the preferred day. 8. Student clicks confirm to place the booking. 		
Sub Flow: -		

Alternate Flow: -

Table 4.11: Use Case of Arrange Discussion Time

Name: Arrange Discussion Time	ID: 11	Priority: Moderate
Actor: System	Type: Detail, Essential	
Stakeholder Information: System → the processing unit of the platform.		
Summary: This use case illustrates the flow of system to arrange a group discussion time.		
Trigger: All the students within the group had provided their own preferred discussion time.		
Relationship: <ul style="list-style-type: none"> • Association: System • Include: Send Notification (Use Case ID: 22) 		
Main Flow: <ol style="list-style-type: none"> 1. Compute the most frequent day among the students. 2. Get the overlap time of the results. 3. Update the group's database about the decided date and time. 4. Send a notification to all the team members. 		
Sub Flow: -		
Alternate Flow: -		

Table 4.12: Use Case of Create New Assignment

Name: Create New Assignment	ID: 12	Priority: High
Actor: Lecturer	Type: Detail, Essential	

<p>Stakeholder Information: Lecturer → User conducted programming course and using the platform for assignment management.</p>
<p>Summary: This use case illustrates the steps necessary for a lecturer to create a new assignment for one of his/her course.</p>
<p>Trigger: Lecturer wishes to add an assignment for one of his/her course.</p>
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Lecturer • Include: Upload Assignment Question (Use Case ID: 14)
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. Lecturer navigates to the assignments page by clicking on the corresponding navigation link of the sidebar. 2. Lecturer clicks on the Add an Assignment button and a popup will be shown. 3. Lecturer fills in the description of the assignment. 4. Lecturer selects a course that he/she wants the assignment to be added for. 5. Lecturer chooses a programming language in which the assignment will be based on. 6. Lecturer specifies the number of maximum students per group. 7. Lecturer specifies the assignment number for the current assignment. 8. Lecturer chooses a due date for the assignment. 9. After all details is set, lecturer clicks on the confirm button to add the assignment.
<p>Sub Flow: -</p>
<p>Alternate Flow: -</p>

Table 4.13: Use Case of Edit Assignment Details

Name: Edit Assignment Details	ID: 13	Priority: Medium
Actor: Lecturer	Type: Detail, Essential	
<p>Stakeholder Information: Lecturer → User conducted programming course and using the platform for</p>		

collecting assignments.
<p>Summary: This use case illustrates the steps taken to edit assignment's detail.</p>
<p>Trigger: The lecturer wishes to correct or update the information of the assignment that he/she has created.</p>
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Lecturer • Include: -
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. Lecturer clicks on the 'Assignments' link on the navigation sidebar. 2. Lecturer clicks on the details link of the assignment and navigates into the detail page. 3. The details of the assignment will be shown and readily accept any changes that the lecturer wishes to make. 4. After making suitable changes, lecturer clicks on the save button beneath the section. 5. Upon successful update, a success message in green will be shown beneath.
<p>Sub Flow: -</p>
<p>Alternate Flow: -</p>

Table 4.14: Use Case of Upload Assignment Question

Name: Upload Assignment Question	ID: 14	Priority: Moderate
Actor: Lecturer	Type: Detail, Essential	
<p>Stakeholder Information: Lecturer → User that is in charge of the assignment and the course.</p>		
<p>Summary: This use case shows the steps taken by lecturer to upload an assignment question.</p>		

<p>Trigger: After lecturer creating the assignment, he/she wishes to upload the assignment question for that particular assignment.</p>
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Lecturer • Include: Send Notification (Use Case ID: 22)
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. Lecturer clicks on the ‘details’ link to navigate to the newly added assignment’s detail page. 2. Lecturer clicks on the choose file button at the bottom of the page under the assignment question section. 3. Lecturer chooses the assignment file and upload. 4. The uploading process takes a few moments. 5. Upon successful uploads, the name of the file will be appeared in the middle of the section. 6. Lecturer may click on the file name to download and verify the file uploaded.
<p>Sub Flow: -</p>
<p>Alternate Flow: -</p>

Table 4.15: Use Case of Download Assignment Question

Name: Download Assignment Question	ID: 15	Priority: Moderate
Actor: Student	Type: Detail, Essential	
Stakeholder Information: Student → User that are working on a group assignment as a team.		
Summary: This use case outlines the different steps taken by student to download an assignment question.		
Trigger: Students wish to download assignment questions from the platform.		
Relationship: <ul style="list-style-type: none"> • Association: Student • Include: - 		

<p>Main Flow: <u>Student</u></p> <ol style="list-style-type: none"> 1. Student navigates into the assignment list page. If user is in the group information page, perform Sub Flow 1.1 2. Student locates the assignment on which he/she wishes to download the question for. 3. Student checks the download icon that is located right most on the assignment item of the list. If the icon is outlined and the tooltip shows ‘No questions uploaded’, perform Alternate Flow 3.1. 4. Student clicks on the download icon 5. The file will be downloaded into the default downloads folder.
<p>Sub Flow:</p> <ol style="list-style-type: none"> 1.1. Student clicks on the download button on the toolbars of the group information page.
<p>Alternate Flow:</p> <ol style="list-style-type: none"> 3.1 Student could not download the question as the lecturer haven’t upload it yet.

Table 4.16: Use Case of Submit Assignment

Name: Submit Assignment	ID: 16	Priority: High
Actor: Student	Type: Detail, Essential	
Stakeholder Information: Student → User that is a group leader and has completed the group assignment.		
Summary: Submission of assignment files by the student.		
Trigger: <ol style="list-style-type: none"> 1. Students that are working together on the group assignment had completed the assignment and wish to submit the assignment. 2. The due date for the assignment has been reached. 		
Relationship: <ul style="list-style-type: none"> • Association: Student • Include: Send Notification (Use Case ID: 22) 		
Main Flow: <u>Student</u> <ol style="list-style-type: none"> 1. Student navigates into the assignment upload page. 2. Student prepares the assignment file in a zip format. 3. Students click on the submit assignment drop box or drag the file into 		

<p>the provided drop box.</p> <p>4. Upon successful submission, a congratulation message will be shown.</p>
<p>Sub Flow:</p> <p>-</p>
<p>Alternate Flow:</p> <p>-</p>

Table 4.17: Use Case of Download Student Group Assignments

Name: Download Student Group Assignments	ID: 17	Priority: High
Actor: Lecturer	Type: Detail, Essential	
<p>Stakeholder Information: Lecturer → conducted programming course and using the platform for collecting assignments.</p>		
<p>Summary: This use case illustrates the steps taken by a lecturer to retrieve assignments.</p>		
<p>Trigger: The lecturer that wishes to mark the student's work.</p>		
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Lecturer • Dependency: Student Upload 		
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. Lecturer navigates to the assignment list page that shows all assignments that he/she is in charge of. 2. Lecturer clicks on the desired assignment to go to the groups list page. 3. Lecturer verifies the submission status of the individual group as shown on the rightmost side. 4. Lecturer clicks into the group details page. 5. Lecturer finds and clicks on the uploaded file link. 6. The downloaded file will be in the browser's default download folder. 		
<p>Sub Flow:</p> <p>-</p>		

Alternate Flow: -

Table 4.18: Use Case of View Statistics

Name: View Statistics	ID: 18	Priority: Medium
Actor: Lecturer	Type: Detail, Essential	
Stakeholder Information: Lecturer → User conducted programming course and using the platform for collecting assignments.		
Summary: This use case illustrates the steps taken to view statistics.		
Trigger: The lecturer wishes to get an overview about the current state of assignments that is in charged by him/her.		
Relationship: <ul style="list-style-type: none"> • Association: Lecturer • Include: - 		
Main Flow: <ol style="list-style-type: none"> 1. Lecturer clicks on the ‘Analytics’ link on the navigation sidebar. 2. The charts will be shown grouped by assignments. 		
Sub Flow: -		
Alternate Flow: -		

Table 4.19: Use Case of Request Matchmake Group

Name: Request Matchmake Group	ID: 19	Priority: Medium
Actor: Student	Type: Detail, Essential	

<p>Stakeholder Information: Student → User conducted programming course and using the platform for collecting assignments.</p>
<p>Summary: This use case illustrates the steps taken to view statistics.</p>
<p>Trigger: The student who currently has no group for an assignment and wishes to participate in a group.</p>
<p>Relationship:</p> <ul style="list-style-type: none"> • Association: Student • Include: Matchmake Assignment Group (Use Case ID: 20)
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. Student navigates to the Assignment List. 2. Student clicks on the same icon of the workspace but is outlined which indicates there is no group for that particular assignment yet. 3. Student clicks the 'Matchmake' option at the popup shown.
<p>Sub Flow: -</p>
<p>Alternate Flow: -</p>

Table 4.20: Use Case of Matchmake Assignment Groups

Name: Matchmake Assignment Groups	ID: 20	Priority: Medium
Actor: System	Type: Detail, Essential	
<p>Stakeholder Information: System → The processing unit of the platform.</p>		
<p>Summary: This use case the steps for system to matchmake students.</p>		
<p>Trigger: The assignment queue had gathered the maximum allowed student for that particular assignment.</p>		

<p>Relationship:</p> <ul style="list-style-type: none"> • Association: System • Include: Send Email (Use Case ID: 21), Send Notification (Use Case ID: 22)
<p>Main Flow:</p> <ol style="list-style-type: none"> 1. Remove the students from the queue. 2. Gather the emails of the students. 3. Randomly selects a leader. 4. Creates a group for the students. 5. Update the students' profile. 6. Send the notification to affected students. 7. Send the email to the affected students.
<p>Sub Flow:</p> <p>-</p>
<p>Alternate Flow:</p> <p>-</p>

Table 4.21: Use Case of Send Email

Name: Send Email	ID: 21	Priority: Low
Actor: System	Type: Detail, Essential	
Stakeholder Information: System → the processing unit of the platform.		
Summary: This use case illustrates the flow of system to send an email.		
Trigger: The student who does not have a group are successfully being matchmake by the system or a student had joined another group successfully.		
Relationship: <ul style="list-style-type: none"> • Association: System • Include: - 		
Main Flow: <ol style="list-style-type: none"> 1. Gather the email of the affected students, evidently the group members. 2. Send the email. 		
Sub Flow: -		

Alternate Flow: -

Table 4.22: Use Case of Send Notification

Name: Send Notification	ID: 22	Priority: Medium
Actor: System	Type: Detail, Essential	
Stakeholder Information: System → The processing unit of the platform.		
Summary: This use case the conditions of system to send notifications.		
Trigger: <ol style="list-style-type: none"> 1. Group leader (Student) submitted the assignment. 2. Lecturer uploaded the assignment question. 3. New member (Student) joined a group. 4. Matchmake of students are successful. 5. Discussion session determined. 		
Relationship: <ul style="list-style-type: none"> • Association: System • Include: - 		
Main Flow: <ol style="list-style-type: none"> 1. Listen to various events that will trigger this action. 2. Determine the actor of the event. 3. Obtain the name, user ID and avatar Uri of the actor. 4. Parse the activities into a notification object. 5. Push the parsed notification object to the database. 6. Users affected will be able to see the notification in notification page. 		
Sub Flow: -		
Alternate Flow: -		

Table 4.23: Use Case of View Group List

Name: View Group List	ID: 23	Priority: Moderate
Actor: Lecturer	Type: Detail, Essential	
Stakeholder Information: Lecturer → User that is in charge of the assignment and the course.		
Summary: This use case shows the steps taken by lecturer view group list of an existing assignment that he/she is in charge of.		
Trigger: <ol style="list-style-type: none"> 1. Lecturer wishes to get an overview of the existing group of that particular assignment. 2. Lecturer wishes to interact with students in their workspace. 		
Relationship: <ul style="list-style-type: none"> • Association: Lecturer • Include: - 		
Main Flow: <ol style="list-style-type: none"> 1. Lecturer clicks on the 'Assignments' link to navigate to the assignments list page. 2. Lecturer clicks on the group list link of the desired assignment. 3. Lecturer will be brought to the assignment group list page where all groups of the assignment are outlined. 		
Sub Flow: -		
Alternate Flow: -		

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

In this chapter, the system's overall design such as architecture design, user interface design and data modelling.

5.2 System Architecture Design

As most of the web applications out there that applies three-tier architecture as the building block, this project is no exception. The three tiers involved are presentation tier, application tier and data tier. In other words, presentation is the user interface, application tier is the server that holds computation logics and lastly data tier is the database. The tools used for each tier for this project is shown below.

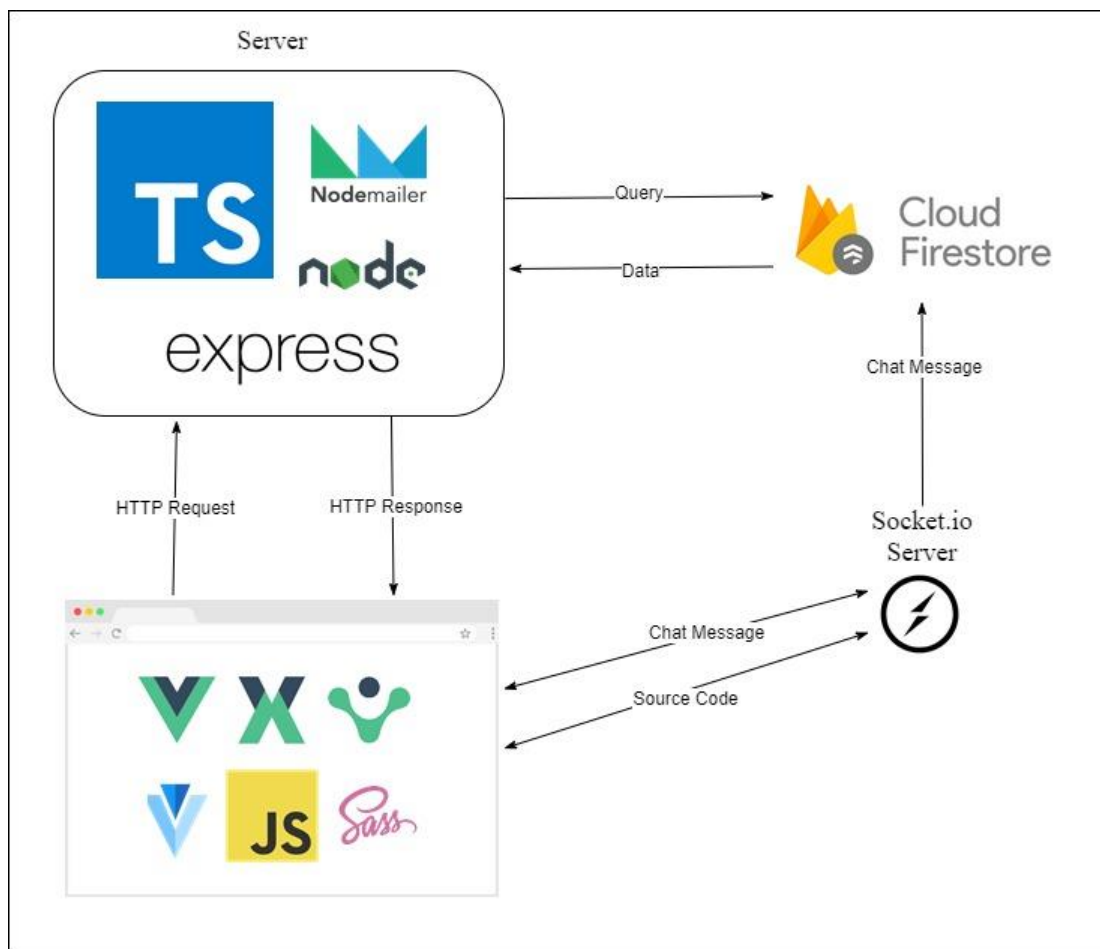


Figure 5.1: System Architecture

In presentation layer, Vue.js is used as the front-end framework to quickly and efficiently develop the UI. A front-end framework is used to code faster as many of the trivial and low-level stuffs are being taken care already. Hence, we need not to spend or waste our time to implement every essential aspect of a webpage or web application from the ground-up. Vuex and Vue Router are used for the extension of feature where Vuex allow sharing of globalised state throughout the application and Vue Router takes care of the routing needed for the application. Vuetify is the material design library for Vue.js which provides a lot of useful components out of the box. Sass is a CSS preprocessor which is cleaner to write and more powerful.

In the application layer, Node.js is used as the server-side language for the system. Basically, Node.js is the runtime JavaScript. The server was written in TypeScript to provide extra security and error avoidance as the code that is ambiguous are strictly not allowed. Express.js is used alongside with Node.js as it is very easy to deploy API endpoints with just a few lines of code. Nodemailer is a smtp package that is used for sending emails in Node.js. Nonetheless, Socket.io was used as a second server that is used primarily to handle real-time data such as chat message.

In the data layer, Firestore of Firebase is used for storing all of the data that is used within the system. Firestore is a document-based database.

5.3 Database Design

Under this section, numerous diagrams will be presented to demonstrate the design logic of the database.

5.3.1 Entity Relationship Diagram

5.3.1.1 Physical Entity Relationship Diagram

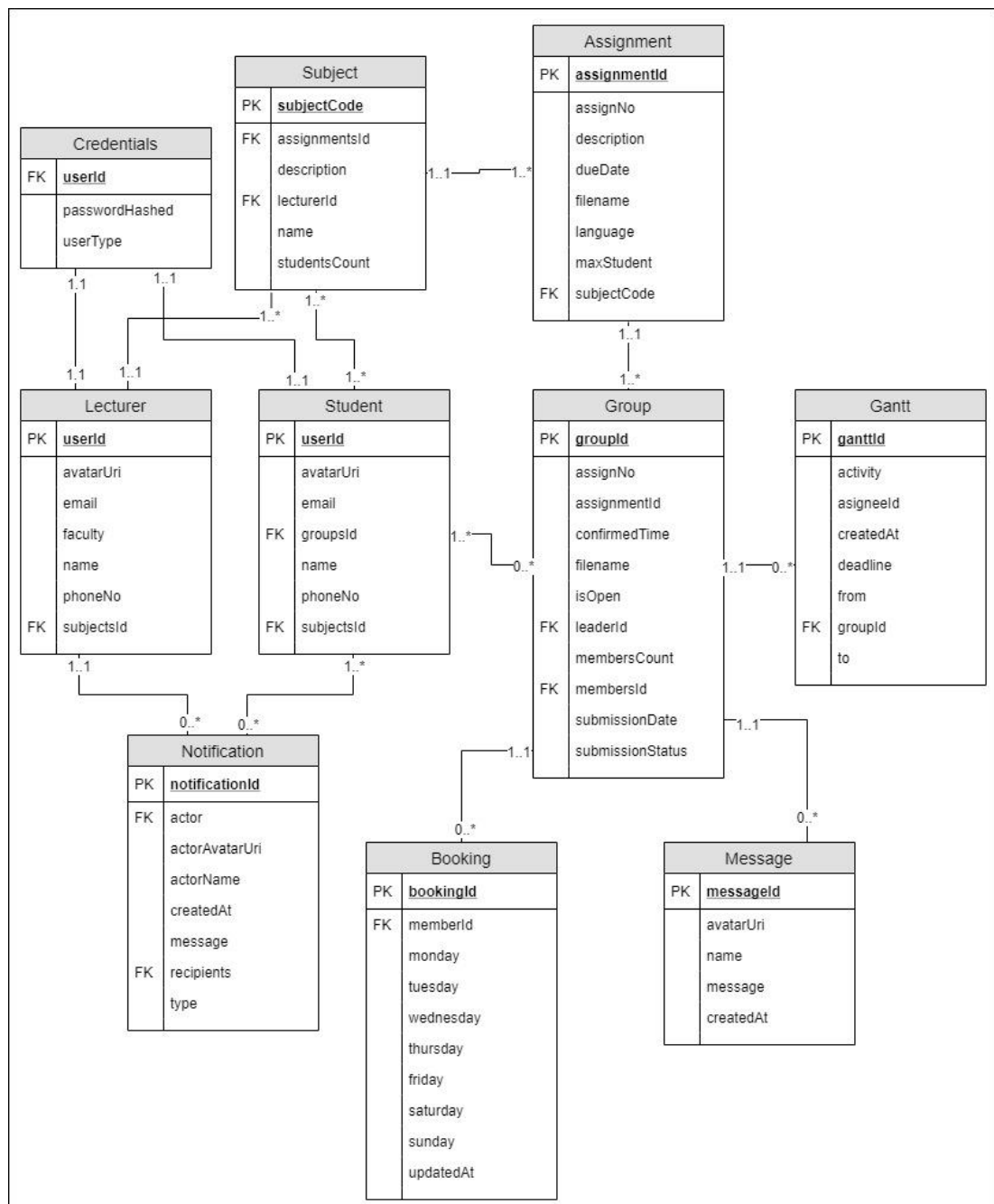


Figure 5.2: Physical Entity Relationship Diagram

5.3.1.2 Logical Entity Relationship Diagram

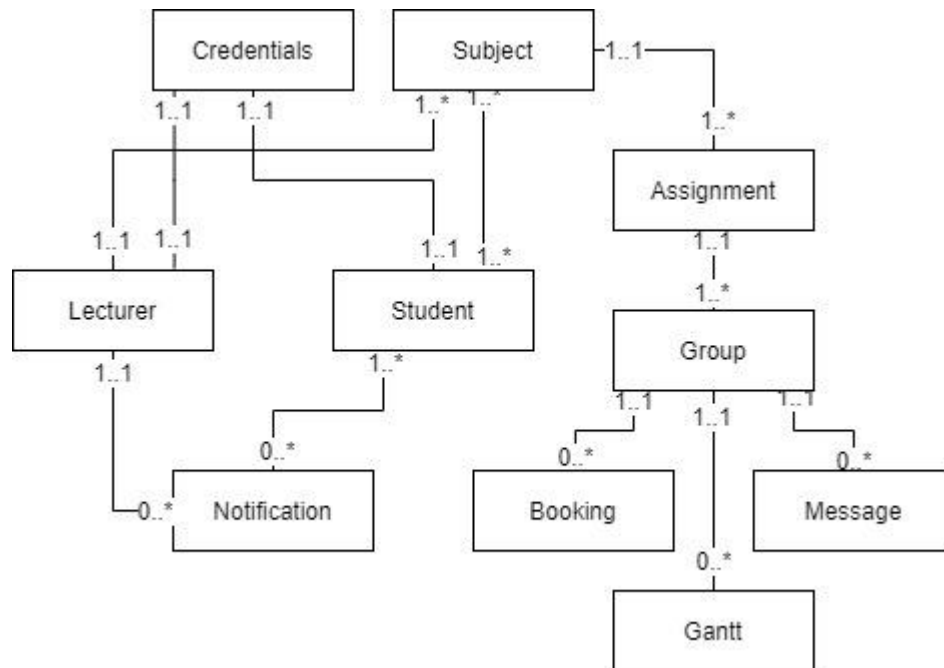


Figure 5.3: Logical Entity Relationship Diagram

5.3.1.3 Entities Description

Table 5.1: Entities Description Table

Entity	Description
Credentials	Stores login credentials for both students and lecturers
Subject	Stores information about each programming subject or course offered on the semester.
Assignment	Stores details of assignments of the existing courses
Lecturer	Database for all lecturers that is conducting courses.
Student	Database for all students involved in programming courses
Group	Stores information of assignment groups
Booking	Stores all booking information of a group.
Gantt	Stores all Gantt Charts of a group.
Message	Stores all text messages of a group.
Notification	Stores all notifications of the users.

Since a NoSQL database (Firestore) is used, the entities are stored inside collections of documents instead of conventional tables used in SQL databases. All

the primary keys are not stored inside the data record itself but is indexed in Firebase directly so that direct calls and queries can still access the results.

5.3.1.4 Data Dictionary

Table 5.2: Data Dictionary for Credentials collection

Attribute	Description	Data Type	PK/FK	NULL
userId	The owner of this credential	string	FK	
passwordHashed	The MD5 hashed password string	string		
userType	The user type of the credential	string		

Table 5.3: Data Dictionary for Subjects collection

Attribute	Description	Data Type	PK/FK	NULL
subjectCode	The unique identifier for subject	string	PK	
assignmentsId	The id of all assignment for this subject	string[]	FK	
description	The description of the course	string		
lecturerId	The lecturer that in charge of the course	string	FK	
name	The title of the course	string		
studentsCount	The total number of students enrolled.	number		

Table 5.4: Data Dictionary for Assignments collection

Attribute	Description	Data Type	PK/FK	NULL
assignmentId	The unique identifier for assignment	string	PK	
assignNo	The latest assignment number of the course. (A1, A2, etc.)	number		
description	The description of the assignment	string		
dueDate	The deadline for the assignment submission	Firestore Timestamp		
filename	The name of the uploaded	string		

	question file			
language	The programming language used	string		
maxStudent	The maximum number of members within one group	number		
subjectCode	The subject which this assignment belongs	string	FK	

Table 5.5: Data Dictionary for Lecturers collection

Attribute	Description	Data Type	PK/FK	NULL
userId	The unique identifier for user	string	PK	
avatarUri	The profile image link of lecturer	string		
email	The email address of the lecturer	string		
faculty	The faculty of the lecturer.	string		
name	The name of the lecturer	string		
phoneNo	The phone number of the lecturer	string		
subjectsId	The subjects that the lecturer is in charged in the semester.	string[]	FK	

Table 5.6: Data Dictionary for Students collection

Attribute	Description	Data Type	PK/FK	NULL
userId	The unique identifier for user	string	PK	
avatarUri	The profile image link of student	string		
email	The email address of the student	string		
groupsId	The groups that the student had joined.	string[]	FK	
name	The name of the student	string		
phoneNo	The phone number of the student	string		
subjectsId	The subjects that the student is enrolled in the semester.	string[]	FK	

Table 5.7: Data Dictionary for Groups collection

Attribute	Description	Data Type	PK/FK	NULL
groupId	The unique identifier for group	string	PK	
assignNo	The assignment number the group belongs to in the course	number		
assignmentId	The assignment that the group belongs	string		
confirmedTime	The time when a discussion session is confirmed	Object		
filename	The name of the submitted assignment file	string		
isOpen	The status of the group whether still accepts new members or not	boolean		
leaderId	The leader (student) of the group	string[]	FK	
membersCount	The number of current members	number		
membersId	The id of other members	string[]		
submissionDate	The data of assignment submission	Firestore Timestamp		
submissionStatus	The submission status of the group	boolean		

Table 5.8: Data Dictionary for Gantts collection

Attribute	Description	Data Type	PK/FK	NULL
ganttId	The unique identifier for Gantt	string	PK	
activity	The activity of the Gantt Chart	string		
assigneeId	The student that in charge of this activity	string	FK	
createdAt	The time when this Gantt is created	Firestore Timestamp		
deadline	The self-set deadline of this Gantt	string		
from	The starting week of this Gantt	number		
to	The ending week of this Gantt	number		
groupId	The group that owns this Gantt	string		

Table 5.9: Data Dictionary for Notifications collection

Attribute	Description	Data Type	PK/FK	NULL
notificationId	The unique identifier for notification	string	PK	
actor	The user (userId) that initiates this notification	string	FK	
actorAvatarUri	The user avatar of the user	string		
actorName	The name of the user	string		
createdAt	The time of initiation	Firestore Timestamp		
message	The message of the notification	string		
recipients	The users that receive this notification	string[]	FK	
type	The type of this notification	string		

Table 5.10: Data Dictionary for Bookings collection

Attribute	Description	Data Type	PK/FK	NULL
bookingId	The unique identifier for booking	string	PK	
memberId	The student (userId) that owns this booking	string	FK	
monday	The user available time on Monday	Object		Yes
tuesday	The user available time on Tuesday	Object		Yes
wednesday	The user available time on Wednesday	Object		Yes
thursday	The user available time on Thursday	Object		Yes
friday	The user available time on Friday	Object		Yes
saturday	The user available time on Saturday	Object		Yes
sunday	The user available time on Sunday	Object		Yes

	Sunday			
updatedAt	The last update time for this booking	Firestore Timestamp		

Table 5.11: Data Dictionary for Messages collection

Attribute	Description	Data Type	PK/FK	NULL
messageId	The unique identifier for message	string	PK	
avatarUri	The profile image link of user	string		
name	The name of the user	string		
message	The message of the user	string		
createdAt	The time when the message was sent	Firestore Timestamp		

