SAGE: A COMMUNITY EMPOWERED UNIVERSITY E-LEARNING APPLICATION

RONALD LIM SHENG WEI

Bachelor of Computer Science
(Software Engineering) with Honours

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	RONALD LIM SHENG WEI
Date of Birth	:	
Title	:	SAGE: A COMMUNITY EMPOWERED UNIVERISITY E-LEARNING APPLICATION
Academic Session	:	2019/2023

I declare that this thesis is classified as:

□	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
□	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☒	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

____
(Student's Signature)

____

Date: 5th February 2023

____
(Supervisor's Signature)

____
Dr. Nabilah Filzah binti Mohd Radzuan
Date: 10/02/2023

Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering) with Honours.

____
(Supervisor's Signature)

Full Name : Dr. Nabilah Filzah binti Mohd Radzuan
Position : Senior Lecturer
Date : 10/02/2023

UMP Universiti Malaysia PAHANG
Engineering • Technology • Creativity

STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student’s Signature)

Full Name : Ronald Lim Sheng Wei
ID Number : CB19052
Date : 5 February 2023

SAGE: A COMMUNITY EMPOWERED UNIVERISITY E-LEARNING APPLICATION

RONALD LIM SHENG WEI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

.

FEBRUARY 2023

ACKNOWLEDGEMENTS

Throughout the journey of writing this thesis, I am grateful to have received a tremendous amount of support and assistance from families, friends, and educators alike.

First and foremost, I would express my sincere gratitude to my supervisor, Dr. Nabilah Filzah binti Mohd Radzuan, whose invaluable and knowledgeable insight and professional opinion has propelled my work to even greater heights. Her dedication, heartening and selflessness in times of need has deeply motivated me to persevere on and complete the project.

In addition, I would like to thank my parents for their constant rapport throughout my academic studies. They have shown great support in times of need both financially and in spirit.

Finally, I would like to show my appreciation to my three dear friends, Tan Chee Kin, Tan Yi Wee, and Wong Sung Sum who provided their critical criticism of my written work as well and their subtle encouragement in times of adversity.

ABSTRAK

Paradigma telah berubah untuk sektor pendidikan tinggi kerana ia memasuki era baharu, di mana universiti beradaptasi dengan norma baharu iaitu pembelajaran dalam talian atau pembelajaran hibrid. Dalam proses ini, elemen interaksi antara manusia dalam pembelajaran telah terjejas kerana pelajar dianggap sebagai individu dan bukan sebagai sebuah komuniti. Oleh hal sedemikian, pelajar memerlukan platform yang memberikan mereka peluang untuk berfungsi sebagai komuniti untuk menggalakkan pembelajaran, perbincangan dan perkongsian akademik bersama-sama sebagai satu komuniti akademik.

ABSTRACT

The paradigm has shifted for the tertiary education sector as it is diving into a new era, where universities are embracing a new norm which is online learning or hybrid learning. In this process, the human interaction element of learning is lost as students are treated as an individual as opposed to a community. Now more than ever, students require a platform where they can function as a community to encourage mutual learning, discussions and sharing as an academic community together.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

MOODLE	Modular Object-Oriented Dynamic Learning Environment
UAT	User Acceptance Testing
SRS	Software Requirement Specification
SDD	Software Design Document

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Most educational systems have been forced to embrace alternatives to face-to-face teaching and learning in response to the COVID-19 situation. Numerous educational systems have transitioned to activities online to maintain instruction in the event of classroom cancellations. When compared to the option of not attending university, online education has proven a critical tool for sustaining skill development during critical times of the pandemic.

Before COVID-19, according to Research and Markets, the online education industry is expected to reach $350 billion by 2025; however, these figures may be adjusted after studying the influence of COVID-19 on the online education business. The versatility of online education is one embraced by many. (Koksal, 2020). Hence, it is clear that there is a demand in the field of online education.

Online education brings about flexibility to not only students but educators alike. The ability to be able to follow online courses at one's own pace is something that has benefited education seekers around the globe. Considering the recent events, even world-renowned universities globally such as Stanford University and Harvard University have started to offer their courses online, in subjects ranging from Computer Science, Mathematics, and also engineering, embracing digital learning as the next step in the history of education.

However, Agarwal and Dewan (2022) asserted that there was insufficient time to educate academicians and students on how to conduct online teaching-learning processes or to create best practices and standard operating procedures. Online education, which was first marketed as a cure for all ills, now requires more thought to be put into it. Due to the sloppy implementation and execution, it brought about chaos and distraught when it was first introduced.

Furthermore, there are still concerns that access to online learning facilities is different for students of different socioeconomic backgrounds. According to the Department of Statistics Malaysia's general report published in 2021, the Tawau region's mean household income is less than half that of the national average, at RM7,901. As a result, residents are likely to face a greater digital gap than residents in other locations. (Sarimah, 2021)

In conclusion, as promising as it seems, online learning still has its flaws and hurdles to overcome. As a solution to the problem, educational institutions have adopted blended learning and hybrid learning widely to reap benefits from both online learning and traditional learning.

1.2 Problem Statement

Since the pandemic shook the world, the educational sector has taken a large hit as it has always depended on traditional pedagogical methods to deliver educational content to its seekers from all around the world. Transitioning from a traditional based teaching method to a fully digital learning mode of learning was done out of necessity due to the threat of COVID-19. Digital learning certainly has come across as attractive to forward-thinking educators as they embrace the process that takes place in digital learning.

Transitioning into a post-pandemic future, the remnants of online learning still has a profound impact on educators globally. Admittedly, online learning is less superior to a face-to-face method of education. However, there are some lessons we can learn from online learning. A solution of digital learning paired with the time-proven methods of the trade, can give birth to a technology-enhanced form of teaching and learning, with the consideration flexibility, empowerment, professionalisation and strategic decision making. (Rapanta et al., 2021).

Despite the flexibility, convenience, and practicality of online learning, there are hidden challenges that come with online learning. The lack of interaction is a significant drawback of the adoption of online learning. Teachers and students are segregated throughout the online learning process. (Watson et al., 2012 as cited in Agarwal & Dewan,2022). In many circumstances, input from students and professors is derailed in the online mode. Even acquiring fundamental information might become difficult at times owing to a lack of human connection. (Bodzin & Park,2000 as cited in Agarwal & Dewan,2022).

In addition to that, an important aspect of learning which is a peer-to-peer learning is not practised as commonly since the adoption of online learning. This can be attributed to the lack of interactions between teachers and students during the online learning process. The sharing of personal opinions and understandings of a subject matter can verify and validate one's depth of knowledge for the parties involved. To further highlight the importance of peer learning, Choi et al. (2021) asserted that peer learning strategies that supplement students' individual learning experiences with peer evaluation are successful at enhancing students' accountability and capacity to acquire professional skills.

In addition, the restrictions on gatherings and the need for social distancing have made it difficult for students to socialize in person. As a result, many students have reported feeling isolated and disconnected from their peers (Li et al., 2021).

The lack of face-to-face interaction can have negative consequences for student well-being and academic success. According to a review published in the Journal of Positive Psychology, social connectedness is an important predictor of mental health and well-being (Shamionov et al., 2021). In addition, research has shown that social support from peers can enhance academic performance (Dupont et al., 2015).

Even though current solutions in the market provide innovative solutions to the issue of the transfer of knowledge, and improving educator to student interactions, however, there is a lack of emphasis towards improving the interactions between students and encouraging peer learning through academic discussions on e-learning applications.

Students undergoing their education are often not highlighted as a community but evaluated as individuals.

Furthermore, students do not have a platform where they can come together to discuss a solution to a question aside from asking lecturers. Every student is on their own as they do not have a past repository to search for the answers. Hence there is a dire need to provide students with a voice to be able to take part in academic discussions together to develop a knowledge-seeking culture among Malaysian tertiary students.

In conclusion, the significant element of allowing students to function as a community of knowledge seekers must be embedded into e-learning applications as a way forward for the online learning field.

1.3 Objectives

i. To determine the existing mobile applications and design a new community empowered e-learning application for Malaysian universities.
ii. To develop a mobile application for e-learning and academic discussions among tertiary students and lecturers digitally.
iii. To validate the functionality of the developed e-learning application.

1.4 Scope

- User Scope
  i. Tertiary students undergoing their studies at public Malaysian universities.
  ii. Lecturers

- System Scope
  i. Covers mobile phone based e-learning tools for community empowered problem-solving, retrieval of educational resources, and lesson planning.

- Development Scope
    1. Contains multimedia elements such as sound, text, and graphics.
    2. Using flutter as the framework, Firebase as the cloud storage, and GitHub for version control.

1.5 Significance of The Project

i. University Students
Students can gain access to educational resources and discuss problems together as a community which improves student to student interaction and encourages peer learning.

ii. Lecturers
Lecturers can plan their lectures for the entire semester for their courses with activities such as providing lecture links and educational resources. They can enable chat groups as well within their course sections.

1.6 Thesis Organization

This thesis contains five chapters. The first chapter will explain the project's introduction. The introduction describes the context of the project and how it might be utilized to tackle real-world challenges. It also contains the project's purpose and objectives, scope and significance, which dictates the development of the project's outcome.

In chapter two, a literature review of the similar systems is presented. Each existing system has its own unique characteristics, and functions. The advantages and disadvantages of each existing system are analysed in depth as well.

In chapter three, the development process of the project is discussed. The project requirements are described in depth in Appendix A, which is the Software Requirement Specification (SRS), whereas Appendix B, which is the Software Architecture Description, describes the system design in detail (SDD).

Chapter four talks about the implementation of the project from beginning to end. It also talks about how the project’s testing is conducted and how it impacts the future of the project. The Usability Test form is attached in Appendix B while the User Acceptance Test (UAT) form is attached in Appendix C.

Finally, chapter five describes the constraints faced when developing the project. Mentions of future works are also briefly touched on to further improve the project in the near future to increase the practicality and functionality of the system.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explains about the current university e-learning systems that are available on the market. A detailed analysis is done on the systems to pinpoint the strengths and weaknesses of the current systems on the market.

To fulfil the demand for online learning management systems on the market, there are many on the shelf solutions developed to cater for the increase in demand for digital learning experiences. The three main systems that will be analysed in depth will be KALAM, Google Classroom as well as Edmodo.

2.2 KALAM

2.2.1 Discussion

KALAM is an online learning management system that is developed and managed by Universiti Malaysia Pahang (UMP) on top of the MOODLE platform. It is designed as a platform to help students access course materials and lecturers to conduct their courses. It utilises the open-source learning management system (LMS) framework provided by Moodle to develop the system. Many well-known universities are utilising this platform such as Monash Malaysia, The University of Nottingham Malaysia, and Curtin University Malaysia.

Since KALAM is a system based on the Moodle framework, it supports modern and widely used browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge on the desktop platform. Additionally, for mobile platforms, it can also accommodate the Mobile Safari and Google Chrome mobile browsers. Furthermore, Moodle also has a dedicated mobile application on the world’s two most popular mobile operating systems namely, Android and IOS. (Moodle, 2021). This makes KALAM widely accessible to users who do not have many internet-enabled devices.

KALAM is mainly written in the programme development language of PHP, JavaScript as well as SQL databases. It supports the use of various well-known databases such as depicted in Table 2.1. (Moodle, 2021)

Table 2.1 Supported Databases

Database	Minimum Version	Recommended
PostgreSQL	9.6	Latest
MySQL	5.7	Latest
MariaDB	10.2.29	Latest
Microsoft SQL Server	2017 (increased since Moodle 3.10)	Latest
Oracle Database	11.2	Latest

Source: Moodle (2021).

The mobile platform of KALAM (Moodle) uses technologies such as the angular and ionic framework to develop the application. The whole communication between the app and a site occurs through a layer of web services. Each time a user logs into the app, a new session starts, and that session is what the idea of a "site" embodies in the application. As a result of this, you might log several times into the same site and from the point of view of the mobile app, those would be separate sites. (Moodle, n.d.)

There are many advantages to the adoption of the Moodle framework for KALAM. Academic professionals and students are able to communicate in real-time, and students receive prompt responses. Moodle also makes document management and editing simple. Chung and Ackerman (2015) substantiated this argument by stating that students believe MOODLE to be user-friendly. After that, the process of creating backup copies and restoring data is straightforward. Grades can be exported to spreadsheets.

Finally, access to archived material from other academic personnel is made simple. (Ayanda, 2020, as cited in Chicioreanu & Cosma, 2017). On the technical side, it is based on Moodle, an open-source platform managed by an open-source community. Hence there are no charges incurred for the use of its platform. Moodle employs a modular system that supports many plugins that can enhance its functionalities. Hence, Moodle supports smooth communication, text formatting and management, data exporting and the ability to add many plugins.

In spite of the amazing strengths that KALAM possesses, there are some downsides to KALAM. There is no genuine guarantee that learners studied the material assigned to them. Academic personnel sometimes struggle to assess students' talents and capabilities in areas such as creativity and critical thinking. Copying and pasting can be used to complete tasks. Finally, there is no certainty regarding the results of the final testing. (Ayanda, 2020, as cited in Petrovici & Ciobanu, 2016). Thus, KALAM struggles to evaluate a learner’s cognitive abilities through conventional means.

2.2.2 Graphical User Interface and Functionality

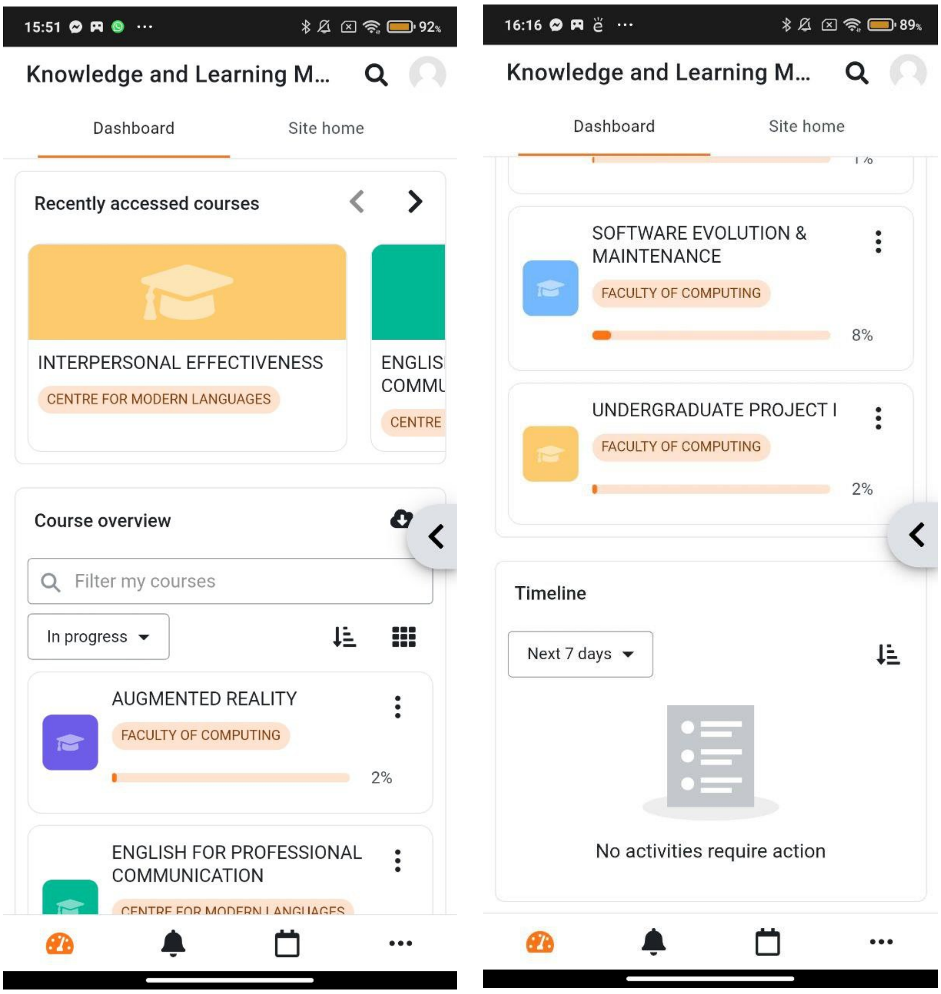
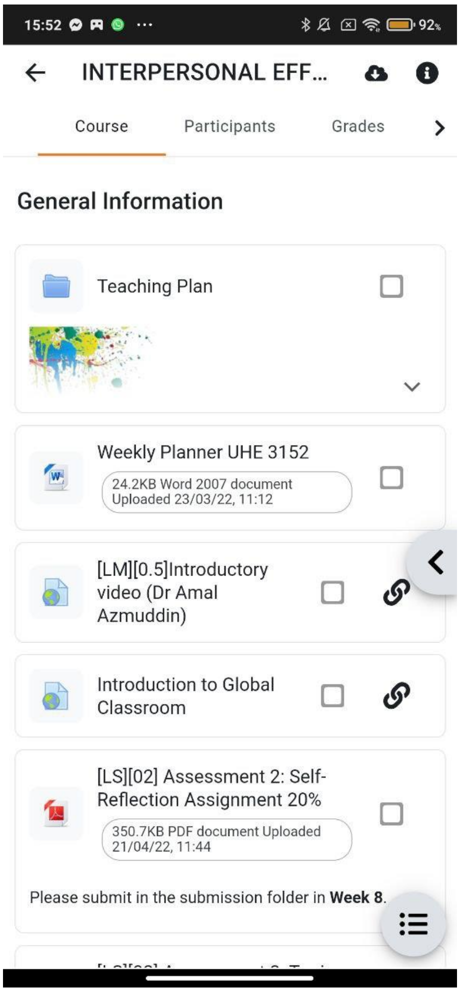

15:51
92%
Knowledge and Learning M...
Dashboard
Site home
Recently accessed courses
INTERPERSONAL EFFECTIVENESS
CENTRE FOR MODERN LANGUAGES
ENGLIS
COMMU
CENTRE
Course overview
Filter my courses
In progress
AUGMENTED REALITY
FACULTY OF COMPUTING
2%
ENGLISH FOR PROFESSIONAL COMMUNICATION
CENTRE FOR MODERN LANGUAGES
16:16
89%
Knowledge and Learning M...
Dashboard
Site home
SOFTWARE EVOLUTION & MAINTENANCE
FACULTY OF COMPUTING
8%
UNDERGRADUATE PROJECT I
FACULTY OF COMPUTING
2%
Timeline
Next 7 days
No activities require action

Figure 2.1 KALAM Dashboard

Figure 2.1 depicts the dashboard of KALAM. In the figure, there is the "recently accessed" courses which show the user's last accessed course contents for the convenience of the user. Next, there is also the course overview in which the user can view all courses undertaken by the student for the current semester. Furthermore, there is a timeline in which the user can view upcoming quizzes and assignment due dates.



15:52 ··· 92%

INTERPERSONAL EFF...

Course | Participants | Grades

General Information

Teaching Plan □

Weekly Planner UHE 3152 □
24.2KB Word 2007 document Uploaded 23/03/22, 11:12

[LM][0.5]Introductory video (Dr Amal Azmuddin) □

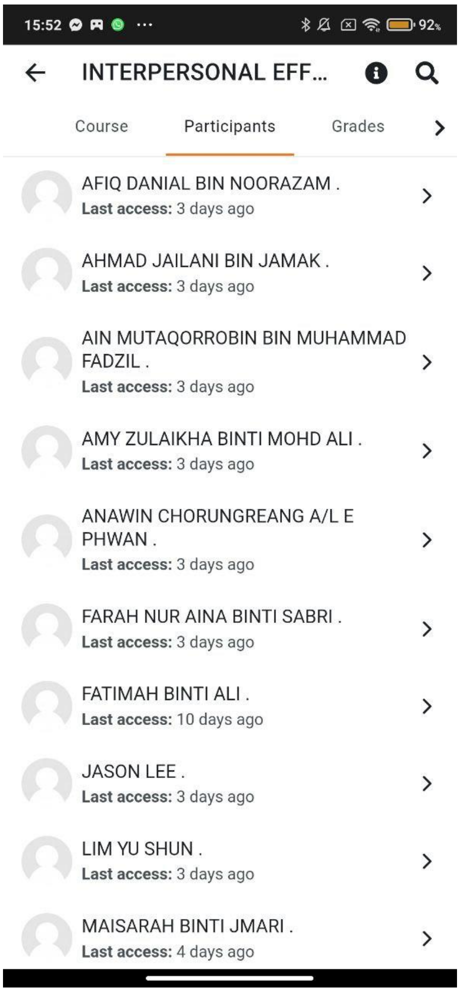
Introduction to Global Classroom □

[LS][02] Assessment 2: Self-Reflection Assignment 20% □
350.7KB PDF document Uploaded 21/04/22, 11:44

Please submit in the submission folder in **Week 8**.

Figure 2.2 Course Page

Figure 2.2 depicts the course contents page. This section allows students to access educational content provided by the course lecturer. For instance, students can take quizzes, write on a discussion board, answer questionnaires, download educational resources and upload assignments. Students can also view video content based on what is provided by the lecturer. The educational resources are divided the allocated weeks. For example, week 1 ranges from 1st March 2021 to 5th March 2021, excluding the weekends. During this period, the lecturer uploads specific educational reading material, quizzes, or assignment upload links.



AFIQ DANIAL BIN NOORAZAM .
Last access: 3 days ago

AHMAD JAILANI BIN JAMAK .
Last access: 3 days ago

AIN MUTAQORROBIN BIN MUHAMMAD FADZIL .
Last access: 3 days ago

AMY ZULAIKHA BINTI MOHD ALI .
Last access: 3 days ago

ANAWIN CHORUNGREANG A/L E PHWAN .
Last access: 3 days ago

FARAH NUR AINA BINTI SABRI .
Last access: 3 days ago

FATIMAH BINTI ALI .
Last access: 10 days ago

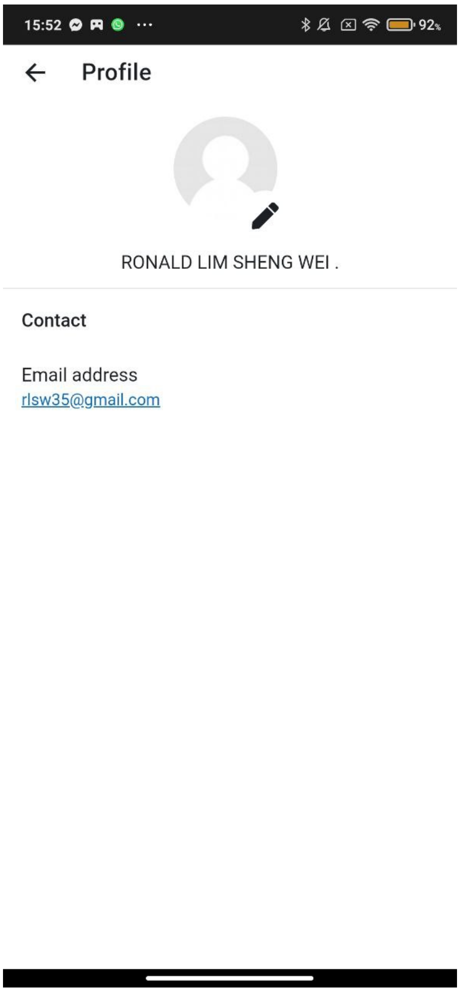
JASON LEE .
Last access: 3 days ago

LIM YU SHUN .
Last access: 3 days ago

MAISARAH BINTI JMARI .
Last access: 4 days ago

Figure 2.3 Course Participants

Figure 2.3 depicts the course participants in the selected course. From here, we can see when the students have last accessed the course.



RONALD LIM SHENG WEI .

Contact

Email address
rlsw35@gmail.com

Figure 2.4 Profile Settings Interface

Figure 2.4 shows the profile page of KALAM. From this page, there are user and contact details. There is also an option to change the profile picture.

2.3 Google Classroom

2.3.1 Discussion

Google Classroom is a product developed by the Google company in 2014 as Google's one-stop solution to digital learning classes for educators. It makes use of the existing productivity suite of Google, including Google Docs, Google Slides, and Google Sheets and incorporates them into a digital learning application – Google Classroom. What distinguishes Google Classroom from the standard Google Drive experience is the instructor interface, which Google has engineered for how instructors and students think and interact. (Okmawati, 2020).

Google states that its product, Google Workspace for Education, which includes Google Classroom, will work on the latest version of modern browsers such as Google Chrome, Firefox, as well as Safari and Microsoft Edge. Google classroom is also supported on android devices with Android 5.0 Lollipop or later and iPhones and iPads with IOS 11 or later.