5.3.2 Data Flow Diagram

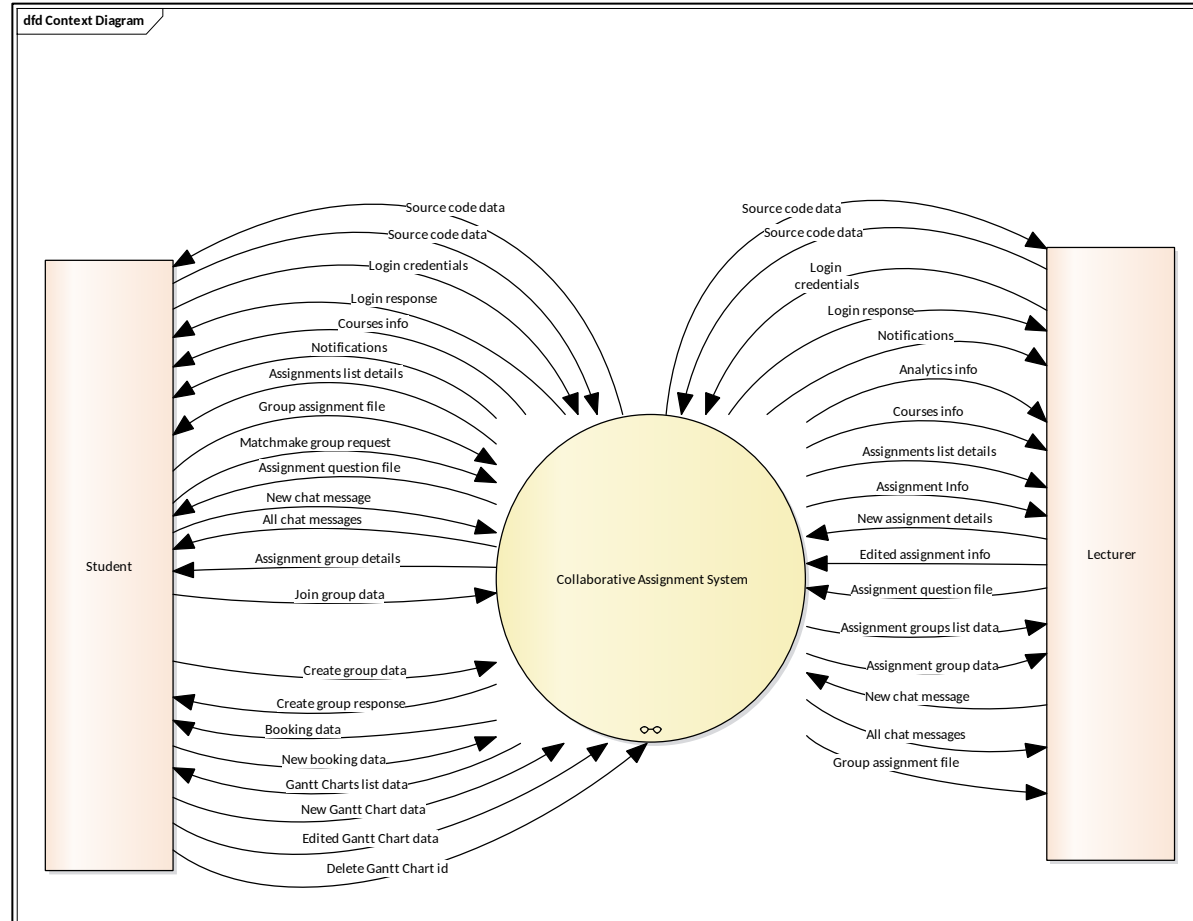


Figure 5.4: Context Diagram

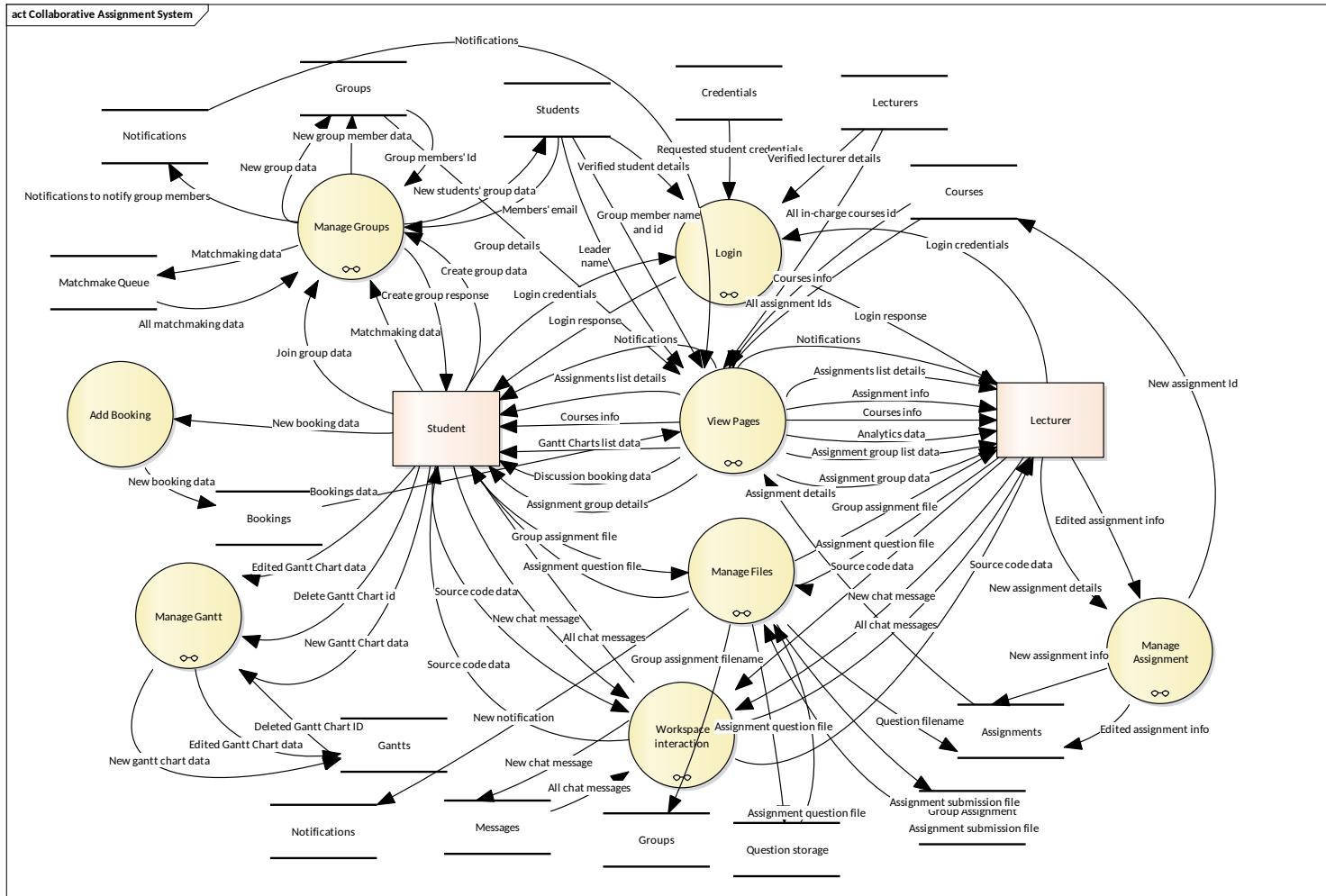


Figure 5.5: A fairly complicated Level 0 Data Flow Diagram

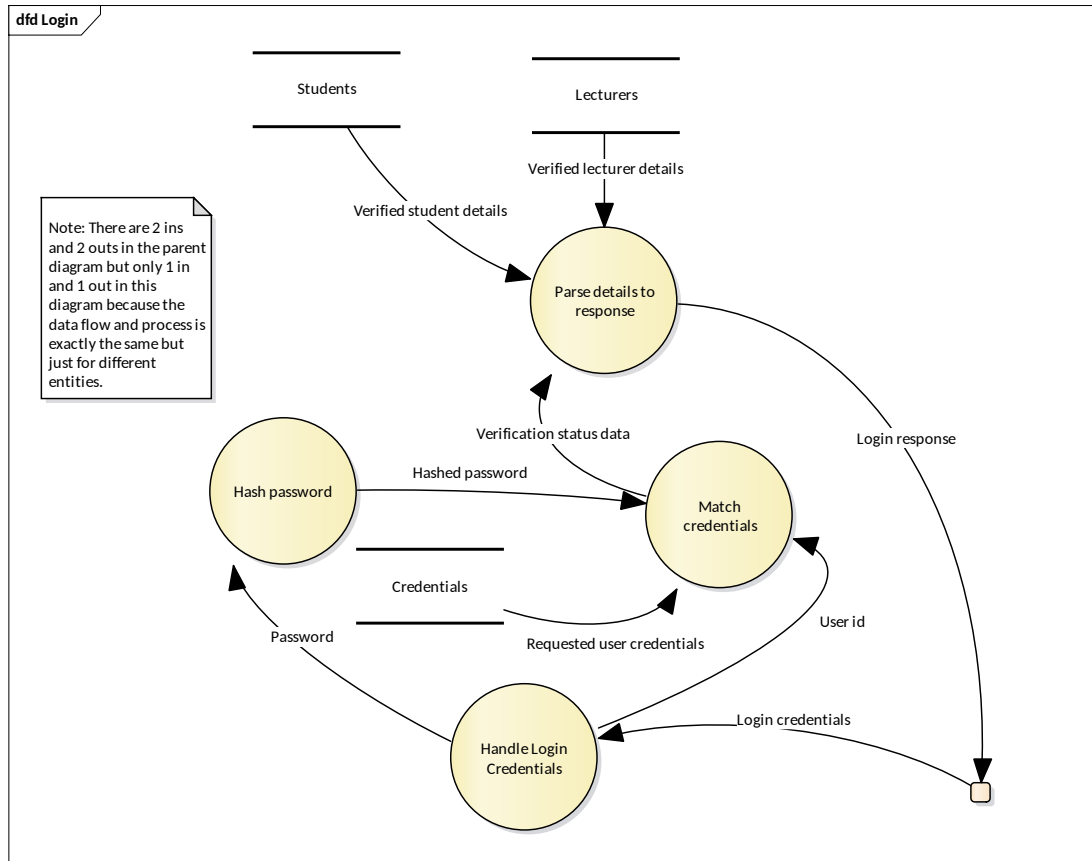


Figure 5.6: Level 1 DFD for Login process

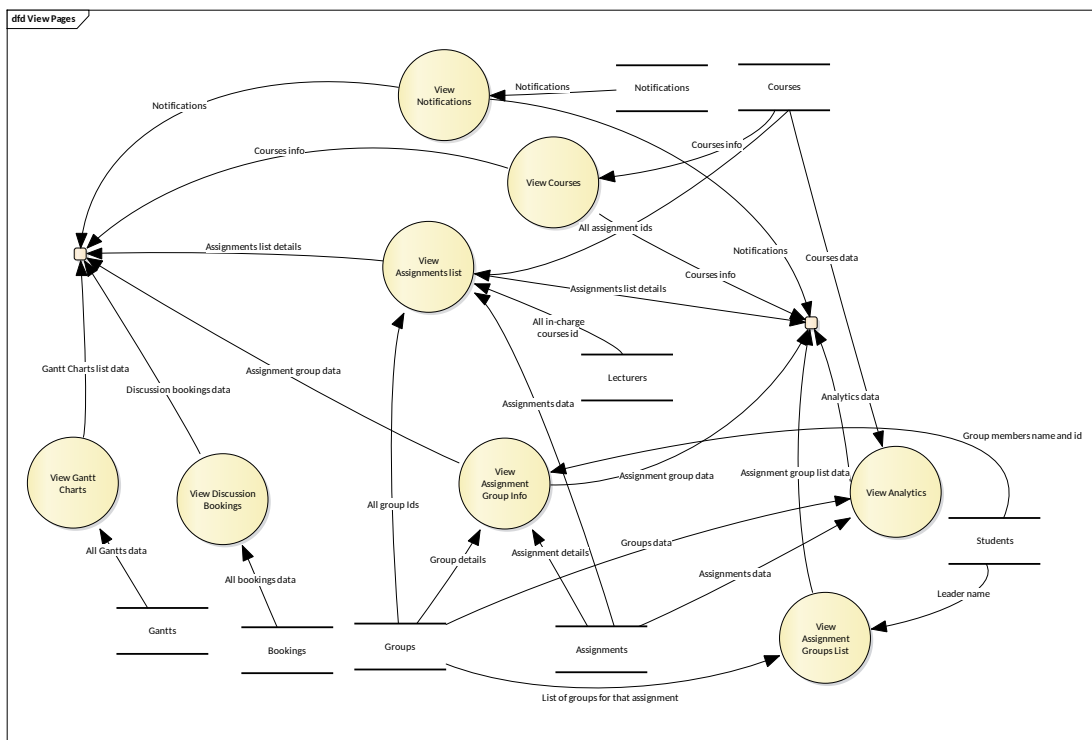


Figure 5.7: Level 1 DFD for View Pages process

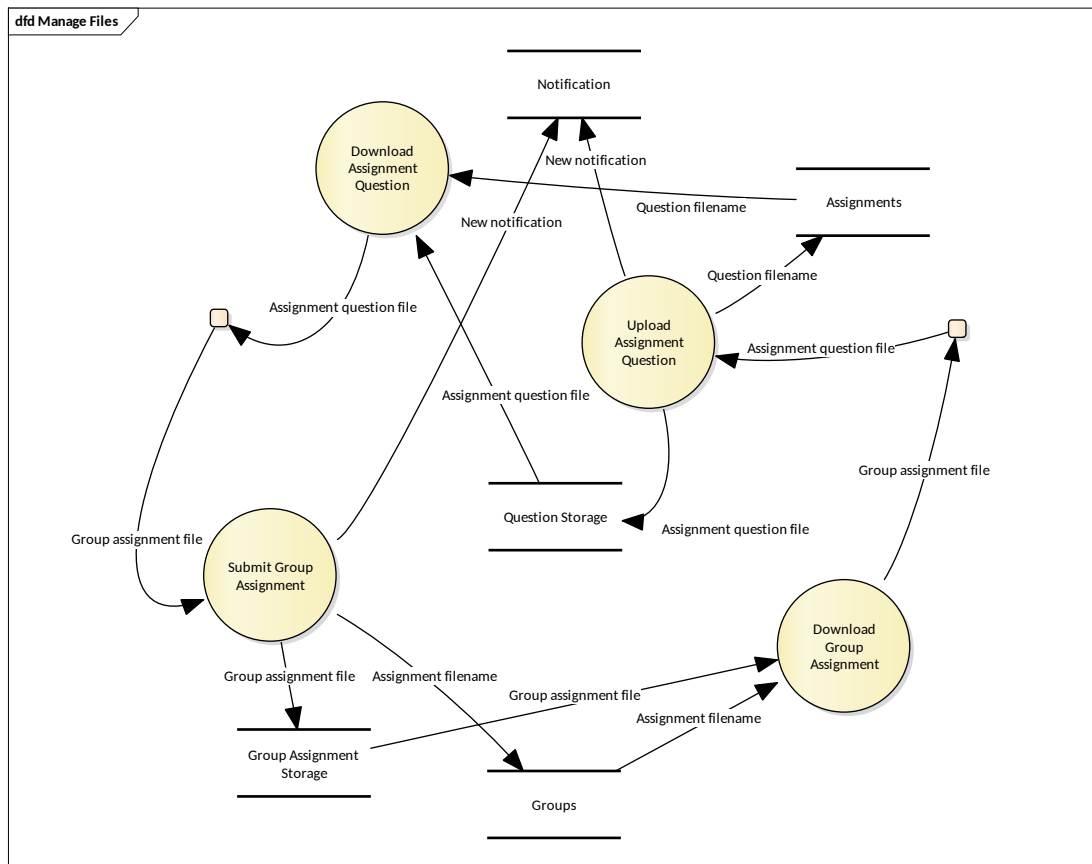


Figure 5.8: Level 1 DFD for Manage Files process

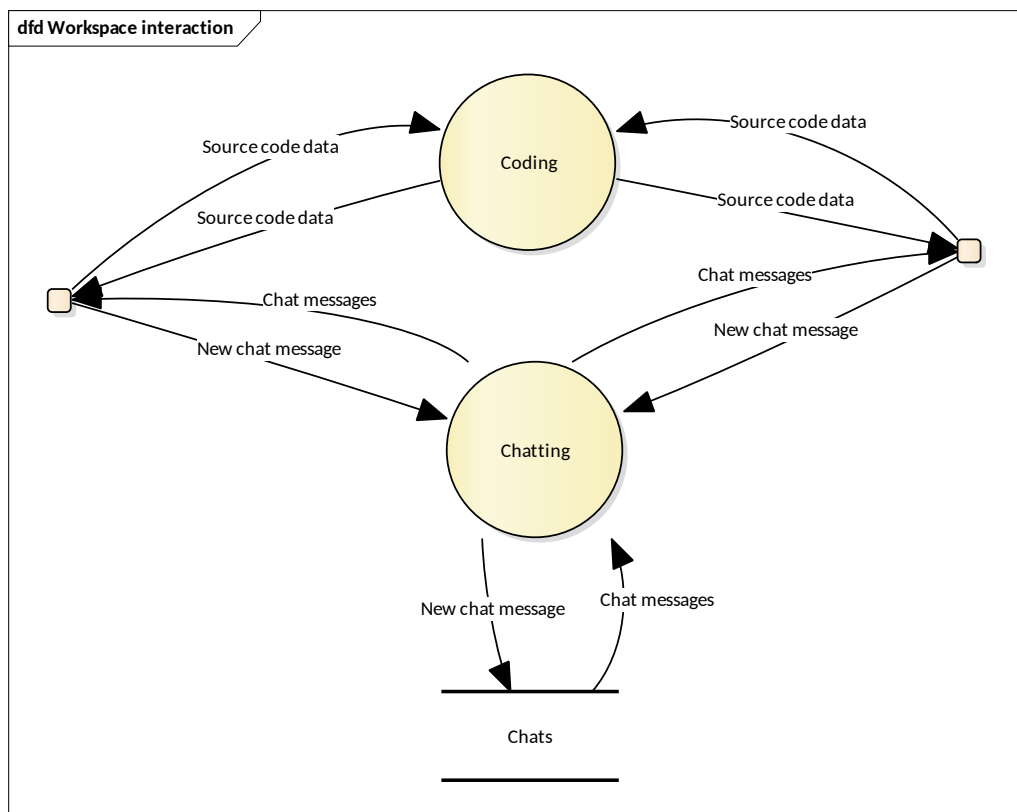


Figure 5.9: Level 1 DFD for Workspace Interaction process

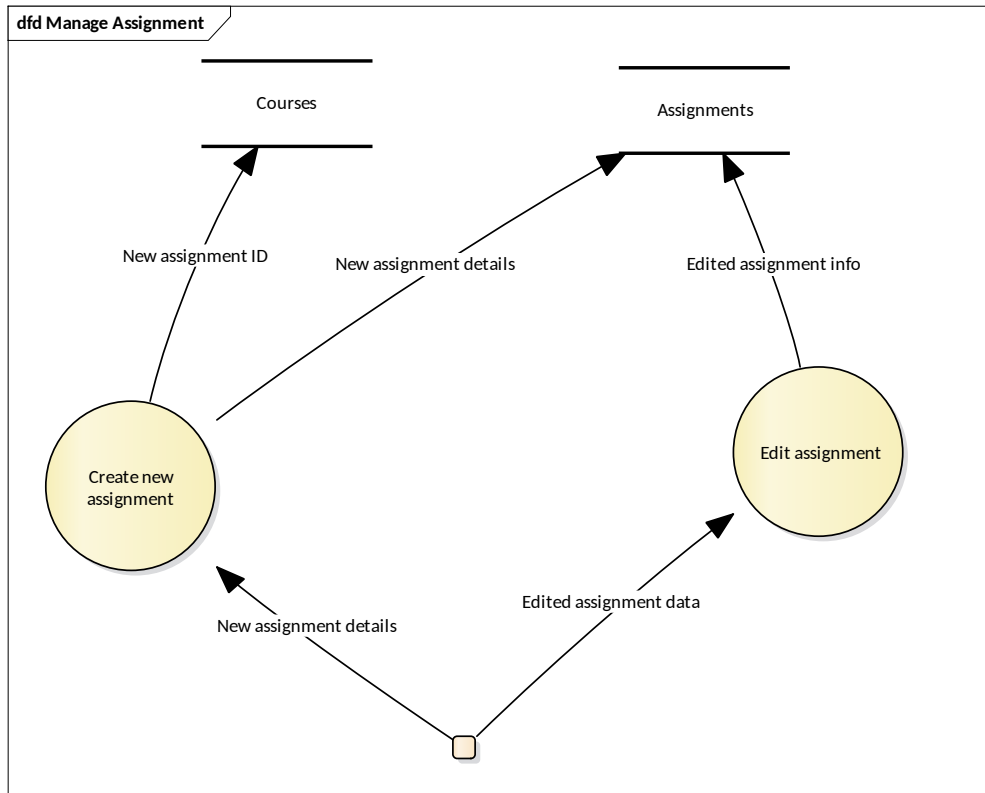


Figure 5.10: Level 1 DFD for Manage Assignment process

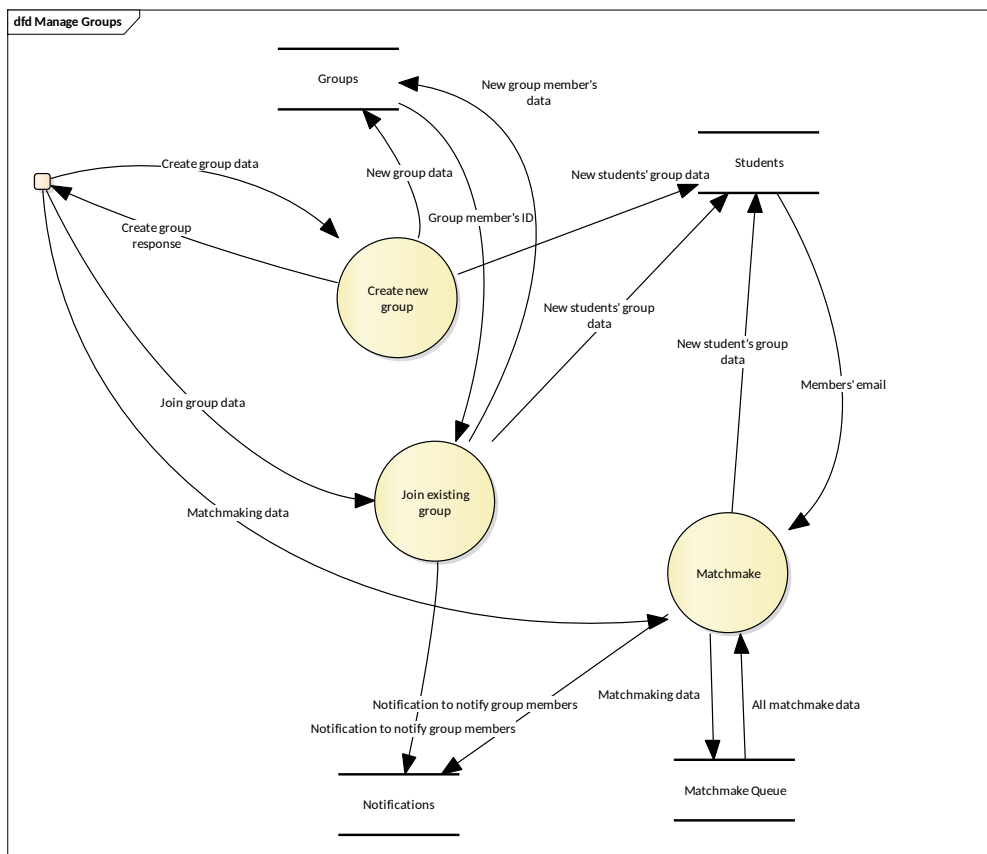


Figure 5.11: Level 1 DFD for Manage Groups process

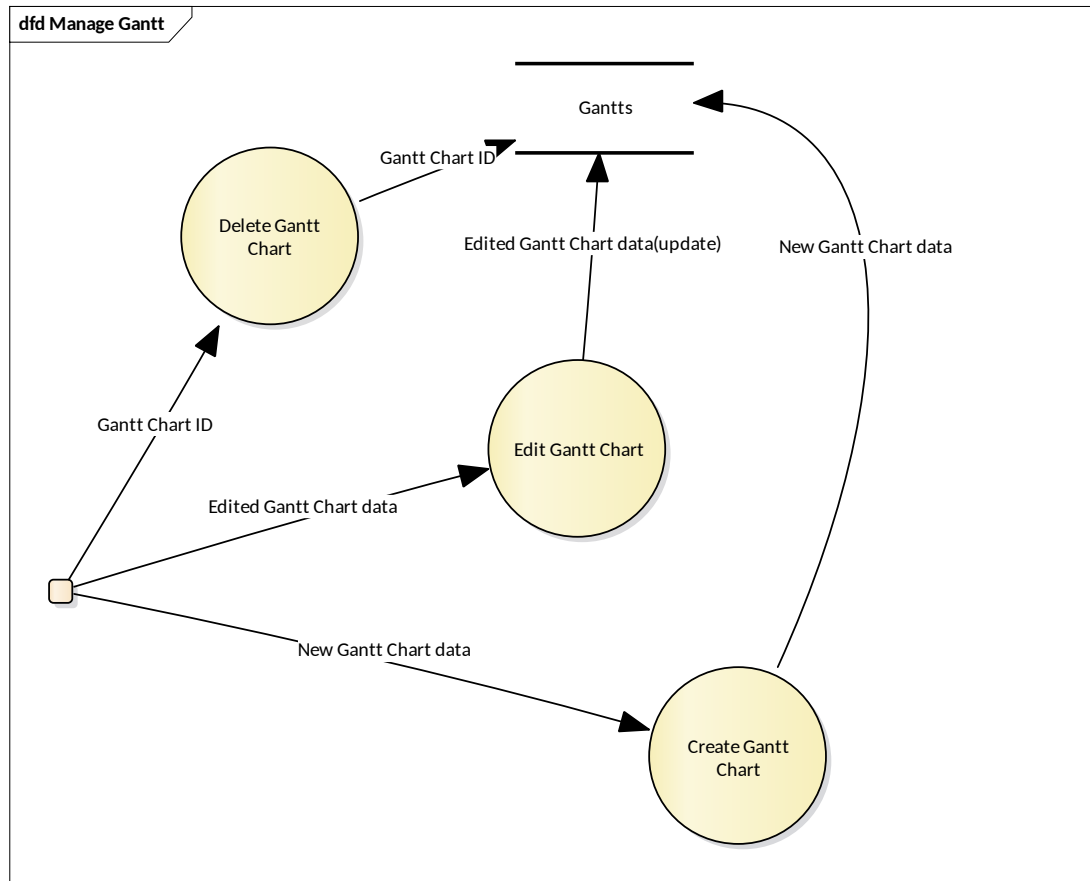


Figure 5.12: Level 1 DFD for Manage Gantt process

5.4 Wireframe for User Interface Design

The initial mock-up of the different screens within the application will be shown in this section, and is grouped into three different categories, namely student view, lecturer view and shared view. The actual implementation might be a bit different in terms of the layout and screens design, but still

5.4.1 Shared View

Shared view is the view or screens that will be shared among student and lecturer as both of the modules uses it for same purposes.

The screenshot shows the LEED Group Assignment Portal login page. At the top left, the word "LEED" is displayed. At the top right, there are navigation links: "About", "Code of Conduct", "Contact", and "Help". The main content area is centered and contains the following elements:

- Group Assignment Portal** (Section Header)
- User ID:
- Password:
- [Forgot Password?](#) (Link)

Figure 5.13: Login Page

The screenshot shows the LEED homepage. At the top left, the word "LEED" is displayed. At the top right, there are navigation links: "Assignments", "Courses", "Notification", "Book", and a user profile icon labeled "Username" with a dropdown arrow. The main content area is centered and contains the following elements:

- Welcome XXX.** (Text)
- This is the home page.** (Text)
- A large, light gray rectangular placeholder box below the text.