First and foremost, Google Classroom is a free application that may be utilised by any university that lacks the means to develop its own learning management system (Learning Management System). It alleviates instructors' administrative burdens and aids in classroom management. Additionally, it contributes to the improvement of student-teacher contact and communication. (Azhar & Iqbal, 2018). Google Classroom has the potential to save a significant amount of time for both students and teachers due to the ease with which it can be set up and used. (Ketut Sudarsana et. al., 2019).

It requires no paid subscriptions to have full access to the basic functionalities in the application. Next, it supports cross-platform access because it has a dedicated mobile application as well as a web-based application. On top of that, it is integrated within Google's own ecosystem of applications. This means that important word processing, presentation slide making as well as managing excel sheets can be done seamlessly on the cloud using Google's suite of applications for assignments and homework with just an Internet connection.

The main medium of communication for Google Classroom users is over the "stream". A stream can be compared to a news feed as found in Facebook. Each class has a stream which allows educators to put out announcements, educational content, and graphical content. Students can provide responses by leaving comments on the posts on the stream. As compared to instant messaging, communication over a stream encourages users to avoid unnecessary conversations and texts.

Not to mention, it offers a feedback function for assignments. This allows educators to seamlessly provide their students with immediate feedback or comments to improve their work. Google Classroom also provides many access controls such as making posts read-only. Overall, Google's integration of existing productivity applications dramatically improves Google Classroom's capability as an educational tool for users with minimal technical background.

Based on the research findings of Okmawati (2020), from the perspective of effective communications based on the theory presented by Hardjana (2003), Okmawati (2020) demonstrated the effective of Google Classroom during the pandemic. From the perspective of the effectiveness of the message recipient, the message recipient was determined to be in accordance with the intended receivers. This implies that when the instructor desired to distribute information or assign a task, he or she did so directly on the accounts of students who are bound by the learning process, ensuring that the postings were instantly visible to the students.

On the other hand, the downsides of using a Google developed application involves the requirement of a Google account to access the services provided by Google Classroom as well as its suite of productivity tools. Next, the interface of Google Classroom is not user-friendly. According to Azhar and Iqbal (2018), their survey indicated that a lot of teachers initially struggled to adjust to the operation of Google Classroom. On the other hand, it also lacks the functionality for real-time communication between educators and students. Users are required to refresh the page of wait for a few seconds for new comments or posts to appear. Hence, Google Classroom requires some improvements to the system's responsiveness and user interface.

2.3.2 Graphical User Interface and Functionalities

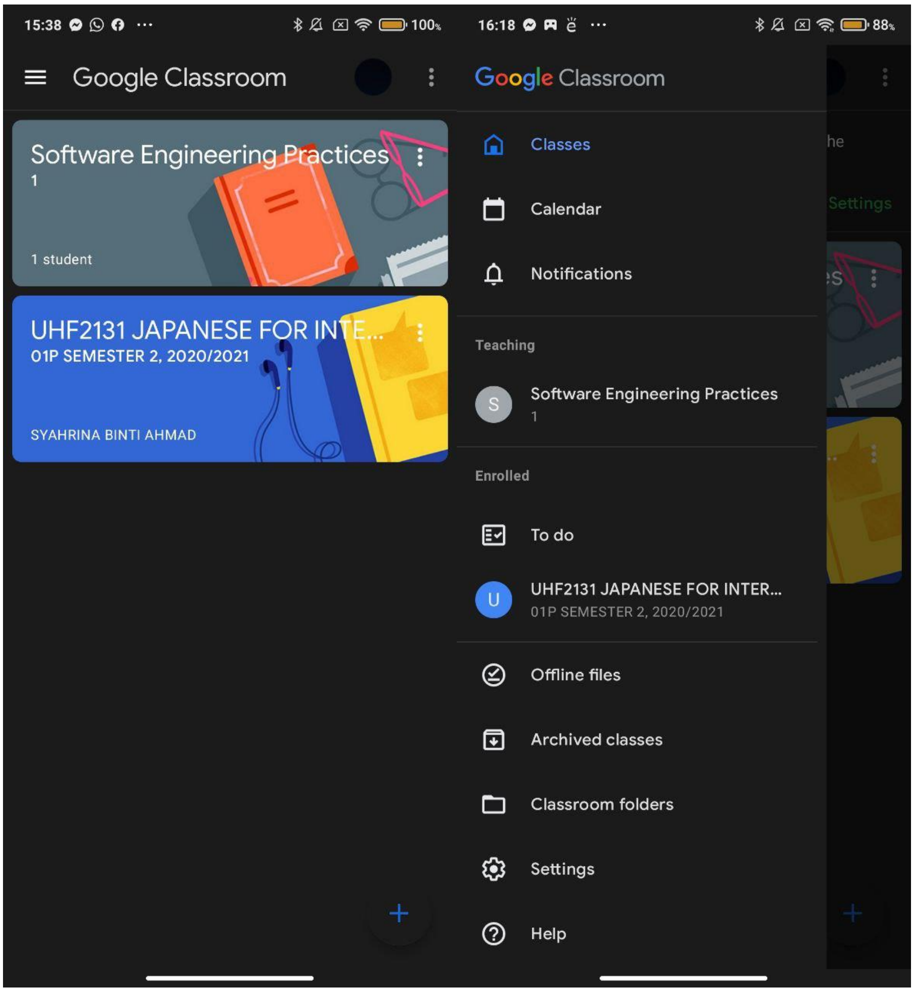
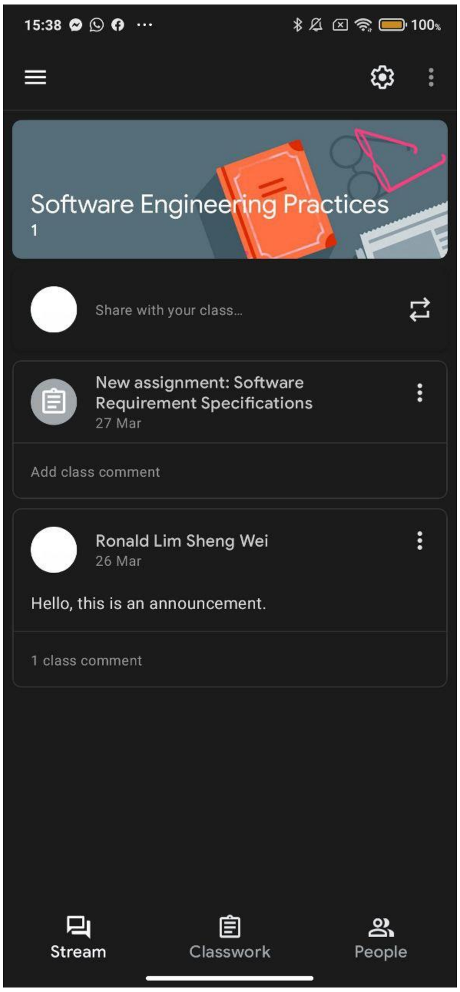


Figure 2.5 Dashboard Interface

Figure 2.5 shows the dashboard of Google Classroom. In this interface, users can view the classes that they are teaching or the classes that they are enrolled in. Additionally, they can access tools such as a to-do list, to-review list as well as Google Calendar to plan their schedules.



Software Engineering Practices
1

Share with your class...

New assignment: Software Requirement Specifications
27 Mar

Add class comment

Ronald Lim Sheng Wei
26 Mar

Hello, this is an announcement.

1 class comment

Figure 2.6 Class Interface

Figure 2.6 shows the class interface of a new class. In this interface, the creator of the classroom can obtain the class code for other users to join via a 7 alphanumerical code, post announcements which are not limited to text, but supports links, and images as well. Finally, there is a functionality to reply to comments as well. On the bottom navigation bar, there are a few tabs that can be accessed such as the class's coursework, class attendees and also assignments.

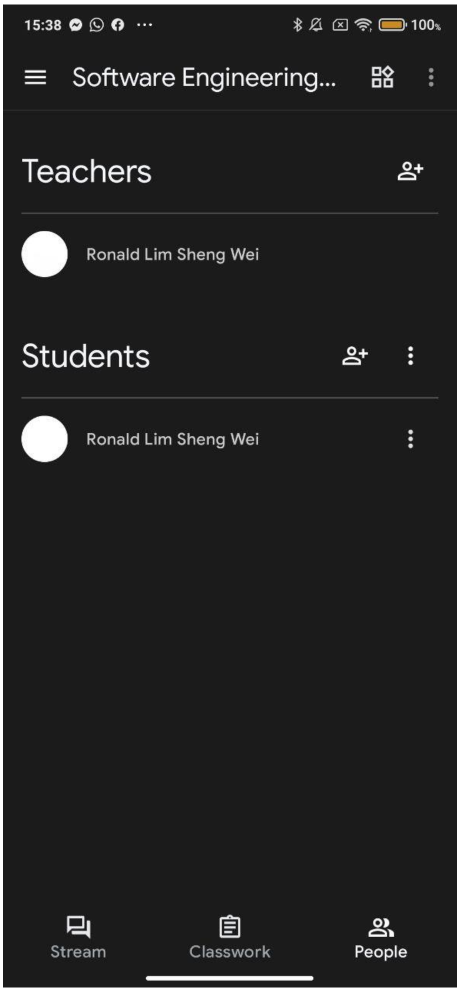


Figure 2.7 People Interface

Figure 2.7 shows the people tab of a class. The people tab contains information about participants in a class as well as educators involved in the class. Here, the creator of the class can manage teachers as well as students in the class.

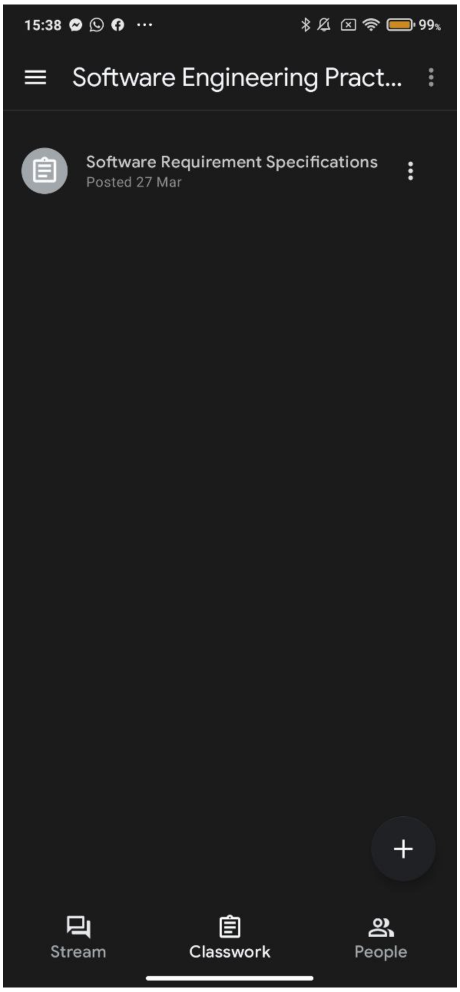

15:38
99%
Software Engineering Pract...
Software Requirement Specifications
Posted 27 Mar
+
Stream
Classwork
People

Figure 2.8 Classwork Interface

Figure 2.8 shows the classwork tab of a class. The classwork tab allows the user to manage assignments that are assigned to students from a teacher’s perspective. From a student’s perspective, they can manage their upcoming or ongoing assignments.

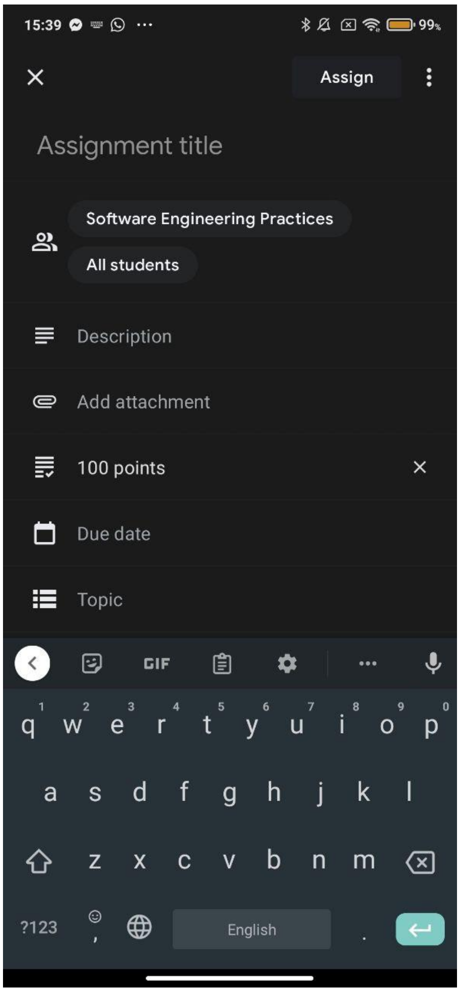
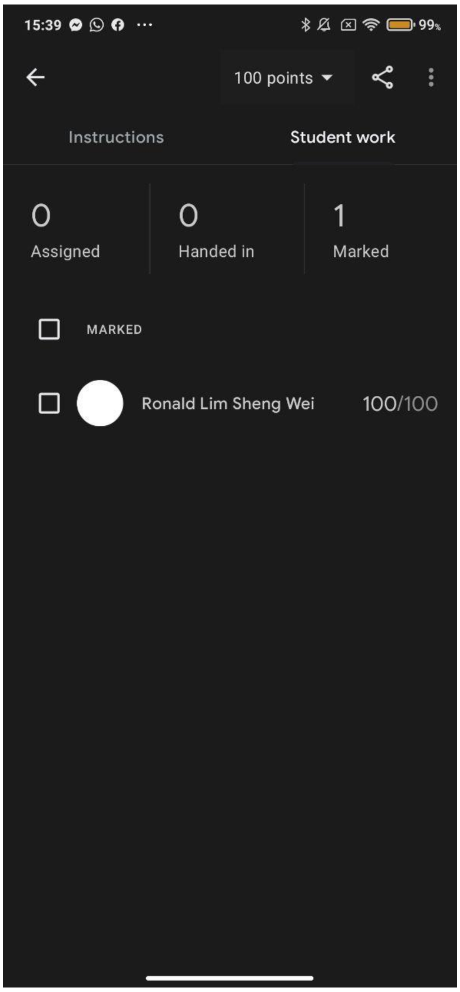


Figure 2.9 Create A New Assignment Interface

Figure 2.9 depicts the interface for the creation of a new assignment. There are a few input fields to fill before an assignment can be assigned. For instance, the targeted groups, points allocated, due date, the topic, the rubric, the title and instructions to follow.



0 Assigned

0 Handed in

1 Marked

□ MARKED

□ Ronald Lim Sheng Wei 100/100

Figure 2.10 Grade Interface

Figure 2.10 shows the grades tab of a class. Teachers can grade a student’s assignment and key in their marks. Then, the assignment can be returned to the student along with an optional feedback message. The class average is also calculated by the system.

2.4 Edmodo

2.4.1 Discussion

Edmodo was founded by Nic Borg and Jeff O'Hara in 2018 to improve classroom learning potential by utilising social media tools. In response to the restrictions set in place by schools, they created a platform where the class can connect and collaborate. (Wiebe, n.d.). Edmodo currently supports the latest version of Google Chrome, Firefox and Safari on computer browsers as well as mobile web browsers. Additionally, the Edmodo phone application is android devices with 5.0 Lollipop and iOS 11.0 and above.

First and foremost, Edmodo is a learning platform that is frequently compared to Facebook in the education space. (Gay & Sofyan, 2017). It has a business model of "Freemium" which means that basic functionalities are entirely free to use while other features require a subscription. The pro plan provides additional administrative tools and built-in Zoom meetings for subscribers. Next, Edmodo has cross-platform support which means that classes and the content can be accessed from either a mobile browser, a computer browser, or Edmodo's dedicated mobile application from an Android or iOS device. This means that Edmodo still retains the accessibility as found in the other two applications.

Overall, the system has all the basic features similar to KALAM and Google Classroom. It can create a classroom in which students can join via a specially generated code that is distributed by the class creator. This makes it easy and seamless for students to join created classes. An announcement page is the main medium for communication between the student and the educator. Furthermore, dedicated groups can be created from within the classroom. This provides the flexibility to give specific instructions. Next, it also boasts a folder management system. The creator can create folders that can store educational resources, quizzes, and links. A robust file management is essential for classroom management.

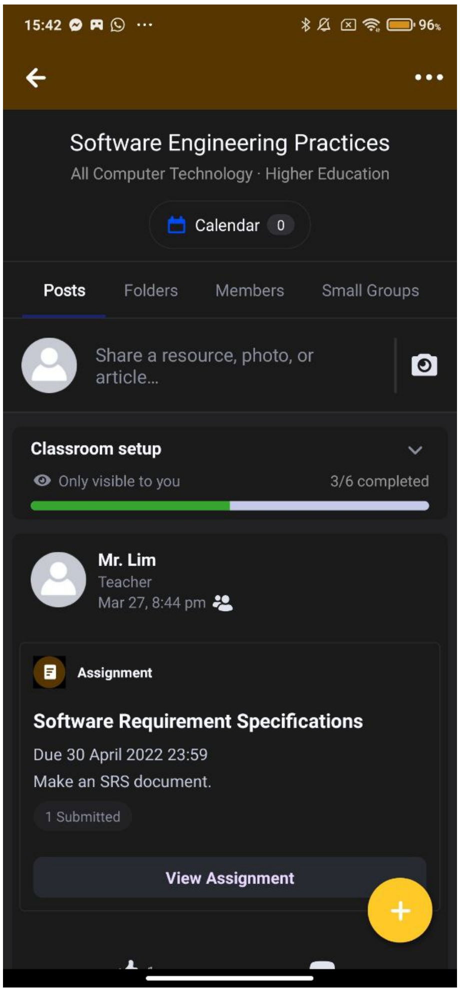
Edmodo allows students and educators to communicate through a stream. Similar to Google Classroom, Edmodo's very own stream allows educators to post announcements, assignments, quizzes and graphical content. Students with enquiries can choose to

One of the advantages of Edmodo is that it has OneDrive and Google Drive integration. Users can choose to share files directly from their cloud-based storage. It also integrates Microsoft’s online productivity suite such as the famous Microsoft Word, Microsoft PowerPoint, and Microsoft Excel. Next, Edmodo is more inclusive as it comes with a set of parental features. It allows the student to be connected with their parents for progress tracking. On top of that, the creator of the class can change the student’s password and remove their profile pictures. Hence, Edmodo has good integrations with famous productivity application providers as well as inclusivity of students who require parental guidance.

However, Edmodo lacks an instant messaging function. The messaging function is not instantaneous as users will have to reload the page to receive the latest messages. Furthermore, it does not currently support video conferencing tools such as Moodle, Zoom, Skype, or Microsoft Team. Edmodo's features include the ability to submit content, share videos (but not conferencing), links, grades, alerts, and assignments (Etfita, 2019). One of Edmodo's shortcomings is the absence of video conferencing for direct interaction in online learning, but the benefits exceed the drawbacks (Ekayati, 2018).

In conclusion, Edmodo, despite its shortcomings, can be concluded as a good learning medium that could potentially increase student learning outcomes based on the comprehensive research done by Nurhayati (2019).

2.4.2 Graphical User Interface and Functionality



Mr. Lim
Teacher
Mar 27, 8:44 pm

Assignment

Software Requirement Specifications

Due 30 April 2022 23:59

Make an SRS document.

1 Submitted

Figure 2.11 Class Interface

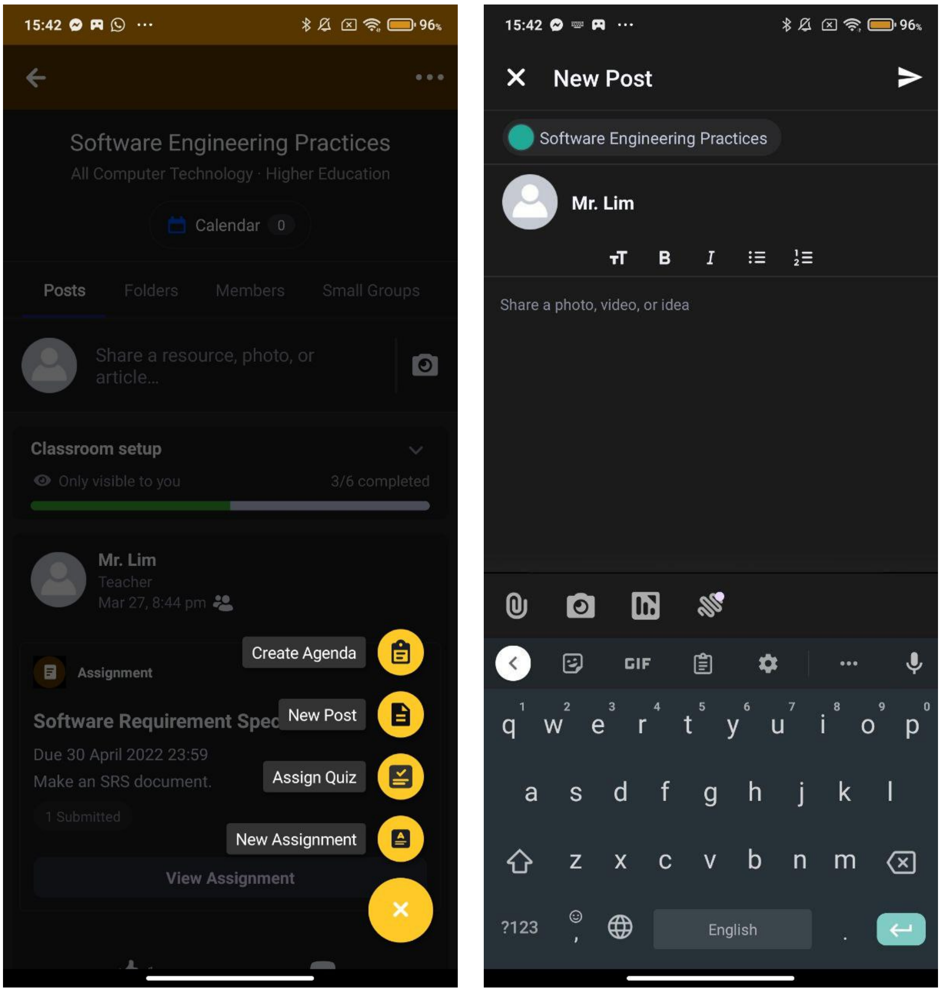
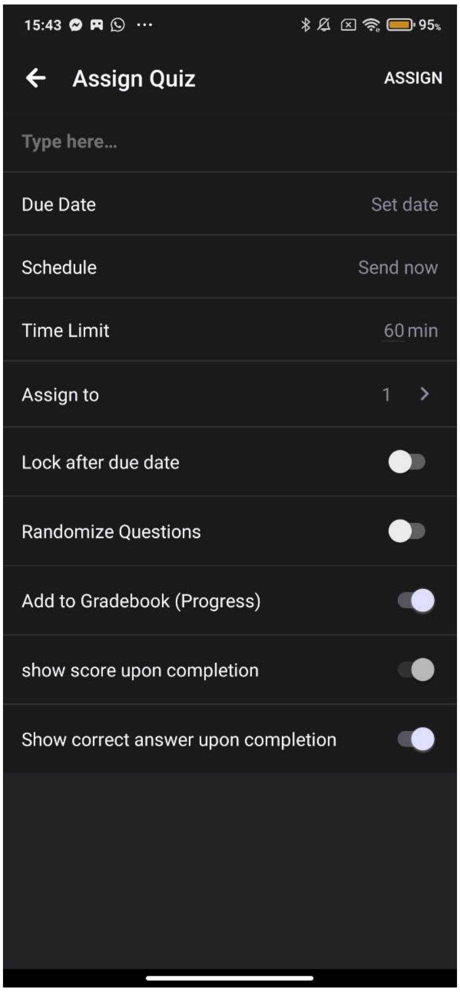


Figure 2.12 Create New Post

Figure 2.11 and Figure 2.12 shows the interface of a class. In the interface, the user can post new announcements, create new polls, or share educational resources in the form of images or text. Other functionalities such as folders, classes, members can be accessed from this interface as well.



15:43 ··· 95%

Assign Quiz ASSIGN

Type here...

Due Date Set date

Schedule Send now

Time Limit 60 min

Assign to 1 >

Lock after due date ○

Randomize Questions ○

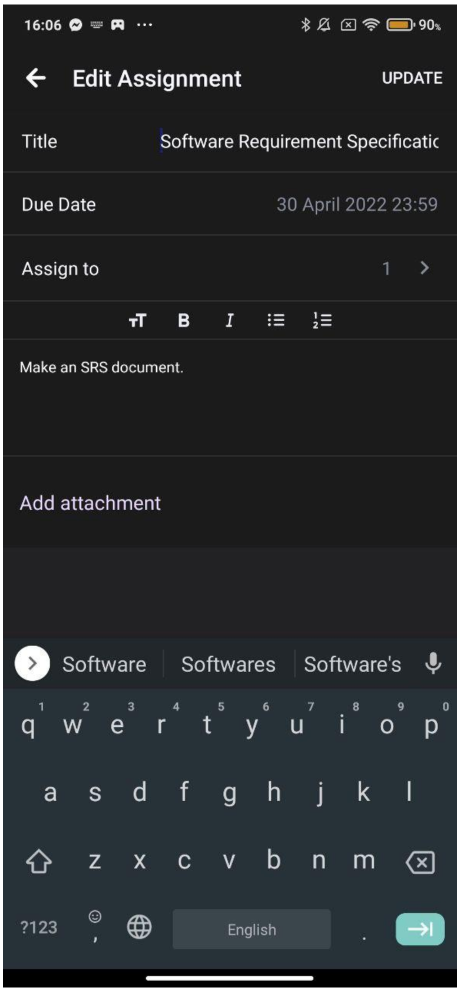
Add to Gradebook (Progress) ●

show score upon completion ●

Show correct answer upon completion ●

Figure 2.13 Create New Assignment Interface

Figure 2.13 depicts the interface that is used to create new assignments. The interface is simple as it provides input fields such as the title and instructions. There are many options to format the text such as the option to bold text, and toggle bullet points.



Edit Assignment

UPDATE

Title: Software Requirement Specificatic

Due Date: 30 April 2022 23:59

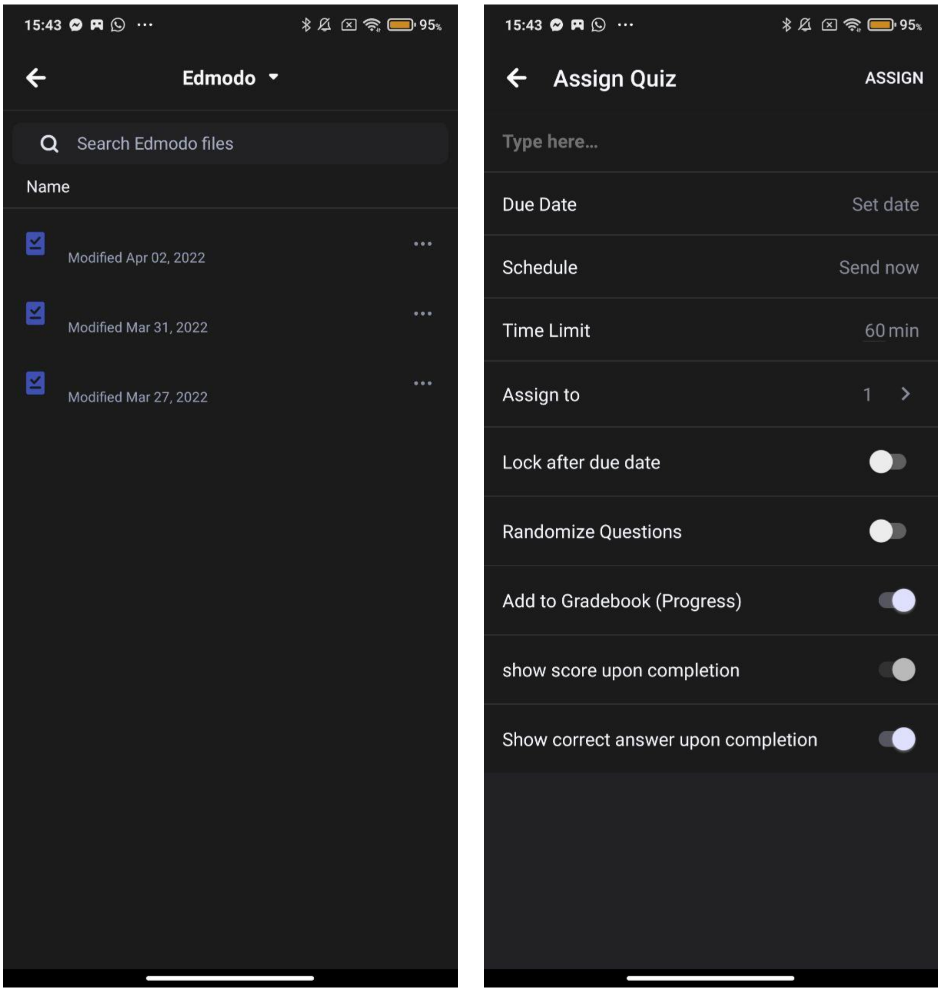
Assign to: 1

Make an SRS document.

Add attachment

Figure 2.14 Assignment Settings Interface

Furthermore, additional settings for creating new assignments can be found in Figure 2.14. This interface requires the user to set a due date. Users can also choose to lock submissions after the designated date.



Edmodo

Search Edmodo files

Name

Modified Apr 02, 2022

Modified Mar 31, 2022

Modified Mar 27, 2022

Assign Quiz

ASSIGN

Type here...

Due Date Set date

Schedule Send now

Time Limit 60 min

Assign to 1

Lock after due date

Randomize Questions

Add to Gradebook (Progress)

show score upon completion

Show correct answer upon completion

Figure 2.15 Create New Quiz Interface

Figure 2.15 shows the quiz function of Edmodo. Users can create quizzes with different types of questions such as matching, multiple-choice questions, as well as short answers. Images or links can also be attached to the questions.

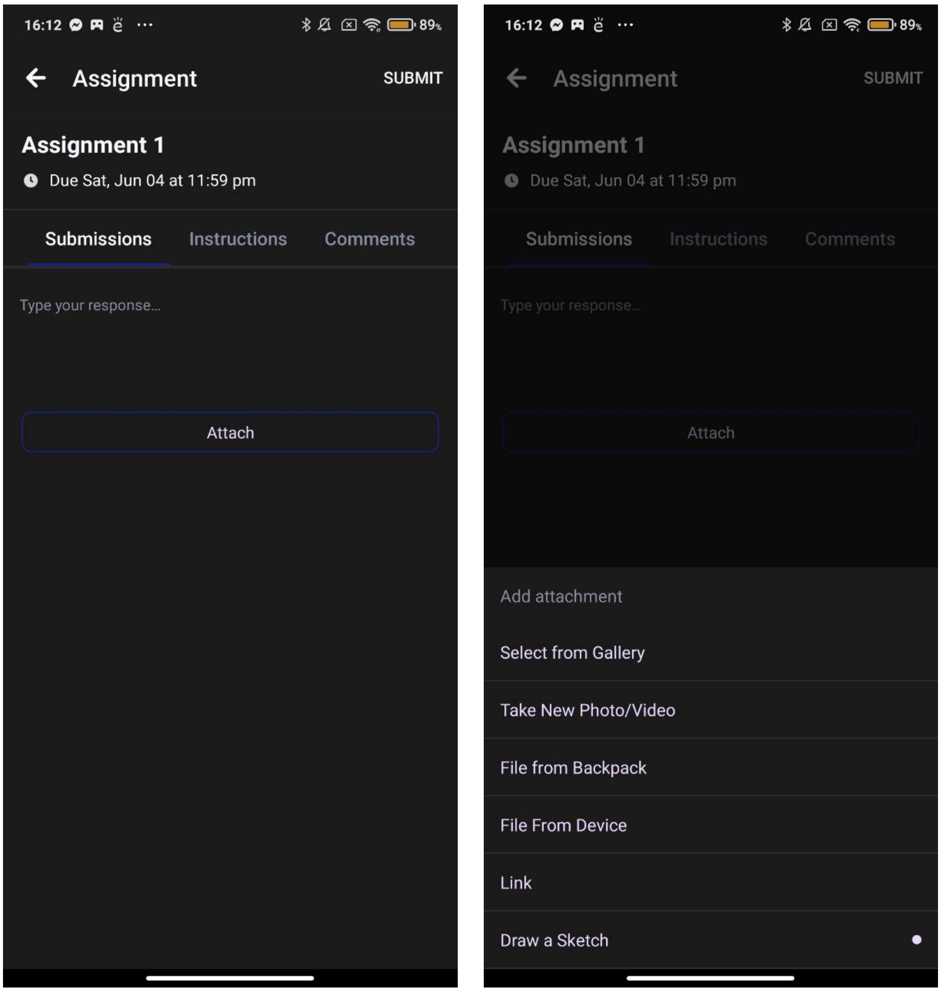
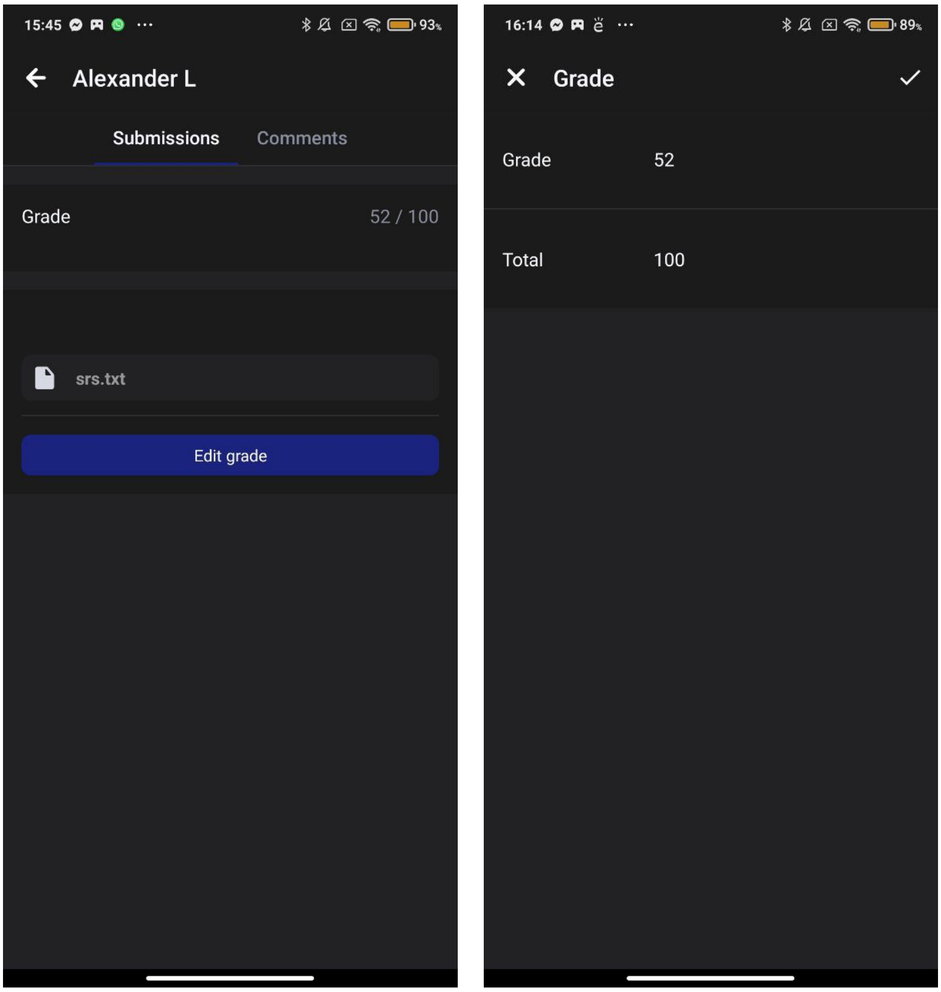


Figure 2.16 Assigment Submission Interface

Figure 2.16 shows the interface that will be seen by students in a class. Assignments can be uploaded and submitted here.



Grade 52 / 100

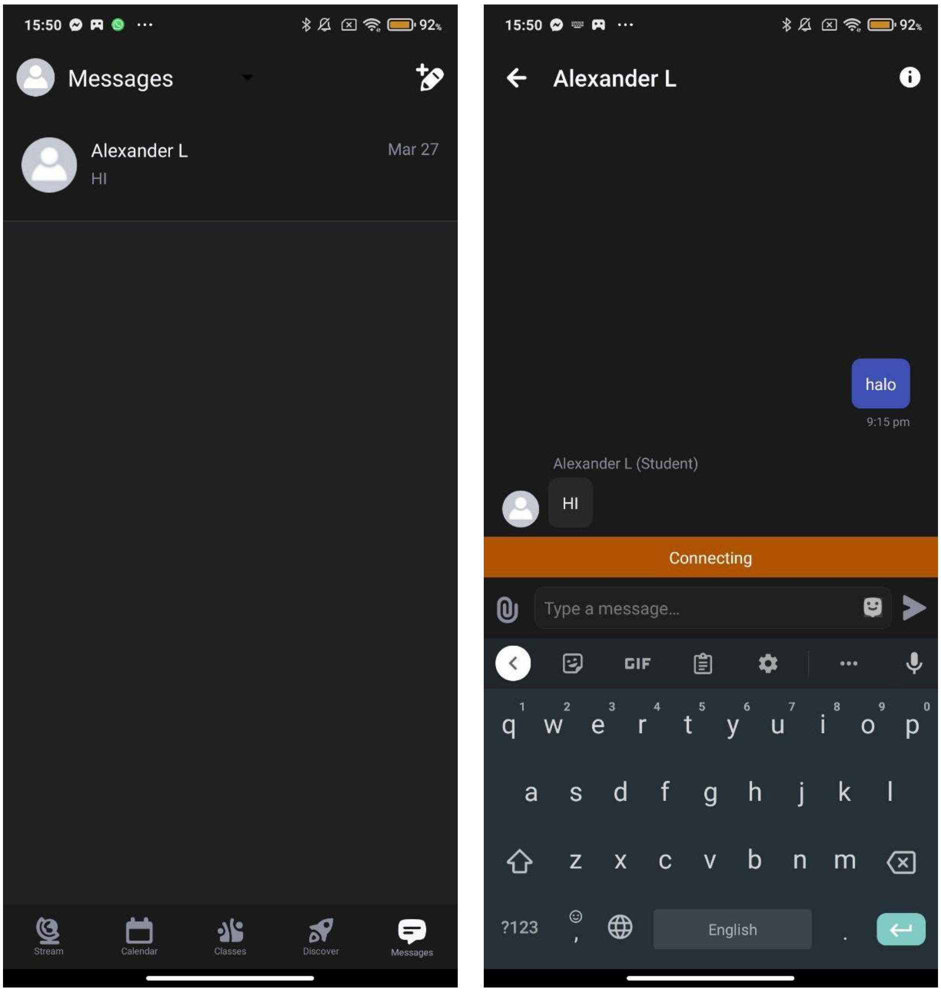
srs.txt

Grade 52

Total 100

Figure 2.17 Assignment Grading Interface

In figure 2.17, teachers-in-charge of the class can grade assignments in this interface. They can see who has not turned in their assignments and the average grade score as well.



15:50

Messages

Alexander L Mar 27
HI

Alexander L

halo
9:15 pm

Alexander L (Student)
HI

Connecting

Type a message...

Figure 2.18 Messages Interface

In figure 2.1.8, Edmodo users can send messages to each other through the messages tab. However, the messages sent here are not instantaneous as there is a delay when receiving new messages which requires a reload.

2.5 Comparison between 3 Existing Systems

Table 2.1 Overall System Comparison and Proposed App

Criteria	**KALAM**	**Google Classroom**	**Edmodo**	**Proposed App**
Active Users	311m	150m	100m	1000

Type of Software	Open-Source	Proprietary	Proprietary	Proprietary
Pricing	Free	Freemium, requires payment for more administrative controls.	Free, requires payment for school and district use.	Free
Usability	Small learning curve.	Higher learning curve	Small learning curve.	Small learning Curve
Customisability	High	Low	Low	Low
Third-Party Plugin Support	Supported	Unsupported	Unsupported	Unsupported
Desktop Browsers	Supported	Supported	Supported	Unsupported
Mobile Browsers	Supported	Supported	Supported	Unsupported
Set-up	Requires IT professionals and web hosting.	Cloud-based, can be used instantly.	Cloud-based, can be used instantly.	Cloud-based, can be used instantly.
Cloud Storage	Unimplemented	Google Drive	OneDrive, Google Drive	Firebase
Mobile Application	Available on play store and Apple App Store	Available on play store and Apple App Store	Available on play store and Apple App Store	Available on play store

The open-source platform used by KALAM, Moodle, has over 311 million active users as compared to Google Classroom (150 million) and Edmodo (100 million). This speaks volume about the appeal of the application to modern users. Based on statistics alone, it is safe to say that Moodle is the preferred system by the education field.

The appeal of KALAM (Moodle) is not only due to the fact that it is an entirely free system, but the fact that it is designed to accommodate various in-house plugins or third-party plugins that the other two systems fail to provide. It's customisability and open-source concept is what constitutes the high adoption of the system across universities.

On the other hand, all three systems support modern mobile and desktop browsers. Edmodo and Google Classroom can be used instantly while Edmodo requires web hosting and professional configuration to set up. Cloud integrations is also a modern feature that is adopted by Google Classroom and Edmodo while KALAM has not implemented the feature even though Moodle supports it.

Table 2.2 Function Comparison Including The Proposed App

Functions	**KALAM**	**Google Classroom**	**Edmodo**	**Proposed App**
Dashboard	Blocks can be rearranged, hidden, and deleted. It contains recently accessed courses, course overview and timeline.	Classes can be moved, copied, edited, or archived.	Classes can be accessed from the dashboard. It contains a dedicated news feed as well.	Contains recently accessed classes and managing classrooms.
Calendar	Able to add new events to calendar and	Able to view upcoming tasks and	Able to add new events to calendar and	Able to view calendar and weather.

	view upcoming tasks and assignments. There is an option to export the calendar as well.	assignments only.	view upcoming tasks and assignments.	
Content	Content is organized based on a weekly basis.	Content is organized using a stream.	Content is organized using a stream. Resources can be stored in folders as well.	Content is organized based on a weekly basis.
Messaging	Non real time messaging.	Email or through assignment feedback.	Non real time messaging. Supports file transfer.	Real time messaging within classes.
Thread	Only the educator can start a thread for discussions.	Unavailable	Unavailable	Student or educators can contribute or create threads.
Grading	Educator can grade assignments.	Educator can grade assignments.	Educator can grade assignments.	Educator can grade assignments.
Assignment	Educator can submit or create assignments.	Educator can submit or create assignments.	Educator can submit or create assignments.	Educator can submit or create assignments.

Quizzes	Users can take quizzes or create quizzes.	Users can take quizzes or create quizzes.	Users can take quizzes or create quizzes.	N/A

With reference to Table 2.3, for the dashboard function, Google Classroom and KALAM boasts a more straightforward access to classes while Edmodo has the addition of a general stream on its dashboard which not only include content from all classes, but advertisements from Edmodo as well. Hence, KALAM has the better implementation of dashboard as it is a balance between simplicity and functionality.

Next, KALAM's calendar is more feature packed as compared to the other two applications. It provides the extra functionality to export the calendar which is absent on the other two applications. KALAM and Edmodo are able to create new events on the calendar as well.

The organization is better on KALAM as well. By implementing a system in which content is organized according to the week of the semester, it is convenient for students as well as educators to navigate around the system to obtain educational resources. In comparison to the stream system, which is adopted by Google Classroom and Edmodo, it is more disorganized and difficult to find uploaded resources. This issue can be rectified by implementing a search function.

Messaging is an essential component for most modern applications now. KALAM does not have the chat function; however, Moodle supports it. Google Classroom facilitates communication via e-mail while Edmodo has a dedicated messaging page. In terms of messaging, Edmodo has the best implementation among the three applications. The short delays between communications are negligible as it can function as intended.

Threads are where important discussions between students and educators take place. Currently, only KALAM can open new threads while the functionality is non-existent in the other two applications. However, the thread opening functionality is only available to accounts with educator access levels. In addition to that, the interface looks unpolished which might discourage users from using it. Hence, it is important that the forum functionality is accessible to all users to facilitate academic discussions.

The grading, assignment and quizzes are available on all three platforms. While written assignments must be graded manually by the educator, the marking of quizzes is automated for multiple choice questions.

2.6 Conclusion

As a result of the comparisons in Table 2.2 and Table 2.3, these three systems mostly possess the same functionalities ranging from the dashboard, classes, calendar, and assignment grading. All three systems provide basic functionalities that can facilitate online learning.

However, most of these systems often neglect student-to-student interactions and focuses on the delivery of instructions and the grading of assignments only. There is no dedicated function such as an implementation of a forum for learners to gather and discuss theoretical questions and solutions to various problems. For instance, only KALAM has the functionality to start forum threads for questions. However, the thread can only be initiated by the lecturer or person-in-charge of the class. Hence, it is very clear that there needs to innovation in the field to introduce elements of student-to-student interactions to e-learning systems.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter describes the methodology used to develop the SAGE system. Every successful system requires a good and comprehensive plan in order to maintain the quality of the developed system, the requirements of the system are met, and high user satisfaction is achieved.

Software Development Life Cycles (SDLC) have advanced over the decades and new methodologies have been introduced to cater to rising demands of proprietary software. In this project, the Rapid Application Development (RAD) with respect to the given time to complete the project. As substantiated by Beynon-Davies et. al. (1999), most RAD projects appear to be focused on highly interactive apps with a well-defined user group and little computational complexity.

3.2 Methodology

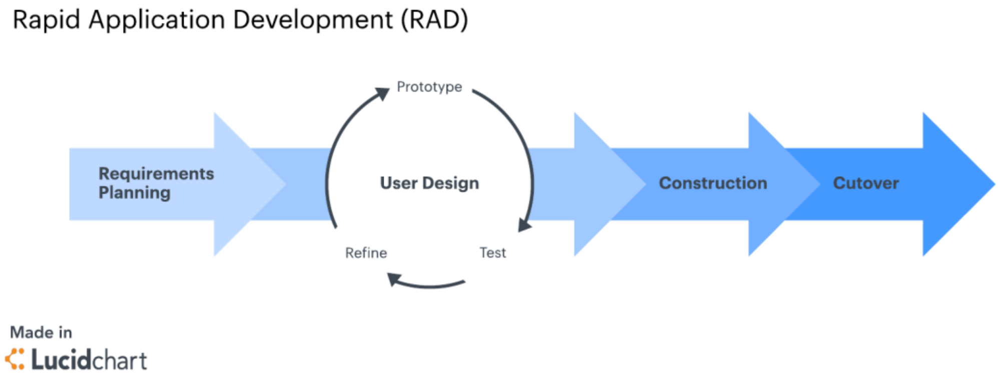


Figure 3.1 Rapid Application Development Phases
Source: LucidChart (2018)

Rapid application development (RAD) appears to have gained popularity following the publication of a book by James Martin of the same title. Martin characterises the primary goals of rapid application development as high-quality systems, rapid development and delivery, and cheap costs. These goals may be summarised in a single sentence: the commercial compulsion to produce functional business applications in shorter timeframes and with lower expenditure. (Beynon-Davies et al., 1999). Thus, RAD is chosen as the main SDLC to be implemented in this project.

In the RAD SDLC, there are four major phases that constitutes the RAD process, namely, requirements planning, user design, construction and implementation. Furthermore, RAD applies two types of methodologies, which are phased development and prototyping. (Fatima et al., 2014)

Firstly, the requirement planning phase involves procuring a broad range of requirements from stakeholders. For requirements elicitation, the technique of interface analysis. As mentioned in Chapter 2, three systems are chosen and compared to retrieve the best implementation of features among the three and propose improvements.

Next, the second step of the Rapid Application Development methodology is the user design. It entails obtaining user feedback and then developing many prototypes of the project under development utilising developer tools. Instead of working with a fixed set of criteria, RAD developers generate a variety of prototypes with diverse features and functionality. All of these prototypes are then assessed by the client to select what to keep and what to reject. The user description step comprises the re-examination and validation of the data acquired during the first phase. This step also covers the identification and clarification of the dataset characteristics.

The construction phase is where the prototypes generated in the preceding phase are refined. During this third step of the RAD Model, all gathered additions and alterations are implemented. This phase provides feedback on what is good, what is poor, what to maintain and what to eliminate. During the building process, input is not limited to functionality, but also to aesthetics, interface, and so on. The prototype process is then resumed, with all obtained comments taken into account. Both prototype and feedback are carried out until a final product that is most closely aligned with the client's needs is established.