Figure 5.14: Homepage

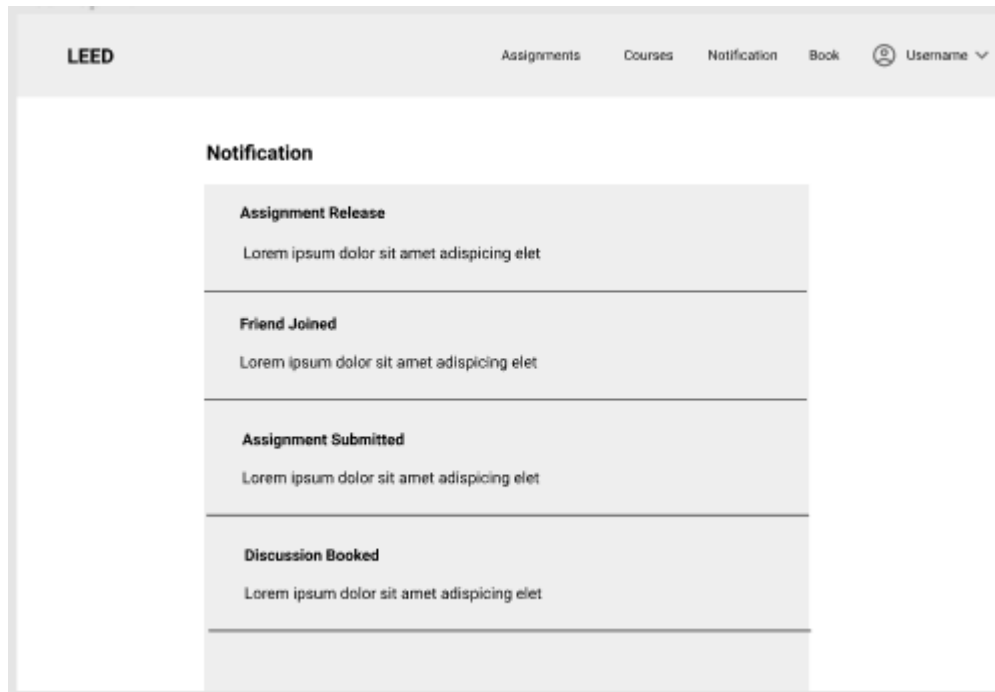


Figure 5.15: Notification Page

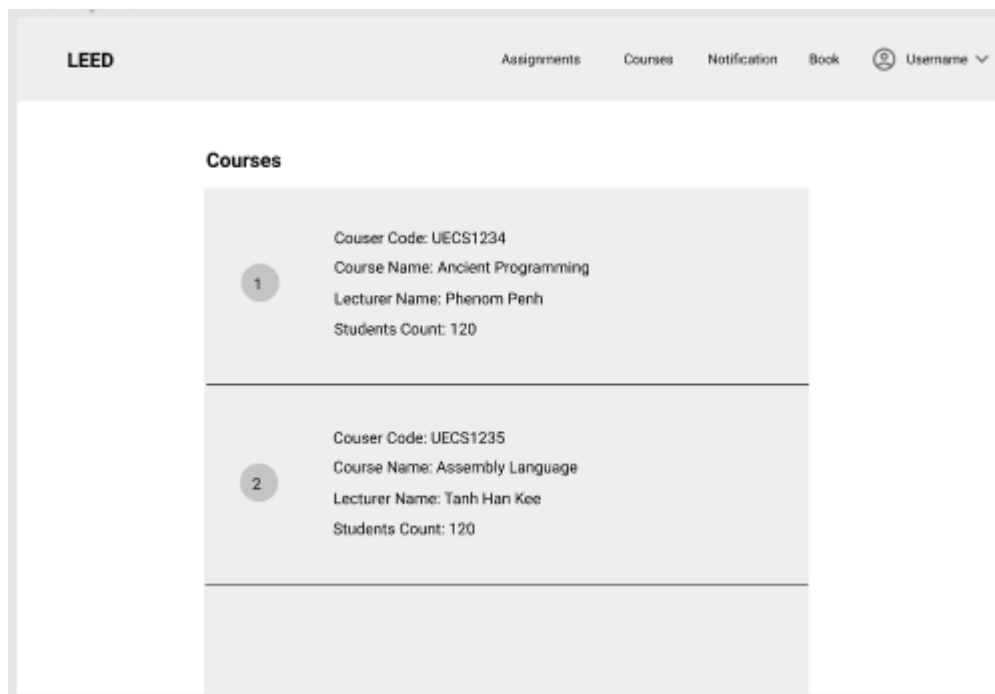


Figure 5.16: Course Listing Page

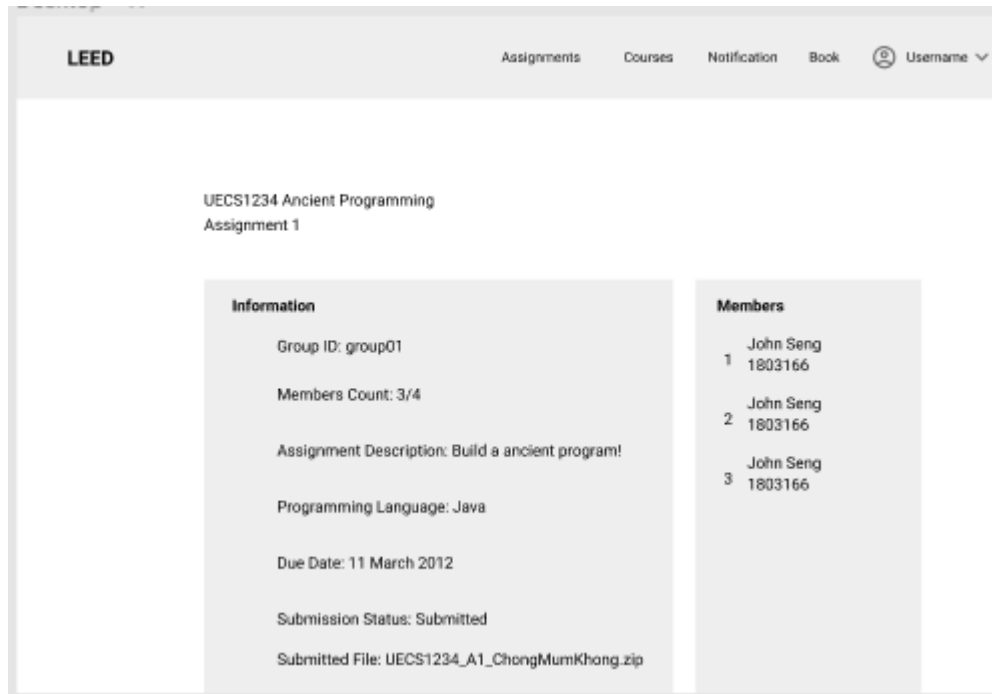


Figure 5.17: Group Info Page

5.4.2 Student View

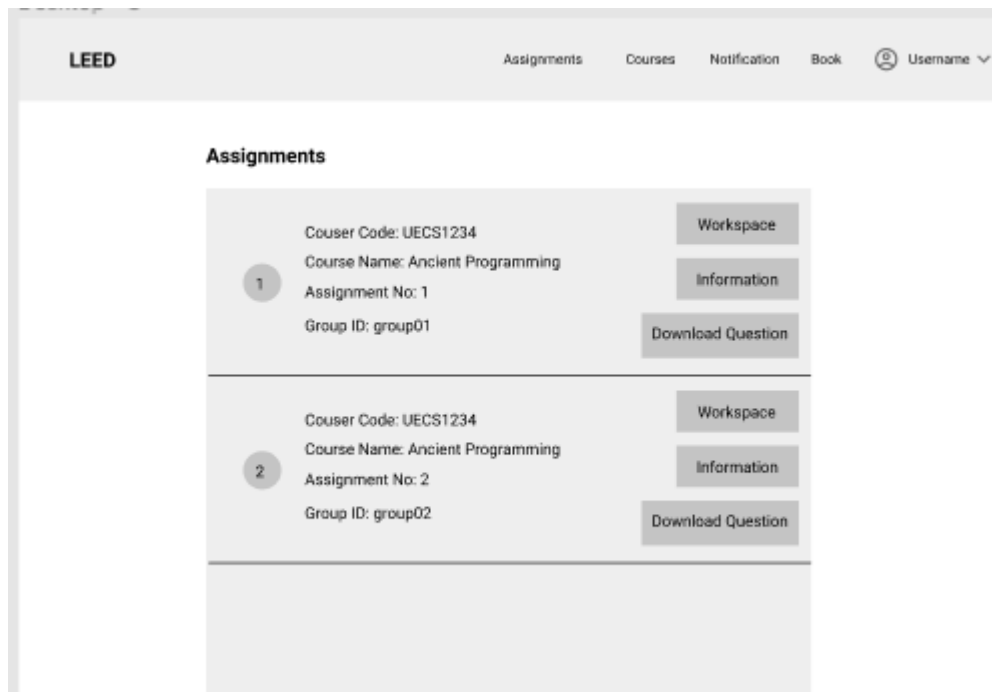


Figure 5.18: Assignments List Page

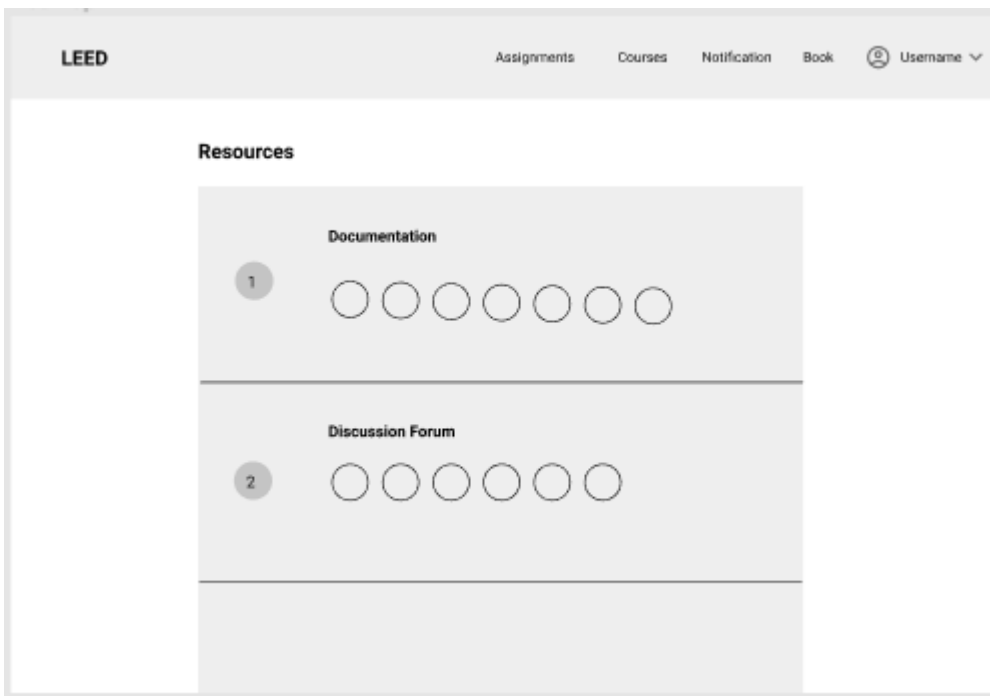


Figure 5.19: Resources Page



Figure 5.20: Group Workspace

LEED

Assignments Courses Notification Book Username

UECS1234 Ancient Programming
Assignment 1

New Task

No	Task	W1	W2	W3	Actions
1	Task number one				
1	Task number one				
1	Task number one				
1	Task number one				
1	Task number one				
1	Task number one				
1	Task number one				

Figure 5.21: Gantt Chart Page

LEED

Assignments Courses Notification Book Username

UECS1234 Ancient Programming
Assignment 1

Book Discussion

Next Discussion: Wednesday 4pm-6pm, 21 Feb

<p>John</p> <p>Monday: 4-9pm Wednesday: 0-12am</p> <p>Sammy</p> <p>Monday: 4-9pm Wednesday: 0-12am</p>

Figure 5.22: Discussion Booking Page

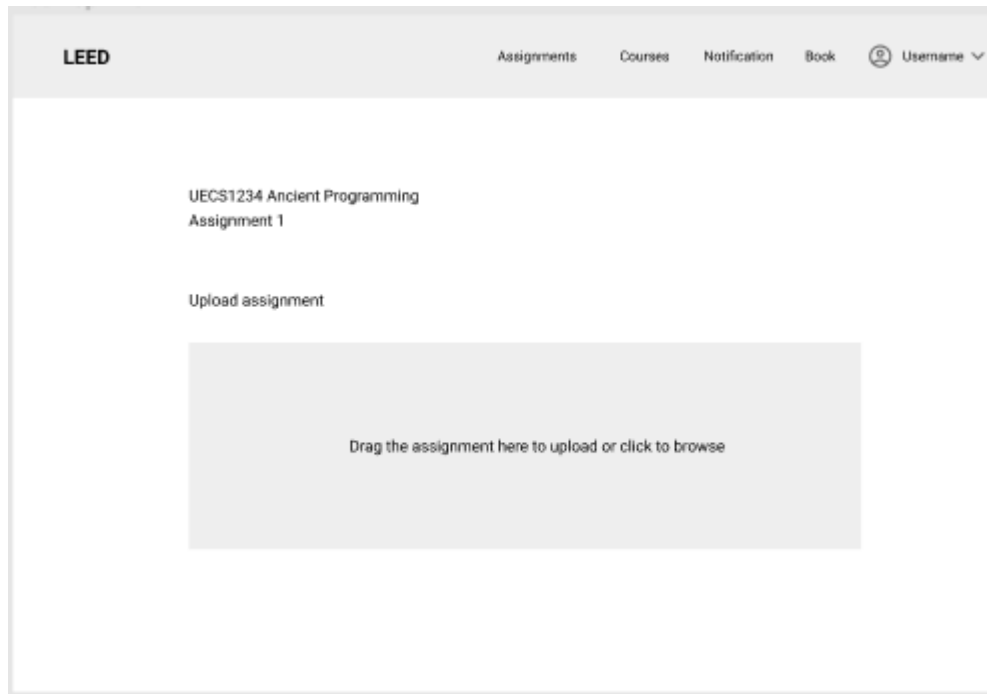


Figure 5.23: Submit Assignment Page

5.4.3 Lecturer View

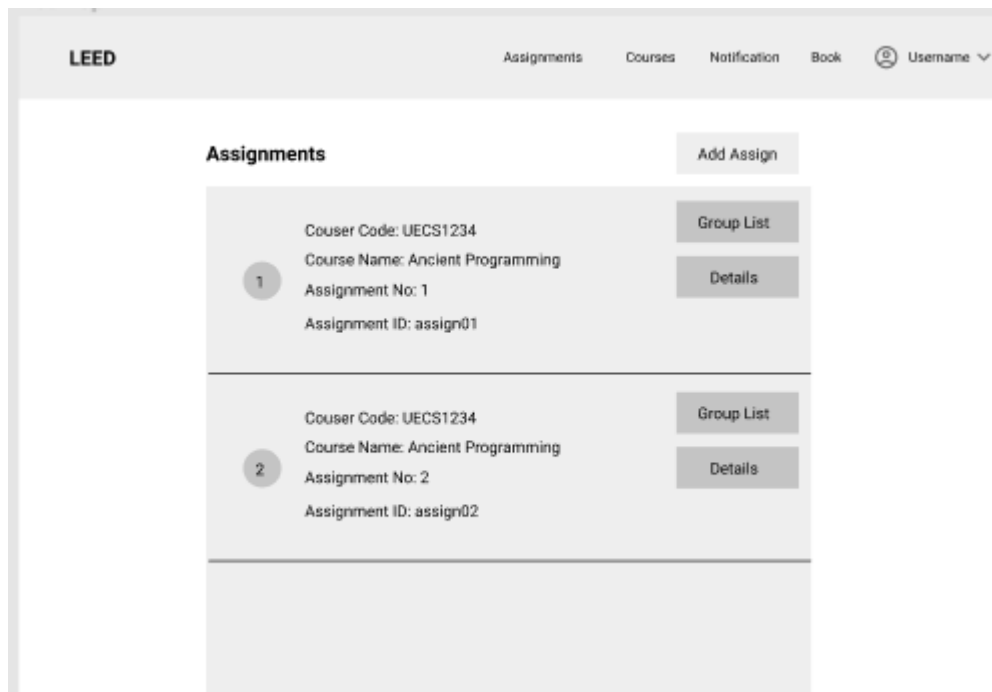


Figure 5.24: Assignments List Page

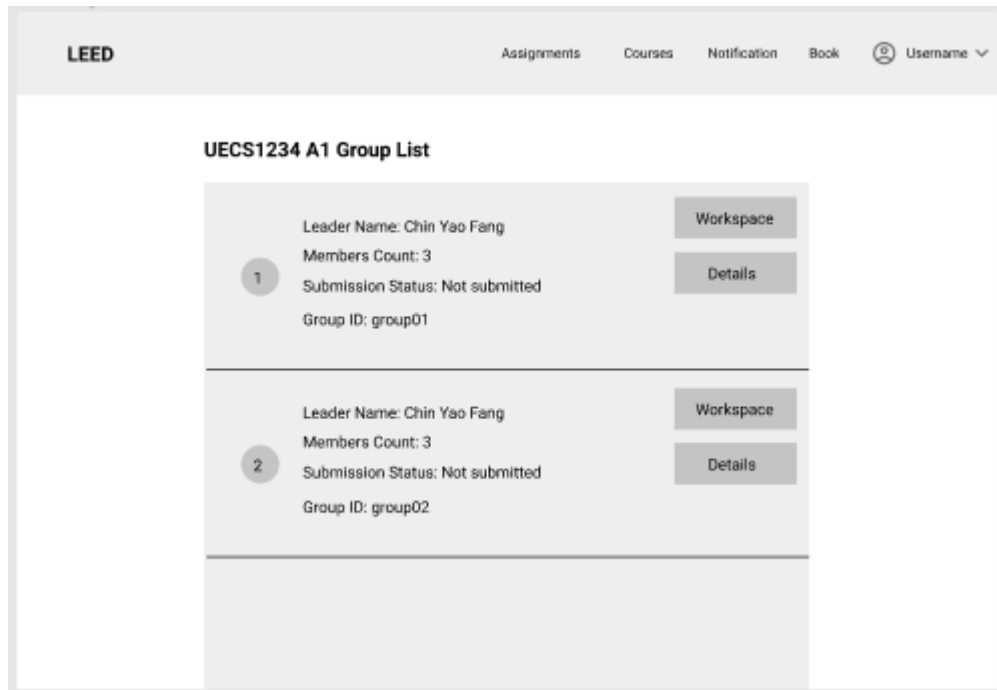


Figure 5.25: Assignment Groups List Page

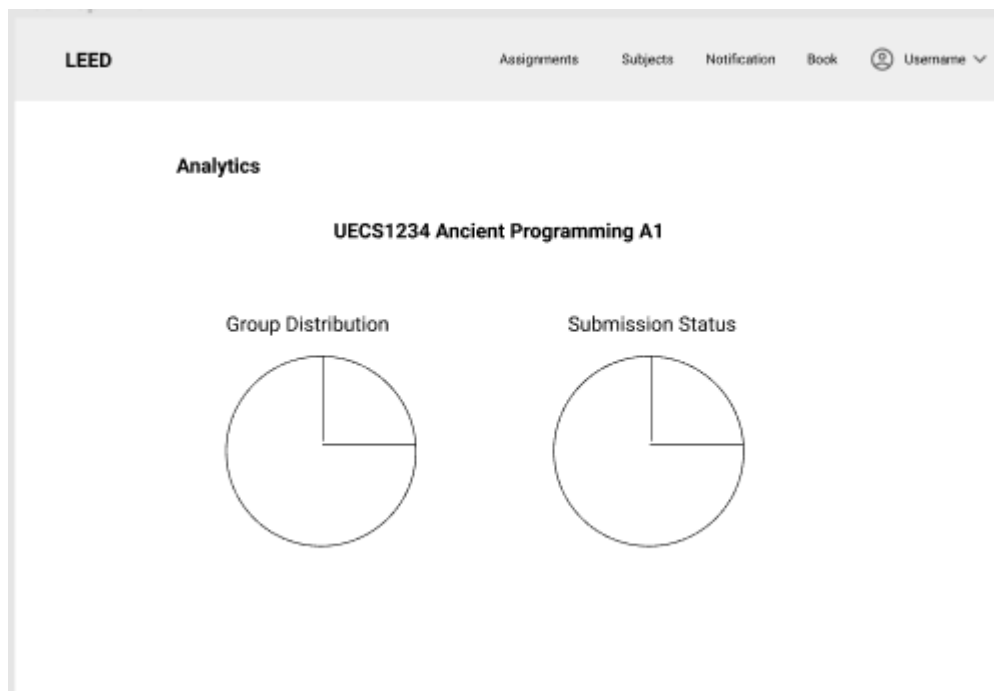


Figure 5.26: Analytics Page

The screenshot shows a web interface for LEED. At the top, there is a navigation bar with the LEED logo on the left and links for Assignments, Courses, Notification, Book, and a user profile icon labeled Username. Below the navigation bar, the page title is "UECS1234 Ancient Programming Assignment 1". The main content area features a grey box titled "Information" containing the following details:

- Group ID: group01
- Maximum student: 4 (with an edit icon)
- Assignment Description: Build a ancient program! (with an edit icon)
- Programming Language: Java (with an edit icon)
- Due Date: 11 March 2012 (with an edit icon)
- Upload question: Choose a file (with a button)
- Uploaded File: UECS1234_A1_Question.pdf

Figure 5.27: Assignment Info Page

5.5 Page Navigation Flow (Site Map)

5.5.1 Student Flow

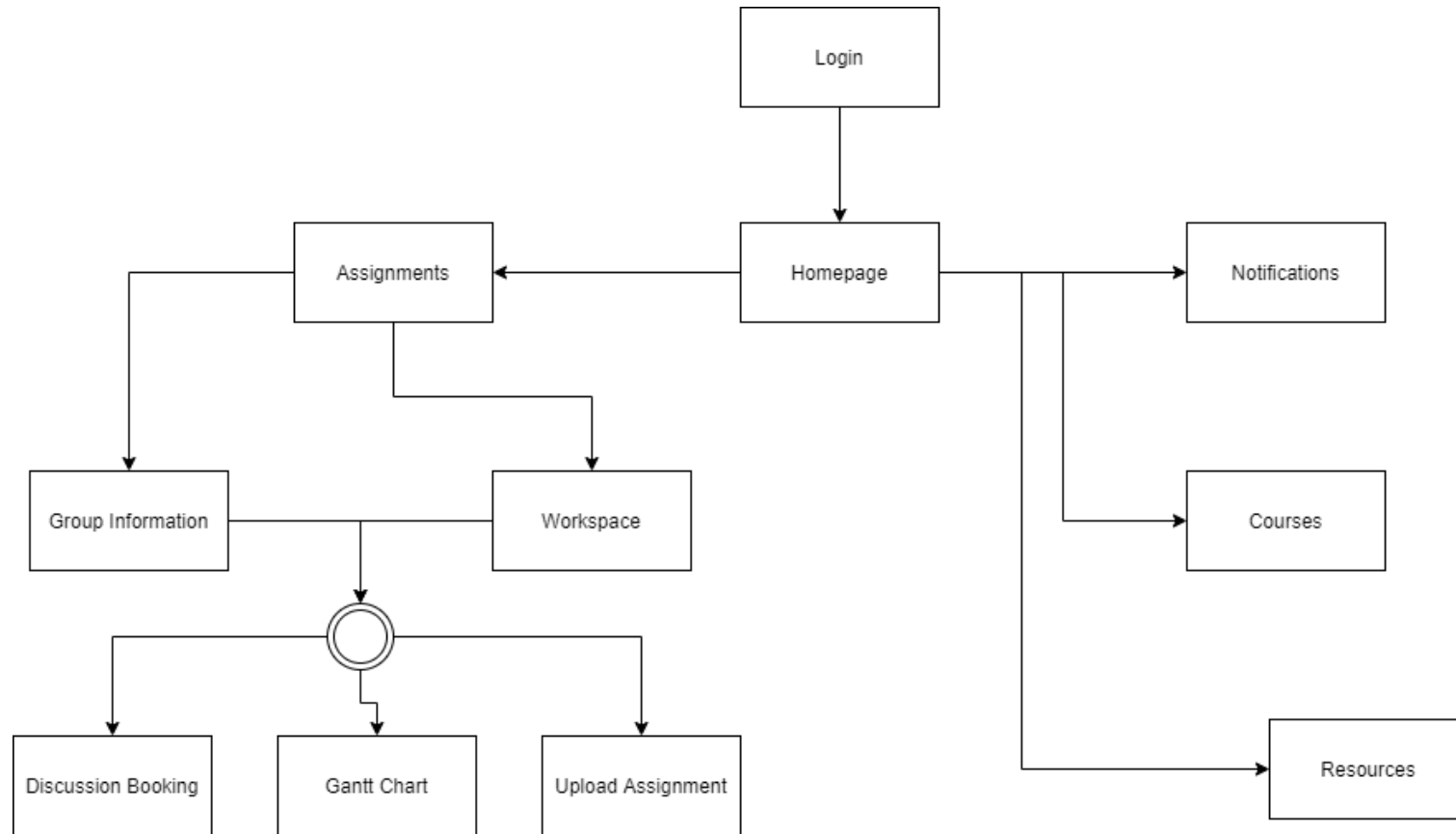


Figure 5.28: Site Map for Student Module

5.5.2 Lecturer Flow

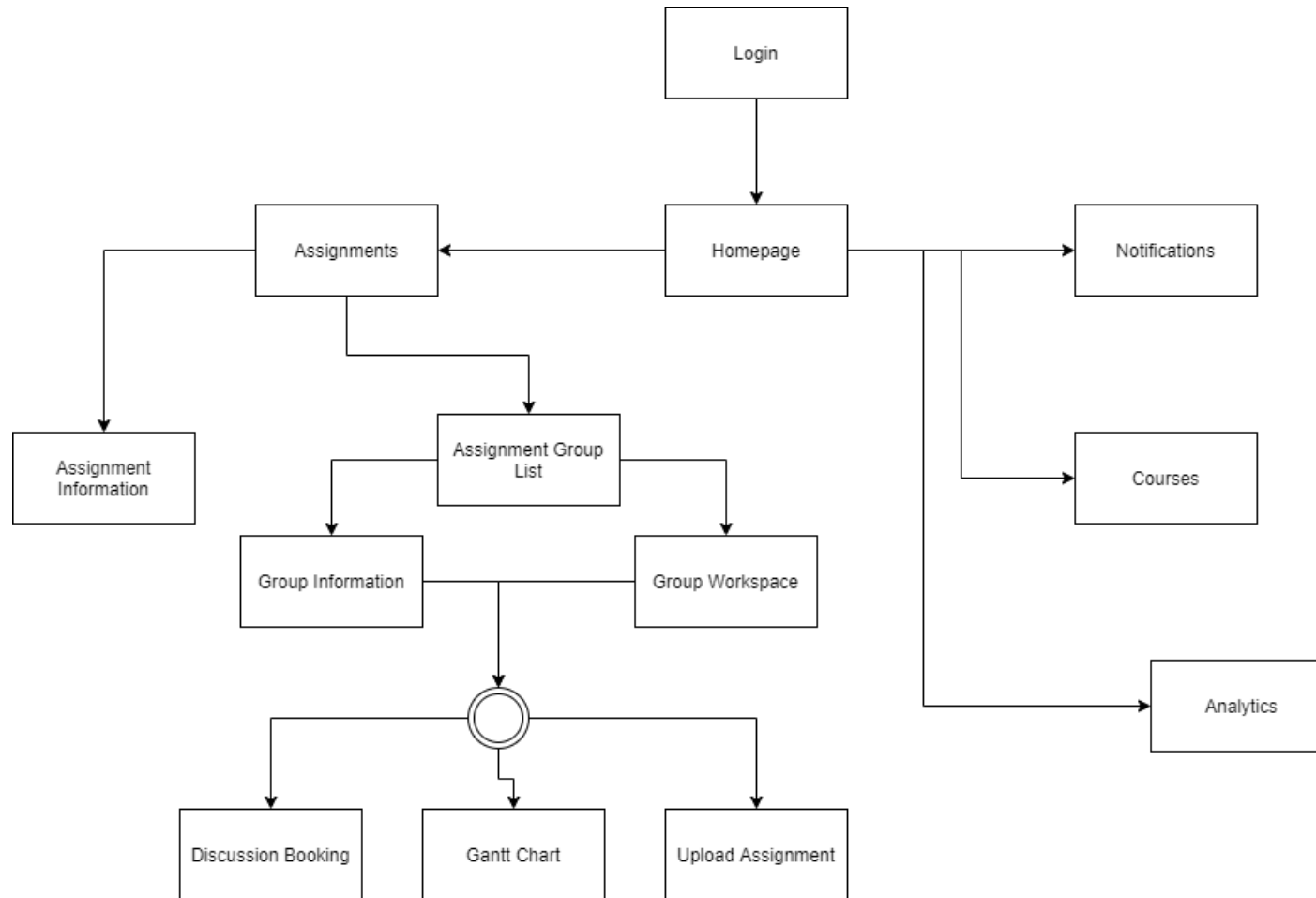


Figure 5.29: Site Map for Lecturer Module

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

This chapter will outline each aspects and steps of the actual implementation of the system.

6.2 Project Setup

First of all, two separate empty folders were created. They are the repositories for front-end and back-end server. The repository name for the front-end is called “leedv2.1” whereas the name for the back-end is called “leed-backend”.



Figure 6.1: NPM (Node Package Manager)

Node Packet Manager is a command-line tool that comes along with Node.js installation that is extremely useful to pull libraries or tools that others created from the internet into local machine with just one line of command. NPM was used to install @vue/cli for Vue.js command line tool for front-end development and TypeScript and Express.js for back-end development separately.

```
npm Install <package_name>
```

After that Git is initiated in both of the repositories and committed for the first time. The repositories were then pushed into a remote server called GitHub to provide backup and backtracking for the project development.

6.3 Front-end Development

6.3.1 Application of UI/UX Theories

When it comes to design and implement the actual screens for the web application, numerous UI/UX theories and trends was taken into account in order to provide the best experience in terms of aesthetic value and ease of use to most of the users as the user retention is highly influenced by their experience and the value that they get from using the system. For example, space theory including white space (also known as negative space), active space and passive space is applied extensively within the application to make it tidy and less crowded.