The final phase involves finalising the aesthetics, features, functionalities, and interface of the software project, as well as everything else associated with it. Interfaces between distinct modules must be well tested. This is accomplished during the cutover phase. It is followed by client acceptability testing. Prior to providing the final product to the client, it is critical to ensure that the generated software is maintainable, stable, and usable.

3.3 Project Requirements

3.3.1 Functional Requirements and Non-Functional Requirements

Table 3.1 Functional Requirements of The Proposed Application

No.	Functional Requirements
1.	The system shall be able to add, edit or delete classes.
2.	The system shall display weekly class content.
3.	The system shall support real-time messaging within classes.
4.	The system shall be able to add, edit or delete threads.
5.	The system shall be able to grade and return assignments.
6.	The system shall be able to add, edit or delete assignments.
7.	The system shall be able to add, edit, delete or join events.
8	The system shall be able to add, edit or delete communities.

Table 3.1 shows the functional requirements of the proposed application. Within the application contains requirements that are elicitated from the comparison of the three existing systems from Chapter 2, Table 2.2. The system mainly contains the features that most e-learning system possess such as management of classroom, educational content management, real-time messaging, community and thread management, assignment management, and event management.

Table 3.2 Non-Functional Requirements of The Proposed Application

Quality Attribute	**Non-Functional Requirements**
Usability	The time taken to get familiar with the system should not be more than 30 minutes. The registration shall not take more than 5 minutes.
Scalability	The system shall use firebase as its cloud database
Reliability	The system shall not have downtime for more than 2 hours.

Table 3.2 depicts the non-functional requirements of the proposed application. Non-functional requirements are indirectly related to the services provided by the system to its users. Since it is more important than functional requirements, failing to meet either one of the non-functional requirements may lead to an unusable system.

3.3.2 Constraints and Limitations

Table 3.3 Constraints of the Proposed Application

No.	Constraints
1	The system must be connected to the internet as long as the user is using the system as the system's database uses a cloud database for sending, receiving and storing data across all functionalities.
2	The system is developed based on author's view on Universiti Malaysia Pahang's current education structure, which is presumed to be the same across all local government universities in Malaysia. The system structure may not be practical to other universities domestically or internationally.
3	The system must be completed before the author has completed the course "BCC3024 Undergraduate Project II".
4	There are no financial resources allocated to the project by the educational institution. Thus, the system relies on the author's financial ability to support the used services. Hence, the author will be using free tier subscriptions whenever possible to reduce cost.
5	The system will only be developed by one person. Hence, the time constraints to develop the system is short, resulting in limited functionality of the system.

Table 3.3 depicts the constraints of the system as a result of external factors. The constraint of the system comprises of internet connectivity, limited exposure, time constraints as well as financial constraints.

Table 3.4 Limitations of the Proposed Application

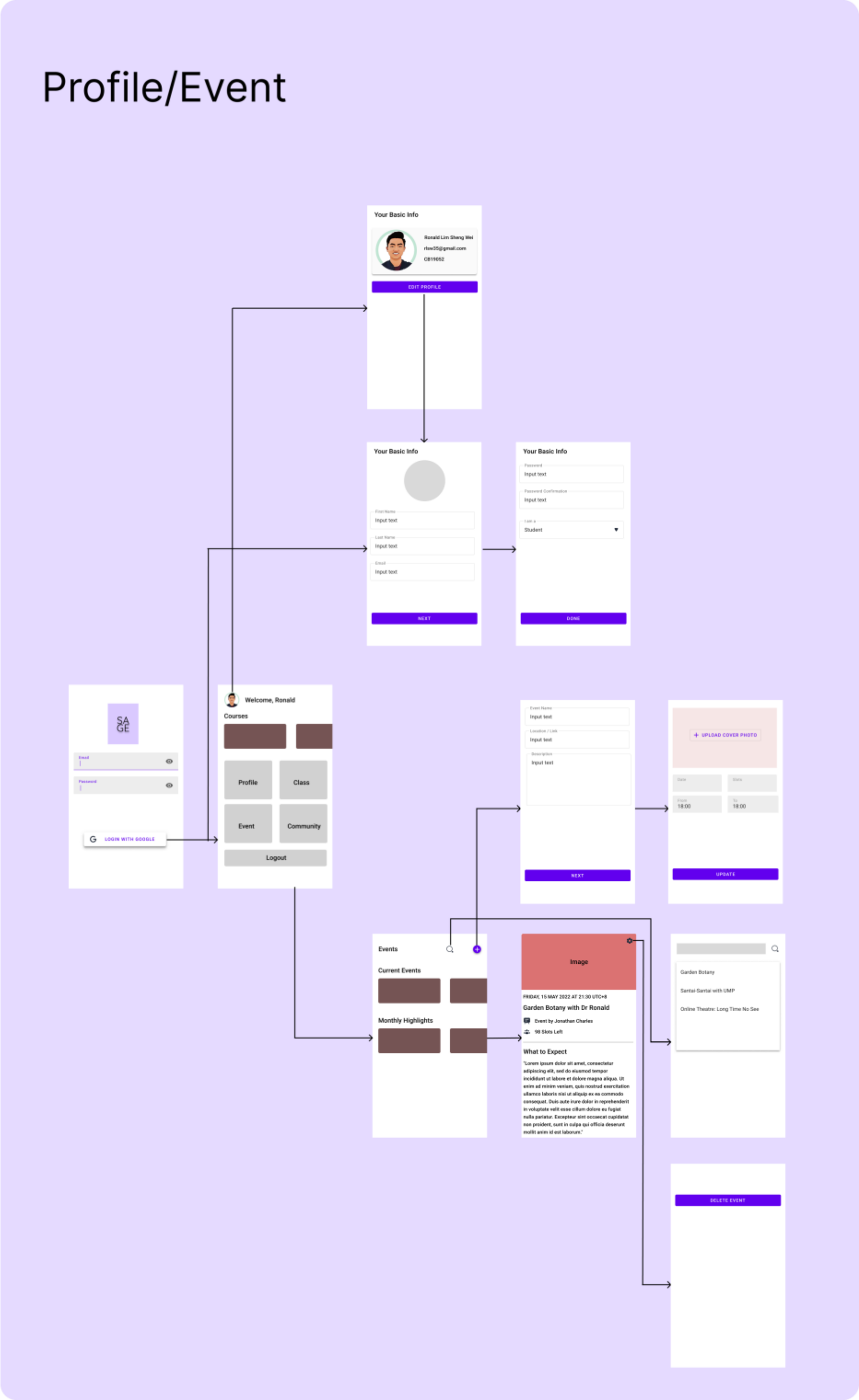
No.	Limitations
1	The system will not support IOS platform, web platform, and windows platform.
2	The system will not have a module for system admin to manage users.
3	The system will only have one tester for each user type to gain feedback.

Table 3.4 depicts the technical limitations of the system. The system is limited to the android platform and also pc web browsers. In addition to that, there is no module for

system admins to manage users. Finally, due to time constraints, the system will have limited testers, mainly only one tester for each user type.

3.3.3 Proposed Design

3.3.3.1 Prototype



"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Figure 3.2 Profile and Events Module Interfaces

Figure 3.2 depicts the flow of the Manage Profile and Manage Event modules. Starting with the Manage Profile, users can access this module from the home page of the system. From there, users can choose to edit their details at the edit interface.

For the Manage Event module, users can explore the latest events on the index page. Next, users can also choose to host their own events. By filling the event creation form, a new event will be created. Once created, the user hosted event can be managed as well. Users can view information regarding the participants, edit the details of the event as well as delete it.

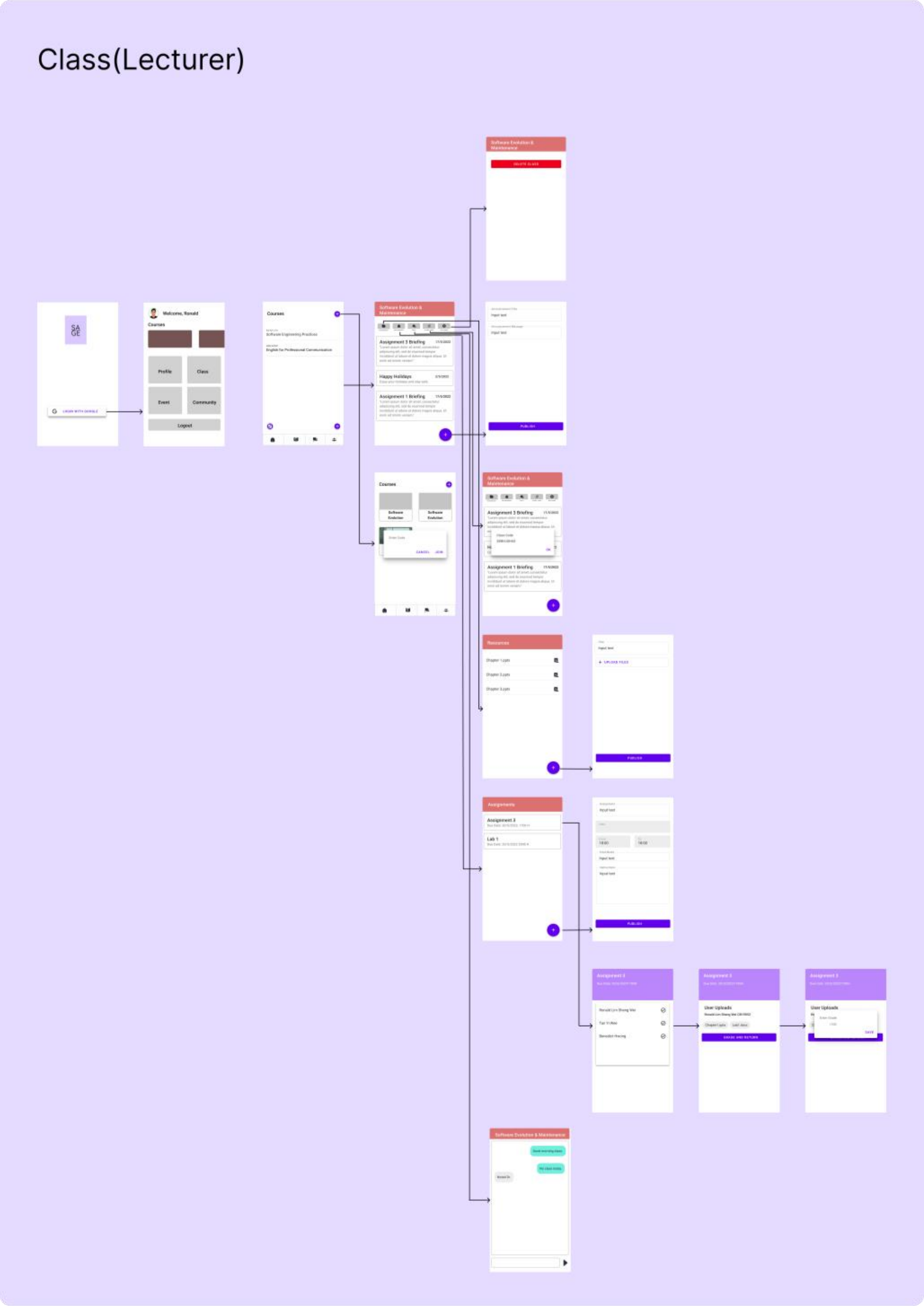


Figure 3.3 Class Module Interfaces – Lecturer View

Figure 3.3 depicts a lecturer's view of the Manage Class. Lecturer will have full access to the administrative functions across the Manage Class module. Lecturers are able to create, edit, delete announcement, assignments, resources, participate in the chatroom, as well as manage the classroom. In addition, lecturers can also view, and grade submitted assignments by students.

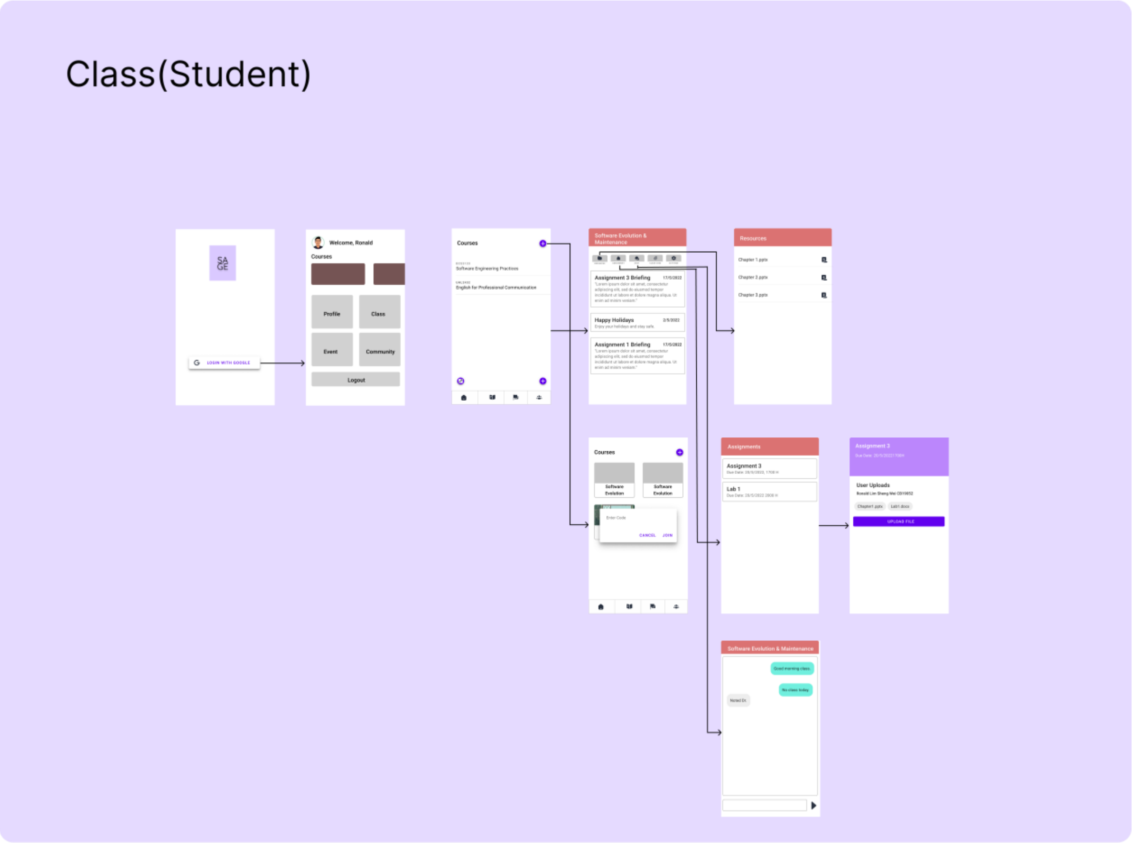
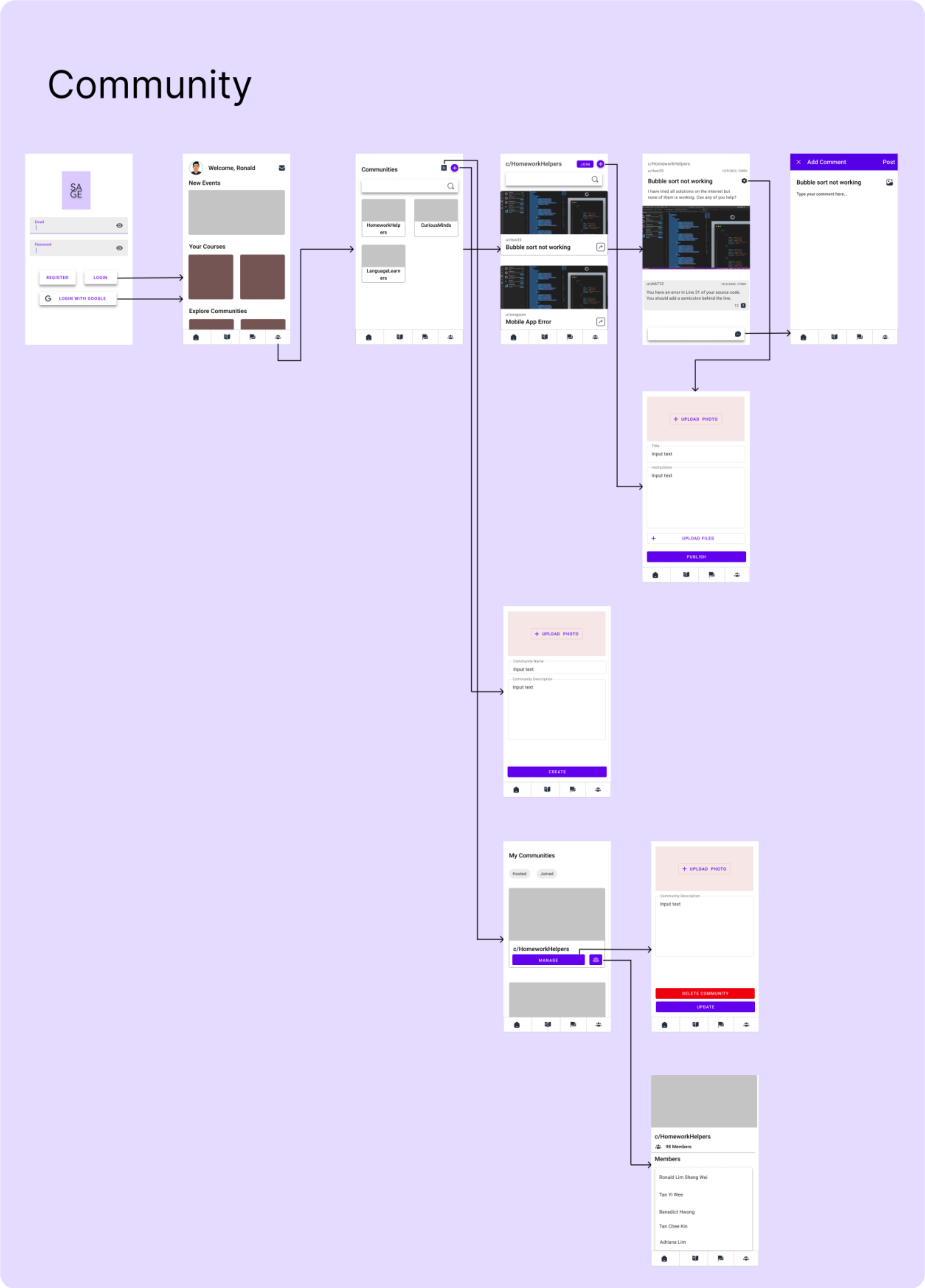


Figure 3.4 Class Module Interfaces – Student View

Figure 3.4 depicts a student’s view of the Manage Class module. In this view, many of the administrative functions are stripped off as they are only limited to the lecturer or creator of the class. Meanwhile, users can still view announcements, chat with their classmates and educator in the group chat. There is also an interface to input the class code to join a class. The resources tab can be accessed to read the instructions and download the required files. Finally, students can submit their assignments in the assignments tab. Once the files are uploaded, they can press submit for the lecturer’s evaluation.



I have tried all solutions on the internet but none of them is working. Can any of you help?

You have an error in Line 31 of your source code. You should add a semicolon behind the line.

Figure 3.5 Community Module Interfaces

Figure 3.5 depicts the interface of Manage Community module. Starting with the login screen, the flow of the interfaces starts when the user presses on the community icon to access the module. Users can create their own communities by using the provided community creation form. The community can be edited or deleted.

Upon accessing an existing community, the user can view many threads that have been submitted by other users. The thread list contains a picture of the problem and the preview of the title and description of the problem. After accessing a thread, users can view the full content of the thread which includes the replies. Users can then submit a reply to an existing thread by using the reply bar on the bottom of the interface. Next, users can also create a new thread within a community by using the provided thread form. Users will need to include the title, description and image of the problem.

3.3.3.2 Use Case Diagram

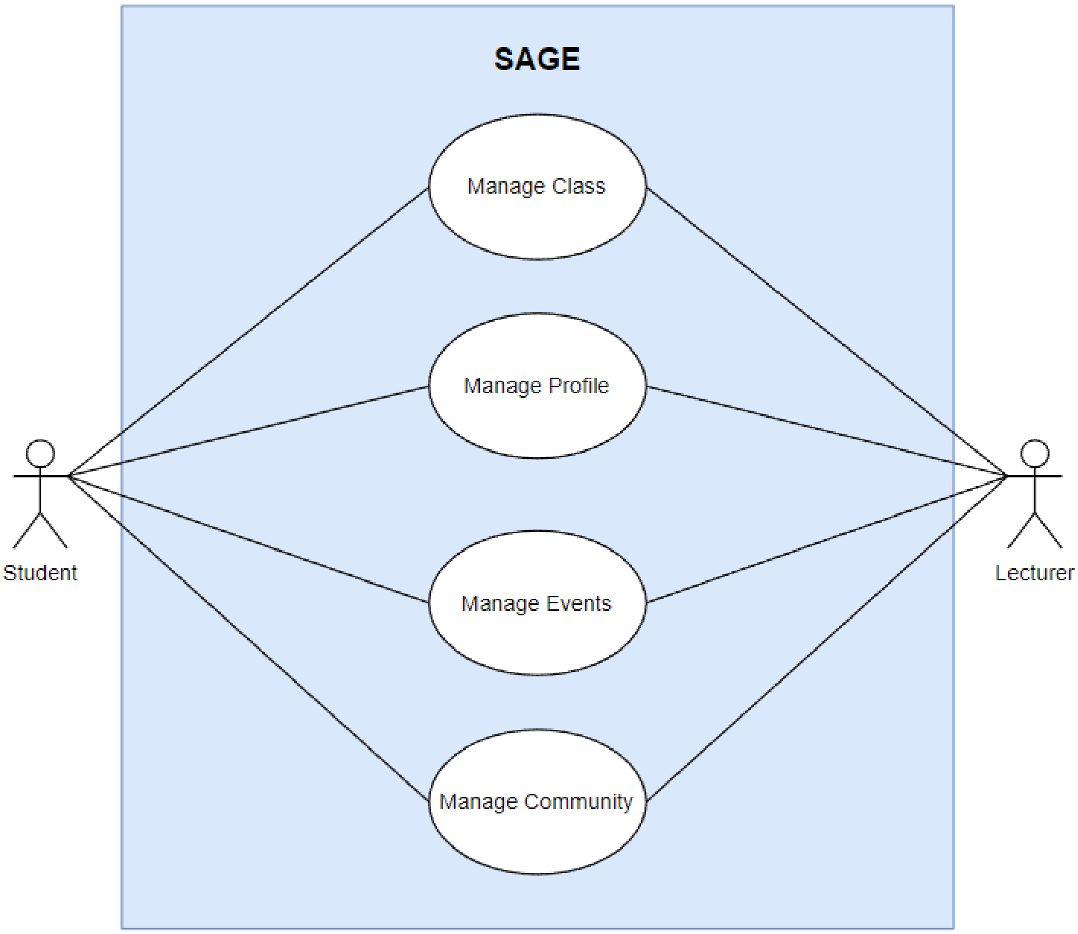


Figure 3.6 Use Case Diagram

Figure 3.5 depicts the use case diagram of the system. There are two stakeholders in the system, namely, the student and the lecturer. Both students have access to the four functions or modules which are Manage Class, Manage Profile, Manage Events, Manage Community. However, functions within the modules will be limited based on the role of the user. Students will only be able to join classes, participate in class chats, and also retrieve educational resources while the lecturers will have the full access to the functions. Lecturers can manage classes, assignments, and even announcements. Secondly, both users will be able to change their profile information. Managing events are fully accessible to both users. In the Manage Events module, users can create, delete, join and discover new events around the campus. Last but not least, Manage Community connects students across the campus to discuss academic hurdles and achievements by creating personal communities and threads and by replying to threads.