Consistency is also heavily practised throughout the application for the colour schemes, font sizes, font families and element arrangement. However, there might be the case where the consistency rule is not to the point, but generally everything is kept as consistent as possible.

As a bad designer, it would take an indefinite amount of time to design a user-friendly interface from scratch. Hence, the Twitter's user interface is taken for reference as the backbone for the UI design of this project.

6.3.2 Login Page

For the users to authenticate themselves and use the system, a login page is necessary, in this case, the login form is provided in the main/root page for the system. Fallback page and access denied page are used to handle eccentric routes.

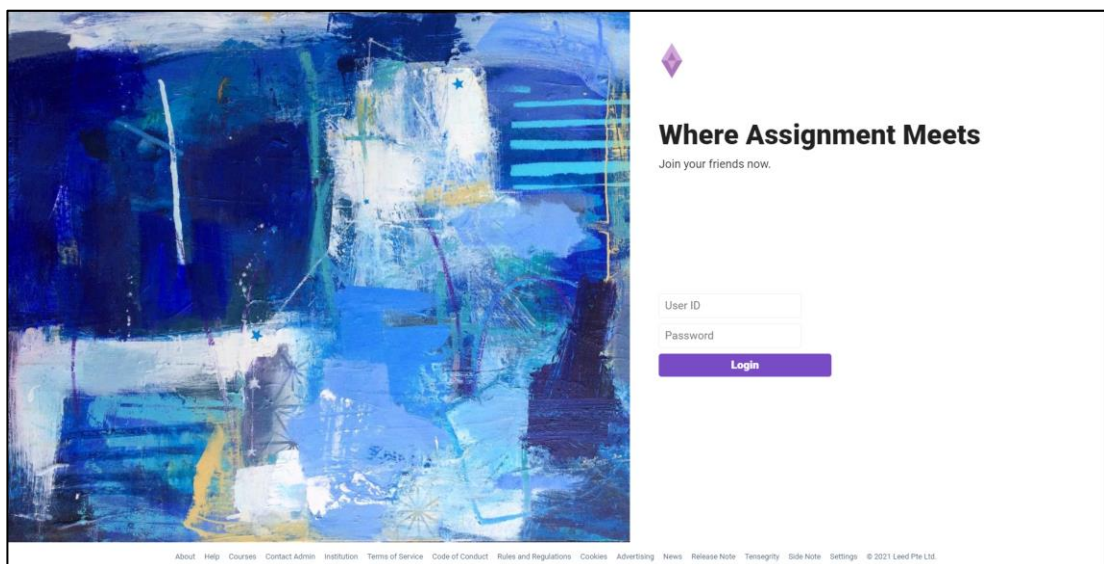
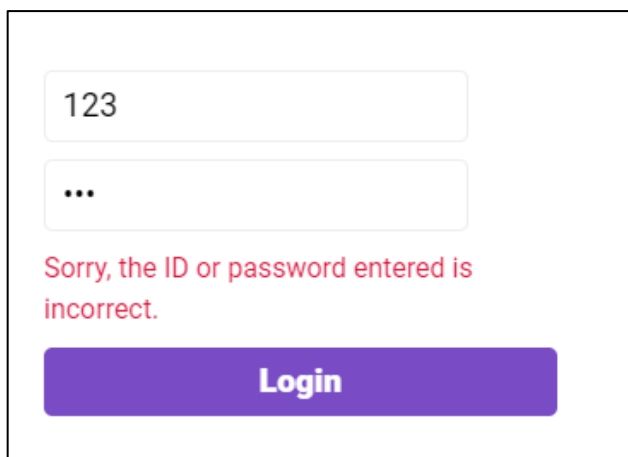


Figure 6.2: Main Login Page



A login form with two input fields. The first field contains the text '123'. The second field contains three dots '...'. Below the fields is a red error message: 'Sorry, the ID or password entered is incorrect.' At the bottom is a purple button with the text 'Login'.

Figure 6.3: Error message for unmatched credentials

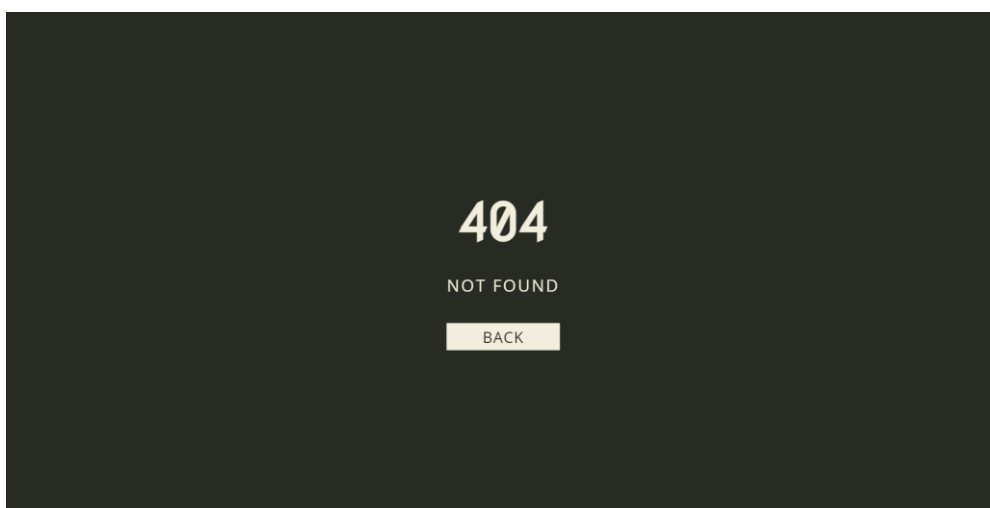


Figure 6.4: Fallback Page

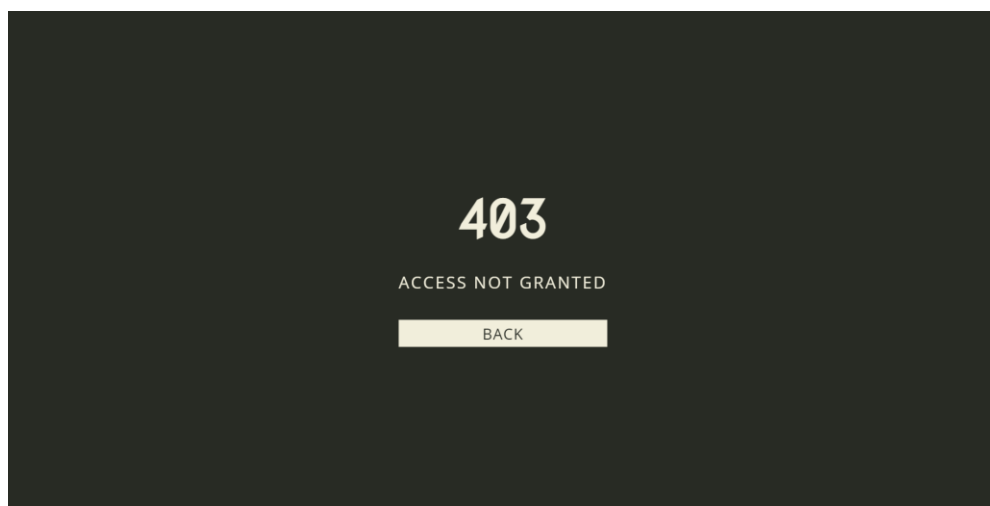


Figure 6.5: Not Authorized Page

6.3.3 Student Module

6.3.3.1 Homepage

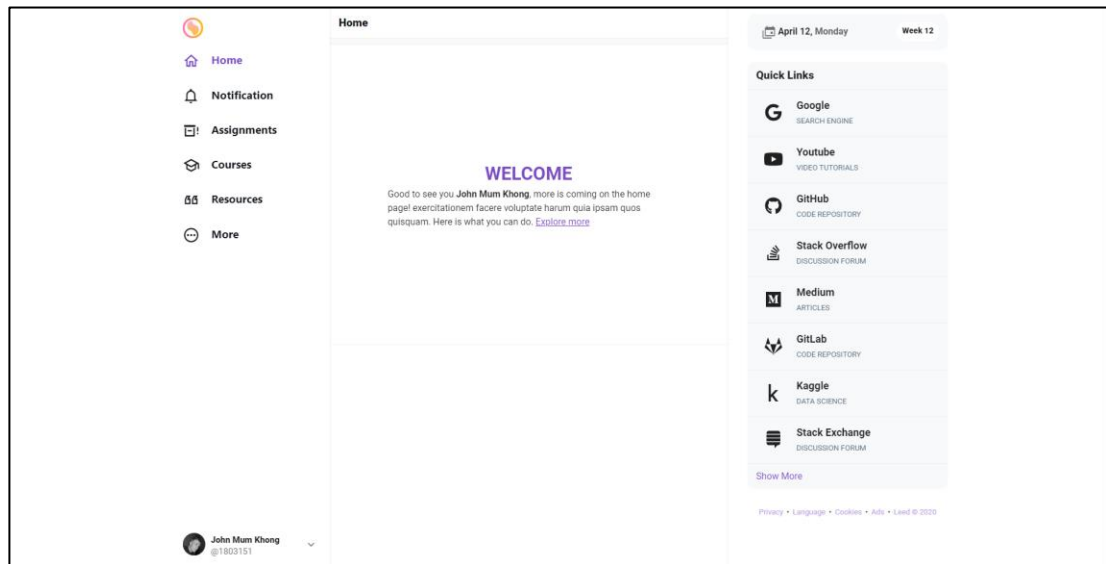


Figure 6.6: Homepage

After logging into the system as a student, the welcome page is first presented onto the screen. The left part contains the navigation panel and profile card for logout, middle section is the content section in which the content will change when other page is visited and lastly the miscellaneous panel that provide helpful quick links to popular websites for learning and programming related manners.

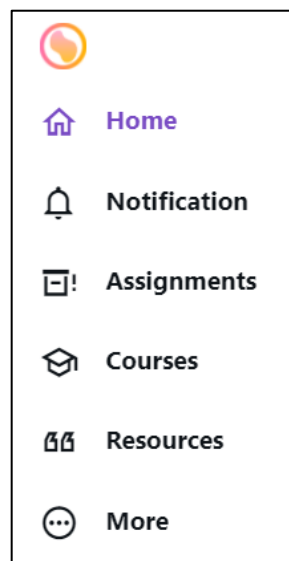


Figure 6.7: Navigation Bar

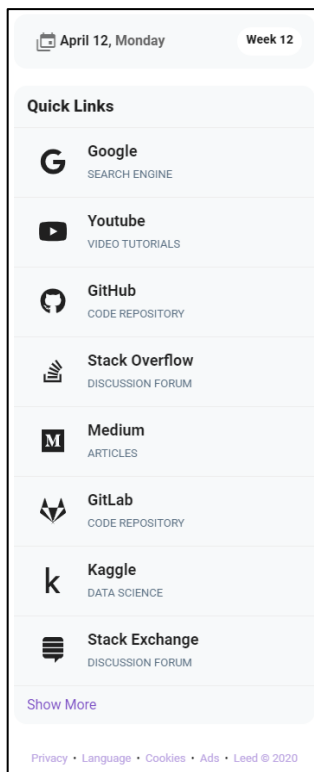


Figure 6.8: Miscellaneous Panel

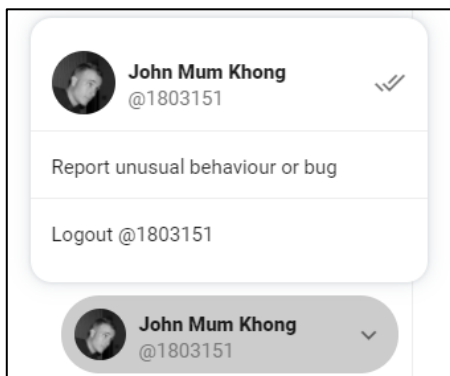


Figure 6.9: User Panel

6.3.3.2 Notification Page

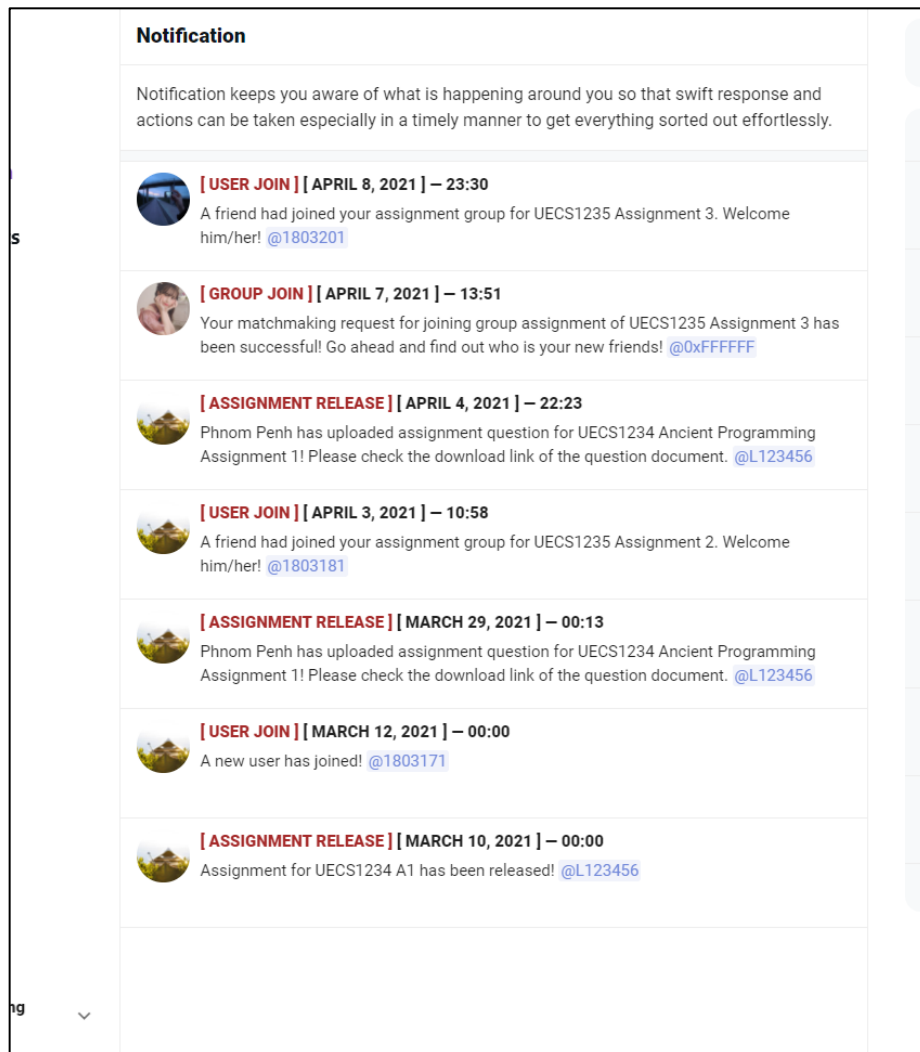


Figure 6.10: Notification Page

Notification page contains the recent notifications that a user has, with a maximum of 25 notification items presented only. The reason for such implementation is to prevent the notifications to flood the page if the user has over a hundred or thousand notifications. This will severely slow down the browser and cause unfavourable experience to users.

The title of the notification, coloured with brown indicated the type of activity or action that caused such notification, followed by the date and time it is initiated. Currently, there are 5 supported types of notification, namely assignment question upload, assignment submit, new student join, successful matchmake and confirmation of discussion time.

6.3.3.3 Courses Page

Courses

All the programming courses that is enrolled is shown on below. Click on the individual course to view the details of it.

UECS1234 Ancient Programming

Course Description
 Legacy system support is crucial skill to have. Mainframe computing talent is on high demand.

Lecturer
 Phnom Penh

Difficulty
 Easy

Students Count
 120

UECS1235 Software Testing

Course Description
 This subject is about 100 ways to test a project. Penetration with brute force testing is also covered.

Lecturer
 Phnom Penh

Difficulty
 Easy

Students Count
 12

Figure 6.11: Courses Page

Courses Page shows all the programming courses enrolled by a student on the semester. The details such as course description, lecturer's name and the total students enrolled is shown.

6.3.3.4 Resources Page

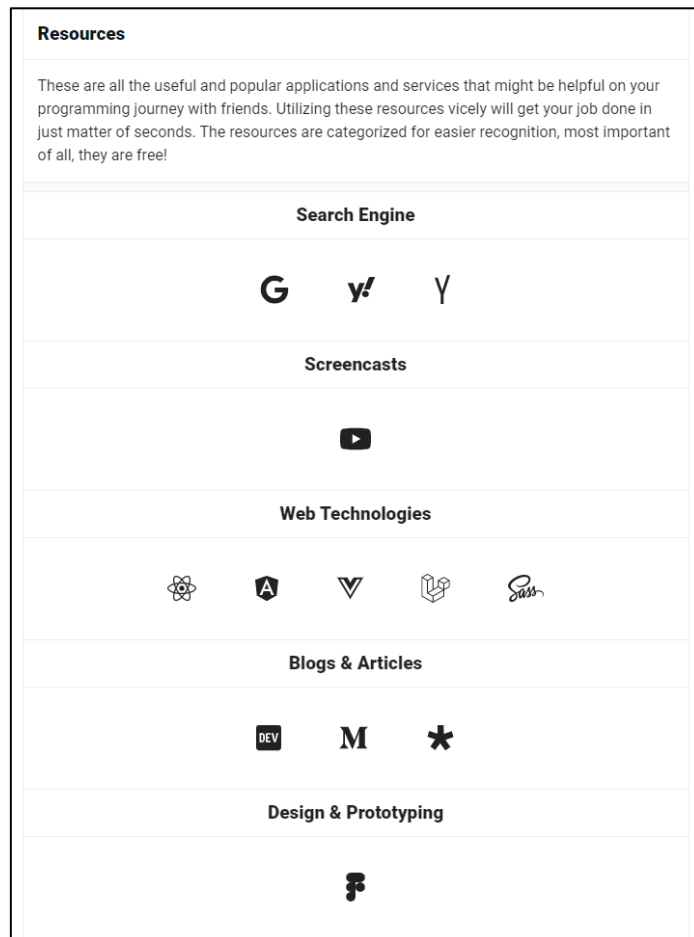


Figure 6.12: Resources Page

Resources page is a comprehensive list of useful websites gathered that is deemed to bring value for students on their programming journey. It is the expanded version of the miscellaneous panel. Students can use the shortcut to the webpages provided by clicking on their icon. A new tab will be launched with the desired webpage. This is the extra feature provided that was not being mentioned much in other parts of the proposal and report.

6.3.3.5 Assignments Page

Assignments		
The below are the assignments of the enrolled courses that are available for this semester. To direct to the workspace for the assignment, click on the button. To navigate to the assignment and group details, click on the star button.		
UECS1234 Ancient Programming 1	★	ⓘ ↓
UECS1234 Ancient Programming 2	★	ⓘ ↓
UECS1235 Software Testing 1	★	ⓘ ↓
UECS1235 Software Testing 3	★	ⓘ ↓
UECS1235 Software Testing 2	★	ⓘ ↓

Figure 6.13: Assignments List Page for Students

Assignments list page shows a list of assignments that a student has in the semester. It is based on the courses that they have enrolled. If a course take is taken by the student is having a new assignment, the new assignment will appear here.

The star icon represents the workspace of the student where they can interact with their teammates through real-time code editor and chats. The empty star icon represents the student does not have a group for that particular assignment yet. The information icon will lead to a page that shows all the essential details for the group and assignment. If a student does not have a group yet for the assignment, he/she will not be able to access to the information page as well. The download icon is the download link to the assignment question file uploaded by the lecturer. The empty download icon shows that the lecturer has yet to upload the question file.

Student can click on the empty workspace icon to either create, join or matchmake group for the same assignment.

Ways for New Group

Join a group
Please provide a valid group ID to join.

Group Id

@ Join

Create a group
You will be the leader for this group.

+ Create

Matchmake with others
The system will arrange your group.

Matchmake

Figure 6.14: Popup for New Assignment Group

Join a group
Please provide a valid group ID to join.

Group Id

@ Join

Group ID cannot be empty!

Figure 6.15: Error Message for Empty Group ID

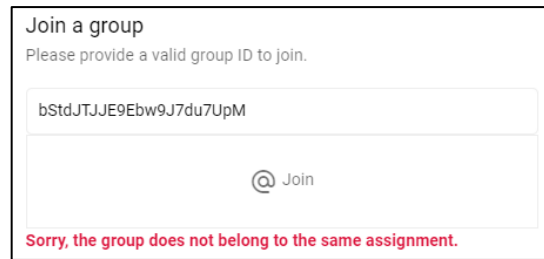
Join a group
Please provide a valid group ID to join.

ffsfasdf

@ Join

Sorry, the id does not belong to any group.

Figure 6.16: Error Message for Non-existing Group

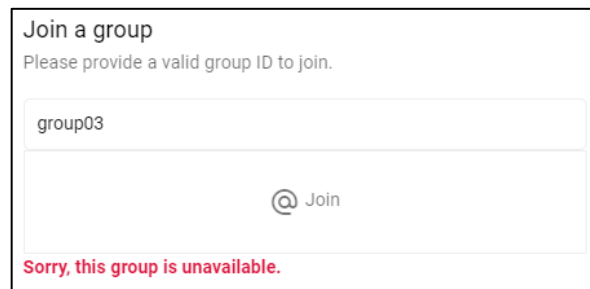


Join a group
Please provide a valid group ID to join.

Sorry, the group does not belong to the same assignment.

Figure 6.17: Error Message for Wrong Assignment

When students try to join the assignment group that does not belong to the same assignment title, they will not be able to do it. The logics for all the checking is done on the back-end server.



Join a group
Please provide a valid group ID to join.

Sorry, this group is unavailable.

Figure 6.18: Error Message for Unavailable Group

Moreover, when students tried to join a group that is unavailable, due to the group is full or the group members had explicitly prohibit other people to join, the error message on above will be shown.

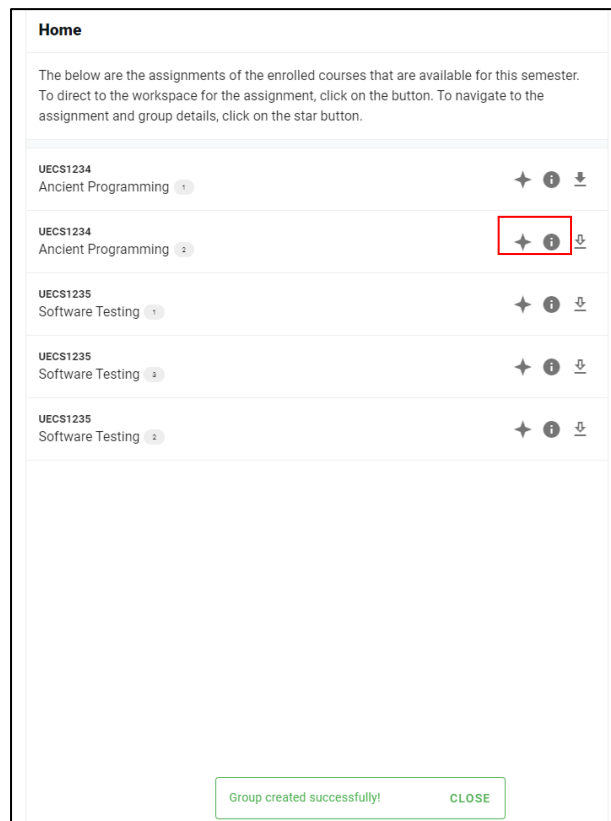


Figure 6.19: New Group Created

The above figure shows the screen when the ‘Create’ button is pressed. Successful creation of group is indicated by the snackbar below the screen that reads ‘Group created successfully!’ along with the workspace and information icon that changed from outlined into filled.

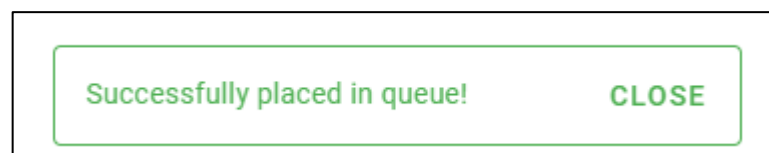
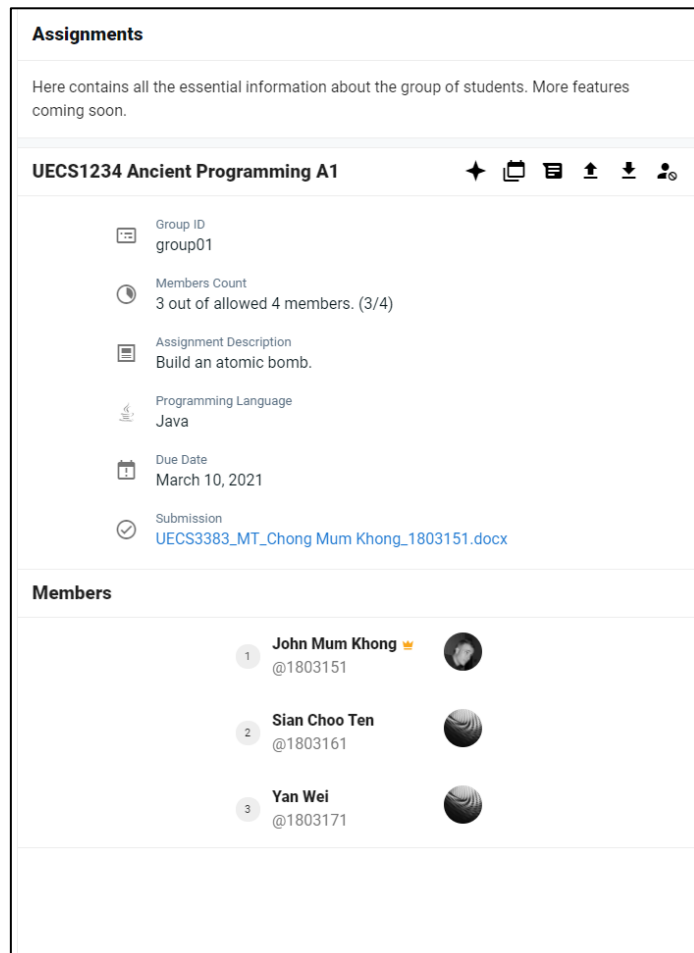


Figure 6.20: Snackbar for Matchmake Request Successful

The matchmaking mechanism does not provide immediate feedback as it is waiting for other students to fill in the queue as well. Once the number of students for the assignment is equal to the maximum allowed group members, matchmake happens and a notification and email will be sent. Student might need to refresh the page to see the effect.

6.3.3.6 Group and Assignment Info Page



Assignments

Here contains all the essential information about the group of students. More features coming soon.

UECS1234 Ancient Programming A1 ✦ 📄 📋 ⬆️ ⬇️ 👤

- Group ID: group01
- Members Count: 3 out of allowed 4 members. (3/4)
- Assignment Description: Build an atomic bomb.
- Programming Language: Java
- Due Date: March 10, 2021
- Submission: [UECS3383_MT_Chong Mum Khong_1803151.docx](#)

Members

- John Mum Khong** 👑 @1803151
- Sian Choo Ten** @1803161
- Yan Wei** @1803171

Figure 6.21: Group and Assignment Info Page

This page shows the necessary information for the group and assignment such as group ID, total members, assignment description, programming language used, due date of the assignment and the submission status for this group assignment. It also shows the info of the members within the group whereby the leader is sided by a mini yellow crown icon.

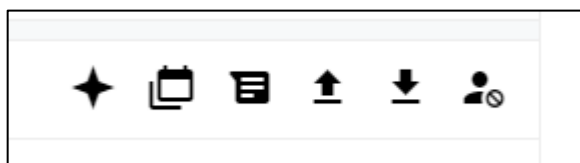


Figure 6.22: Actions Shortcut

There is a shortcut bar beside the course title which allows students to perform various actions. The first one is the shortcut to join the workspace in case

the student is not on the assignment list page. This redundancy is part of the UI/UX design where it reduces the clicks of the user to perform an action. Imagine if this button is not available, the student will need to click back to the original list page and then click on the workspace icon to navigate into the workspace. That might not much, but it is enough to upset a user.

The second button is used to go to the Gantt Chart Page. The third button is for the discussion booking session. The upload icon is where only group leader is accessible. It is used to submit the assignment to the system. The download icon is used to download the assignment question file uploaded by the lecturer. Lastly, the right most icon indicates the status of the group whether they accept new member to join or not. They can toggle the icon to switch on and off for that.