3.3.3.3 Context Diagram

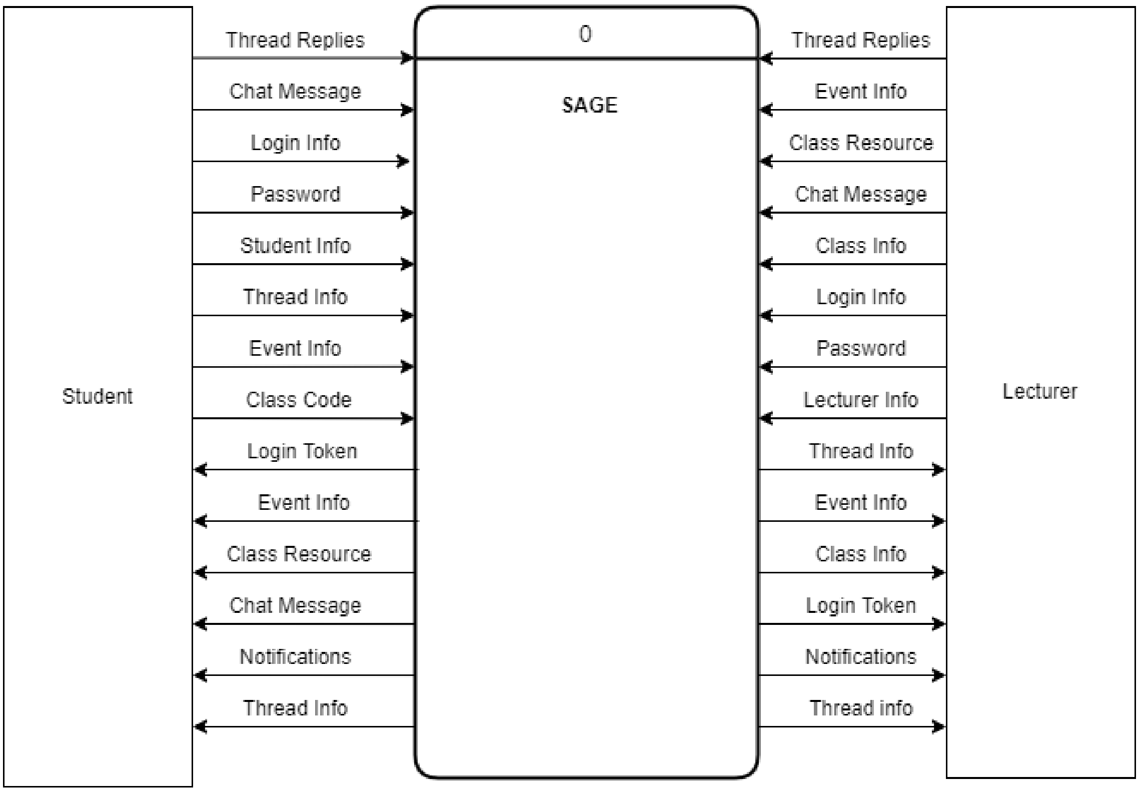


Figure 3.7 Context Diagram

The context diagram contains two entities which are the student and the lecturer who will interact with the system. From the two entities to the system, there will be data flowing from multiple modules from within the system. For the login function of the system, user credentials or login information will be sent to the system while a login token will be returned to the user. Next, for the Manage Class Module, there will be class information and class code and also class messages going into the system and out from the system to the students. Class code will be used to join new classes while class information for students include the submission of class assignments. For the lecturers, class information indicates the upload of educational resources, creating announcements, as well as creating and marking of assignmnets. Besides that, for the Manage Event module, both entities will be required to provide event information to the system for the creation, editing, and deleting of events. The system will also provide users with the ability to browse events as seen as the event information flowing from the system to the entities. After that, for the Manage Community module, both entities will be expected to send and receive community information, thread info and also thread replies from the system. This allows the managing of communities, threads and also thread replies.

3.3.3.4 Data Flow Diagram

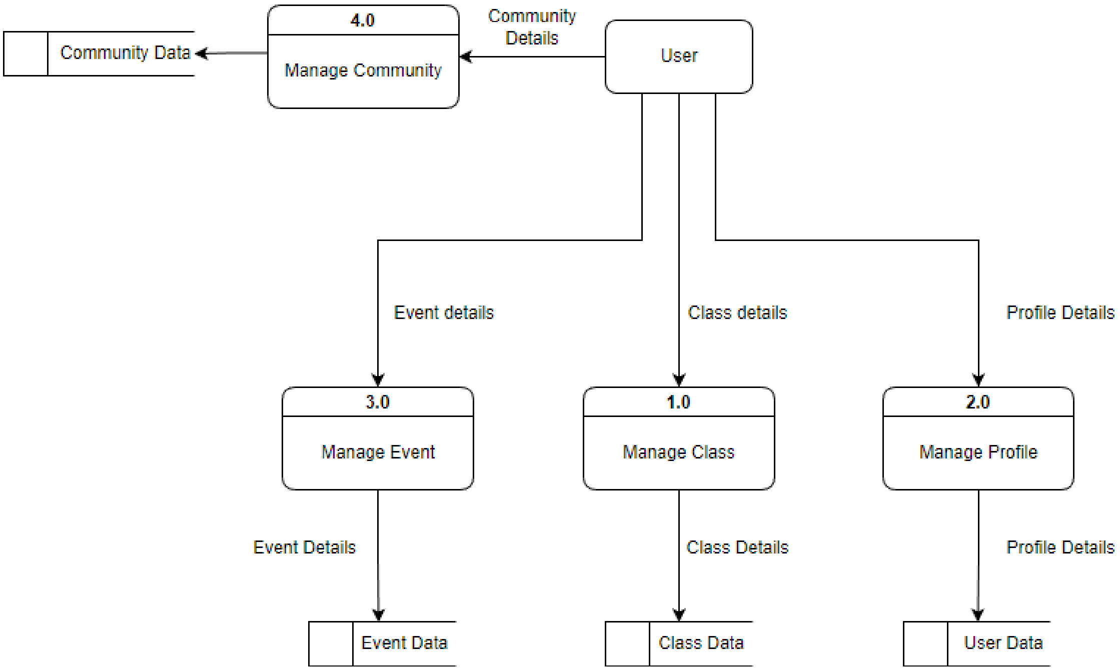


Figure 3.8 Data Flow Diagram

Figure 3.8 depicts the data flow diagram of the system. There are a total of 4 data stores, 4 processes, and one external entity. For the Manage Class process, class details data will flow into the process. Next, for the Manage Event process, it will receive event details from the user. Eventually, the output from the process will have data flowing into the Event datastore. For the Manage Profile, it will receive profile details from the external entity, user and subsequently output profile details to be stored in the user data store. Finally, the Manage Community process receives community details and subsequently outputs data to be stored in the Community data store.

3.3.3.5 Activity Diagram

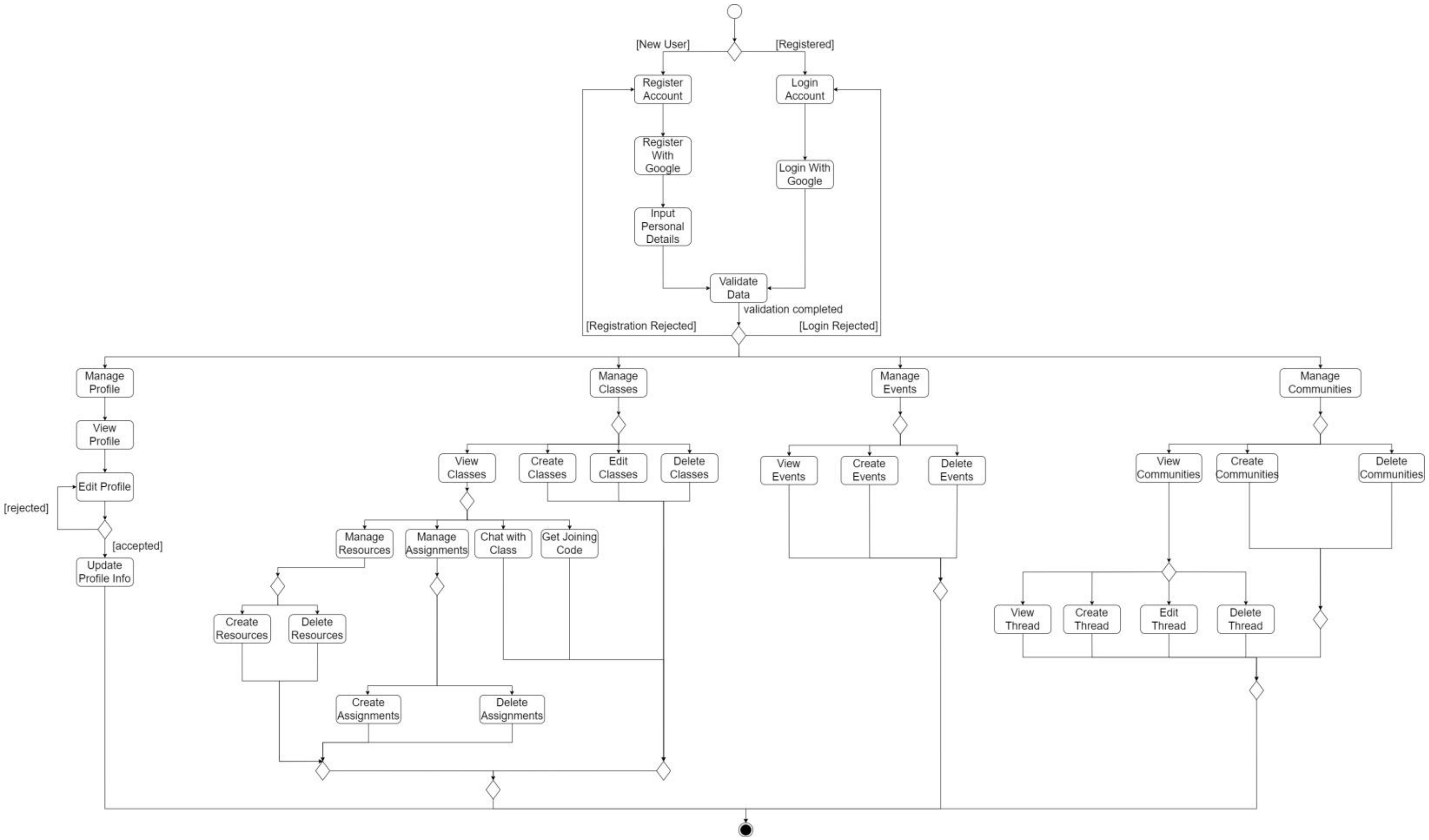


Figure 3.9 Activity Diagram

Figure 3.9 depicts an activity diagram of the system. From the diagram, the flow of the four modules, namely, Manage Class, Manage Profile, Manage Event and Manage Community can be easily understood and visualized.

3.4 Data Design

3.4.1 Firebase Authentication

Firebase Authentication is a modern way of authenticating users through the Google Ecosystem. By using OAuth2 technology, users with google accounts can easily login to the platform effortlessly. Credentials such as email and password or OAuth tokens are easily verified by backend services provided Firebase through its Firebase Authentication SDK.

3.4.2 Firebase Realtime Database

The Firebase Realtime Database is a database stored in the cloud. The data is saved as JSON and is synced in real-time with all connected clients. When creating cross-platform applications using our Apple iOS, Android, and JavaScript SDKs, all of the clients share a single instance of Realtime Database and immediately receive the most recent data changes. Firebase Realtime Database uses Not Only SQL (NoSQL) database instead of the traditional SQL database.

NoSQL uses a non-relational database. NoSQL databases enable developers to store vast quantities of unstructured data, providing them with a great deal of freedom. In addition, the Agile Manifesto was gaining momentum, and software developers were reconsidering their approach to software development. They realised the necessity for swift adaptation to shifting requirements. They required the capacity to rapidly iterate and modify their whole software stack, including the database. NoSQL databases provided them with this versatility.

In this project, out of the two databases offered by Firebase Realtime Database, which are Realtime Database and Cloud Database, Cloud Database will be chosen to serve as the database of this application. Cloud Firestore is the newest mobile app development database from Firebase. It expands upon the achievements of the Realtime

Database by introducing a new, more understandable data model. Cloud Firestore supports more complex, quicker searches and grows more effectively than Realtime Database.

3.4.3 Entity Relational Diagram

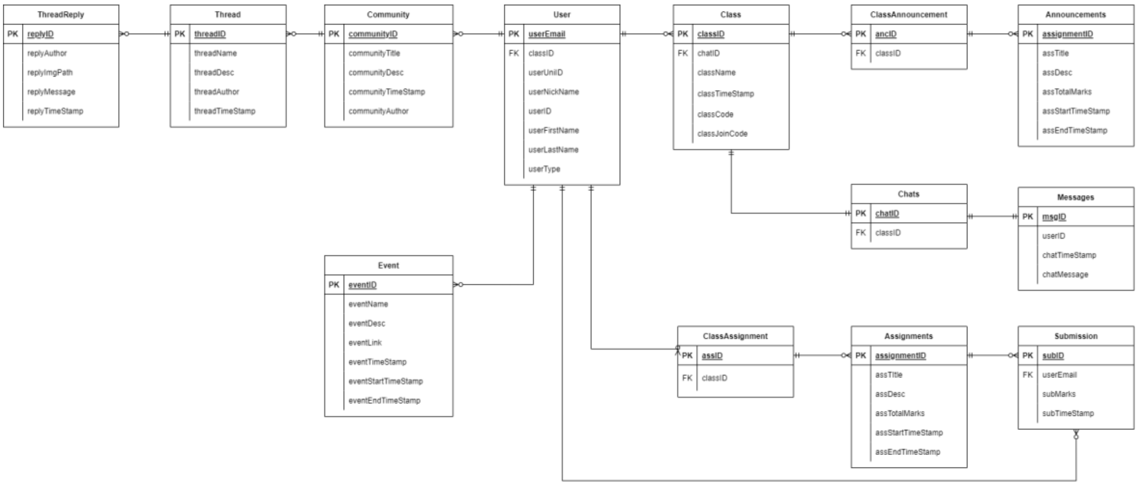


Figure 3.10 Entity Relational Diagram

Figure 3.10 depicts the entity relational diagram of the system. There are a total of 13 entities that stores data in the database. The User entity stores user personal information, as well as joined classes. It includes the credentials of the user as well. Next, the Class entity stores the information of the classes created. It is linked to various other entities such as the ClassAssignment and Submissions entity for the storage of assignments, ClassAnnouncement and Announcements entity for the storage of class announcements and Chats and Messages entity for classroom chat logs. inally, there is Community entity which is linked to the Thread and ThreadReply entities which store community threads and its replies for easy access. The rest of the data such as thread pictures and profile pictures are stored in Firebase Storage.

3.4.4 Data Dictionary

Table 3.5 User Table

Field Name	Data Type	Constraint	Description
userID	String	PK	User Identification Number
classID	Array	FK	Class Identification Number
userUniID	String		User University Identification
userNickName	String		User Community Nickname
userEmail	String		User Email
userFirstName	String		User First Name
userLastName	String		User Last Name
userType	String		User Type
userPassword	String		User Password
userTimeStamp	timestamp		User Timestamp

Table 3.6 Class Table

Field Name	Data Type	Constraint	Description
classID	String	PK	Class Identification Number
ancID	String	FK	Announcement Identification Number
assID	String	FK	Assignment Identification Number
className	String		Class Name
classTimeStamp	Timestamp		Class Created Timestamp
classCode	String		Class Code
classJoinCode	String		Class Joining Code

Table 3.7 Assignment Table

Field Name	Data Type	Constraint	Description
assID	String	PK	Assignment Identification Number
classID	String	FK	Class Identification Number

Table 3.8 ClassAssignment Table

Field Name	Data Type	Constraint	Description
assignmentID	String	PK	Assignment Identification Number
assTitle	String		Assignment Title

assDesc	String		Assignment Description
assStartTimeStamp	String		Assignment Start Timestamp
assEndTimeStamp	String		Assignment End Timestamp
assTotalMarks	Number		Assignment Total Marks

Table 3.9 Submission Table

Field Name	Data Type	Constraint	Description
subID	String	PK	Submission Identification Number
userEmail	String	FK	User Email
subMarks	Number		Submission Returned Marks

Table 3.10 ClassAnnouncement Table

Field Name	Data Type	Constraint	Description
ancID	String	PKFK1	Announcement Identification Number
classID	String	PKFK2	Class Identification Number

Table 3.11 Announcements

Field Name	Data Type	Constraint	Description
ancTimeStamp	Timestamp		Announcement Created Time Stamp
ancTitle	String		Announcement Title
ancMessage	String		Announcement Message

Table 3.12 Chat Table

Field Name	Data Type	Constraint	Description
chatID	String	PKFK1	Chat Identification Number
classID	String	PKFK2	Class Identification Number

Table 3.13 Messages Table

Field Name	Data Type	Constraint	Description
userID	String	FK	User Identification Number
chatMessage	String		Chat Message
chatTimeStamp	Timestamp		Chat Timestamp

Table 3.14 Event Table

Field Name	Data Type	Constraint	Description
eventID	String	PK	Event Identification Number
eventName	String		Event Name

eventDesc	String		Event Description
eventHost	String		Event Host Name
eventLink	String		Event Meeting Link
eventTimeStamp	Timestamp		Event Created Timestamp
eventStartTimeStamp	Timestamp		Event Start Timestamp
eventEndTimeStamp	Timestamp		Event End Timestamp

Table 3.15 Community Table

Field Name	Data Type	Constraint	Description
communityID	String	PK	Community Identification Number
communityTitle	String		Community Title
communityDesc	String		Community Description
communityTimeStamp	Timestamp		Community Created Timestamp
communityAuthor	String		Community Author Name

Table 3.16 Thread Table

Field Name	Data Type	Constraint	Description
threadID	String	PK	Thread Identification Number

threadAuthor	String		Thread Author Name
threadName	String		Thread Name
threadDesc	String		Thread Description
threadImgPath	String		Thread Image Path

Table 3.17 Reply Table

Field Name	Data Type	Constraint	Description
replyID	String		Reply Identification Number
replyAuthor	String		Reply Author Name
replyMessage	String		Reply Message
replyTimeStamp	Timestamp		Reply Created Time Stamp
replyImgPath	String		Reply Image Path

3.5 Testing Plan

The system will be tested by the developer on every iteration during its early phases of the prototype as the project is using the Rapid Application Development (R.A.D.) methodology. When the system is in its final phases, after it has undergone vigorous testing to ensure that the system works smoothly upon usage, bug-free, and has full functionality, the system will undergo User Acceptance Testing (UAT) and usability testing to ensure that the developed system can handle the intended tasks according to its specifications. The testing will be conducted by people with a technical background as well as a non-technical background ranging from students, and lecturers. Upon agreeing to conduct the test, the participants of the UAT will be given a set of instructions, access to the system, as well as an online questionnaire for the usability test. The results will be collected and analysed for future improvements.

3.6 Potential Use

The system is aimed to be deployed at educational institutions such as Malaysian public universities to aid in the e-learning process. Despite its shortcomings, the system caters to the modern needs of lecturers and students, facilitating real-time communication and academic discussions across the entirety of the students at a particular university through the built-in chat system and community module. By encouraging open academic discussions and peer-to-peer learning, students can engage is meaningful discussions in a safe environment. Not only that, but the system also facilitates online classroom management, providing management of educational resources, the grading of assignments and the relay of important announcements. The events module also exposes students to different kinds of talks, encouraging participation across the universities to worthwhile knowledge and thinking maturity.

CHAPTER 4

IMPLEMENTATION, RESULTS AND DISCUSSIONS

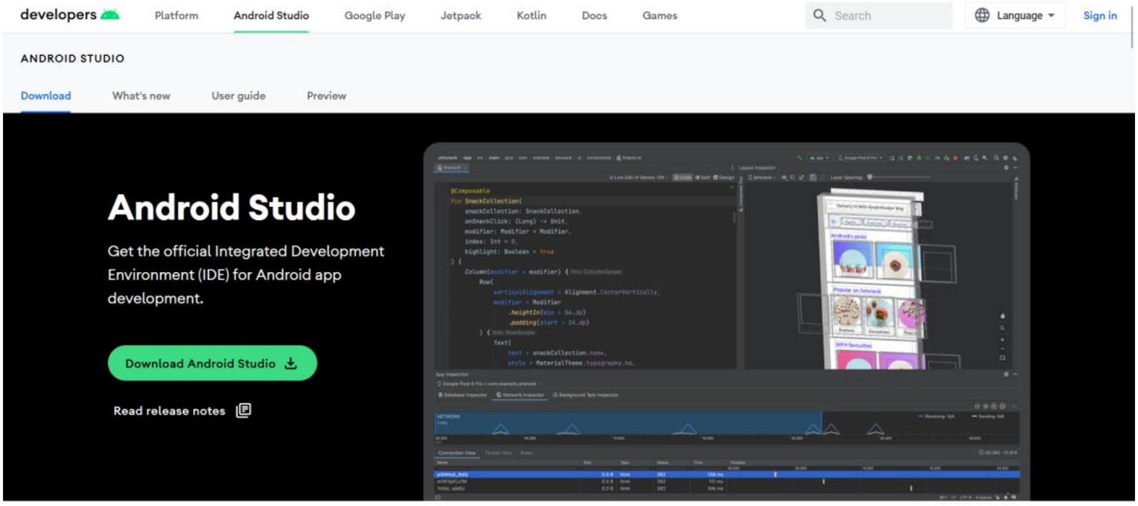
4.1 Introduction

This chapter elaborates on the development and implementation of the system throughout the project. Detailed explanations on the development and implementations are elaborated clearly based on the documented methodology, functionalities, and requirements in the previous chapters. The chapter begins with a discussion on the initial project setup.

4.2 Implementation

4.2.1 Initial Project Setup

4.2.1.1 Integrated Development Environment (IDE)

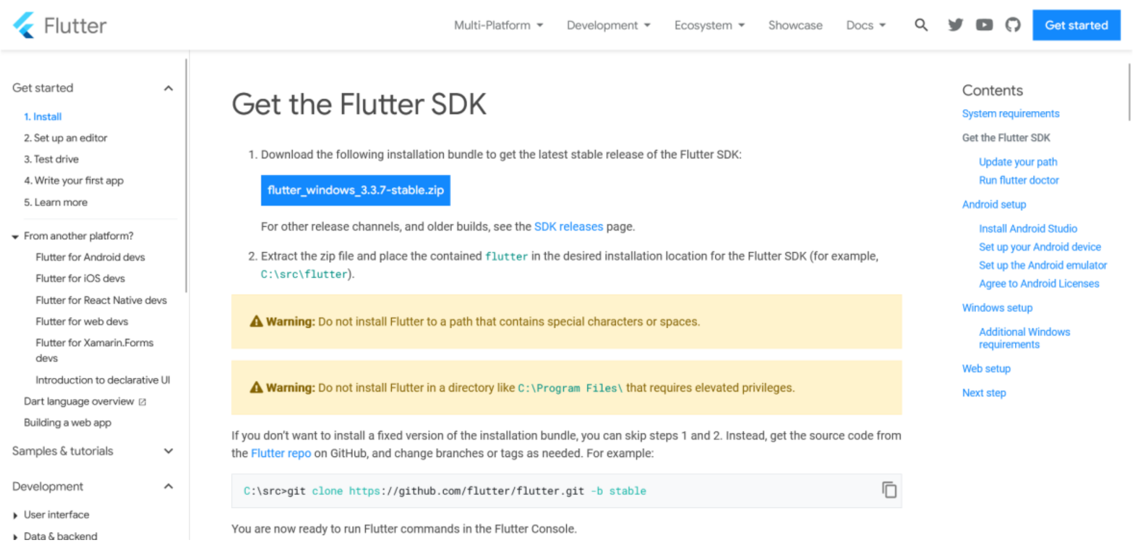


Get the official Integrated Development Environment (IDE) for Android app development.

```
@Composable
fun SnackCollection(
    snackCollection: SnackCollection,
    onSnackClick: (Long) -> Unit,
    modifier: Modifier = Modifier,
    index: Int = 0,
    highlight: Boolean = true
) {
    Column(modifier = modifier) { this: ColumnScope
        Row(
            verticalAlignment = Alignment.CenterVertically,
            modifier = Modifier
                .heightIn(min = 56.dp)
                .padding(start = 24.dp)
        ) { this: RowScope
            Text(
                text = snackCollection.name,
                style = MaterialTheme.typography.h6,
```

Figure 4.1 Android Studio Download Page

First and foremost, the Integrated Development Environment (IDE) has to be downloaded from the official website and installed in order to write code for the system. For this system, Android Studio, an IntelliJ based IDE will be used for the entire development process. The IDE also provides support for the user such as auto-complete, flutter integration as well as android emulator for an efficient development process.



Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

   flutter_windows_3.3.7-stable.zip

   For other release channels, and older builds, see the SDK releases page.

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`).

⚠ **Warning:** Do not install Flutter to a path that contains special characters or spaces.

⚠ **Warning:** Do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges.

If you don't want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the Flutter repo on GitHub, and change branches or tags as needed. For example:

```
C:\src>git clone https://github.com/flutter/flutter.git -b stable
```

You are now ready to run Flutter commands in the Flutter Console.

Figure 4.2 Flutter SDK Website

On top of the IDE, the Flutter SDK has to be downloaded and installed in order to create a new flutter project in the Android Studio IDE. After setting the path to the Flutter SDK in Android Studio settings, a new Flutter project can be created.

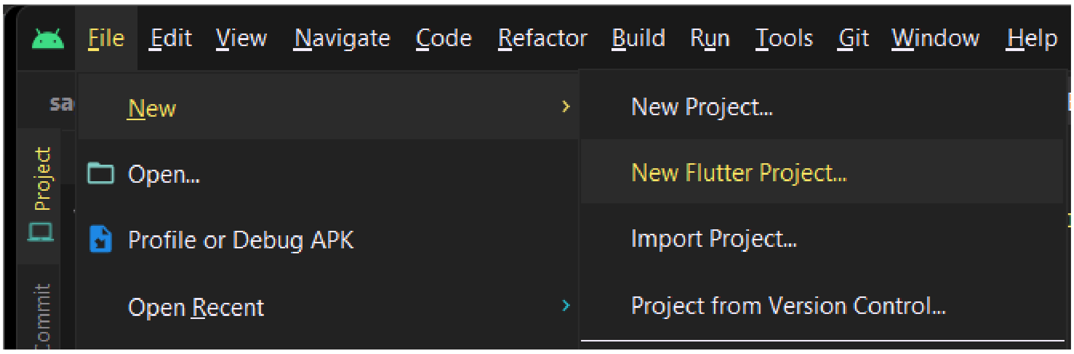


Figure 4.3 Android Studio New Flutter Project Option

4.2.1.2 Backend Services

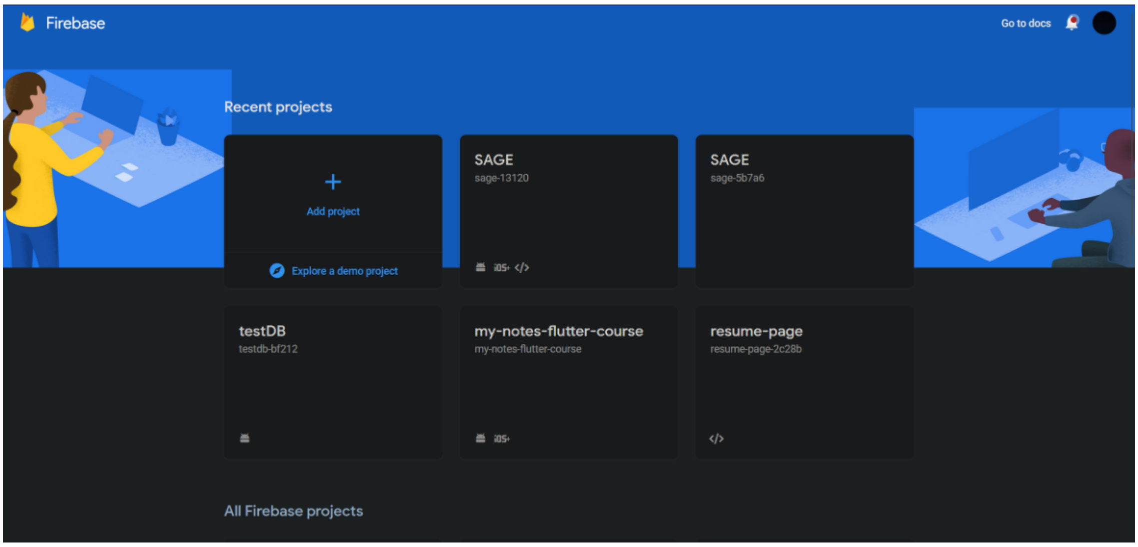
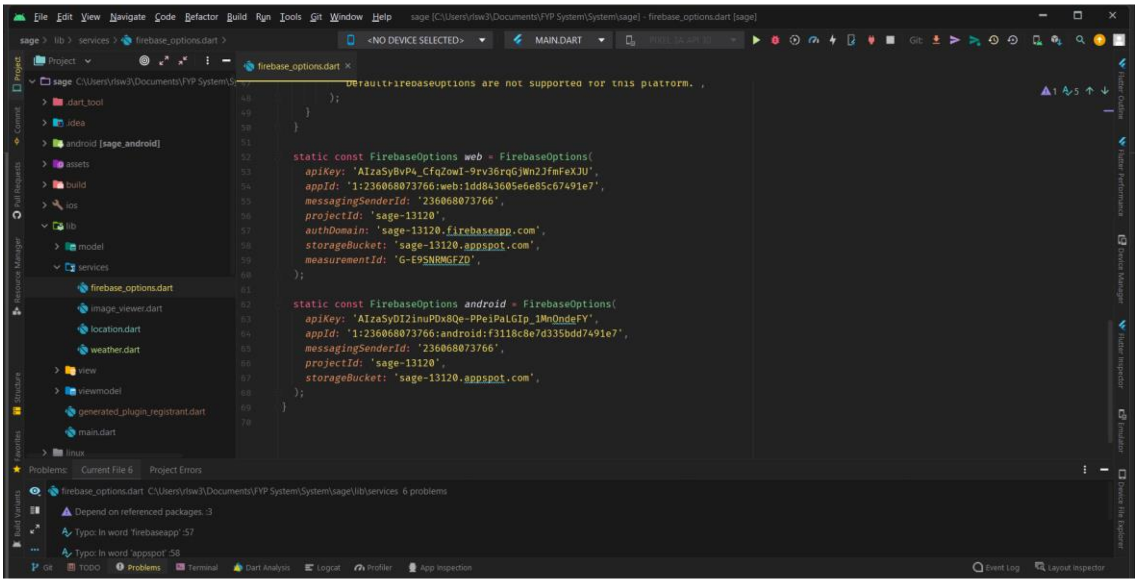


Figure 4.4 Firebase Console

Next, the backend module of the system, which is Firebase by Google is set up. It is done by first creating a new project in the Firebase console. In the process of creating a new project, the platform to be implemented upon, Android, is selected. After accessing the Firebase console, a new project is created by pressing the "Add Project" button. Firebase will initialize the new project and provide the necessary guidelines to follow according to the selected development platform.

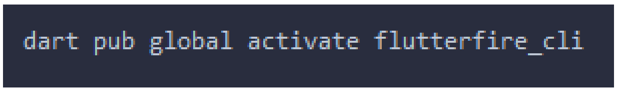


```
          'DefaultFirebaseOptions are not supported for this platform.',
        );
    }
  }

  static const FirebaseOptions web = FirebaseOptions(
    apiKey: 'AIzaSyBvP4_CfqZowI-9rv36rqGjWn2JfmFeXJU',
    appId: '1:236068073766:web:1dd843605e6e85c67491e7',
    messagingSenderId: '236068073766',
    projectId: 'sage-13120',
    authDomain: 'sage-13120.firebaseapp.com',
    storageBucket: 'sage-13120.appspot.com',
    measurementId: 'G-E9SNRMGFZD',
  );

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyDI2inuPDx8Qe-PPeiPaLGIp_1MnOndeFY',
    appId: '1:236068073766:android:f3118c8e7d335bdd7491e7',
    messagingSenderId: '236068073766',
    projectId: 'sage-13120',
    storageBucket: 'sage-13120.appspot.com',
  );
}
```

```
firebase_options.dart C:\Users\rlsw3\Documents\FYP System\System\sage\lib\services 6 problems
  Depend on referenced packages :3
  Typo: In word 'firebaseapp' :57
  Typo: In word 'appspot' :58
```

Figure 4.5 Firebase Configuration File

After that, Firebase will provide instructions on how to set up Firebase by providing a file containing the appropriate information such as API Key, application ID, project ID and authentication information. This file is to be inserted into the project directory and initialized at the start of the system.

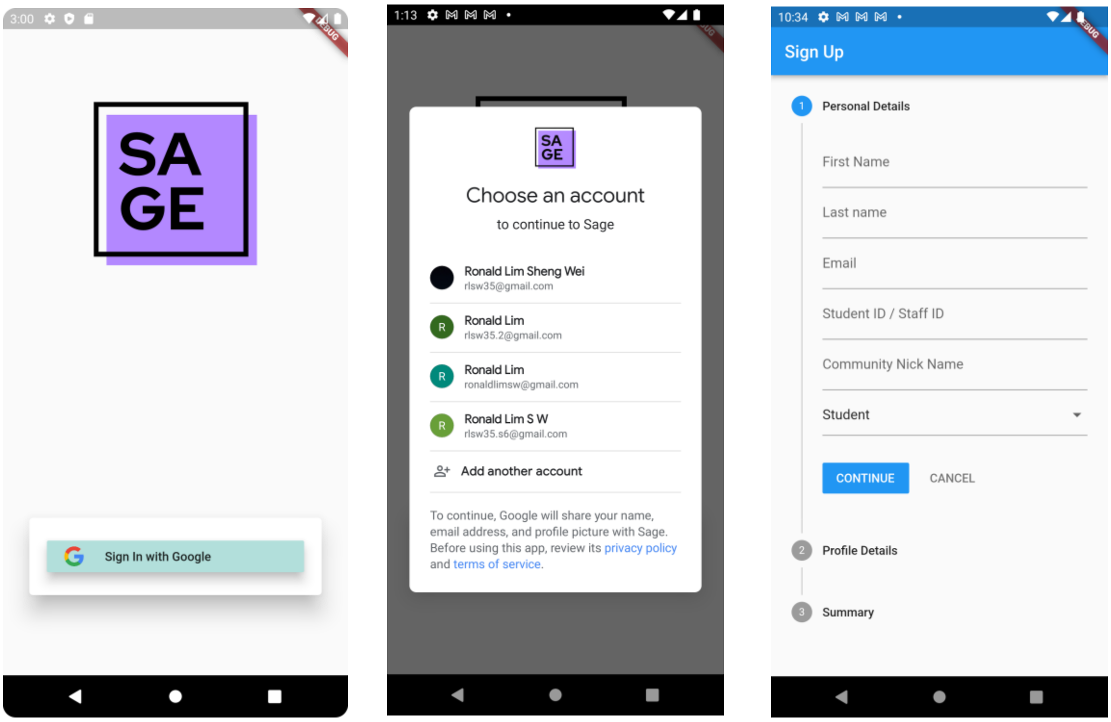


```
dart pub global activate flutterfire_cli
```

Figure 4.6 FlutterFire Installation Command

Installing the FlutterFire command line interface in the project directory's terminal takes precedence in order to access its services.

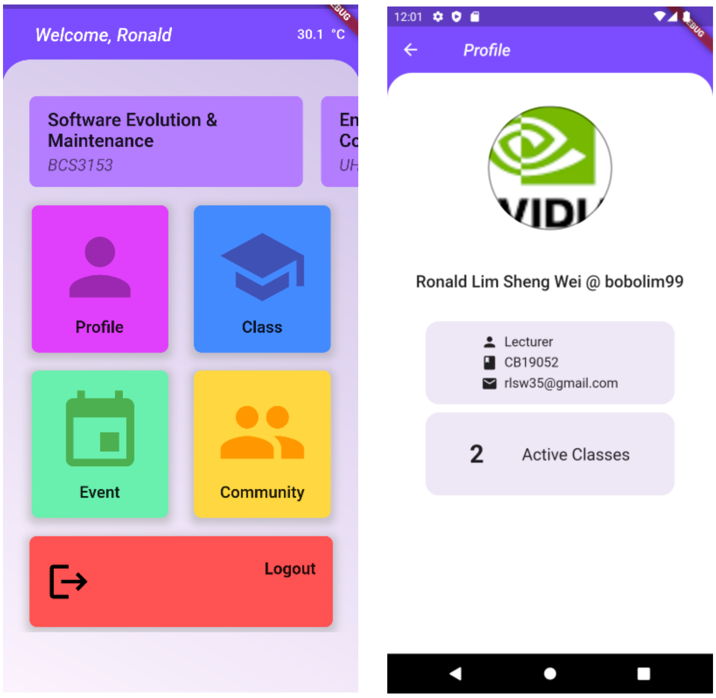
4.2.2 Interfaces



To continue, Google will share your name, email address, and profile picture with Sage. Before using this app, review its privacy policy and terms of service.

Figure 4.7 Login, Google Authentication, and Sign Up Interface

The SAGE system implements Firebase Authentication services by Google to ensure there are no spam bots. First, an account has to be chosen from a list of registered emails on the device. If the selected user email is a first-time user, the user will be directed to a sign-up screen where the user has to fill in personal details and select a profile picture.



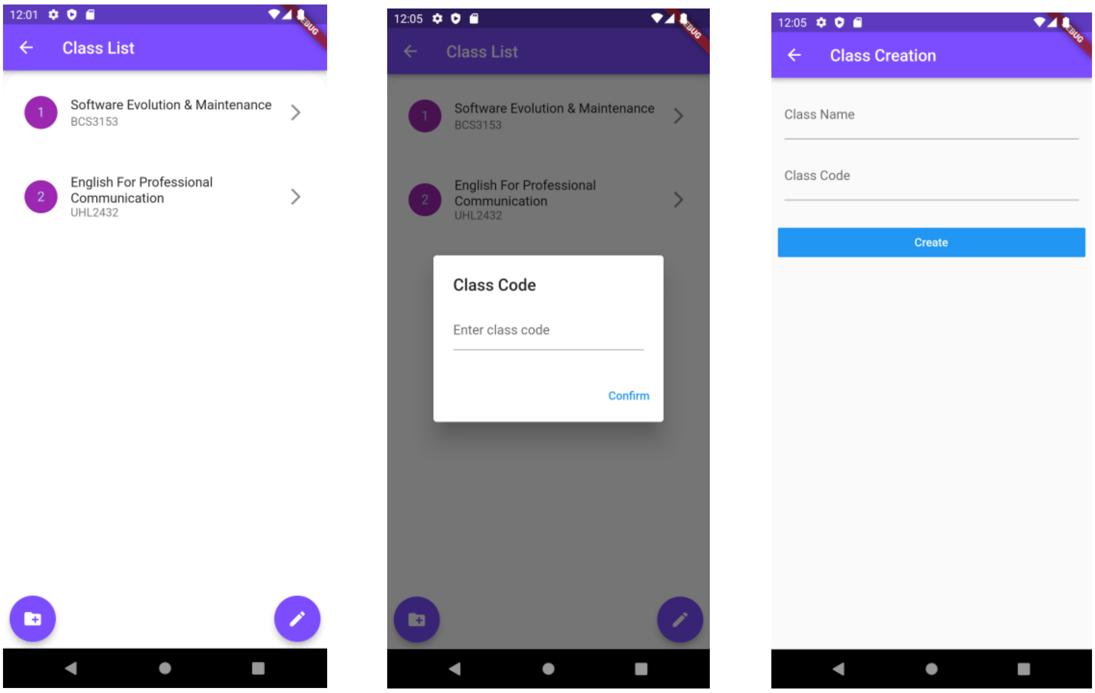
Ronald Lim Sheng Wei @ bobolim99

Lecturer
CB19052
rlsw35@gmail.com

2 Active Classes

Figure 4.8 Home and Profile Interface

After the authentication phase, users will arrive at the dashboard of the system where the system can navigate to different modules of the system. On the second screenshot of Figure 4.8, users can see the profile interface with various personal information displayed along with the active classes.

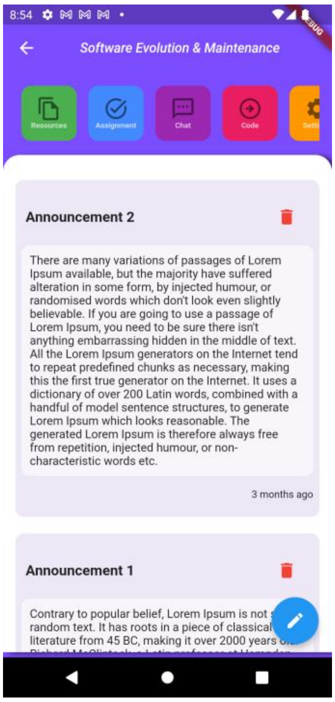


1 Software Evolution & Maintenance
BCS3153

2 English For Professional Communication
UHL2432

Figure 4.9 Class List and Class Creation Interface

Accessing the class module, users will arrive at the class list interface. This interface shows the classes that the student is enrolled in. To enrol in a new class, users can press the button located on the bottom right and enter a class code. This In this interface, the lecturer has additional functionalities such as an option to create a new class as shown in the third interface from Figure 4.9.



Announcement 2

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.

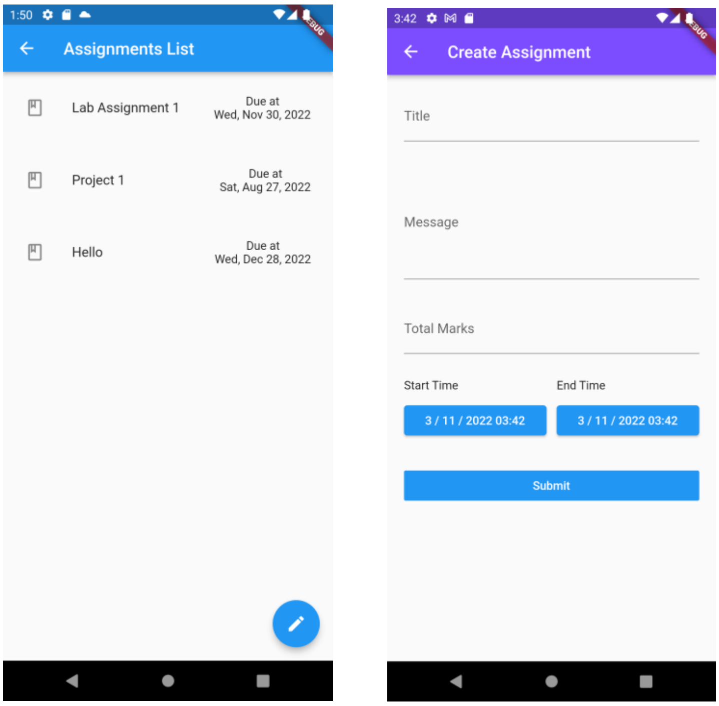
3 months ago

Announcement 1

Contrary to popular belief, Lorem Ipsum is not random text. It has roots in a piece of classical literature from 45 BC, making it over 2000 years

Figure 4.10 Class Home Interface

The class home contains various sub modules such as resources, assignments, chat, code and settings. The announcements can be viewed or deleted on the class home interface. On the other hand. New announcements can be created by pressing the bottom right icon. The announcement creation interface can be seen on the second interface of Figure 4.10.



Assignments List

Lab Assignment 1	Due at Wed, Nov 30, 2022
Project 1	Due at Sat, Aug 27, 2022
Hello	Due at Wed, Dec 28, 2022

Create Assignment

Title

Message

Total Marks

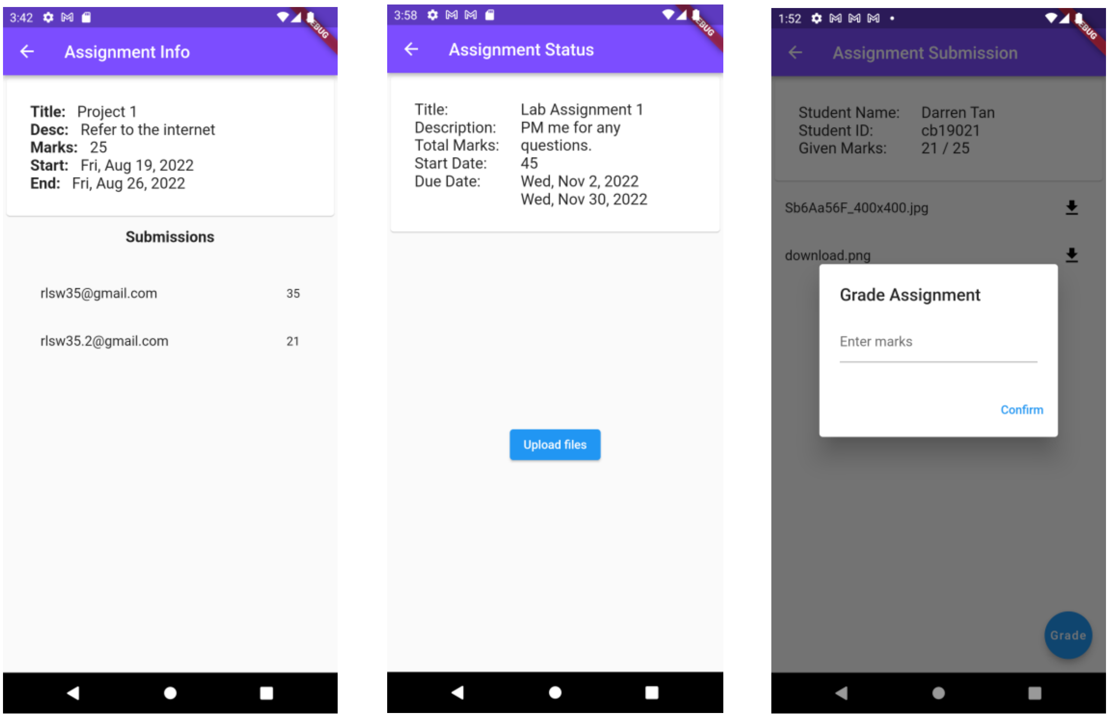
Start Time: 3 / 11 / 2022 03:42

End Time: 3 / 11 / 2022 03:42

Submit

Figure 4.11 Assignments List, Create Assignment Interface

The assignments list shows assignments that can be created by the lecturer of the class. There is an interface on the right which shows text form fields that are required to create an assignment.



Assignment Info

Title: Project 1
Desc: Refer to the internet
Marks: 25
Start: Fri, Aug 19, 2022
End: Fri, Aug 26, 2022

Submissions

rlsw35@gmail.com	35
rlsw35.2@gmail.com	21

Assignment Status

Title:	Lab Assignment 1
Description:	PM me for any questions.
Total Marks:	45
Start Date:	Wed, Nov 2, 2022
Due Date:	Wed, Nov 30, 2022

Upload files

Assignment Submission

Student Name:	Darren Tan
Student ID:	cb19021
Given Marks:	21 / 25

Sb6Aa56F_400x400.jpg

download.png

Grade Assignment

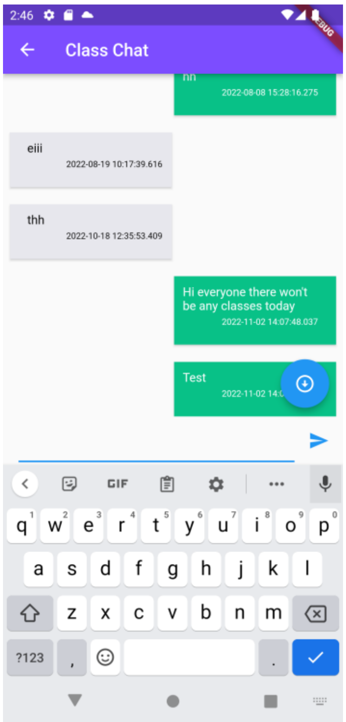
Enter marks

Confirm

Grade

Figure 4.12 Assignment Submissions, Student Submission, Grade Assignment Interface

From the lecturer’s view, student’s submissions can be accessed easily. The uploaded content can be downloaded and viewed. On the other hand, lecturer can grade submissions.



Class Chat

hh
2022-08-08 15:28:16.275

eiii
2022-08-19 10:17:39.616

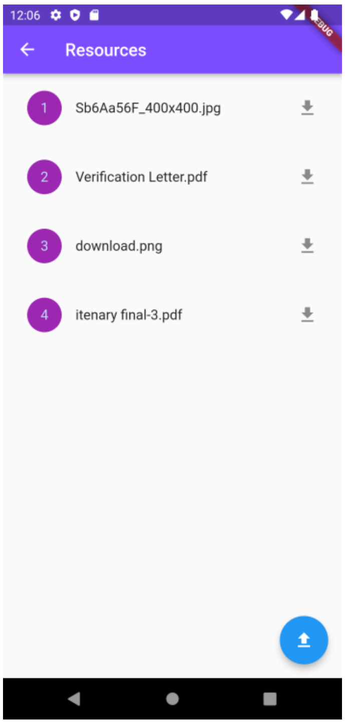
thh
2022-10-18 12:35:53.409

Hi everyone there won't be any classes today
2022-11-02 14:07:48.037

Test
2022-11-02 14:0

Figure 4.13 Class Chat Interface

Within a class, there is an instance of a class chat. This class chat allows students and lecturers within the enrolled class to communicate effectively with each other.