6.3.3.7 Group Workspace

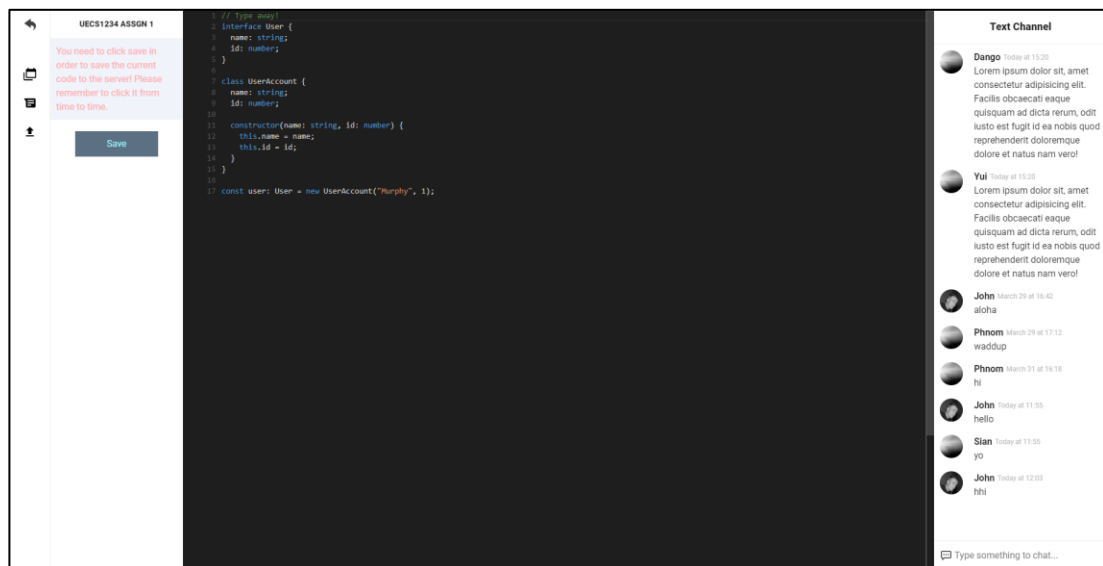


Figure 6.23: Group Workspace

In the workspace, group members can code collaboratively in real-time and discuss among each other with the chat provided. They can access the Gantt Chart page, discussion booking page and leader-exclusive upload page from here as well through the links on the sidebar.

The limitation for this workspace is that the codes are not saved automatically to the server due to the difficulties for implementing the real-time environment. There are limited resources for implementing such features as a good real-time

application needs to take network latency, cursor positioning and a well-established algorithm to handle the ins and outs, which is arduous for my coding experience.

To mitigate this issue, a save button was introduced for the students to manually save their work whenever they reach a certain milestone or before they leave the workspace.

6.3.3.8 Gantt Chart Page

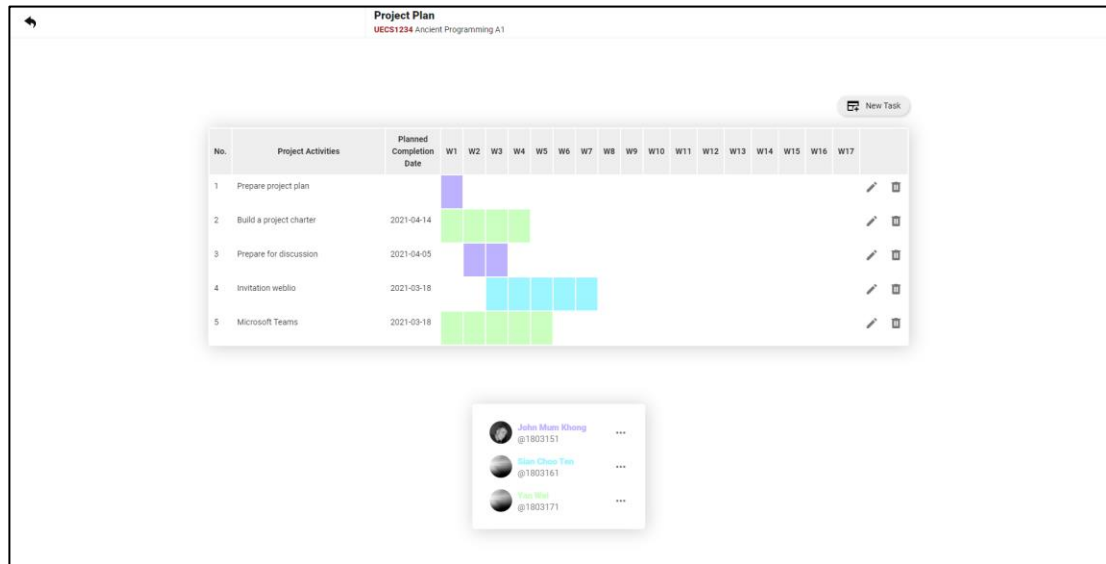


Figure 6.24: Gantt Chart Page

In this page, students can see the tasks that they have planned for this assignment. The tasks or activities are colour-coded with the member's name to show that who is in charge of which tasks. The students can add a new task, edit a task and delete a task.

Add a New Activity

Activity

From week To week Assignee

Self-set deadline

2021
Thu, May 27

< May 2021 >

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

[CLOSE](#) [CONFIRM](#)

Figure 6.25: Popup for adding a new activity

6.3.3.9 Discussion Booking Page

Discussion Booking
UECS1234 Ancient Programming A1 [Add Booking](#)

All members are required to provide suggestions for result!

✓ User has placed his/her bookings
 John Mum Khong @1803151 · March 12, 2021

- Sunday available at 4 – 12
- Tuesday available at –
- Monday available at –
- Wednesday available at –
- Thursday available at –
- Friday available at –
- Saturday available at –

Figure 6.26: Discussion Booking Page

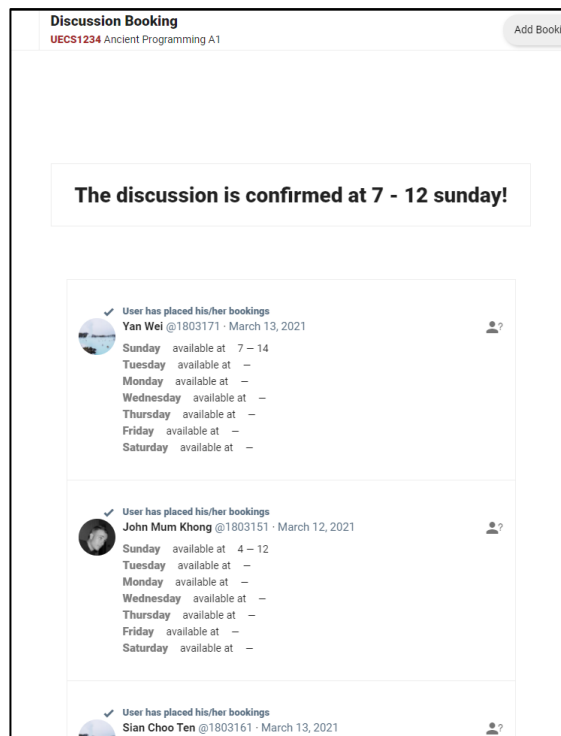


Figure 6.27: Results shown when all members placed their bookings

Booking Discussion Time

Please select a day and a timeframe that you are available to proceed.

Day	Start time	End time
Sunday ▼	0	24

CANCEL
CONFIRM

Figure 6.28: Popup for adding a new discussion time

6.3.3.10 Assignment Submission Page

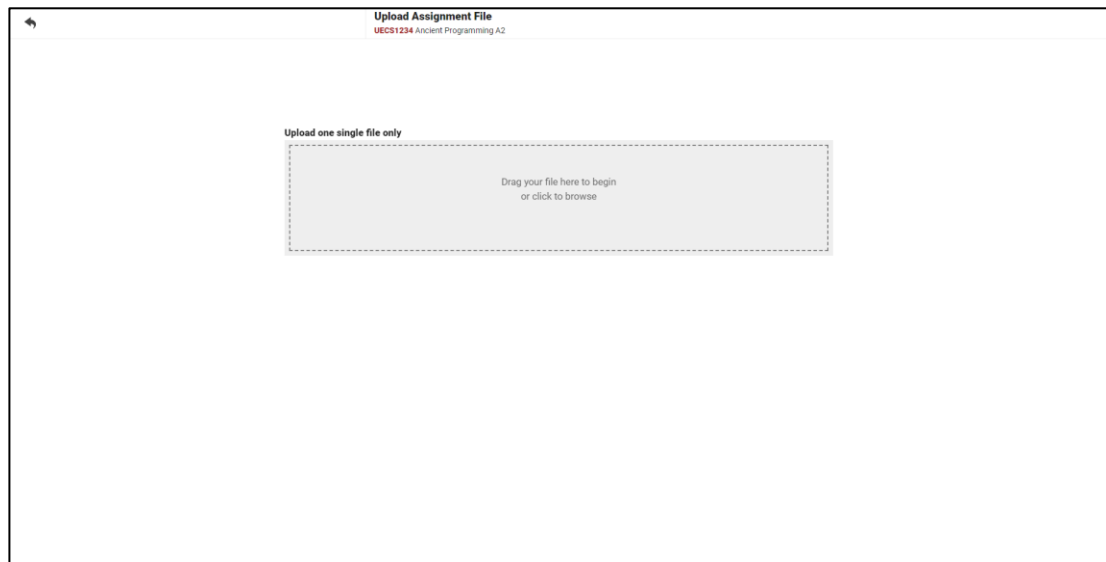


Figure 6.29: Assignment Submission Page

Students can either drag and drop their assignment file or click on the drop box to browse manually.

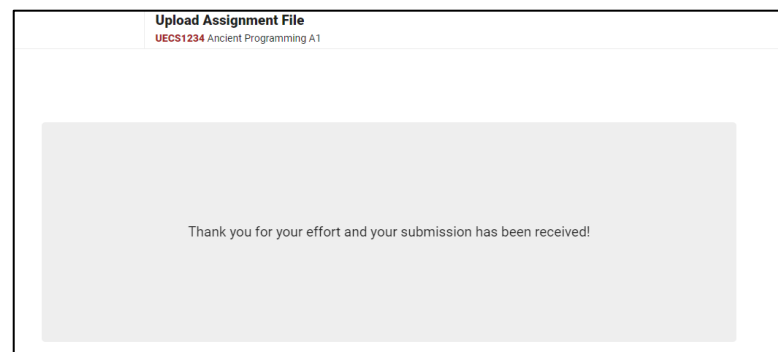


Figure 6.30: Message for Successful Submission

6.3.4 Lecturer Module

6.3.4.1 Homepage

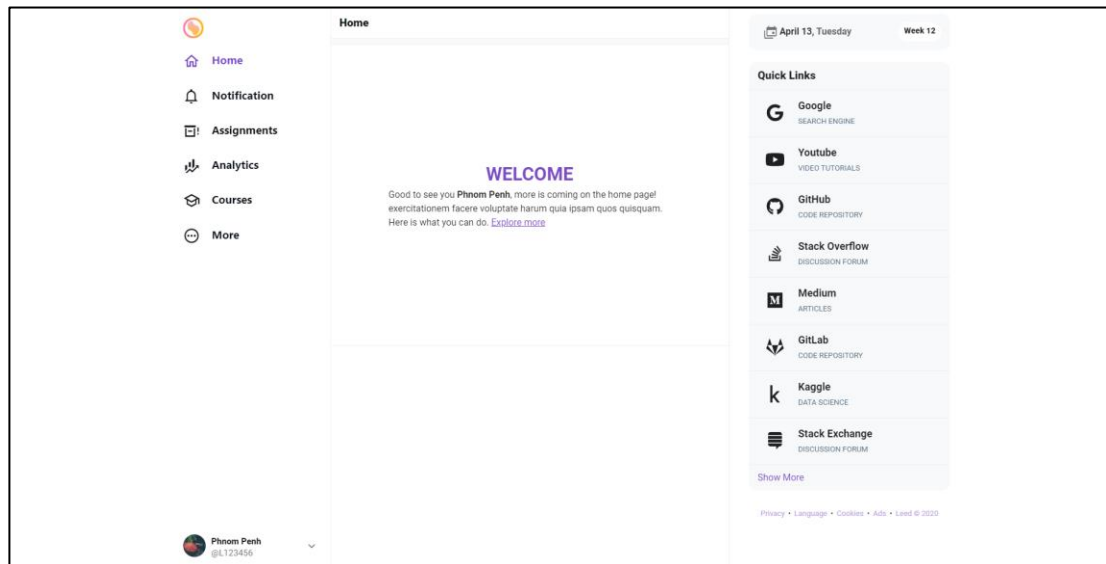


Figure 6.31: Homepage

There are not much different for lecturer's homepage compared to student's homepage. The only difference is item in the navigation bar. Lecturer does not have Resources Page but instead have Analytics Page.

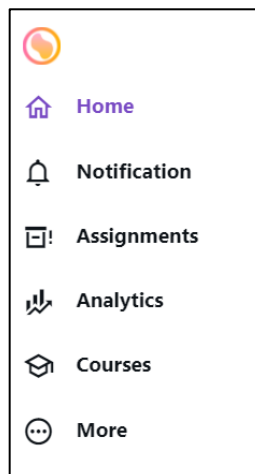


Figure 6.32: Navigation Bar

6.3.4.2 Notification Page

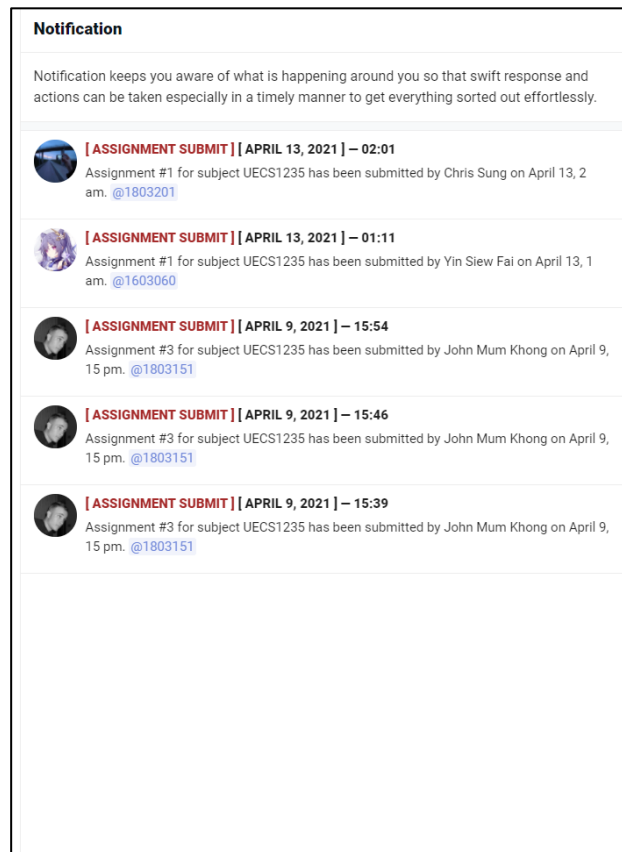


Figure 6.33: Message for Successful Submission


Currently, there are not much notification type for lecturer. Lecturer will only be notified on assignment submission by the team leaders only. More type of notifications can be added in the future.


6.3.4.3 Courses Page


Courses


The below are a listing of subjects being taught by you on the current semester. There are a few actions available, please explore them accordingly.

UECS1234 Ancient Programming


Course Description
 Legacy system support is crucial skill to have. Mainframe computing talent is on high demand.


Lecturer
 Phnom Penh


Difficulty
 Easy

Students Count
 120

UECS1235 Software Testing

Course Description
 This subject is about 100 ways to test a project. Penetration with brute force testing is also covered.

Lecturer
 Phnom Penh

Difficulty
 Easy


Students Count
 12
















Figure 6.34: Courses Page

The courses page is identical to the student's courses page. However, the courses here represent the subjects that the lecturer is in charged of in the semester.

6.3.4.4 Assignments Page

Assignments

The below are the assignments of the enrolled courses that are available for this semester. To direct to the workspace for the assignment, click on the button. To navigate to the assignment and group details, click on the star button.

<p>UECS1234 Ancient Programming </p>	 
<p>UECS1234 Ancient Programming </p>	 
<p>UECS1235 Software Testing </p>	 
<p>UECS1235 Software Testing </p>	 
<p>UECS1235 Software Testing </p>	 


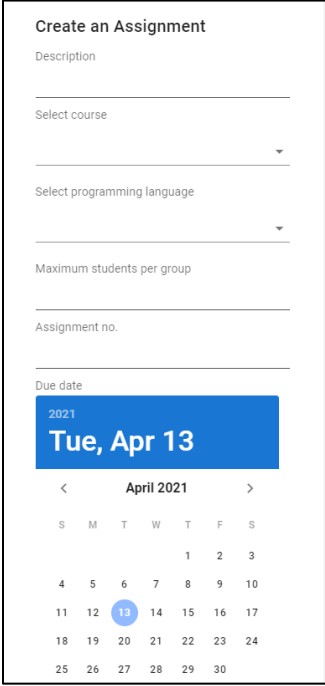
 Add an Assignment

Figure 6.35: Assignments Page

The assignments page shows each assignment that a lecturer has released on behalf of the courses that they in charge. The group icon will navigate lecturer into the list of groups formed for that particular assignment whereas the information icon will bring lecturer into the assignment's detail page.

Lecturers can add a new assignment for any of the course that they in charge via a popup shown below.



Create an Assignment

Description

Select course

Select programming language

Maximum students per group

Assignment no.

Due date

2021
Tue, Apr 13

< April 2021 >

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Figure 6.36: Add Assignment Popup






Lecturer first needs to specify the description of the assignment on what is the assignment about. After that, proceed to select a course for the new assignment, select a programming language, specify maximum students are able to team up in a group, current assignment number, and finally select a due date via the date picker for the assignment.

6.3.4.5 Assignment Detail Page

Assignments

This is a author view of the assignmnet details page. Lorem ipsum dolor sit amet, adispicing valet.

UECS1234 Ancient Programming A1

	Assignment ID assign1
	Description Mainframe Computing with COBOL and Java.
	Due Date 2021-03-09
	Programming Language Java
	Group Capacity 4

Assignment Question File

AWAD_Assignment (1).docx

Assignment upload: No file chosen

Figure 6.37: Assignment Detail Page

Lecturer can view the details that he/she set for this assignment and even edit the details in this page. After suitable changes are applied, simply click on the save button will commit the changes made for the assignment to the server.

Lecturer will have to upload assignment question file via this page as well. After clicking the choose file button, a file explorer will be shown. Once the file is selected, it will automatically upload to the server and notifying each student that is taking the subject about the assignment question release so that they can download and start working on the assignment.

6.3.4.6 Assignment Groups List Page

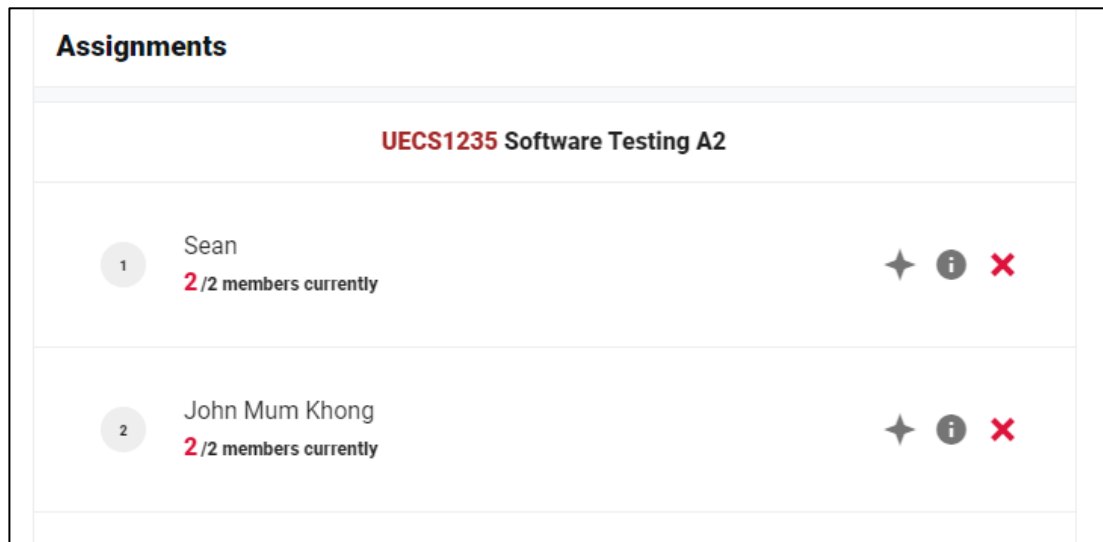


Figure 6.38: Assignment Group List Page

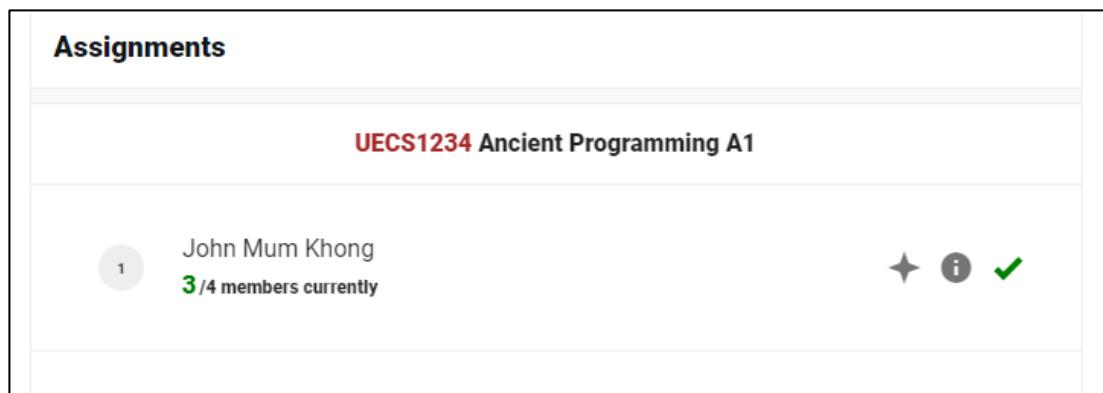


Figure 6.39: Assignment Group List Page

All the assignment group registered will be appearing under this page. It shows the group leader's name and the number of group members inside the group briefly. The first icon is the workspace button that leads lecturer into the group's workspace of the assignment. The information icon is the link to the group's detailed information. Both the workspace and group detail page are identical to the student's module. The last icon represents the state of submission for the assignment. A red cross indicates that the group has not submitted the assignment yet whereas a green tick shows that the group had already submitted their assignment.

6.3.4.7 Analytics Page

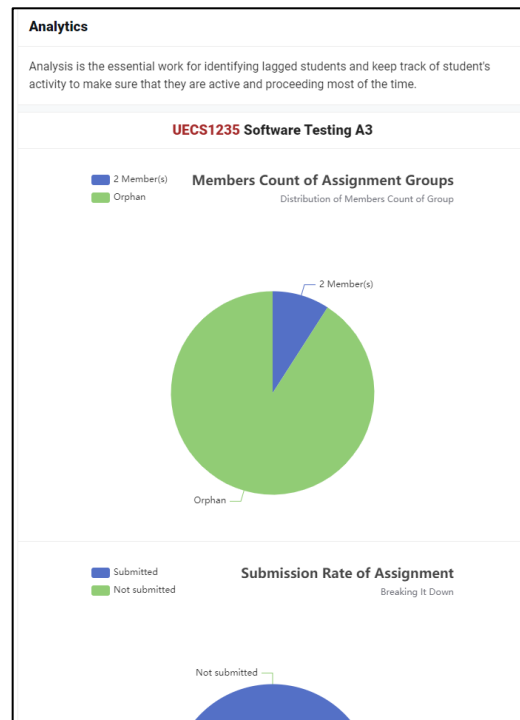


Figure 6.40: Assignment Detail Page

Lecturers can take a quick view of the statistics of all the groups of the assignments. The current supported charts are members count distribution and submission status for the assignment. The charts used is Apache Echarts. The explanations for the charts are as follows:

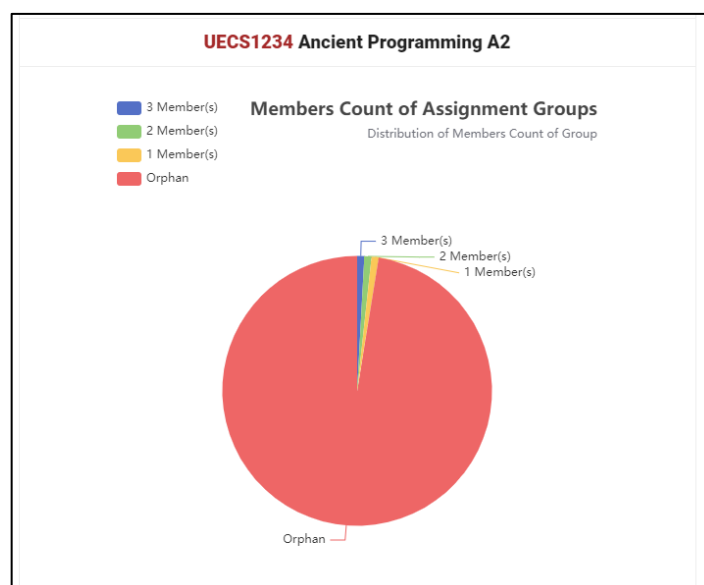


Figure 6.41: Member Distribution Chart for UECS1234 A2

According to the graph, given that there are 120 students participating in UECS1234 course, there are one group with 3 members, one group with 2 members and one group and the remanding 114 students are yet to form a group at the moment.

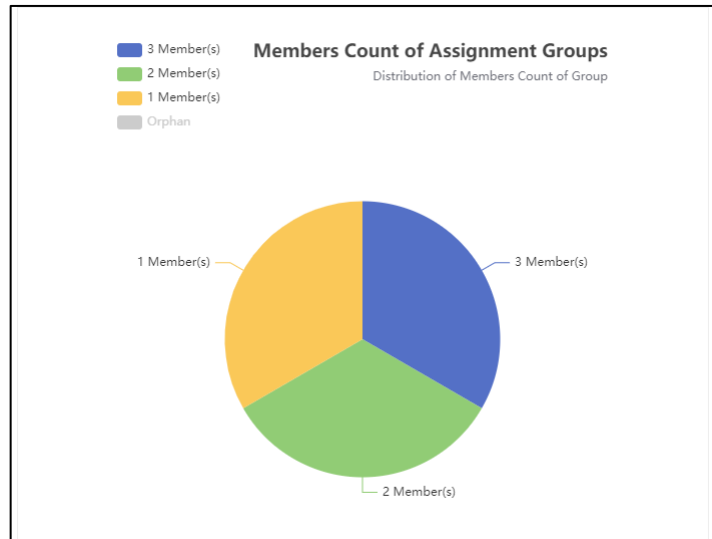


Figure 6.42: Member Distribution Chart after hiding Orphans

The attribute can be turned off to see the other combination for the charts by clicking on the attribute at legends located at top-left side. This is fairly convenient to filter out the information. To get a detailed number for each of the section, simply hover the mouse on the pie and the details will be showed as a tooltip.

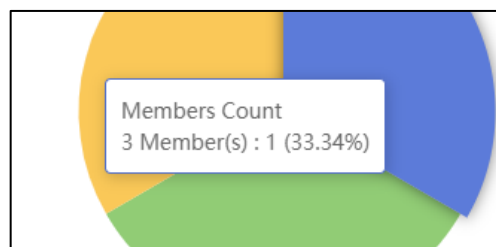


Figure 6.43: Chart with Tooltip Shown

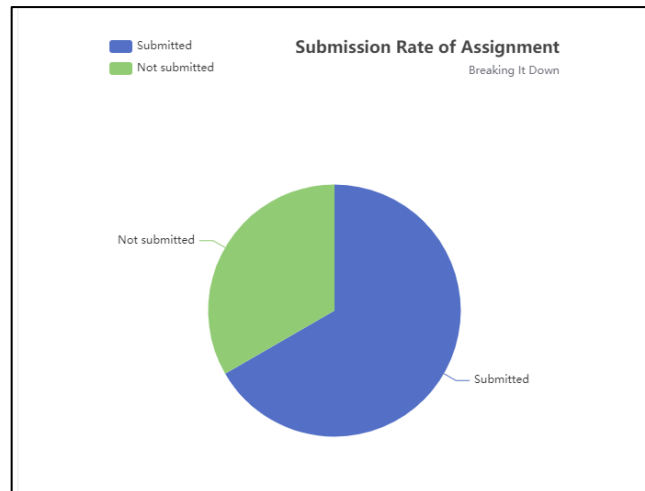


Figure 6.44: Submission Status Chart for UECS1234 A2

The second chart shows the submission status for the assignment. It only has 2 possible values which are submitted and not submitted.