1 Sb6Aa56F_400x400.jpg

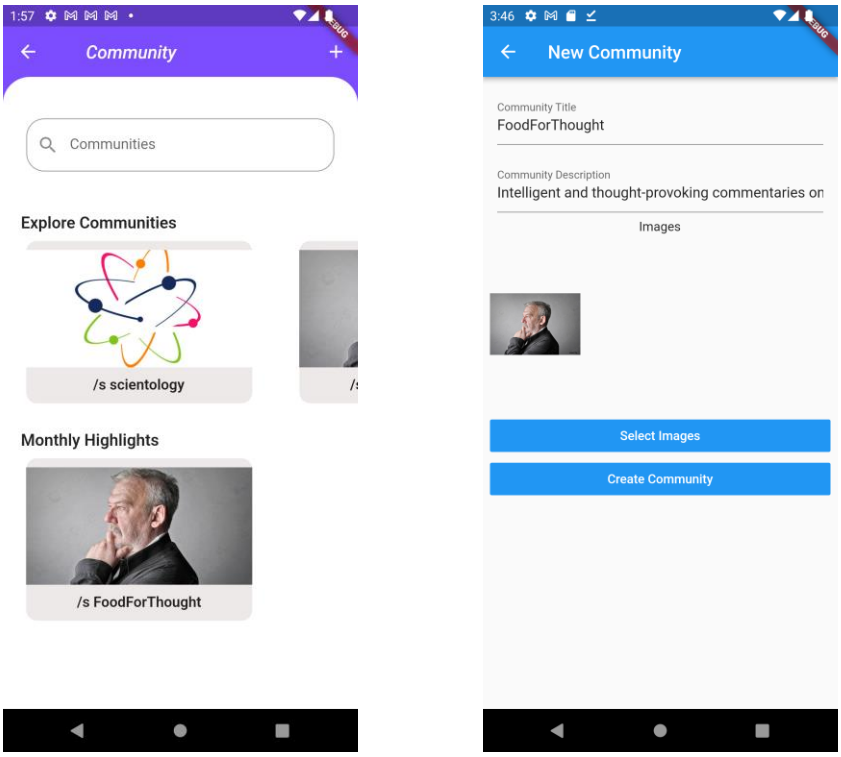
2 Verification Letter.pdf

3 download.png

4 itenary final-3.pdf

Figure 4.14 Resource List Interface

The resource list enables lecturers to uploaded important files for viewing by pressing the floating action button on the bottom right. Lecturers will be redirected to the file explorer of the android operating system.

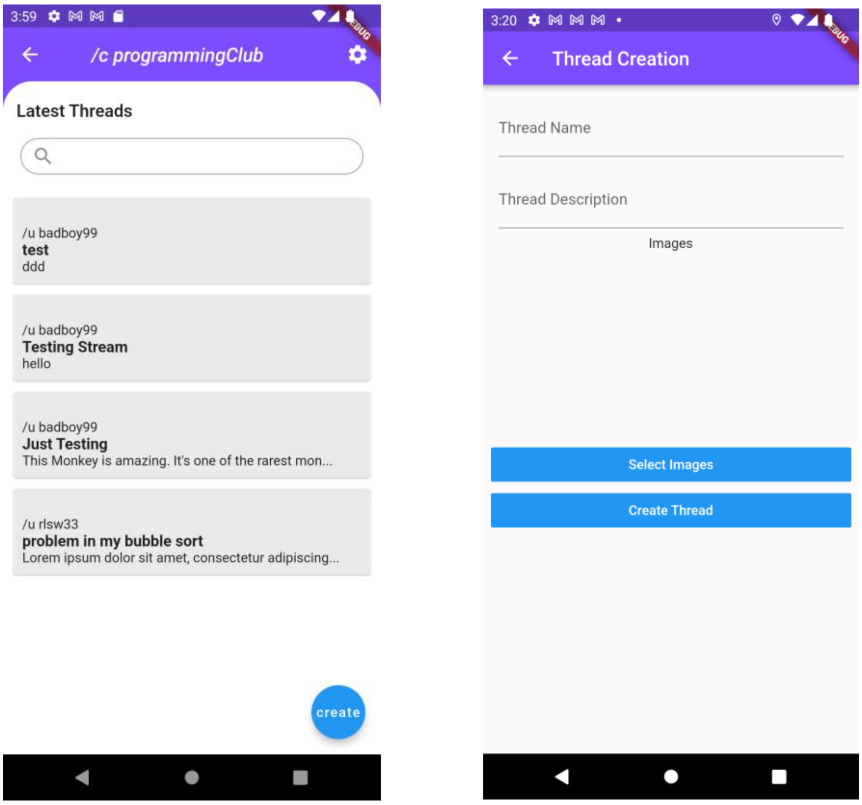


Community Title
FoodForThought

Community Description
Intelligent and thought-provoking commentaries on

Figure 4.15 Community Index and New Community Creation Interface

The community index page allows user to explore new communities and access the threads that are written by users. There is also a monthly highlight section which introduces newly created communities in the month. On the right, there is an interface to create new communities by inputting the title description and uploading the community image.



/u badboy99
test
ddd

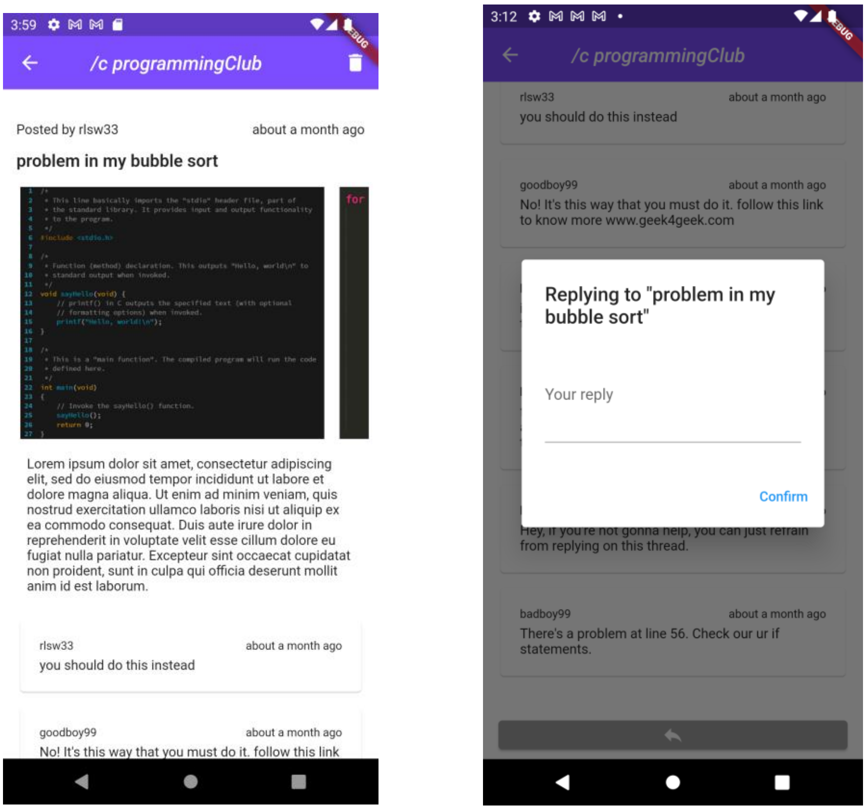
/u badboy99
Testing Stream
hello

/u badboy99
Just Testing
This Monkey is amazing. It's one of the rarest mon...

/u rlsw33
problem in my bubble sort
Lorem ipsum dolor sit amet, consectetur adipiscing...

Figure 4.16 Thread List, Thread Creation Interface

The first screenshot of Figure 4.15 shows the threads created by users participating in the community. The following screenshot shows the thread creation form to create new threads in the selected community.