6.4 Back-end Development

In this section, various tools or Node.js packages used for developing the back-end server will be introduced.

```

"dependencies": {
  "express": "^4.17.1",
  "firebase": "^8.2.4",
  "jshashes": "^1.0.8",
  "multer": "^1.4.2",
  "node-schedule": "^2.0.0",
  "nodemailer": "^6.5.0",
  "short-uuid": "^4.1.0",
  "socket.io": "^3.1.1",
  "uuid": "^8.3.2"
},
"devDependencies": {
  "@types/cors": "^2.8.9",
  "@types/express": "^4.17.11",
  "@types/jest": "^26.0.22",
  "@types/multer": "^1.4.5",
  "@types/node": "^14.14.22",
  "@types/node-schedule": "^1.3.1",
  "@types/nodemailer": "^6.4.0",
  "@types/socket.io": "^2.1.13",
  "body-parser": "^1.19.0",
  "cors": "^2.8.5",
  "dotenv": "^8.2.0",
  "jest": "^26.6.3",
  "nodemon": "^2.0.7",
  "ts-jest": "^26.5.4",
  "ts-node": "^9.1.1",
  "typescript": "^4.1.3"
}

```

Figure 6.45: Dependencies Excerpt from Project's package.json File.

Dependencies in the file lists all third-party packages that is used in the project. Those packages are crucial and the application could not run properly if any of them are missing in the production build. DevDependencies on the other hand is the packages that is used by developer during the development phase. These dependencies are not needed during the production phase.

Express is used as framework for the server to deploy endpoints easily. Firebase is used for the Firestore Cloud Database to store and retrieve data to and fro the front-end. Jshashes is used to hash password to compare the credentials for the users. Multer is the middleware used to store files that is uploaded from the client-side. Node-schedule is used to schedule jobs. Nodemailer is used to send emails. Short-uuid is used to generate a unique identifier for most of the entities. Socket.io is used for real-time data flowing.

For development dependencies, TypeScript is used. All the packages that start with @types are the type support for TypeScript for the packages installed in the project. Cors is used to disable the security for local development. Dotenv is used to manage the environment file that contains all the credentials used in the project such as Firebase ID. Jest is used for testing. Nodemon is used to serve the server so that it restarts automatically when a file is modified. Ts-jest and Ts-node is the runtime for jest and node in TypeScript.

6.4.1 Job Scheduling

Job scheduling is the execution of a job or codes periodically or at a specific time. To schedule a job, Cron Expression is used.

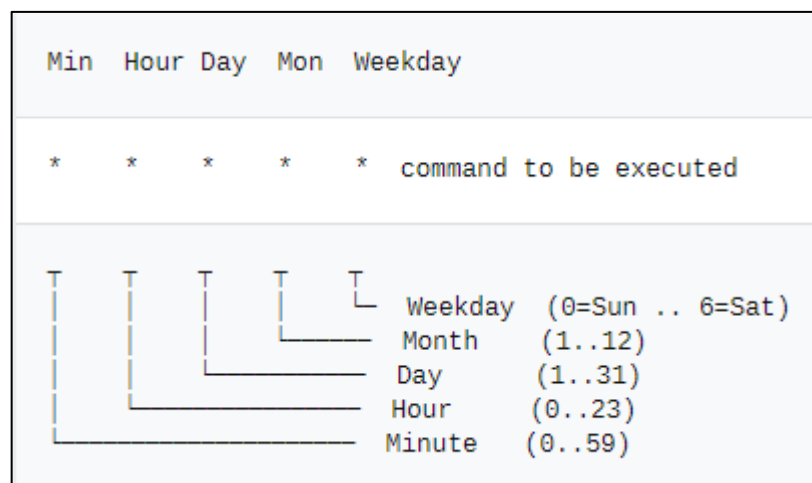


Figure 6.46: Cron Expression

Table 6.1: Sample Cron expression and description

Cron Expression	Description
* * * * *	Every minute
0 0 * * 1-5	Every weekday at 00:00 midnight
1 6 7 5 *	At May 7 th , 6.01am specifically

```

let serverWeek = 1;
schedule.scheduleJob("0 0 0 * * 1", () => {
  // Increment week number
  serverWeek++;

  // Reset discussion time and bookings
  firebase
    .firestore()
    .collection("groups")
    .get()
    .then((groups) => {
      groups.forEach((doc) => {
        // Reset time
        doc.ref.update({
          confirmedTime: {},
        });
        // Remove existing bookings
        doc.ref
          .collection("booking")
          .get()
          .then((bookings) => {
            bookings.forEach((idoc) => {
              idoc.ref.delete();
            });
          });
      });
    });
});
});

```

Figure 6.47: Cron Job in Server via node-schedule.

Job scheduling is used to increment the week number for the semester and clear the discussion booking data so that students can prepare for the next discussion weekly at 12am sharp every Monday.

6.4.2 Real-time Server

Real-time server is used to transmit data concurrently across the active clients. In this project, there are one more small server which serves the front-end application as well, which is the real-time server. The real-time server will be the server on its own because the Express server will be only listening to API endpoint calls. Moreover, the library used for establishing the real-time server (Socket.io) is not fully supported

by TypeScript. Hence, the real-time server will be written in JavaScript and having its own dedicated endpoint (port 5050) different from the main server (port 5000).

6.5 API Endpoints

There are a total of 36 endpoints used in this application. All of the endpoints are separated nicely by grouping them by entity in their respective files. Parent route is the route groups all the relevant endpoints to provide a unified process of accessing the individual endpoint. For example, if the parent route for “/endpoint” is “/api/parent”, the full route would be “/api/parent/endpoint”.

Table 6.2: API List for Assignment Entity

Parent Route:		/api/assignment	
No	Endpoints	Method	Description
1.	/:id	GET	Get assignment details
2.	/	POST	Create new assignment
3.	/overview	POST	Get details for assignments page (student)
4.	/:lecturerId/overview	GET	Get details for assignment page (lecturer)
5.	/:id/upload/ques	POST	Upload assignment file
6.	/:id/upload/:groupId	POST	Submit group assignment file
7.	/submitted/:filename	GET	Retrieve group assignment file
8.	/question/:filename	GET	Retrieve assignment question
9.	/:id	PATCH	Update assignment details
10.	/:id/groups	GET	Get all groups for this assignment

Table 6.3: API List for Group Entity

Parent Route:		/api/group	
No	Endpoints	Method	Description
1.	/:id	GET	Get group details
2.	/:id/gantt	GET	Get all the Gantt Chart of group
3.	/:id/members	GET	Get details of group members
4.	/create	POST	Create a new group
5.	/:id/gantt/create	POST	Add Gantt Chart for group
6.	/:id/booking	POST	Add discussion booking
7.	/:id/booking	GET	Get all booking records
8.	/join	POST	Join an existing group
9.	/matchmake	POST	Matchmake request
10.	/:id/extended/v1	GET	Get all details for the group and assignment for info page
11.	/:id/chats	GET	Get all chat messages
12.	/:id/availability	PATCH	Update availability of the group
13.	/:id/gantt/:ganttId	PATCH	Update Gantt Chart
14.	/:id/gantt/:ganttId	DELETE	Delete Gantt Chart
15.	/:id/code	GET	Retrieve source code

Table 6.4: API List for Lecturer Entity

Parent Route:		/api/lecturer	
No	Endpoints	Method	Description
1.	/:id	GET	Get lecturer details
2.	/:id/subjects	GET	Get subjects details of lecturer
3.	/:id/assignments	POST	Get assignment details of lecturer

Table 6.5: API List for Student Entity

Parent Route:		/api/student	
No	Endpoints	Method	Description
1.	/:id	GET	Get student details
2.	/avatar	POST	Get a list of student's avatar URL
3.	/:id/groups	GET	Get details of all the student's assignment groups

Table 6.6: API List for Subject (Course) Entity

Parent Route:		/api/subject	
No	Endpoints	Method	Description
1.	/:code	GET	Get subject's details
2.	/	POST	Get a list of subject's details

Table 6.7: API List for Other Uses

Parent Route:		No parent route	
No	Endpoints	Method	Description
1.	/api/week	GET	Get semester week
2.	/login	POST	User authentication
3.	/api/notifications/:userId	GET	Get user notifications

6.6 File Structure

6.6.1 Front-end

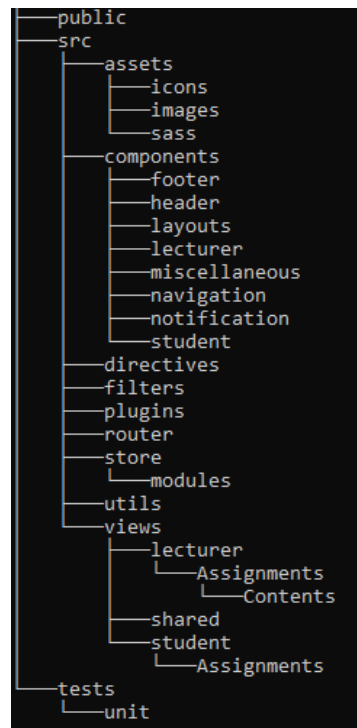


Figure 6.48: File structure of front-end repository

All the development files are located inside the source folder and the test files in the tests folder. The assets folder contains all the pictures and icons used in the project. It also contains the Sass files that contains the CSS styles used in the project. In the components folder, all the Vue component files are located there. It is further grouped into folders for cleaner directory as there are so many big and small components used and it will be a strenuous task to get to one specific component file if everything is mingled under the hood.

```

<div
  class="setting-popup"
  v-if="isPanelClicked"
  v-click-outside="verifyPanelState"
>
  
```

Figure 6.49: A div element using directives

Next, the directive folder contains the custom directives used. Directive is a Vue feature that enables direct manipulation of DOM. Plugin is used to contain declaration of third-party packages that will extend functionalities for the Vue project.

Vuetify, Vuex, Vue router are some of the examples. The router folder contains all the declaration for the routes of the UI. Store folder is used by Vuex to store the global state data so that it can be shared to other components as well. Utils folder contains all the utility functions used across the project. Lastly, all the pages are located inside the views folder.

6.6.2 Back-end

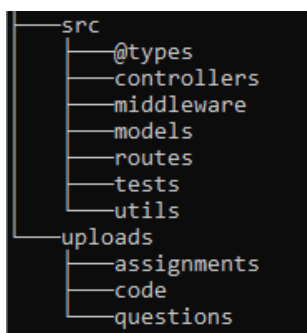


Figure 6.50: File structure for back-end repository

Meanwhile at the server side, all the development files including test files are included within the source folder. The @types folder are the dummy type declaration for the third-party packages that does not have support for TypeScript. Without the type declaration, TypeScript will complain about the code and inhibits the execution for the whole project.

The controller folder contains all the logic for computing, fetching and retrieving data. It is the core folder for the backend server. The middleware folder contains the middleware used for API. Models folder contains the entities declaration only similar to a data dictionary. It is not used in the project. Routes folder contains all the API endpoints for the server. The utilities folder supplies all necessary utility function in the back-end.

The uploads folder outside the source folder contains all the uploaded files from the user as well as source code file for the group. This folder is untracked in Git.

6.7 Version Control System

Git is used as the version control system for both the front-end and back-end development. Each changes or new feature, regardless big or small are committed shortly after. There are many types of commit, some of which are for bug fix, some of them are for new feature. Committing with regular string messages can sometime

be ambiguous. In order to properly show the type of changes being made, commit prefixes are used. An example would be “feat: Added exception handling for controller.”

Table 6.8: Common Commit Prefixes (Tasevski, 2019)

Prefix	Description
feat	A new feature introduced
fix	A bug fix
refactor	Change of code that does not fix bug nor add a new feature
chore	Updating gulp tasks
docs	Changes to documentation.
style	Code formatting such as white space, semicolons etc.
perf	Changes in production code that boosts the performance overall
vendor	Installation or uninstallation of third-party packages
tests	Adding and fixing of test files

6.7.1 Remote Hosting with GitHub

Both the source codes for [front-end](#) and [back-end](#) can be reach via the links provided at GitHub.

The screenshot shows a GitHub repository page for 'data-miner00'. At the top, there are navigation buttons: 'Go to file', 'Add file', and 'Code'. Below this, the repository name 'data-miner00' is displayed along with the commit message 'fix: More detailed tooltip' and the commit hash 'a60dcf7' and '2 hours ago'. There are also '323 commits' indicated. The main content is a list of files and folders, each with a commit message and a date:

- public: feat: Added a funny comment (2 months ago)
- src: fix: More detailed tooltip (2 hours ago)
- tests/unit: test: Rename test suite (3 days ago)
- .browserslistrc: init (5 months ago)
- .env: vendor: Installed socket.io (2 months ago)
- .env.example: vendor: Installed socket.io (2 months ago)
- .eslintrc.js: vendor: Added jest (13 days ago)
- .gitignore: init (5 months ago)
- README.md: init (5 months ago)
- babel.config.js: init (5 months ago)
- jest.config.js: vendor: Added jest (13 days ago)
- package-lock.json: vendor: Added jest (13 days ago)
- package.json: vendor: Added jest (13 days ago)
- vue.config.js: Initial commit (3 months ago)

Figure 6.51: Remote Repository for Front-end on GitHub

File	Commit Message	Time Ago
src	fix: One member group ignored	8 hours ago
.env.example	feat: Added example env file	21 days ago
.gitignore	chore: Removed .env	11 days ago
README.md	docs: Added readme file	2 hours ago
jest.config.js	chore: Config jest use ts preprocessor	11 days ago
package-lock.json	chore: Added types file for node-schedule	9 days ago
package.json	chore: Added types file for node-schedule	9 days ago
tsconfig.json	chore: Added ES2019 support	last month

Figure 6.52: Remote Repository for Back-end on GitHub

As of 14 April 2021, there are a total of 323 commits for the front-end and 128 commits on the server side.

6.8 Summary

In a nutshell, this chapter outlined the setup process for both side of the project. Next, the implementation in the front-end was demonstrated with screenshots of the actual system. The logics used in back-end servers were discussed as well. File structures for this project was thoroughly explained and finally the use of version control system and remote hosting were illustrated.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

This chapter will be providing detailed information with regards to the unit testing operations that was done with the application. It encompasses two big sections, namely automated test and manual test. A balanced combination of automated test and manual test is essential to ensure the testing quality and efficiency.

7.2 Automated Testing

For automated test, the test case written must be transformed into codes so that the test runner can pick up the state and condition for the dedicated test. The test engine that is used is called Jest which is a mature JavaScript testing framework known for its simplicity and rich feature.

Vue Test Utils, a Vue testing tools that can be used with Jest was used for testing the UI of the application. In the backend which is written in TypeScript, 'ts-jest' extension is installed for directly running the test without having to compile into native JavaScript codes for Jest to work.

7.2.1 User Interface Test

When it comes to UI test, it is decided that not every single screens and components will be tested thoroughly. The test was non-exhaustive. Instead, the core features such as render, lifecycle hooks, methods, interaction offered by Vue which impacts the UI will only be tested. This is to ensure that the UI testing process is short, effective and non-redundant.

Several screens or components that used the core features was selected to test out the behaviour. If everything runs as expected, it is pretty safe to assume that each and every screens and components on the front-end will be behaving perfectly.

7.2.1.1 Screen Rendering

Screen rendering is the process done by Vue engine to convert the Vue markup into HTML so that the browser can understand and display on the screen.

Table 7.1: Test Case 1 - Render

Test Case #	1	Test Case Name	Render	Module	App
Test Case Description	To test whether the screens or components is rendered correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Ensure the Login.vue is rendered properly.	1. Checks for string in HTML	Where Assignment Meets	Login.vue should render correctly.	Login.vue rendered correctly	Pass
	2. Checks for string in HTML	Join your friends now.			
	3. Checks for string in HTML	Login			
Ensure the Resources.vue with child component is rendered properly.	1. Checks for string in HTML	Search Engine	Resources.vue should render correctly.	Resources.vue rendered correctly.	Pass
	2. Checks for string in HTML	Screencasts			
	3. Checks for string in HTML	Web Technologies			

7.2.1.2 Props

Props is the data passed down to children components from the parent component. The props data may be used to render on screen as well as doing computations down the line.

Table 7.2: Test Case 2 - Props

Test Case #	2	Test Case Name	Props	Module	App
Test Case Description	To test whether the screens or components using props is rendered correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Ensure props passed into AssignmentItem.vue is rendered correctly.	1. Provide prop data.	(Relevant data)	AssignmentItem.vue should render correctly with the provided props.	AssignmentItem.vue rendered correctly with the provided props.	Pass
	2. Checks for string in HTML.	UECS1234			
	3. Checks for string in HTML.	Legend Programming			
	4. Assert existence.	.download			
	5. Assert non-existence.	.no-download			
Ensure props passed into GroupItem.vue	1. Provide prop data.	(Relevant data)	GroupItem.vue should render according to	GroupItem.vue rendered according	Pass
	2. Checks for string in HTML.	<div>1</div>			

is rendered nicely.	3. Checks for string in HTML	Sandra	props.	to props provided.	
	4. Assert existence.	.bad			
	5. Assert non-existence.	.good			

7.2.1.3 Data

Data, as weird as it is named this way, is the internal state of the Vue components that stores data that is not shared among other components within the application. It is responsible for dynamic rendering on screens. Ways to alter the data includes interaction on screens, methods invocation and lifecycle hooks.

Table 7.3: Test Case 3 - Data

Test Case #	3	Test Case Name	Data	Module	App
Test Case Description	To test whether the screens or components with internal data storage is rendered correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Ensure data within Popup2.vue is detected correctly.	1. Checks 'errored' attribute.	false	Default data state of Popup2.vue should be correct.	Default data state of Popup2.vue is correct.	Pass
	2. Checks 'errorMessage' attribute.	Unknown Error Occurred!			
Ensure data in	1. Changes 'errored' attribute.	true	The data state should be	The data state	Pass

Popup2.vue is changed and rendered correctly.	2. Changes 'errorMessage' attribute	A very serious problem occurred!	changed into newest state and render correctly on screen.	changed into newest state and rendered correctly on screen.
	3. Checks equivalence in 'errored' attribute.	true		
	4. Checks equivalence in 'errorMessage' attribute.	A very serious problem occurred!		
	5. Asserts the error message has been rendered on screen.	A very serious problem occurred!		

7.2.1.4 Lifecycle Hooks

Lifecycle hooks are the stages that a component will went through from created to destroyed. Whenever the component is transitioning to the next lifecycle, its respective hooks will be invoked.

Table 7.4: Test Case 4 - Lifecycle Hooks

Test Case #	4	Test Case Name	Lifecycle Hooks	Module	App
Test Case Description	To test whether the screens or components invoked lifecycle hooks accordingly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)

To ensure that the lifecycle hooks within NotiItem.vue is called correctly.	1. Create a spy on the created hook.	N/A	The lifecycle hooks should be called in order.	The lifecycle hooks were called correctly in order.	Pass
	2. Create a spy on the mounted hook.	N/A			
	3. Create a spy on the beforeDestroy hook.	N/A			
	4. Mount NotiItem.vue.	N/A			
	5. Checks invocation of all hooks	N/A			

7.2.1.5 Interaction and Methods

Interaction here means that the activity done on the UI such as clicking event, keyup event etc. Methods are the functions that are only available within the component. Hence, a combo of clicking event to call a particular method can be tested simultaneously.

Table 7.5: Test Case 5 – Interaction and Methods

Test Case #	5	Test Case Name	Interaction and Methods	Module	App
Test Case Description	To test whether the click events and method invocation is done correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		

Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
To ensure that join function is invoked when join button is clicked	1. Create a spy on join method.	N/A	Join should be invoked successfully.	Join was invoked successfully.	Pass
	2. Trigger click on join button,	N/A			
	3. Assert join is invoked.	N/A			
To ensure that create method is called when clicked.	1. Create spy on create method.	N/A	Create should be invoked successfully.	Create was invoked successfully.	Pass
	2. Invoke click create button.	N/A			
	3. Assert create is invoked.	N/A			
To ensure that matchmake button is called when clicked.	1. Create matchmake spy.	N/A	Matchmake should be invoked successfully.	Matchmake was invoked successfully.	Pass
	2. Invoke click matchmake.	N/A			
	3. Assert matchmake called.	N/A			

7.2.1.6 Emit

Emit is the passing back of data from children component to parent component. It is the direct opposite of props. Emit is treated as an event.

Table 7.6: Test Case 6 – Emit

Test Case #	6	Test Case Name	Emit	Module	App
Test Case Description	To test whether the emit is behaving correctly.				
Pre-Conditions	User needs to load up the home/login page.				

Executed By		Chong Mum Khong	Execution Date		10 April 2021
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
To ensure that emit is behaving properly in Popup2.vue	1. Mount Popup2.vue	N/A	The emitted should behaves properly with the data passed correctly.	The emitted behaves properly with the data passed correctly.	Pass
	2. Triggers emit.	done			
	3. Check 'done' has been emitted.	N/A			
	4. Check the data emitted is the same.	assignmentId, assignNo			

7.2.2 Utility Functions Test

There are modular utility functions on both the front-end and the server-side. This test will test exhaustively all the functions available in the utility file.

7.2.2.1 Front-end Utilities Test

Table 7.7: Test Case 7 – Snippet

Test Case #	7	Test Case Name	Snippet	Module	App
Test Case Description	To test whether the snippet function works correctly.				

Pre-Conditions		N/A			
Executed By		Chong Mum Khong	Execution Date	10 April 2021	
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
To test whether the returned value is correct	1. Provide test string.	Long string	Should obtain string “Lon...”	Obtained the expected string.	Pass
	2. Provide desired length.	3			
	3. Click on Login button	N/A			

Table 7.8: Test Case 8 – Alias

Test Case #	8	Test Case Name	Alias	Module	App
Test Case Description	To test whether the snippet function works correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
String is provided as input.	1. Provide test string.	1803151	Should obtain string “@1803151	Obtained the expected string.	Pass

Table 7.9: Test Case 9 – Python Range Function

Test Case #	9	Test Case Name	Python Range Function	Module	App
Test Case Description	To test whether the Python Range Function works correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Only end parameter is provided.	1. Provide end parameter.	5	[0, 1, 2, 3, 4]	Obtained the expected result.	Pass
Start and end parameter is specified.	1. Provide start parameter.	2	[2, 3, 4]	Obtained the expected result.	Pass
	2. Provide end parameter.	5			
Start, end and step is specified.	1. Provide start parameter.	2	[2, 4]	Obtained the expected result.	Pass
	2. Provide end parameter.	5			
	3. Provide step parameter.	2			

Table 7.10: Test Case 10 – Random Colour

Test Case #	10	Test Case Name	Random Colour	Module	App
Test Case Description	To test whether the random colour function is correct.				

Pre-Conditions		N/A			
Executed By		Chong Mum Khong	Execution Date	10 April 2021	
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
No parameters needed to test the function.	1. Run on test suite.	N/A	String matching the regular expression /^\#\{d\{6\}\}\$/	Obtained the expected result.	Pass

Table 7.11: Test Case 11 – Message Wrapper

Test Case #	11	Test Case Name	Message Wrapper	Module	App
Test Case Description	To test whether the message wrapper works as intended.				
Pre-Conditions	User needs to load up the home/login page.				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Provide all sufficient data for the function	1. Provide all relevant data	(All data including message)	The returned message object should contain all information passed with extra auto	The returned message object contains all information passed	Pass
	2. Checks the return ID is truthy	N/A			

	3. Checks equivalence of the message returned.	N/A	generated id.	with extra auto generated id.	
--	------------------------------------------------	-----	---------------	-------------------------------	--

7.2.2.2 Back-end Utilities Test

Table 7.12: Test Case 12 – Date to Timestamp Method

Test Case #	12	Test Case Name	dateToTimestamp	Module	App
Test Case Description	To test whether the conversion of string date to timestamp works.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Provide valid string date.	1. Run with test runner.	2012-01-31	{seconds: 1327939200, nanoseconds: 0}	Obtained the expected result.	Pass
Invalid string date is provided.	1. Run with test runner.	2012-01-51 99999-01-31 abc	{seconds: NaN, nanoseconds: NaN}	Obtained the expected result.	Pass

Table 7.13: Test Case 13 – Timestamp to Date

Test Case #	13	Test Case Name	timestampToDate	Module	App
Test Case Description	To test whether the conversion from timestamp to string date works correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Test with a timestamp object provided.	1. Run with test runner.	{seconds: 1327939200, nanoseconds: 0}	January 31, 2012	Obtained the expected result.	Pass

Table 7.14: Test Case 14 – Timestamp to Date 2

Test Case #	14	Test Case Name	timestampToDate2	Module	App
Test Case Description	To test whether the second conversion from timestamp to string date works correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Test with a	1. Run with test runner.	{seconds:	2012-01-30	Obtained the	Pass

timestamp object provided.		1327939200, nanoseconds: 0}	expected result.	
----------------------------	--	--------------------------------	------------------	--

Table 7.15: Test Case 15 – Get String Month

Test Case #	15	Test Case Name	getStringMonth	Module	App
Test Case Description	To test whether the function returns month in name given index.				
Pre-Conditions	User needs to load up the home/login page.				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Valid index is used. (0-11)	1. Run with test runner.	1 3	Should get February and April.	Obtained the expected result.	Pass
Invalid index is used. (negative, 12+)	1. Run with test runner.	-99 16	Should get undefined.	Obtained the expected result.	Pass

Table 7.16: Test Case 16 – Get AM PM

Test Case #	16	Test Case Name	getAMPM	Module	App
Test Case Description	To test whether the am pm methods returns correctly.				

Pre-Conditions		N/A			
Executed By		Chong Mum Khong	Execution Date	10 April 2021	
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Tested with hours less than 12 (valid hours and invalid hours)	1. Run with test runner.	4 -10	Should get “am”.	Obtained the expected result.	Pass
Tested with hours more than 12 (valid hours and invalid hours)	1. Run with test runner.	16 999	Should get “pm”.	Obtained the expected result.	Pass

Table 7.17: Test Case 17 – Timestamp Get Time

Test Case #	17	Test Case Name	timestampGetTime	Module	App
Test Case Description	To test whether the conversion from timestamp into time works fine.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status

					(PASS/FAIL)
Provided with valid timestamp.	1. Run with test runner.	{seconds: 1327939200, nanoseconds: 0}	Should get “00:00”.	Obtained the expected result.	Pass

Table 7.18: Test Case 18 – Is Object Empty

Test Case #	18	Test Case Name	isEmpty	Module	App
Test Case Description	To test the function that checks for empty object works correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Empty object is provided.	1. Run with test runner.	{ }	true	Obtained the expected result.	Pass
Provided with object with properties.	1. Run with test runner.	{ a: 1 }	false	Obtained the expected result.	Pass
Login with invalid ID and invalid password	1. Enter a random ID.	asdf	Unsuccessful login with an error message below the password textbox.	Unsuccessful login with an error message below the	Pass
	2. Enter a random password.	1234			
	3. Click on Login button	N/A			

			password textbox.	
--	--	--	-------------------	--

Table 7.19: Test Case 19 – Compute Time

Test Case #	19	Test Case Name	computeTime	Module	App
Test Case Description	To test the function that compute time interval works correctly.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Test for valid computation.	1. Run with test runner.	[{ startTime: 1, endTime: 5, }, { startTime: 3, endTime: 6, },]	{ startTime: 3, endTime: 5, }	Obtained the expected result.	Pass

Table 7.20: Test Case 20 – Booking Algorithm

Test Case #	20	Test Case Name	bookingAlgorithm	Module	App
Test Case Description	To test the booking algorithm of the system works as intended.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Test for valid booking requests.	1. Run with test runner.	(A very long data, refer to source file)	{ startTime: 19, endTime: 22, day: "wednesday", }	Obtained the expected result.	Pass

Table 7.21: Test Case 21 – Random Choice

Test Case #	21	Test Case Name	randomChoice	Module	App
Test Case Description	To test the JavaScript implementation of Python's random_choice.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		

Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Test the behaviour of picking a value.	1. Mock the Math.random to always return a fixed number.	0.123	1	Obtained the expected result.	Pass
	2. Provide an array.	[1, 2, 3]			
	3. Restore Math.random before end.	N/A			

Table 7.22: Test Case 22 – Random Pop

Test Case #	22	Test Case Name	randomPop	Module	App
Test Case Description	To test the implementation of randomChoice plus removing the selected from the original.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Test the behaviour of picking and removing a value.	1. Mock the Math.random to always return a fixed number.	0.123	{ removedItem: 1, arr: [2, 3] }	Obtained the expected result.	Pass
	2. Provide an array.	[1, 2, 3]			
	3. Restore Math.random before	N/A			

	end.			
--	------	--	--	--

7.2.3 Automated Test Results

```

PASS tests/unit/emit.spec.js
PASS tests/unit/lifecycle.spec.js
PASS tests/unit/interaction.spec.js
PASS tests/unit/data.spec.js
PASS tests/unit/props.spec.js
PASS tests/unit/render.spec.js
Test Suites: 6 passed, 6 total
Tests:      11 passed, 11 total
Snapshots: 0 total
Time:       4.66s
Ran all test suites.

```

Figure 7.1: Automated test results for UI

Above figure shows the console output from running the UI test with Jest and six files or test suites was successfully executed and passed. With this, it is confident to say that all UI that built with Vue is pretty solid as long as the fundamentals are applied correctly.

```

PASS tests/unit/utils.spec.js
  All functions within utilities
    ✓ should get correct snippet string (2ms)
    ✓ should add an leading @ to string (1ms)
    ✓ should return # with 6 trailing numbers
    ✓ should wrap the params into message object (2ms)
  Python range function
    ✓ should give list from 0 to 4 (1ms)
    ✓ should give list from 2 to 4
    ✓ should give 2 to 5 with one jump (1ms)
Test Suites: 1 passed, 1 total
Tests:      7 passed, 7 total
Snapshots: 0 total
Time:       1.545s, estimated 2s
Ran all test suites matching /utils.spec.js/i.

```

Figure 7.2: Automated test results for front-end utility

The utility functions on both the front-end and back-end was tested exhaustively as it is modular and suitable for extensive tests. The figure above shows that all test cases had been executed without a hitch.

```
PASS src/tests/utils.test.ts
Testing every functions in utils file
  ✓ should convert from Timestamp to string (1 ms)
  ✓ should convert from Timestamp to string 2
  ✓ should get the Month from index
  ✓ should get AM PM according to hours (1 ms)
  ✓ should get padded time only from Timestamp object
  ✓ should validate the emptyness of an object (1 ms)
  ✓ should select the correct timeperiod
  ✓ should return majority booking time and date correctly (1 ms)
  ✓ should work as the same as Python random_choice function
  ✓ should randomly remove an item from array
Testing dateToTimestamp
  ✓ should convert from string into Timestamp (3 ms)
  ✓ should return NaN for invalid date (1 ms)

Test Suites: 1 passed, 1 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        3.439 s, estimated 4 s
Ran all test suites.
```

Figure 7.3: Automated test results for back-end utility

The same goes for the back-end utilities function as all of the test cases are passed. There are much more utilities function at the back-end compared to front-end as most of the computations and logics resides here.

7.3 Manual Testing

Manual testing is done for the part of the application where it is difficult or expensive to write automated test such as asynchronous calls from the front-end to the server and into the cloud database. In this part, feature testing, screen navigation test and API test were conducted.

7.3.1 Feature Testing

Table 7.23: Test Case 23 – Login

Test Case #	23	Test Case Name	Login	Module	App
Test Case Description	To test whether the login of the system is functioning as intended.				
Pre-Conditions	User needs to load up the home/login page.				
Executed By	Chong Mum Khong	Execution Date	10 March 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Login as student with correct ID and password.	1. Enter student ID.	1803151	Login successful with redirection to student homepage.	Login successful with redirection to student homepage.	Pass
	2. Enter password.	D1onald			
	3. Click on Login button	N/A			
Login as lecturer with correct ID and password.	1. Enter lecturer/staff ID.	L123456	Login successful with redirection to lecturer homepage.	Login successful with redirection to student homepage.	Pass
	2. Enter password.	D1onald			
	3. Click on Login button.	N/A			
Login with invalid	1. Enter a random ID.	asdf	Unsuccessful login with	Unsuccessful login	Pass

ID and invalid password	2. Enter a random password.	1234	an error message below the password textbox.	with an error message below the password textbox.
	3. Click on Login button	N/A		

Table 7.24: Test Case 24 – Join Assignment Group

Test Case #	24	Test Case Name	Join Assignment Group	Module	Student
Test Case Description	To test the join function.				
Pre-Conditions	Student needs to have at least one ungrouped assignment available.				
Executed By	Chong Mum Khong	Execution Date	8 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Join an existing assignment group.	1. Enter the groupId.	Valid GroupID	Updated UI for the group and show a join success notification.	Updated UI for the group and show a join success notification.	Pass
	2. Click on Join button	N/A			
Join an existing assignment group that is full or explicitly declared	1. Enter the groupId.	Valid GroupID	Join not successful. The popup persists and an error message in red is shown.	Join not successful. The popup persists and an error message in red is	Pass
	2. Click on Join button	N/A			

unavailable.				shown.	
Join an existing group but is from different assignment.	1. Enter the groupID	Valid GroupID	Join not successful. The popup persists and an error message in red is shown.	Join not successful. The popup persists and an error message in red is shown.	Pass
	2. Click on Join button	N/A			
Join a non-existing assignment group.	1. Enter the groupID	Invalid GroupID	Join not successful. The popup persists and an error message in red is shown.	Join not successful. The popup persists and an error message in red is shown.	Pass
	2. Click on Join button	N/A			

Table 7.25: Test Case 25 – Create/Matchmake Group

Test Case #	25	Test Case Name	Create/Matchmake Group	Module	Student
Test Case Description	To verify the create assignment feature and flows				
Pre-Conditions	Student needs to have at least one ungrouped assignment available.				
Executed By	Chong Mum Khong	Execution Date	8 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)

Create an assignment group.	1. Click on the Create button.	N/A	Shows successful message and the UI has been updated.	Shows successful message and the UI has been updated.	Pass
Matchmake for assignment group	1. Click on the Matchmake button	N/A	Shows successful message.	Shows successful message.	Pass

Table 7.26: Test Case 26 – Send Text Messages

Test Case #	26	Test Case Name	Send Text Messages	Module	Student
Test Case Description	To test the functionality of the chat.				
Pre-Conditions	User needs to get into assignment workspace.				
Executed By	Chong Mum Khong	Execution Date	10 March 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Send text message to the group.	1. Enter text messages.	Hello	“Hello” shown on the UI with name, avatar and timestamp.	“Hello” shown on the UI with name, avatar and timestamp.	Pass
	2. Hit Enter key.	N/A			
Send empty message to the group	1. Hit Enter key.	N/A	Nothing is sent to the channel.	Nothing is sent to the channel.	Pass

Table 7.27: Test Case 27 – Gantt Chart

Test Case #	27	Test Case Name	Gantt Chart	Module	Student
Test Case Description	To test whether the Gantt Chart's functionality is intact.				
Pre-Conditions	User needs to navigate to the Gantt Chart's page within their assignment workspace.				
Executed By	Chong Mum Khong	Execution Date	10 March 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Adding Gantt activity.	1. Enter activity name.	Sample activity.	The popup should close with a new entry on the Gantt Chart is shown.	The popup closed with a new entry on the Gantt Chart is shown.	Pass
	2. Enter valid from week.	1			
	3. Enter valid to week.	3			
	4. Assign task to a teammate.	(Select teammate)			
	5. Select a deadline.	(Any date)			
	6. Click on Confirm	N/A			
Editing Gantt activity	1. Change activity name.	Changed activity.	The activity change into "Changed activity" and extend to week 5.	The activity change into "Changed activity" and extend to week 5.	Pass
	2. Change valid to week.	5			
	3. Click on Confirm button.	N/A			
Gantt activity	1. Choose on a Gantt activity.	(last activity)	The last activity should	The last activity is	Pass

deletion	2. Click on the delete icon.	N/A	be removed.	removed.	
	3. Click Confirm upon confirmation.	N/A			

Table 7.28: Test Case 28 – Add Discussion Time

Test Case #	28	Test Case Name	Add Discussion Time	Module	Student
Test Case Description	To test the adding of discussion time is successful.				
Pre-Conditions	User needs to load up the discussion booking page.				
Executed By	Chong Mum Khong	Execution Date	10 March 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Add booking without changing any data of the prompt.	1. Click the Confirm button	N/A	A default placement of Sunday starting from 12am to 11.59pm is made.	A default placement of Sunday starting from 12am to 11.59pm is made.	Pass
Add booking with personalized time.	1. Select a day.	Wednesday	A placement of Wednesday from 14:00	A placement of Wednesday from	Pass
	2. Select a starting time.	14			

3. Select an ending time.	17	to 17:00 is made.	14:00 to 17:00 is
3. Click on Confirm button	N/A		made.

Table 7.29: Test Case 29 – Submit Assignment

Test Case #	29	Test Case Name	Submit Assignment	Module	Student
Test Case Description	To test whether the assignment submission works correctly.				
Pre-Conditions	User needs to be the group leader for the assignment group.				
Executed By	Chong Mum Khong	Execution Date	9 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Upload assignment file through browse.	1. Click on the upload button.	N/A	Successful upload with the drop box turned into a successful message and notify team members and lecturer.	Upload successful with successful message and notified team members and lecturer.	Pass
	2. Select a file.	(Any file)			
	3. Click on Open button.	N/A			
Upload assignment file through drag and drop.	1. Select a file.	(Any file)	Successful upload with the drop box turned into a successful message	Upload successful with successful message and	Pass
	2. Drag the file with the cursor.	N/A			
	3. Drop the file on the drop	N/A			

box.		and notify team members and lecturer.	notified team members and lecturer.
------	--	---------------------------------------	-------------------------------------

Table 7.30: Test Case 30 – Create Assignment

Test Case #	30	Test Case Name	Create Assignment	Module	Lecturer
Test Case Description	To test whether the login of the system is functioning as intended.				
Pre-Conditions	User must be lecturer and needs to have at least one subject in charged.				
Executed By	Chong Mum Khong	Execution Date	10 March 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Provide sufficient information to create an assignment.	1. Enter assignment description.	Program a software	Assignment creation is successful and a snackbar notification with successful message is shown.	Assignment creation is successful and a snackbar notification with successful message is shown.	Pass
	2. Select course of assignment.	UECS1234			
	3. Select programming language.	C#			
	4. Enter max group member.	4			
	5. Enter assignment number.	1			
	6. Select a due date.	April 2, 2021			

	7. Click on Confirm	N/A			
Provide information but one field is deliberately left empty.	1. Enter assignment description.	Program a software	The confirm button is disabled and nothing happened.	The confirm button is disabled and nothing happened.	Pass
	2. Select course of assignment.	UECS1234			
	3. Enter assignment number.	1			
	6. Select a due date.	April 2, 2021			
	7. Click on Confirm	N/A			
Deliberately leave empty for all fields.	1. Click on Confirm	N/A	The confirm button is disabled and nothing happened.	The confirm button is disabled and nothing happened.	Pass

Table 7.31: Test Case 31 – Edit Assignment Details

Test Case #	31	Test Case Name	Edit Assignment Details	Module	Lecturer
Test Case Description	To verify the edit assignment functionality.				
Pre-Conditions	User needs to be lecturer and is in charge of the assignment to edit.				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
Changed all the	1. Changes description.	Build a cool app.	Update is successful	Update is	Pass

fields of assignment	2. Changes due date.	2021-03-14	with a green successful message appeared below the fields.	successful with a green successful message appeared below the fields.	
	3. Changes programming language.	Haskell			
	4. Changes group capacity.	5			
	5. Click on Save button.	N/A			
Upload an assignment question file.	1. Click on Choose File.	N/A	Upload successful with the file name appeared in the question file section.	Upload successful with the file name appeared in the question file section.	Pass
	2. Choose a file.	(Any file)			
	3. Click on Open.	N/A			

7.3.2 Screen Navigation Testing

Table 7.32: Test Case 32 – Student Navigation

Test Case #	32	Test Case Name	Student Navigation	Module	Student
Test Case Description	To verify the navigation of student screens starting from Homepage.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)

View Notification Page.	1. Click on notification link.	N/A	Should navigates to Notification Page	Navigated to Notification Page	Pass
View Assignments Page.	1. Click on Assignments link.	N/A	Should navigates to Assignments Page	Navigated to Assignments Page	Pass
View Courses Page.	1. Click on Courses link.	N/A	Should navigates to Courses Page	Navigated to Courses Page	Pass
View Resources. Page	1. Click on Resources link.	N/A	Should navigates to Resources Page	Navigated to Resources Page	Pass
View Assignment Group Info Page.	1. Click on Assignments link.	N/A	Should navigates to Group Info Page	Navigated to Group Info Page	Pass
	2. Click on Information icon link.	N/A			
View Group Workspace.	1. Click on Assignments link.	N/A	Should navigates to Workspace	Navigated to Workspace	Pass
	2. Click on Workspace icon.	N/A			
View Gantt Chart.	1. Click on Assignments link.	N/A	Should navigates to Gantt Chart Page.	Navigated to Gantt Chart Page.	Pass
	2. Click on Workspace icon.	N/A			
	3. Click on Gantt Chart icon.	N/A			
View Discussion Page	1. Click on Assignments link.	N/A	Should navigates to Workspace.	Navigated to Workspace.	Pass
	2. Click on Workspace icon.	N/A			
	3. Click on Discussion icon.	N/A			

View Discussion Page	1. Click on Assignments link.	N/A	Should navigates to Discussion Page.	Navigated to Discussion page.	Pass
	2. Click on Workspace icon.	N/A			
	3. Click on Discussion icon.	N/A			
View Assignment Submit Page	1. Click on Assignments link.	N/A	Should navigates to Assignment Submit Page.	Navigated to Assignment Submit Page.	Pass
	2. Click on Workspace icon.	N/A			
	3. Click on Upload icon.	N/A			

Table 7.33: Test Case 33 – Lecturer Navigation

Test Case #	33	Test Case Name	Lecturer Navigation	Module	Lecturer
Test Case Description	To verify the navigation of lecturer screens starting from Homepage.				
Pre-Conditions	N/A				
Executed By	Chong Mum Khong	Execution Date	10 April 2021		
Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (PASS/FAIL)
View Notification Page.	1. Click on notification link.	N/A	Should navigates to Notification Page	Navigated to Notification Page	Pass
View Assignments Page.	1. Click on Assignments link.	N/A	Should navigates to Assignments Page	Navigated to Assignments Page	Pass
View Courses Page.	1. Click on Courses link.	N/A	Should navigates to	Navigated to	Pass

				Courses Page	Courses Page	
View Analytics. Page	1. Click on Analytics link.	N/A		Should navigates to Analytics Page	Navigated to Analytics Page	Pass
View Assignment Group List Page.	1. Click on Assignments link.	N/A		Should navigates to Group List Page	Navigated to Group List Page	Pass
	2. Click on Group icon link.	N/A				
View Assignment Info Page.	1. Click on Assignments link.	N/A		Should navigates to Assignment Info Page.	Navigated to Assignment Info Page.	Pass
	2. Click on Information icon.	N/A				
View Assignment Group Info Page.	1. Click on Assignments link.	N/A		Should navigates to a group's information page.	Navigated to a group's information page.	Pass
	2. Click on Group icon link.	N/A				
	3. Select a group.	N/A				
	4. Click on Information icon.	N/A				
View Assignment Group Workspace.	1. Click on Assignments link.	N/A		Should navigates to Workspace.	Navigated to Workspace.	Pass
	2. Click on Workspace icon.	N/A				
	3. Select a group.	N/A				
	4. Click on Workspace icon.	N/A				

7.3.3 API Testing

The API testing is done manually instead of writing automated test in order to save calls to the Firebase database. For this testing, the summary for each API test will be presented to simplify the enormous amount of API endpoints involved.

7.3.3.1 Assignment Module

Table 7.34: Test Case 34 - 44

Test Case	Expected Result	Status
GET request to retrieve a single assignment.	Should retrieve assignment data with status code 200.	Pass
POST request to update an assignment.	Should update assignment with status code 200.	Pass
POST request to retrieve overview of student's assignments.	Should retrieve the required data with status code 200.	Pass
GET request to retrieve overview of lecturer's assignment.	Should retrieve the required data with status code 200.	Pass
POST request to upload assignment question.	Should upload the assignment just fine with status code 200.	Pass
POST request to submit group assignment.	Should submit the group assignment file with status 200.	Pass
GET request to download student's group assignment.	Should download the file with status 200.	Pass
GET request to download assignment question file.	Should download the file with status 200.	Pass
GET request to get assignment data for analytics.	Should get the data just fine with status 200.	Pass
PATCH request to update an assignment.	Should update the assignment with status 200.	Pass
GET request to retrieve all groups under an assignment.	Should get all the groups' data with status 200.	Pass

7.3.3.2 Credentials Module

Table 7.35: Test Case 45

Test Case	Expected Result	Status
POST request to get verification response.	200 on valid credentials, 400 on unmatched credentials.	Pass

7.3.3.3 Group Module

Table 7.36: Test Case 46 – 59

Test Case	Expected Result	Status
GET request to retrieve a single group info.	Should retrieve group data with status code 200.	Pass
GET request to retrieve all Gantt activity of a group.	Should get all the Gantt activity of the group status code 200.	Pass
GET request to retrieve member information of a group.	Should retrieve all members' data with status code 200.	Pass
POST request to create an assignment.	Should create new assignment with status code 200.	Pass
POST request to create a Gantt activity in a group.	Should create a new Gantt just fine with status code 200.	Pass
POST request to add booking in a group.	Should add a new booking with status 200.	Pass
GET request to get all bookings of a group.	Should retrieve all the data with status 200.	Pass
POST request to join assignment group.	200 on success, 404 on group not found, 403 on unavailable.	Pass
POST request to matchmake assignment group.	Matchmake request approved and queued with status 200.	Pass
GET request to get mixed information on group and assignment.	Should retrieve all necessary data with status 200.	Pass
GET request to retrieve all groups chats.	Should get all the chat history with status 200.	Pass
PATCH request to update group	Should update the availability of	Pass

availability.	group with status 200.	
PATCH request to update a Gantt activity.	Should update successfully with status 200.	Pass
DELETE request to remove a Gantt activity.	Should remove the selected activity without problem at 200.	Pass

7.3.3.4 Lecturer Module

Table 7.37: Test Case 60 – 62

Test Case	Expected Result	Status
GET request to get lecturer's data.	Should return lecturer's info with status 200.	Pass
GET request to get lecturer's subject.	Should return lecturer's subject data only with status 200.	Pass
POST request to get lecturer's assignments data.	Should return lecturer's assignments data with status 200.	Pass

7.3.3.5 Notification Module

Table 7.38: Test Case 63

Test Case	Expected Result	Status
GET request to get user's notification.	Should return at max 25 notifications with status 200.	Pass

7.3.3.6 Student Module

Table 7.39: Test Case 64 -66

Test Case	Expected Result	Status
GET request to retrieve a student's data.	Should get the student's data with status 200.	Pass
POST request to get several students' avatar.	Should get listed students' avatar with status 200.	Pass
GET request to get all the assignment groups of a student.	Should retrieve the data with status 200.	Pass

7.3.3.7 Subject Module

Table 7.40: Test Case 67 – 68

Test Case	Expected Result	Status
GET request to retrieve one single course's data.	Should get the course's information with status 200.	Pass
POST request to get several courses' data.	Should get list of courses with status 200.	Pass

7.3.3.8 System Module

Table 7.41: Test Case 69

Test Case	Expected Result	Status
GET request to get current semester week.	Should get the week accurately with status 200.	Pass

7.4 User Acceptance Test

User Acceptance Test is helpful for developers to get feedback on the agreed functionalities for the software or application that they are working on. In this project, a total of 7 users which included 6 students and 1 lecturer were invited to participate in the UAT. In this test, a total of 16 different test scenario was given to the tester and their action, whether they pass or failed the test and the time taken for execution was observed and noted.

Table 7.42: UAT Listing for Student

UAT ID	Modules	Test Scenario	Results (Pass/Fail)
UAT1	Login	<ul style="list-style-type: none"> • Login with student role • Logout the system 	
UAT2	Pages	<ul style="list-style-type: none"> • View notifications • View courses • View resources • View assignments 	

UAT3	Assignment Group	<ul style="list-style-type: none"> • View assignment group details • Create a group • Join a group • Send for matchmaking 	
UAT4	Download	<ul style="list-style-type: none"> • Download assignment question. • Download own assignment. 	
UAT5	Workspace	<ul style="list-style-type: none"> • View the group workspace • Code on editor • Send text messages 	
UAT6	Gantt Chart	<ul style="list-style-type: none"> • View group Gantt Chart. • Add an item. • Edit an item. • Delete an item. 	
UAT7	Discussion	<ul style="list-style-type: none"> • View discussion page. • Add a discussion time. 	
UAT8	Upload	<ul style="list-style-type: none"> • View the upload assignment page • Upload an assignment. 	

Table 7.43: UAT Listing for Lecturer

UAT ID	Modules	Test Scenario	Results (Pass/Fail)
UAT9	Login	<ul style="list-style-type: none"> • Login with lecturer role • Logout the system 	
UAT10	Pages	<ul style="list-style-type: none"> • View notifications • View courses • View analytics • View assignments 	

UAT11	Assignment	<ul style="list-style-type: none"> • Add new assignment • Edit assignment • Upload assignment question 	
UAT12	Groups	<ul style="list-style-type: none"> • Navigate to group list page. • Join one of the group's workspace • Type some code • Send chat message 	
UAT13	Group	<ul style="list-style-type: none"> • View one of the assignment group's details. • Download group's assignment. • View Gantt Chart Page. • Add/Edit/Delete Gantt Chart. • View discussion booking page 	

Overall, the invited participants for the UAT passed all the test cases successfully. However, it was noted that some of the participants are having issues searching for certain button when a test scenario was given. Some of them could not find logout button whereas some of them could not find the discussion booking and Gantt Chart button. This shows that the design of the user interface still has a lot of room for improvement. The full UAT results were attached in Appendix D.

7.5 Usability Test

The usability test was conducted after the invited user for the UAT had done the overall scenario of the system. By then, they would have sufficient knowledge and feedbacks to the system already. Hence, a usability test survey was provided to the invited guest to fill up before ending the entire testing session.

7.5.1 Usability Test Template

The survey template used was adapted directly from System Usability Scale (Brooke. J, 1986) called user satisfaction survey. The content are as follows:

Table 7.44: User Satisfaction Survey (Brooke. J, 1986)

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1. I think that I would like to use this website/system for viewing information and related matter.					
2. I found the website/system unnecessarily complex.					
3. I thought the website/system was easy to use					
4. I think that I would need the support of a technical person to be able to use this website/system.					
5. I found the various functions in this system were well integrated					
6. I thought there was too much inconsistency in this website/system.					
7. I would imagine that most people would learn to use this website/system very quickly.					
8. I found the system/website very awkward to use.					
9. I felt very confident					

using the system/website.					
10. I needed to learn a lot of things before I could get going with this website/system.					

7.5.2 User Satisfaction Survey Result

Among the 7 individuals invited for UAT, only 6 of them had successfully provided feedback via the User Satisfaction Survey. The full response for the survey can be found in Appendix E.

Table 7.45: Summary of User Satisfaction Survey

Participant	Score by Question										SUS Score
	1	2	3	4	5	6	7	8	9	10	
1	4	1	5	1	4	1	4	1	5	4	85
2	5	5	5	2	5	2	5	1	5	2	82.5
3	4	2	4	2	4	2	5	1	4	1	82.5
4	-	-	-	-	-	-	-	-	-	-	-
5	5	1	4	2	5	1	4	1	5	1	92.5
6	5	2	4	2	5	1	2	1	5	2	82.5
7	5	2	4	1	5	1	5	1	5	1	95
Average	4.7	2.2	4.3	1.7	4.7	1.3	4.2	1	4.8	1.8	86.7

CHAPTER 8

CONCLUSION AND RECOMMENDATION

8.1 Conclusion

To sum up, this report had documented the entire process of development for the project entitled “E-Learning Platform for Collaborative Coding Assignments.” The delivery of the system had successfully achieved the objectives for the project.

It had provided student an insurmountable of communicative power in programming group assignments via the collaborative code editor, chat system, Gantt Chart and discussion booking feature. It also provides students a conducive environment for the assignments by allowing students to create, join or matchmake for assignment group according to their preference. It also provides a few quick links for them to get access to helpful coding resources to enhance their workflow. The system also provides transparency of student’s progress by allowing lecturers to check on their students’ work from time to time. The information accessible are source codes, chat histories, Gantt Charts and discussions booked.

Besides, the system has the potential in the university’s context especially within Malaysia as the survey results shown. The shortcomings of the Malaysian university’s e-learning portal or platform that are lackluster in terms of collaborative programming group assignment is pretty much resolved with the solution and features provided by the system.

Students that use the platform for group programming assignment can be more productive and collaborative that will produce high quality work. Lecturers will also be more convenient to manage assignments and groups in the long run.

8.2 Limitations and Future Work

Firstly, this project does not cover the Admin module. The proposed solution only includes Student and Lecturer module to narrow down the scope and time of development as developing the Admin module will be too heavy and have high risk of failing to meet the stipulated deadline for the project. Second, the system’s code editor is not delicate as it contains bugs and limitations due to the difficulties of implementing real-time algorithm. The cursor position within the code editor could

not be retrieved. There are also lack of comprehensive project or resources that could be referred when it comes to Operational Transformation algorithm, the correct way of implementing real-time typing application. Not only that, there are no resources that points the correct way of how to save the source code into the server and properly deal with network latency, which can be devastating if not handled properly.

Moreover, the system is using Firebase as the database, which is a document-based database and the service will be charged by Google if a certain amount of API calls exceeded the free threshold given. Since the design of the database is object-oriented and relational approach, it will be expensive as one API calls from the front-end to the server may invoke more than one API calls to the Firebase database. Take the assignment information page for example, the data needed for the page requires calls to the assignment, group collection to get the data, student collection to get the group member's name. As you can see, a total of 3 API calls was invoked just to fill in the page. At this rate, the calls for the database may quickly exceed the threshold given and being charged by the usage.

For the system to be more useful, Admin module can be developed to manage students, courses, lecturers and other administrative actions such as monitoring and server manipulation. More researches have to be made on the concurrent text editor for the proper implementation that takes the OT algorithm, saving operation, latency into consideration.

To solve the database dilemma, two solutions may be used for future work to rectify the problem. The first solution is to design the database with a data model diagram. This way, the calls for certain page can be done with just one single call. However, the document tends to be massive with this approach. GraphQL can be used to query the Firebase collection to reduce the size of the data retrieved. If the document contains 50 fields and for some page or actions, only a few of the field data are required but not all, retrieving the entire document just to use some of the field data is expensive. GraphQL allows to query the exact fields needed only that will reduce the size of data for better optimization. The second way to solve this is to migrate the database into a local MySQL database that is optimized for relational queries such as joint operation and it is more importantly, free.

Other future works include the rework of UI design to better match the user's expectation. Furthermore, adding responsiveness to the application can support more users of different screen resolutions. Localization using Vue-i18n can be done as

well to provide more language options such as Chinese and Malay languages for the user's preference.

REFERENCES

- Aldiab, A., Chowdhury, H., Kootsookos, A. and Alam, F., 2017. Prospect of eLearning in Higher Education Sectors of Saudi Arabia: A Review. *Energy Procedia*, [online] 110(December 2016), pp.574–580. Available at: <<http://dx.doi.org/10.1016/j.egypro.2017.03.187>>.
- Alhabeeb, A. and Rowley, J., 2017. Critical success factors for eLearning in Saudi Arabian universities. *International Journal of Educational Management*, 31(2), pp.131–147.
- Alsaqqa, S., Sawalha, S. and Abdel-nabi, H., 2020. Agile Software Development : Methodologies and Trends Agile Software Development : Methodologies and Trends. *International Journal of Interactive Mobile Technologies (iJIM)*.
- Alshamrani, A. and Bahattab, A., 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, [online] 12(1), pp.106–111. Available at: <https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model>.
- Bouroumi, A. and Fajr, R., 2014. Collaborative and cooperative e-learning in higher education in Morocco: A case study. *International Journal of Emerging Technologies in Learning*, 9(1), pp.66–72.
- Bucher, B., 2020. Messaging App Usage Statistics Around The World | Messengerpeople. [online] MessengerPeople. Available at: <<https://www.messengerpeople.com/global-messenger-usage-statistics/>> [Accessed 15 August 2020].
- Cohen, J., 2020. Swine Flu Strain With Human Pandemic Potential Increasingly Found In Pigs In China. [online] Science | AAAS. Available at: <<https://www.sciencemag.org/news/2020/06/swine-flu-strain-human-pandemic-potential-increasingly-found-pigs-china>> [Accessed 4 August 2020].
- Fan, H., Li, K., Li, X., Song, T., Zhang, W., Shi, Y. and Du, B., 2019. CoVSCode: A novel real-time collaborative programming environment for lightweight IDE. *Applied Sciences (Switzerland)*, 9(21).
- Gentle, J., n.d. Sharejs – Live Concurrent Editing In Your App. [online] Sharejs.org. Available at: <<https://sharejs.org>> [Accessed 27 August 2020].

- Gill, V., 2020. Coronavirus: This Is Not The Last Pandemic. [online] BBC News. Available at: <<https://www.bbc.com/news/science-environment-52775386>> [Accessed 4 August 2020].
- Giurgiu, L., 2017. Microlearning an Evolving Elearning Trend. *Scientific Bulletin*, 22(1), pp.18–23.
- Göger, S., 2019. Why Programmers Need To Collaborate And How To Help Them. [online] Simple Programmer. Available at: <<https://simpleprogrammer.com/collaboration-software-development/>> [Accessed 14 August 2020].
- Goldman, M., Little, G. and Miller, R.C., 2011a. Collabode: Collaborative coding in the browser. *CHASE 2011 - Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering, Co-located with ICSE 2011*, 2011-Janua, pp.65–68.
- Goldman, M., Little, G. and Miller, R.C., 2011b. Real-time collaborative coding in a web IDE. *UIST'11 - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pp.155–164.
- GOYAL, S., 2012. E-Learning: Future of Education. *Journal of Education and Learning (EduLearn)*, 6(4), p.239.
- Khawas, C. and Shah, P., 2018. Application of Firebase in Android App Development-A Study. *International Journal of Computer Applications*, 179(46), pp.49–53.
- Kumar, G. and Bhatia, P.K., 2014. Comparative analysis of software engineering models from traditional to modern methodologies. *International Conference on Advanced Computing and Communication Technologies, ACCT*, pp.189–196.
- Kumar, M. and Rashid, E., 2018. An Efficient Software Development Life cycle Model for Developing Software Project. *International Journal of Education and Management Engineering*, 8(6), pp.59–68.
- Kurniawan, A., Soesanto, C. and Wijaya, J.E.C., 2015. CodeR: Real-time Code Editor Application for Collaborative Programming. *Procedia Computer Science*, [online] 59(Iccsci), pp.510–519. Available at: <<http://dx.doi.org/10.1016/j.procs.2015.07.531>>.
- Lauron, A.G., 2008. Fostering collaboration to enhance online instruction. *Turkish Online Journal of Distance Education*, 9(2), pp.109–121.
- Lawless, C., 2018. What Is Elearning? A Complete Guide For Your Business.

[online] LearnUpon. Available at: <<https://www.learnupon.com/blog/what-is-elearning/>> [Accessed 4 August 2020].

Lawson, H.A., 2004. The logic of collaboration in education and the human services. *Journal of Interprofessional Care*, 18(3), pp.225–237.

Li, C. and Lalani, F., 2020. The COVID-19 Pandemic Has Changed Education Forever. This Is How. [online] World Economic Forum. Available at: <<https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/>> [Accessed 4 August 2020].

Lu, O.H.T., Huang, J.C.H., Huang, A.Y.Q. and Yang, S.J.H., 2017. Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course. *Interactive Learning Environments*, [online] 25(2), pp.220–234. Available at: <<http://dx.doi.org/10.1080/10494820.2016.1278391>>.

Nieminen, A. and Lautamäki, J., 2013. *Collaborative coding environment on the web: A user study*. Available at: <http://cored.cs.tut.fi/pdf/tucs13_user.pdf>.

Nugroho, S., Hadi, S. and Hakim, L., 2017. Comparative Analysis of Software Development Methods between Parallel, V-Shaped and Iterative. *International Journal of Computer Applications*, 169(11), pp.7–11.

Mozilla Foundation, 2011. *TogetherJS*. [library] Mozilla. Available at: <<https://togetherjs.com/>> [Accessed 29 August 2020].

OpenJS Foundation, 2009. *Node.js (14.9.0)*. [computer program]. Available at: <<https://nodejs.org/en>> [Accessed 29 August 2020].

Patel, P., 2018. What Exactly Is Node.Js?. [online] freeCodeCamp.org. Available at: <<https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/>> [Accessed 29 August 2020].

Perry, Y., 2018. Applications Of Cloud Computing In Education: Trends And Challenges. [online] Cloud.netapp.com. Available at: <<https://cloud.netapp.com/blog/cloud-computing-in-education-trends-and-challenges>> [Accessed 5 August 2020].

Petersen, K., Wohlin, C., and Baca, D., 2009. The Waterfall Model In Large Scale Development.

Powell-Morse, A., 2016. Iterative Model: What Is It And When Should You Use It?. [online] Airbrake Blog. Available at: <<https://airbrake.io/blog/sdlc/iterative-model>> [Accessed 12 August 2020].

- Powell-Morse, A., 2017. Rational Unified Process: What Is It And How Do You Use It?. [online] Airbrake Blog. Available at: <<https://airbrake.io/blog/sdlc/rational-unified-process>> [Accessed 13 August 2020].
- Rantala, M., 2015. Real-time Collaborative Coding - Technical and Group Work Challenges.
- Sajewski, M., 2019. Top 6 Software Development Methodologies - Blog | Planview. [online] Planview Blog. Available at: <<https://blog.planview.com/top-6-software-development-methodologies/>> [Accessed 13 August 2020].
- Tang, K.H.D., 2020. Movement control as an effective measure against Covid-19 spread in Malaysia: an overview. *Journal of Public Health (Germany)*, pp.17–20.
- Tasevski, D., 2019. How are you writing a commit message?. [online] DEV Community. Available at: <<https://dev.to/puritanic/how-are-you-writing-a-commit-message-1ih7>> [Accessed 13 April 2021].
- Tavangarian, D., Leypold, M.E., Nölting, K., Röser, M. and Voigt, D., 2004. Is e-Learning the solution for individual learning? *Electronic Journal of e-Learnin. Electronic Journal of e-Learning*, [online] 2(2), pp.273–280. Available at: <<http://wwwra.informatik.uni-rostock.de>>.
- Yakubu, M.N. and Dasuki, S.I., 2018. Assessing eLearning systems success In Nigeria: An application of the Delone And Mclean information systems success model. *Journal of Information Technology Education: Research*, 17, pp.183–203.

APPENDICES

APPENDIX A: Comparison Between Numerous Malaysian Universities' e-Learning Platform

University	e-Learning Platform	Download Lecture Notes	Assignment Submission	Quiz	Instant Messaging	Discussion Forum	Email	Voice/Video call	Real-time Collaboration
UTAR	WBLE	✓	✓	✓	✗	✓	✗	✗	✗
UCSI	CourseNetworking	✓	✓	✓	✗	✗	✓	✗	✗
UM	Spectrum	✓	✓	✓	✓	✓	✗	✗	✗
TARUC	Google Classroom	✓	✓	✓	✗	✓	✗	✗	✗
MMU	MMLS	✓	✓	✓	✗	✓	✗	✗	✗
WOU	LDM	✓	✓	✓	✗	✓	✓	✓	✓
UTeM	U-learn	✓	✓	✓	✓	✓	✓	✓	✗
USM	eLearn	✓	✓	✓	✗	✓	✗	✗	✓
UKM	UKMFolio	✓	✓	✓	✓	✓	✓	✗	✗
UMT	My Nemo	✓	✓	✓	✗	✓	✓	✗	✓
UTHM	Author	✓	✓	✓	✗	✓	✗	✗	✗
UPSI	Myguru	✓	✓	✓	✗	✓	✗	✓	✓
SEGI	Black Board	✓	✓	✓	✗	✗	✓	✓	✗
Sunway	Elearn	✓	✓	✓	✗	✓	✗	✓	✗
UMP	UMP Kalam	✓	✓	✓	✓	✓	✓	✗	✓
UUM	Webex	✓	✓	✓	✓	✗	✓	✗	✓
UTM	UTM e-Learning	✓	✓	✓	✗	✓	✗	✗	✗
IACT	Pupilpod	✗	✓	✗	✗	✗	✗	✗	✗
*Melbourne	Canvas LMS	✓	✓	✓	✗	✓	✓	✗	✓

APPENDIX B: Questionnaire

University e-learning platform

Hello, I am Chong Mum Khong from Universiti Tunku Abdul Rahman. Firstly, I would like to express my gratitude to you for willing to give this survey a chance. This is a simple questionnaire regarding the e-learning platform that your university is using.

It will take around 1 minute to complete the survey.

Should you contact me (010-941 9641) if you have any enquiries. Thank you for your participation!

***Required**

1. Which university you are currently studying in? *

2. What is your faculty? *

e.g. Engineering, Science, Law, ... (depends on your university)

3. Gender *

Mark only one oval.

Female

Male

4. What is your year of study? *

Mark only one oval.

Foundation

First year

Second year

Third year

Fourth year

Fifth year onwards

5. What is the name of the e-learning platform? (Optional)

E-learning platform is typically, but not limited to a place where you can get access to teaching materials provided by your lecturer/faculty. For example, e-learning platform for UM is called Spectrum and MLLS for MMU

6. Pick the functions that your uni e-learning platform provides. (May choose more than one) *

Tick all that apply.

- Download lecture notes
- Assignment submission
- Quiz
- Instant messaging (like Whatsapp)
- Discussion forum
- Email
- Voice calls/Video calls

Other: _____

7. Pick the functions that you actually use. (based on the previous question) *

Tick all that apply.

- Download lecture notes
- Assignment submission
- Quiz
- Instant messaging (like Whatsapp)
- Discussion forum
- Email
- Voice calls/Video calls

Other: _____

8. Does your uni e-learning portal provide an online platform for students to discuss/edit assignment together in real-time? Something like Google Docs. *

Mark only one oval.

- Yes
 No

9. Is your uni portal user friendly? *

User friendly means that easy to navigate, intuitive and non-confusing.

Mark only one oval.

	1	2	3	4	5	
Not user friendly at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely user friendly

10. Rate the user interface for your uni e-learning platform. *

Mark only one oval.

	1	2	3	4	5	
Extremely ugly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely beautiful

11. Do you think e-Learning is helpful to you or your education? *

e-Learning here refers to learning through internet, images, videos...not necessary from your school.

Mark only one oval.

- Yes
 No

12. Do you think your university's e-learning platform is helpful to you or your education? *

Here, it is the platform that your university provides.

Mark only one oval.

Yes

No

13. Do you think that your uni's e-learning platform can be still improved in terms of functionalities and user interface? *

Mark only one oval.

Yes

No

APPENDIX C: Interview Question and Summary

There are three main questions asked in this short interview. The first one is asking about the difficulties to keep track of the students' progress. Dr Too. Agreed on the statement and noted that unless the students directly consult her, it is almost impossible to get to know their progress.

The second question is that do any of the student forget to submit before or submitted late. Dr. Too denies that students had forgotten to submit assignment but agrees that some students will occasionally submit the assignment late due to the process of compiling and arranging the files and folders.

The last question asks about the usefulness of getting the statistics of students' performance automatically. Dr. Too mentioned that it is tedious for them to visualise the performance as they need to key in the student's mark one by one using the Microsoft Excel application.

APPENDIX D: UAT Results

Tester Number: 1

Tester Name: Yin Siew Fai

Date Tested: 13/4/2021

Test Case #	UAT1	Start Time	1.00am	End Time	1.00am
Module	Login				
Test Scenario		Status		Comments	
1. Login with student role		PASS		-	
2. Logout the system		PASS		-	

Test Case #	UAT2	Start Time	1.01am	End Time	1.02am
Module	Pages				
Test Scenario		Status		Comments	
1. View notifications		PASS		-	
2. View courses		PASS		-	
3. View resources		PASS		-	
4. View assignments		PASS		-	

Test Case #	UAT3	Start Time	1.03am	End Time	1.05am
Module	Assignment Group				
Test Scenario		Status		Comments	
1. Create a group.		PASS		-	
2. Join an existing group		PASS		-	
3. Send for matchmaking.		PASS		-	
4. View assignment group details		PASS		-	

Test Case #	UAT4	Start Time	1.05am	End Time	1.06am
Module	Download				
Test Scenario		Status		Comments	
1. Download assignment question		PASS		-	
2. Download own assignment.		PASS		-	

Test Case #	UAT5	Start Time	1.07am	End Time	1.08am
Module	Workspace				
Test Scenario			Status	Comments	
1. View the group workspace.			PASS	-	
2. Code on editor.			PASS	-	
3. Send text messages.			PASS	-	

Test Case #	UAT6	Start Time	1.09am	End Time	1.11am
Module	Gantt Chart				
Test Scenario			Status	Comments	
1. View group Gantt Chart.			PASS	-	
2. Add an item.			PASS	-	
3. Edit an item.			PASS	-	
4. Delete an item			PASS	-	

Test Case #	UAT7	Start Time	1.11am	End Time	1.12am
Module	Discussion				
Test Scenario			Status	Comments	
1. View group discussion page.			PASS	-	
2. Add a discussion time.			PASS	-	

Test Case #	UAT8	Start Time	1.13am	End Time	1.14am
Module	Upload				
Test Scenario			Status	Comments	
1. View upload assignment page.			PASS	-	
2. Upload/Submit assignment.			PASS	-	

Tester Number: 2

Tester Name: Lim Jing Chun

Date Tested: 13/4/2021

Test Case #	UAT1	Start Time	1.45am	End Time	1.45am
Module	Login				
Test Scenario		Status		Comments	
1. Login with student role		PASS		-	
2. Logout the system		PASS		-	

Test Case #	UAT2	Start Time	1.46am	End Time	1.47am
Module	Pages				
Test Scenario		Status		Comments	
1. View notifications		PASS		-	
2. View courses		PASS		-	
3. View resources		PASS		-	
4. View assignments		PASS		-	

Test Case #	UAT3	Start Time	1.48am	End Time	1.52am
Module	Assignment Group				
Test Scenario		Status		Comments	
1. Create a group.		PASS		-	
2. Join an existing group		PASS		-	
3. Send for matchmaking.		PASS		-	
4. View assignment group details		PASS		-	

Test Case #	UAT4	Start Time	1.52am	End Time	1.53am
Module	Download				
Test Scenario		Status		Comments	
1. Download assignment question		PASS		Button on Info	
2. Download own assignment.		PASS		-	

Test Case #	UAT5	Start Time	1.53am	End Time	1.54am
Module	Workspace				
Test Scenario			Status	Comments	
1. View the group workspace.			PASS		
2. Code on editor.			PASS		
3. Send text messages.			PASS		

Test Case #	UAT6	Start Time	1.54am	End Time	1.56am
Module	Gantt Chart				
Test Scenario			Status	Comments	
1. View group Gantt Chart.			PASS		
2. Add an item.			PASS		
3. Edit an item.			PASS		
4. Delete an item			PASS		

Test Case #	UAT7	Start Time	1.57am	End Time	1.59am
Module	Discussion				
Test Scenario			Status	Comments	
1. View group discussion page.			PASS		
2. Add a discussion time.			PASS	Add button too small	

Test Case #	UAT8	Start Time	2.00am	End Time	2.02am
Module	Upload				
Test Scenario			Status	Comments	
1. View upload assignment page.			PASS		
2. Upload/Submit assignment.			PASS		

Tester Number: 3

Tester Name: Dr. Too Chian Wen

Date Tested: 13/4/2021

Test Case #	UAT1	Start Time	7.58pm	End Time	7.59pm
Module	Login				
Test Scenario			Status	Comments	
1. Login with lecturer role			PASS	-	
2. Logout the system			PASS	-	

Test Case #	UAT2	Start Time	8.00pm	End Time	8.01pm
Module	Pages				
Test Scenario			Status	Comments	
1. View notifications			PASS	-	
2. View courses			PASS	-	
3. View analytics			PASS	-	
4. View assignments			PASS	-	

Test Case #	UAT3	Start Time	8.01pm	End Time	8.05pm
Module	Assignments				
Test Scenario			Status	Comments	
1. Add new assignment			PASS	-	
2. Edit assignment			PASS	-	
3. Upload assignment question			PASS	-	

Test Case #	UAT4	Start Time	8.05pm	End Time	8.08pm
Module	Groups				
Test Scenario			Status	Comments	
1. View assignment groups list			PASS	-	
2. Join one of the group's workspace			PASS	-	
3. Type some code			PASS	-	
4. Send chat message.			PASS	-	

Test Case #	UAT5	Start Time	8.09pm	End Time	8.13pm
Module	Group				
Test Scenario	Status		Comments		
1. View one of the assignment group's details.	PASS		-		
2. Download group's assignment.	PASS		-		
3. View Gantt Chart page	PASS		-		
4. Add/Edit/Delete Gantt Chart	PASS		-		
5. View discussion booking page	PASS		-		

Tester Number: 4

Tester Name: Lim Willy

Date Tested: 13/4/2021

Test Case #	UAT1	Start Time	8.40pm	End Time	8.40pm
Module	Login				
Test Scenario			Status	Comments	
1. Login with student role			PASS	-	
2. Logout the system			PASS	-	

Test Case #	UAT2	Start Time	8.41pm	End Time	8.41pm
Module	Pages				
Test Scenario			Status	Comments	
1. View notifications			PASS	-	
2. View courses			PASS	-	
3. View resources			PASS	-	
4. View assignments			PASS	-	

Test Case #	UAT3	Start Time	8.43pm	End Time	8.45pm
Module	Assignment Group				
Test Scenario			Status	Comments	
1. Create a group.			PASS	-	
2. Join an existing group			PASS	-	
3. Send for matchmaking.			PASS	-	
4. View assignment group details			PASS	-	

Test Case #	UAT4	Start Time	8.46pm	End Time	8.47pm
Module	Download				
Test Scenario			Status	Comments	
1. Download assignment question			PASS	-	
2. Download own assignment.			PASS	-	

Test Case #	UAT5	Start Time	8.48pm	End Time	8.52pm
Module	Workspace				
Test Scenario			Status	Comments	
1. View the group workspace.			PASS	-	
2. Code on editor.			PASS	-	
3. Send text messages.			PASS	-	

Test Case #	UAT6	Start Time	8.52pm	End Time	8.55pm
Module	Gantt Chart				
Test Scenario			Status	Comments	
1. View group Gantt Chart.			PASS	-	
2. Add an item.			PASS	-	
3. Edit an item.			PASS	-	
4. Delete an item			PASS	-	

Test Case #	UAT7	Start Time	8.55pm	End Time	8.57pm
Module	Discussion				
Test Scenario			Status	Comments	
1. View group discussion page.			PASS	-	
2. Add a discussion time.			PASS	-	

Test Case #	UAT8	Start Time	8.57pm	End Time	8.58pm
Module	Upload				
Test Scenario			Status	Comments	
1. View upload assignment page.			PASS	-	
2. Upload/Submit assignment.			PASS	-	

Tester Number: 5

Tester Name: See Yee Choong

Date Tested: 13/4/2021

Test Case #	UAT1	Start Time	9.01pm	End Time	9.03pm
Module	Login				
Test Scenario			Status	Comments	
1. Login with student role			PASS	-	
2. Logout the system			PASS	-	

Test Case #	UAT2	Start Time	9.03pm	End Time	9.03pm
Module	Pages				
Test Scenario			Status	Comments	
1. View notifications			PASS	-	
2. View courses			PASS	-	
3. View resources			PASS	-	
4. View assignments			PASS	-	

Test Case #	UAT3	Start Time	9.04pm	End Time	9.06pm
Module	Assignment Group				
Test Scenario			Status	Comments	
1. Create a group.			PASS	-	
2. Join an existing group			PASS	-	
3. Send for matchmaking.			PASS	-	
4. View assignment group details			PASS	-	

Test Case #	UAT4	Start Time	9.06pm	End Time	9.08pm
Module	Download				
Test Scenario			Status	Comments	
1. Download assignment question			PASS	-	
2. Download own assignment.			PASS	-	

Test Case #	UAT5	Start Time	9.08pm	End Time	9.10pm
Module	Workspace				
Test Scenario			Status	Comments	
1. View the group workspace.			PASS	-	
2. Code on editor.			PASS	-	
3. Send text messages.			PASS	-	

Test Case #	UAT6	Start Time	9.11pm	End Time	9.12pm
Module	Gantt Chart				
Test Scenario			Status	Comments	
1. View group Gantt Chart.			PASS		
2. Add an item.			PASS	-	
3. Edit an item.			PASS	-	
4. Delete an item			PASS	-	

Test Case #	UAT7	Start Time	9.14pm	End Time	9.14pm
Module	Discussion				
Test Scenario			Status	Comments	
1. View group discussion page.			PASS	-	
2. Add a discussion time.			PASS	-	

Test Case #	UAT8	Start Time	9.15pm	End Time	9.16pm
Module	Upload				
Test Scenario			Status	Comments	
1. View upload assignment page.			PASS	-	
2. Upload/Submit assignment.			PASS	-	

Tester Number: 6

Tester Name: Ooi Thean Chun

Date Tested: 14/4/2021

Test Case #	UAT1	Start Time	12.00pm	End Time	12.00pm
Module	Login				
Test Scenario		Status		Comments	
1. Login with student role		PASS		-	
2. Logout the system		PASS		-	

Test Case #	UAT2	Start Time	12.01pm	End Time	12.01pm
Module	Pages				
Test Scenario		Status		Comments	
1. View notifications		PASS		-	
2. View courses		PASS		-	
3. View resources		PASS		-	
4. View assignments		PASS		-	

Test Case #	UAT3	Start Time	12.02pm	End Time	12.03pm
Module	Assignment Group				
Test Scenario		Status		Comments	
1. Create a group.		PASS		-	
2. Join an existing group		PASS		-	
3. Send for matchmaking.		PASS		-	
4. View assignment group details		PASS		-	

Test Case #	UAT4	Start Time	12.03pm	End Time	12.03pm
Module	Download				
Test Scenario		Status		Comments	
1. Download assignment question		PASS		-	
2. Download own assignment.		PASS		-	

Test Case #	UAT5	Start Time	12.04pm	End Time	12.06pm
Module	Workspace				
Test Scenario		Status		Comments	
1. View the group workspace.		PASS		-	
2. Code on editor.		PASS		-	
3. Send text messages.		PASS		-	

Test Case #	UAT6	Start Time	12.06pm	End Time	12.08pm
Module	Gantt Chart				
Test Scenario		Status		Comments	
1. View group Gantt Chart.		PASS		-	
2. Add an item.		PASS		-	
3. Edit an item.		PASS		-	
4. Delete an item		PASS		-	

Test Case #	UAT7	Start Time	12.08pm	End Time	12.08pm
Module	Discussion				
Test Scenario		Status		Comments	
1. View group discussion page.		PASS		-	
2. Add a discussion time.		PASS		Add padding to the button.	

Test Case #	UAT8	Start Time	12.08pm	End Time	12.09pm
Module	Upload				
Test Scenario		Status		Comments	
1. View upload assignment page.		PASS		-	
2. Upload/Submit assignment.		PASS		-	

Tester Number: 7

Tester Name: Ong Wi Yi

Date Tested: 14/4/2021

Test Case #	UAT1	Start Time	1.19pm	End Time	1.19pm
Module	Login				
Test Scenario		Status		Comments	
1. Login with student role		PASS		-	
2. Logout the system		PASS		-	

Test Case #	UAT2	Start Time	1.20pm	End Time	1.20pm
Module	Pages				
Test Scenario		Status		Comments	
1. View notifications		PASS		-	
2. View courses		PASS		-	
3. View resources		PASS		-	
4. View assignments		PASS		-	

Test Case #	UAT3	Start Time	1.21pm	End Time	1.22pm
Module	Assignment Group				
Test Scenario		Status		Comments	
1. Create a group.		PASS		-	
2. Join an existing group		PASS		-	
3. Send for matchmaking.		PASS		-	
4. View assignment group details		PASS		-	

Test Case #	UAT4	Start Time	1.22pm	End Time	1.22pm
Module	Download				
Test Scenario		Status		Comments	
1. Download assignment question		PASS		-	
2. Download own assignment.		PASS		-	

Test Case #	UAT5	Start Time	1.23pm	End Time	1.25pm
Module	Workspace				
Test Scenario			Status	Comments	
1. View the group workspace.			PASS	-	
2. Code on editor.			PASS	-	
3. Send text messages.			PASS	-	

Test Case #	UAT6	Start Time	1.25pm	End Time	1.26pm
Module	Gantt Chart				
Test Scenario			Status	Comments	
1. View group Gantt Chart.			PASS	-	
2. Add an item.			PASS	-	
3. Edit an item.			PASS	-	
4. Delete an item			PASS	-	

Test Case #	UAT7	Start Time	1.25pm	End Time	1.26pm
Module	Discussion				
Test Scenario			Status	Comments	
1. View group discussion page.			PASS	-	
2. Add a discussion time.			PASS	-	

Test Case #	UAT8	Start Time	1.26pm	End Time	1.26pm
Module	Upload				
Test Scenario			Status	Comments	
1. View upload assignment page.			PASS	-	
2. Upload/Submit assignment.			PASS	-	

APPENDIX E: Usability Test Responses

Tester Number: 1

Section I

Please rate the following statements.

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1. I think that I would like to use this website/system for viewing information and related matter.				X	
2. I found the website/system unnecessarily complex.	X				
3. I thought the website/system was easy to use					X
4. I think that I would need the support of a technical person to be able to use this website/system.	X				
5. I found the various functions in this system were well integrated				X	
6. I thought there was too much inconsistency in this website/system.	X				
7. I would imagine that most people would learn to use this website/system very quickly.				X	
8. I found the system/website very awkward to use.	X				
9. I felt very confident using the system/website.					X

10. I needed to learn a lot of things before I could get going with this website/system.	X				
------------------------------------------------------------------------------------------	---	--	--	--	--

Section II

1. Which part of the system/application you like the most?
2. Which part of the system/application you like the least?
3. What is your question on the system?
4. What is your comment/recommendations on the system?

Tester Number: 2

Section I

Please rate the following statements.

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1. I think that I would like to use this website/system for viewing information and related matter.					✓
2. I found the website/system unnecessarily complex.					✓
3. I thought the website/system was easy to use					✓
4. I think that I would need the support of a technical person to be able to use this website/ system.		✓			
5. I found the various functions in this system were well integrated					✓
6. I thought there was too much inconsistency in this website/ system.		✓			
7. I would imagine that most people would learn to use this website/system. very quickly.					✓
8. I found the system/website very awkward to use.	✓				

9. I felt very confident using the system/website.					✓
10. I needed to learn a lot of things before I could get going with this website/system.		✓			

Section II

1. Which part of the system/application you like the most?

Ans: Coding and chatting part.

2. Which part of the system/application you like the least?

Ans: none.

3. What is your question on the system?

Ans: none.

4. What is your comment/recommendations on the system?

Ans: none.

Tester Number: 3

Section I

Please rate the following statements.

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1. I think that I would like to use this website/system for viewing information and related matter.				✓	
2. I found the website/system unnecessarily complex.		✓			
3. I thought the website/system was easy to use				✓	
4. I think that I would need the support of a technical person to be able to use this website/ system.		✓			
5. I found the various functions in this system were well integrated				✓	
6. I thought there was too much inconsistency in this website/ system.		✓			
7. I would imagine that most people would learn to use this website/system. very quickly.					✓
8. I found the system/website very awkward to use.	✓				
9. I felt very confident				✓	

using the system/website.					
10. I needed to learn a lot of things before I could get going with this website/system.	✓				

Section II

1. Which part of the system/application you like the most?

-source code editing and statistic analysis part

2. Which part of the system/application you like the least?

-NIL

3. What is your question on the system?

-NIL

4. What is your comment/recommendations on the system?

-provide some tooltips for certain icon so that user can understand better the function or feature easily.

Tester Number: 5

Section I

Please rate the following statements.

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1. I think that I would like to use this website/system for viewing information and related matter.					√
2. I found the website/system unnecessarily complex.	√				
3. I thought the website/system was easy to use				√	
4. I think that I would need the support of a technical person to be able to use this website/ system.		√			
5. I found the various functions in this system were well integrated					√
6. I thought there was too much inconsistency in this website/ system.	√				
7. I would imagine that most people would learn to use this website/system. very quickly.				√	
8. I found the system/website very awkward to use.	√				

9. I felt very confident using the system/website.					√
10. I needed to learn a lot of things before I could get going with this website/system.	√				

Section II

1. Which part of the system/application you like the most?

-Attractive system design and high consistency of module placements

2. Which part of the system/application you like the least?

-I'm satisfied with every part of the system.

3. What is your question on the system?

-

4. What is your comment/recommendations on the system?

-For the code editor, it should support more programming languages for students to check.

Tester Number: 6

Section I

Please rate the following statements.

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1. I think that I would like to use this website/system for viewing information and related matter.					√
2. I found the website/system unnecessarily complex.		√			
3. I thought the website/system was easy to use				√	
4. I think that I would need the support of a technical person to be able to use this website/ system.		√			
5. I found the various functions in this system were well integrated					√
6. I thought there was too much inconsistency in this website/ system.	√				
7. I would imagine that most people would learn to use this website/system. very quickly.		√			
8. I found the system/website very awkward to use.	√				
9. I felt very confident					√

using the system/website.					
10. I needed to learn a lot of things before I could get going with this website/system.		√			

Section II

1. Which part of the system/application you like the most?

The code editor

2. Which part of the system/application you like the least?

No

3. What is your question on the system?

No

4. What is your comment/recommendations on the system?

Maybe could provide some simple tutorial to teach student on using the system

Tester Number: 7

Section I

Please rate the following statements.

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1. I think that I would like to use this website/system for viewing information and related matter.					✓
2. I found the website/system unnecessarily complex.		✓			
3. I thought the website/system was easy to use				✓	
4. I think that I would need the support of a technical person to be able to use this website/system.	✓				
5. I found the various functions in this system were well integrated					✓
6. I thought there was too much inconsistency in this website/system.	✓				
7. I would imagine that most people would learn to use this website/system very quickly.					✓
8. I found the system/website very awkward to use.	✓				
9. I felt very confident using the system/website.					✓
10. I needed to learn a lot of things before I could get going	✓				

with this website/system.					
---------------------------	--	--	--	--	--

Section II

1. Which part of the system/application you like the most?

The coding page, it feels like a nice ide to use

2. Which part of the system/application you like the least?

Need to hover only able to know the functionality of the icon

3. What is your question on the system?

=

4. What is your comment/recommendations on the system?

Very nice application for students