Posted by rlsw33 about a month ago

problem in my bubble sort

```
/*
 * This line basically imports the "stdio" header file, part of
 * the standard library. It provides input and output functionality
 * to the program.
 */
#include <stdio.h>

/*
 * Function (method) declaration. This outputs "Hello, world\n" to
 * standard output when invoked.
 */
void sayHello(void) {
    // printf() in C outputs the specified text (with optional
    // formatting options) when invoked.
    printf("Hello, world!\n");
}

/*
 * This is a "main function". The compiled program will run the code
 * defined here.
 */
int main(void)
{
    // Invoke the sayHello() function.
    sayHello();
    return 0;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

rlsw33 about a month ago
you should do this instead

goodboy99 about a month ago
No! It's this way that you must do it. follow this link

rlsw33 about a month ago
you should do this instead

goodboy99 about a month ago
No! It's this way that you must do it. follow this link to know more www.geek4geek.com

Replying to "problem in my bubble sort"

Your reply

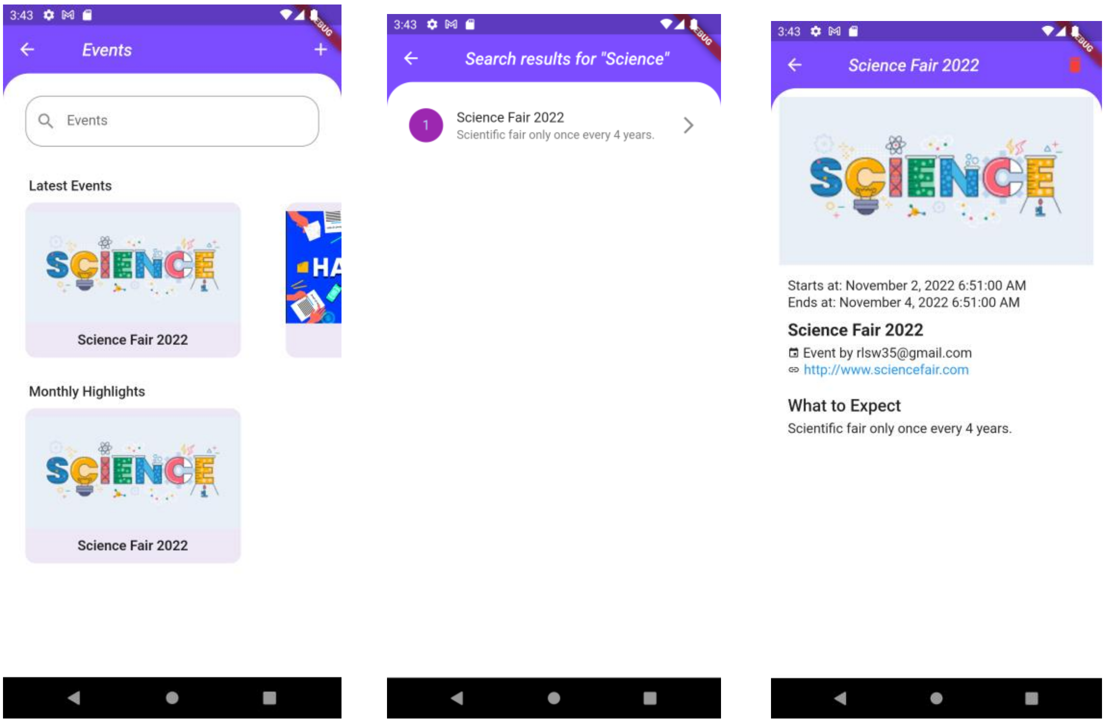
Confirm

Hey, if you're not gonna help, you can just refrain from replying on this thread.

badboy99 about a month ago
There's a problem at line 56. Check our ur if statements.

Figure 4.17 Thread Details, Thread Reply Interface

The thread details page shows pictures uploaded during the thread creation process along with the description of the problem. After that, there is the comments sections where the original poster or other users can comment to help solve the raised issue. The thread reply interface allows users to type in their replies to the selected thread.



1 Science Fair 2022
Scientific fair only once every 4 years.

Starts at: November 2, 2022 6:51:00 AM
Ends at: November 4, 2022 6:51:00 AM

Science Fair 2022

Event by rlsw35@gmail.com
http://www.sciencefair.com

What to Expect

Scientific fair only once every 4 years.

Figure 4.18 Event Home, Event Search, Event Details Interface

Figure 4.17 shows the event module's index page in the first screenshot. This interface allows users to view latest events, create their own events and view monthly highlights and search for events. The second interface shows search results for the selected keyword. Finally, the event details interface. This interface displays the start and end datetime of the event, followed by the host, URL link and description.

4.3 Testing

Two tests were selected and conducted on the SAGE system, namely, the User Acceptance Test (UAT) and Usability Test. The purpose of UAT is to ensure that system can handle real-world scenarios with its current functions. On the other hand, the UAT was done with minimal supervision, with only written instructions available to the targeted tester. On the other hand, the usability test was also done without bias or intervention, ensuring that the interfaces are friendly and suitable for human interaction. The results obtained from the two tests are attached as **APPENDIX B** and **APPENDIX C** for the UAT and Usability Test respectively. Below are the profiles of the testers involved in the two tests mentioned.

Table 4.1 Tester Profile

Portrait Image	Name	User Type
	Hugh John Leong	Lecturer
	Wong Sung Sum	Student

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 5 discusses the summary of the development of the SAGE system in order to fulfill the stated objectives and problem statements as stated in Chapter 1 of this thesis. To reiterate Chapter 1's contents, students of the modern era face the difficulty of maintaining peer-to-peer learning environment outside of their physical time around campus.

In addition, students lack a platform to come together for academic discussions in the university environment. Therefore, this application serves as a solution to the underlying problem that fellow students are facing.

In order to ensure the success of the system, the development of this system has employed the use of software such as Android Studio as the integrated development environment, Flutter as the application's main framework as well as Firebase tools and infrastructure as it's cloud-based database system, data storage system and authentication system.

The methodology used during development is the Rapid Application Development (RAD) methodology to allow for flexible changes to requirements, a shorter development timeframe, and the ability to retrack to previous development phases with less difficulties.

This application has undergone evaluation by two types of users, namely, a lecturer and a student. The user acceptance test (UAT) and Usability Tets has shown the effectiveness, operability, and functionality of the system as well as the items to improve on in the near future.

5.2 Research Constraint

i. Time

The limited time has inhibited more advanced functionalities to be implemented such as moderators in the community module, co-host in the events module, as well as a function to export grades in the class module.

ii. Manpower

Due to the limitation of only a single person planning, developing and testing the project, and preparation of technical documents such as the SRS and the SDD, it requires tremendous time, effort, and focus to balance the workload and quality of the documents.

5.3 Future Works

Even though the developed SAGE application meets the stated requirements, the continued expansion and improvement of the application promises a more featureful, useful and practical revision of the application. A few possibilities of future works are as written below.

i. Improved Interoperability

Since the application uses Flutter, integration to the web platform, IOS platform as well as the Microsoft Windows platform requires little effort and refactoring of code. This means that users can access the application from various devices, improving interoperability as well as accessibility to the masses.

ii. Improved Moderation

The application involves peer-to-peer communication and interaction. Hence, features such as whitelisting of vulgar words and AI based detection of malicious words can be implemented. In addition, a new user type, the moderator, can be added to moderate content and report any activities that violate community guidelines. This ensures a safe learning environment for lecturers as well as students.

iii. Network Usage Optimization

The application involves a lot of data retrieval and data upload to and from the application and the cloud-based services. This usage of data is unsuitable when there are plans to scale the application. Hence, implementing the use of cache to store loaded content throughout the application can help reduce network usage. This in turn reduces the cost of running the application as the read and write usage is drastically reduced.

iv. Additional Testing

The application still requires rigorous testing under real conditions by real users. This is to ensure that the system is up to standards and user’s expectations. Additional testing will involve many more actual users for both user types in order to gain useful feedback and insight on the functionality of the system.

REFERENCES

4 Phases of Rapid Application Development Methodology. (2018, May 23). @Lucidchart. https://www.lucidchart.com/blog/rapid-application-development-methodology

Agarwal, S., & Dewan, J. (n.d.). An Analysis of the Effectiveness of Online Learning in Colleges of Uttar Pradesh during the COVID 19 Lockdown. Journal of Xi'an University of Architecture & Technology, 12(3), 2957-2963. https://www.xajzkjdx.cn/gallery/311-may2020.pdf

Azhar, K.A., Iqbal N. (2018). EFFECTIVENESS OF GOOGLE CLASSROOM: TEACHERS' PERCEPTIONS. Prizren Social Science Journal, 2(2), 52–66. https://www.ceeol.com/search/article-detail?id=940663

Beynon-Davies, P., Carne, C., Mackay, H., & Tudhope, D. (1999). Rapid application development (RAD): an empirical review. European Journal of Information Systems, 8(3), 211–223. https://doi.org/10.1057/palgrave.ejis.3000325

Chicioreanu, T. D., & Cosma, I. (2017). I AM A TEACHER IN THE DIGITAL ERA. WHAT TO CHOOSE: GOOGLE CLASSROOM OR MOODLE?. eLearning & Software for Education, 2.

Choi, J. A., Kim, O., Park, S., Lim, H., & Kim, J.-H. (2021). The Effectiveness of Peer Learning in Undergraduate Nursing Students: A Meta-Analysis. Clinical Simulation in Nursing, 50, 92–101. https://doi.org/10.1016/j.ecns.2020.09.002

Ekayati, R. (2018). Implementasi Metode Blended Learning Berbasis Aplikasi Edmodo. EduTech: Jurnal Ilmu Pendidikan dan Ilmu Sosial, 4(2), 50–56. https://doi.org/10.30596/EDUTECH.V4I2.2277

Etfita, F. (2019). Students' perspective on the use of edmodo as an assessment tool. J-SHMIC : Journal of English for Academic, 6(1), 18–25. https://doi.org/10.25299/jshmic.2019.vol6(1).2516

Gay, E., & Sofyan, N. (2017). The effectiveness of using edmodo in enhancing students outcomes in advance writing course of the fifth semester at FIP - UMMU. Journal of English Education, 2(1), 1–11. https://doi.org/10.31327/jee.v2i1.217

Hardjana, Agus M. (2003). Komunikasi Intrapersonal dan Interpersonal, Yogyakarta : Kanisius

Koksal, I. (2021, December 10). The Rise Of Online Learning. Forbes. https://www.forbes.com/sites/ilkerkoksal/2020/05/02/the-rise-of-online-learning/?sh=69dc65da72f3

Shamionov, R. M., Grigorieva, M. V., Grinina, E. S., & Sozonnik, A. V. (2021). The Role of Personality Characteristics and Social Activity in the Academic Adaptation of University Students with Chronic Diseases. Клиническая и специальная психология, 10(3), 181–207. https://doi.org/10.17759/cpse.2021100310

Moodle 3.11 release notes - MoodleDocs. (2021). Moodle.org. https://docs.moodle.org/dev/Moodle_3.11_release_notes#Browser_support

Moodle 3.11 release notes - MoodleDocs. (2021). Moodle.org. https://docs.moodle.org/dev/Moodle_3.11_release_notes#Browser_support

Moodle App Overview - MoodleDocs. (2021). Moodle.org. https://docs.moodle.org/dev/Moodle_App_Overview

Moodle App Overview - MoodleDocs. (2021). Moodle.org. https://docs.moodle.org/dev/Moodle_App_Overview

Nurhayati, D. A. W. (2019). Students‘ Perspective on Innovative Teaching Model Using Edmodo in Teaching English Phonology: A Virtual Class Development. Dinamika Ilmu, 19(1), 13–35. https://doi.org/10.21093/di.v19i1.1379

Okmawati, M. (2020). The Use of Google Classroom during Pandemic. Journal of English Language Teaching, 9(2), 438–443. http://ejournal.unp.ac.id/index.php/jelt/article/view/109293/103809

Petrovici, A., & Ciobanu, E. P. (2016). THE LESSON, MOODLE TEACHING-LEARNING RESOURCE WITH INTERRACTIVE CONTENT. eLearning & Software for Education, 3.

Rapanta, C., Botturi, L., Goodyear, P., Guàrdia, L., & Koole, M. (2021). Balancing Technology, Pedagogy and the New Normal: Post-pandemic Challenges for Higher Education. *Postdigital Science and Education*, *3*(3), 715–742. https://doi.org/10.1007/s42438-021-00249-1

Sarimah, S. (2021). Digital Divide in Education during COVID-19 Pandemic. Jurnal Ekonomi Malaysia, 55(3), 103–112. https://doi.org/10.17576/jem-2021-5503-07

Sudarsana K., et. al. (2019). The use of Google classroom in the learning process. Journal of Physics: Conference Series. doi:10.1088/1742-6596/1175/1/012165

Teachinghistory.org. (2018). Teachinghistory.org. https://teachinghistory.org/digital-classroom/tech-for-teachers/25425

Dupont, S., Galand, B., & Nils, F. (2015). The impact of different sources of social support on academic performance: Intervening factors and mediated pathways in the case of master’s thesis. *Revue Européenne de Psychologie Appliquée/European Review of Applied Psychology*, *65*(5), 227–237. https://doi.org/10.1016/j.erap.2015.08.003

Web Browser Requirements & Troubleshooting. (2022, January 4). Edmodo Help Center. https://support.edmodo.com/hc/en-us/articles/216676477-Web-Browser-Requirements-Troubleshooting#:~:text=In%20general%2C%20Edmodo%20supports%20the,some%20features%20may%20not%20work.

Li, Y., Wang, A., Wu, Y., Han, N., & Huang, H. (2021). Impact of the COVID-19 Pandemic on the Mental Health of College Students: A Systematic Review and Meta-Analysis. *Frontiers in Psychology*, *12*. https://doi.org/10.3389/fpsyg.2021.669119

APPENDIX A
USER MANUAL FOR SAGE

1.0 Introduction

This section contains information about the usage of the SAGE system. Users can refer to this section as a guideline on the ways to operate the system in a proper manner.

2.0 System Requirements

The SAGE mobile application requires Android 4.1 (API level 16) operating system or higher to support the Flutter framework that it is built upon. Users are also required to have an existing google account to access SAGE's services as well as a fast internet connection of at least 8mbps to ensure a smooth experience.

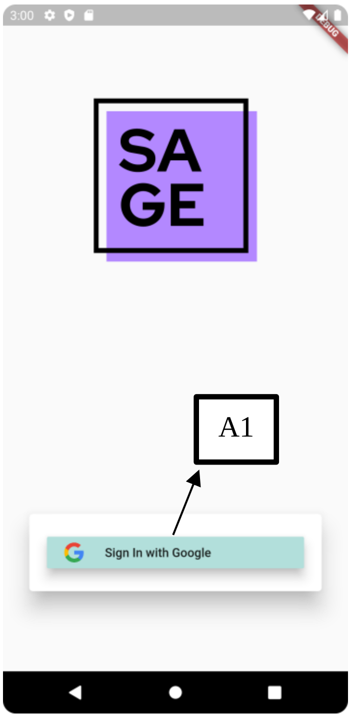
3.0 Getting Started

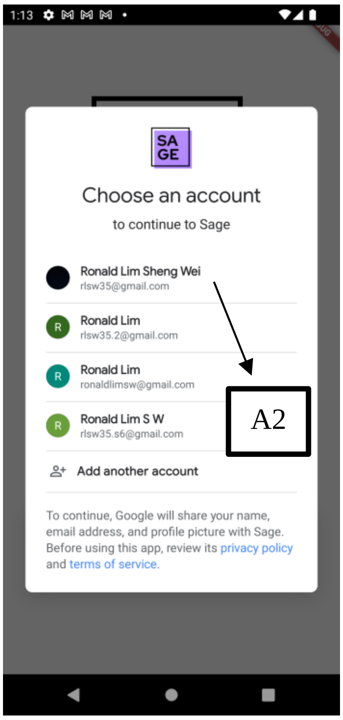
3.1 System Controls

The SAGE application should be operated with touch screen input and an on-screen keyboard.

4.0 User Manual

4.1 System Login and Registration





To continue, Google will share your name, email address, and profile picture with Sage. Before using this app, review its privacy policy and terms of service.

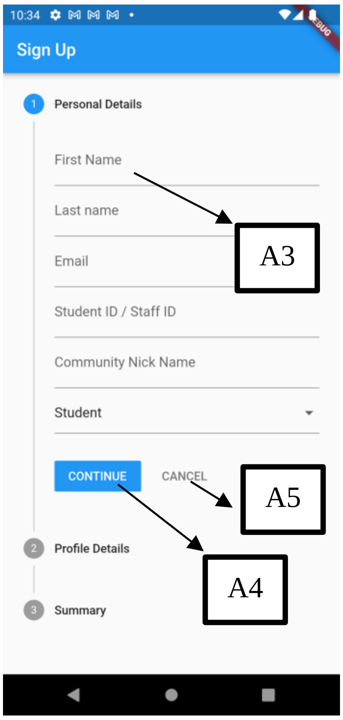


Figure 4.1 System Login and Registration

FUNCTION	DESCRIPTION
A1	Navigate to Google Selection interface
A2	Select Google account to login
A3	Fill in required personal details
A4	Navigate to Profile Details
A5	Cancel registration of new account

4.2 Home Dashboard and Profile

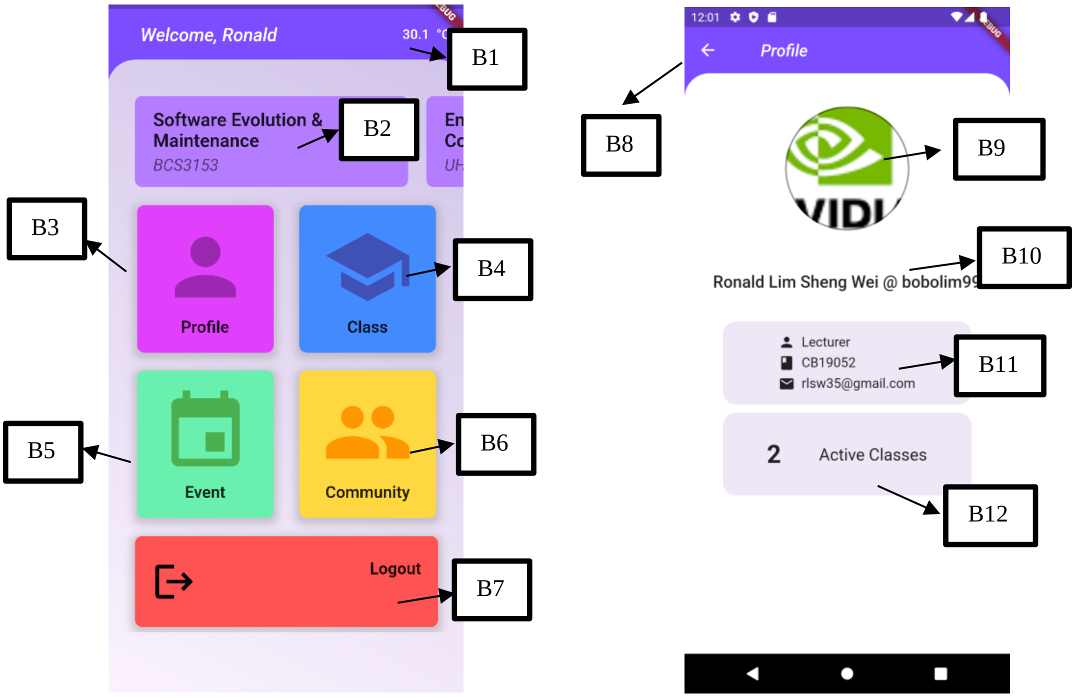


Figure 4.2 Home Dashboard and Profile

FUNCTION	DESCRIPTION
B1	Display weather information
B2	Navigate to recent classes
B3	Navigate to profile page
B4	Navigate to class page
B5	Navigate to event page
B6	Navigate to community page
B7	Logout of current session
B8	Back button to main menu
B9	Display profile picture
B10	Display full name and community name
B11	Display user type, user identification number, and email.
B12	Display number of active classes.

4.3 Class List, Class Code and Class Creation

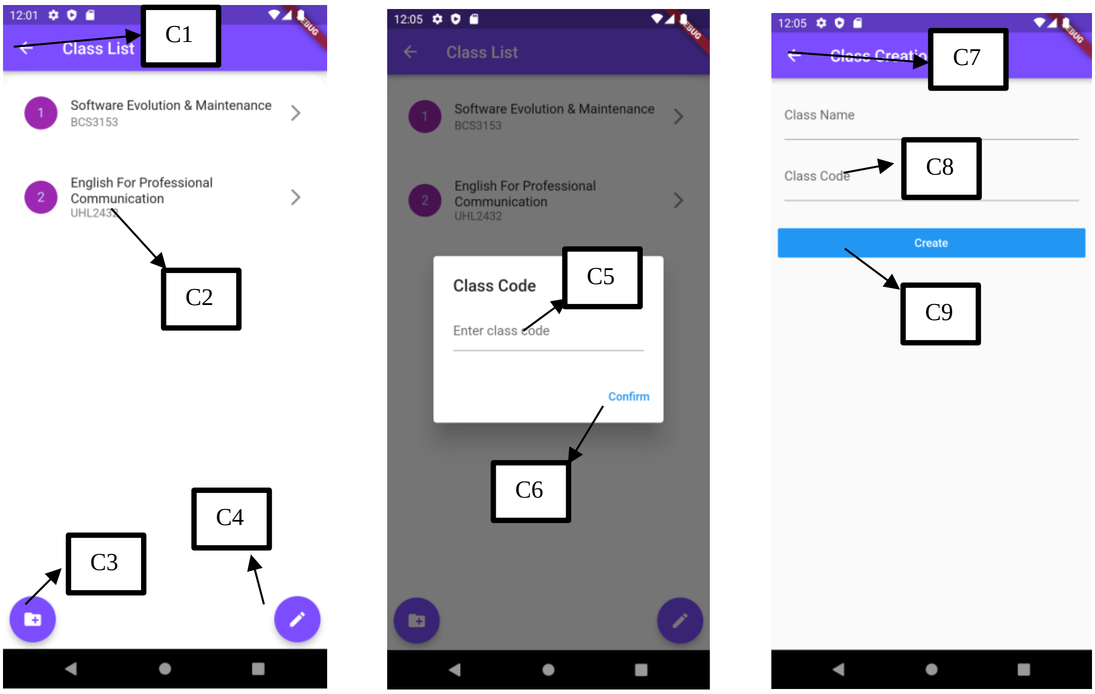
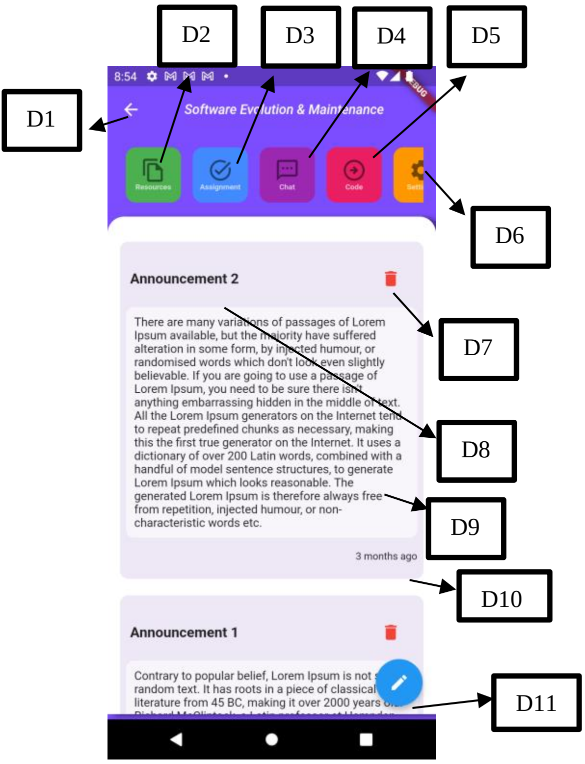


Figure 4.3 Class List, Class Code and Class Creation

FUNCTION	DESCRIPTION
C1	Navigate back to main menu
C2	Navigate to course page
C3	Create new class
C4	Activate pop
C5	Input class code
C6	Submit class code to be added
C7	Navigate back to class list

C8	Input information required to create new class
C9	Submit new class information

4.4 Class Index



Announcement 2

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.

3 months ago

Announcement 1

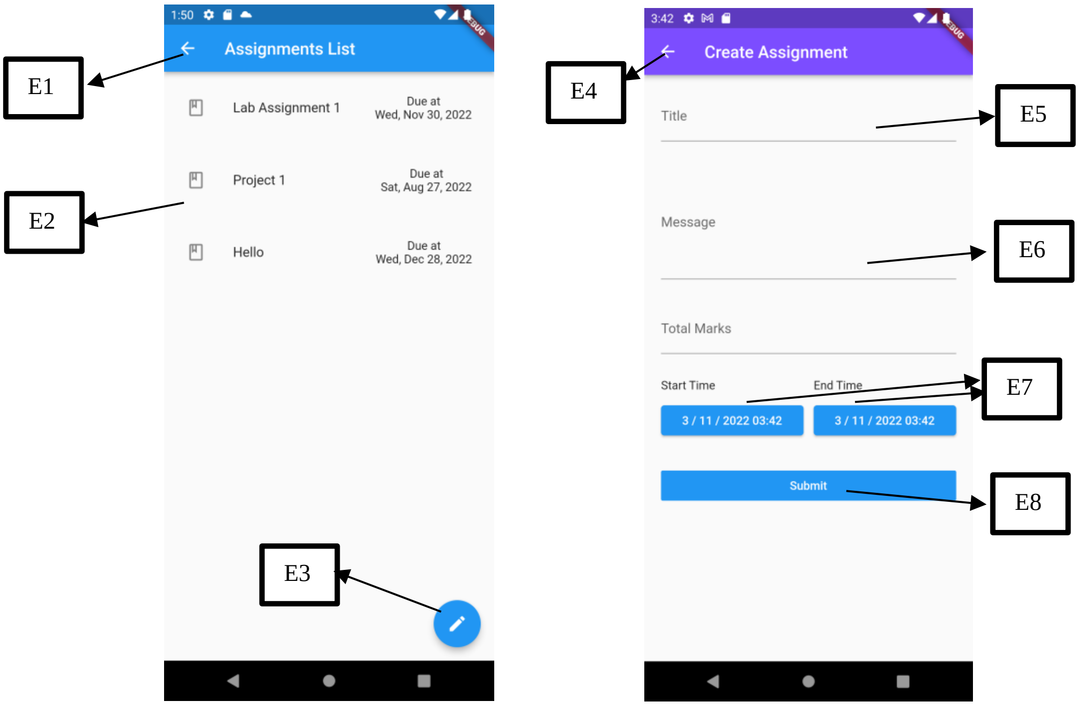
Contrary to popular belief, Lorem Ipsum is not random text. It has roots in a piece of classical literature from 45 BC, making it over 2000 years

Figure 4.4 Class Index

FUNCTION	DESCRIPTION
D1	Navigate back to main menu
D2	Navigate to resource page
D3	Navigate to assignment page
D4	Navigate to chat page

D5	Navigate to class code page
D6	Navigate to class settings page
D7	Delete announcement
D8	Display announcement title
D9	Display announcement content
D10	Display announcement posting date
D11	Navigate to announcement creation page

4.5 Assignments List, Assignment Creation



1:50

Assignments List

Lab Assignment 1 — Due at Wed, Nov 30, 2022

Project 1 — Due at Sat, Aug 27, 2022

Hello — Due at Wed, Dec 28, 2022

3:42

Create Assignment

Title

Message

Total Marks

Start Time: 3 / 11 / 2022 03:42

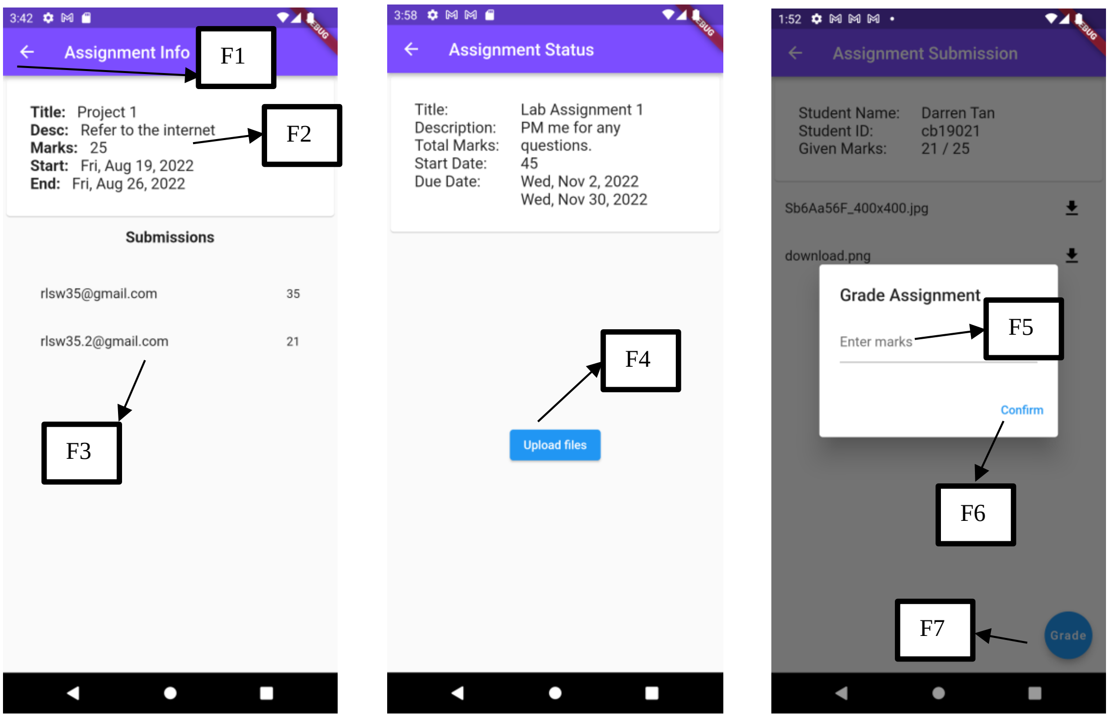
End Time: 3 / 11 / 2022 03:42

Submit

Figure 4.5 Assignments List, Assignment Creation

FUNCTION	DESCRIPTION
E1	Navigate back to class index.
E2	Navigate to assignment details page of selected assignment.
E3	Navigate to assignment creation page.
E4	Navigate to assignment list page.
E5	Input assignment title field.
E6	Input assignment message field.
E7	Input assignment start time and end time field.
E8	Submit creation of new assignment.

4.6 Assignments Information, Assignment Status, Assignment Submission



Title: Project 1
Desc: Refer to the internet
Marks: 25
Start: Fri, Aug 19, 2022
End: Fri, Aug 26, 2022

rlsw35@gmail.com	35
rlsw35.2@gmail.com	21

Title:	Lab Assignment 1
Description:	PM me for any questions.
Total Marks:	45
Start Date:	Wed, Nov 2, 2022
Due Date:	Wed, Nov 30, 2022

Student Name:	Darren Tan
Student ID:	cb19021
Given Marks:	21 / 25

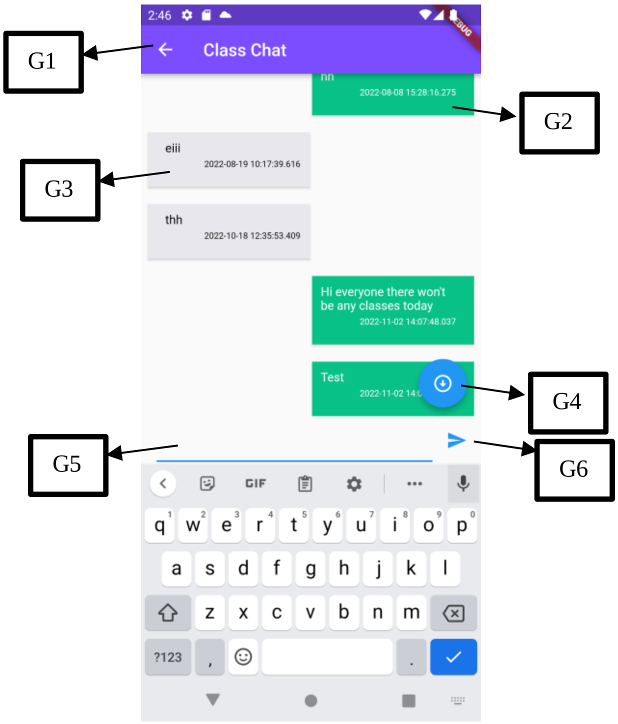
Sb6Aa56F_400x400.jpg

download.png

Figure 4.6 Assignments Information, Assignment Status, Assignment Submission

FUNCTION	DESCRIPTION
F1	Navigate back to class index page.
F2	Display assignment information.
F3	Navigate to assignment status page of selected student
F4	Upload assignment files
F5	Input assignment marks.
F6	Submit assignment marks.
F7	Prompt grade assignment input pop up.

4.7 Class Chat



hh
2022-08-08 15:28:16.275

eiii
2022-08-19 10:17:39.616

thh
2022-10-18 12:35:53.409

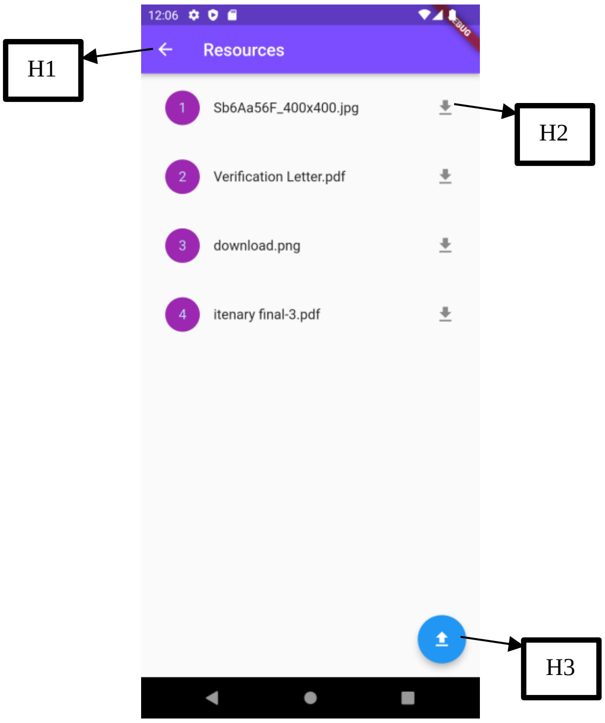
Hi everyone there won't be any classes today
2022-11-02 14:07:48.037

Test
2022-11-02 14:0

Figure 4.7 Class Chat

FUNCTION	DESCRIPTION
G1	Navigate to class index page.
G2	Display sender’s message.
G3	Display recipient’s message
G4	Scroll to the latest chat message.
G5	Input chat message.
G6	Send chat message

4.8 Resource List



Resources

1 Sb6Aa56F_400x400.jpg

2 Verification Letter.pdf

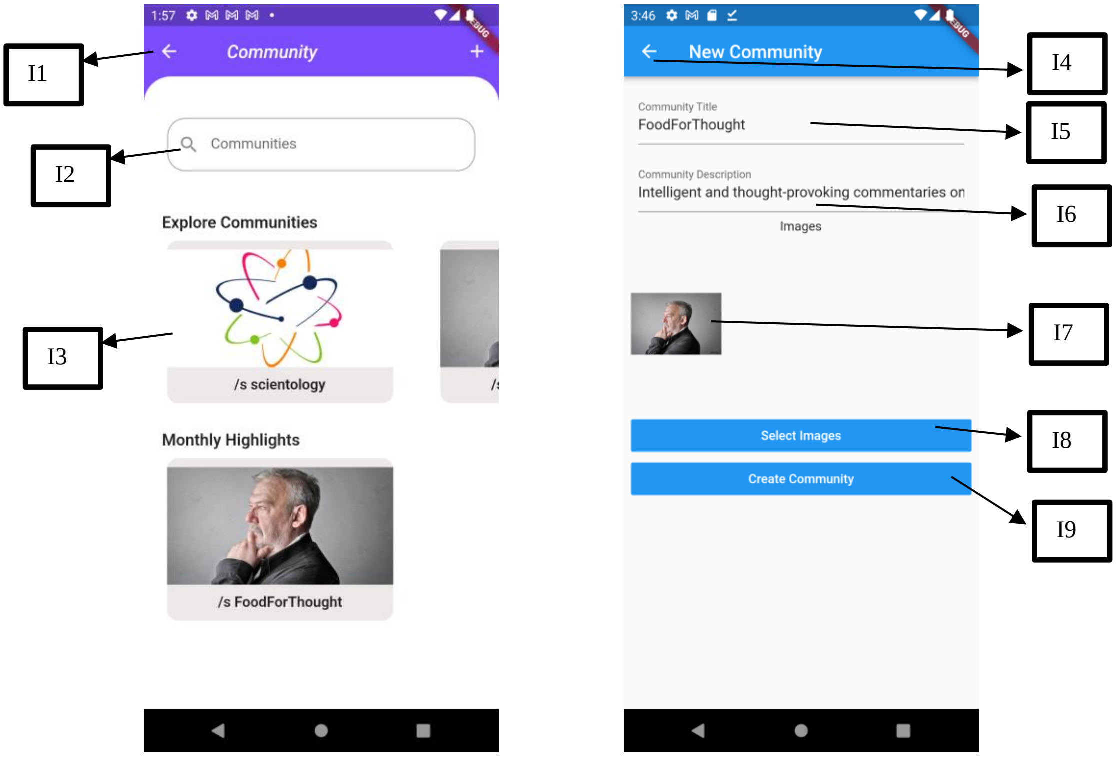
3 download.png

4 itenary final-3.pdf

Figure 4.8 Resource List

FUNCTION	**DESCRIPTION**
H1	Navigate back to class index page.
H2	Download resource
H3	Upload resource

4.9 Community Index, Community Creation



Community Title
FoodForThought

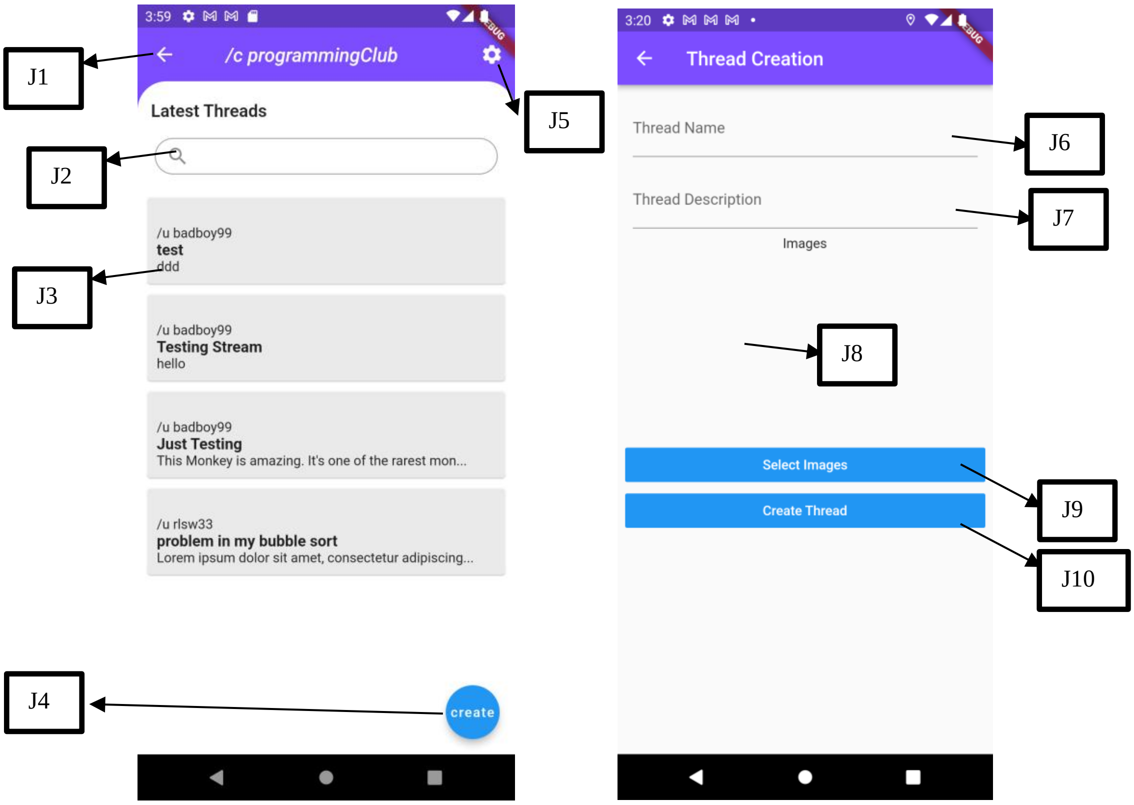
Community Description
Intelligent and thought-provoking commentaries on

Figure 4.9 Community Index, Community Creation

FUNCTION	DESCRIPTION
I1	Navigate back to home page.
I2	Search for communities
I3	Navigate to community thread list
I4	Navigate back to community index page.
I5	Input new community title.
I6	Input new community description.

I7	Uploaded community image
I8	Select image from file browser.
I9	Submit new community creation.

4.10 Community Thread, Thread Creation



/u badboy99
test
ddd

/u badboy99
Testing Stream
hello

/u badboy99
Just Testing
This Monkey is amazing. It's one of the rarest mon...

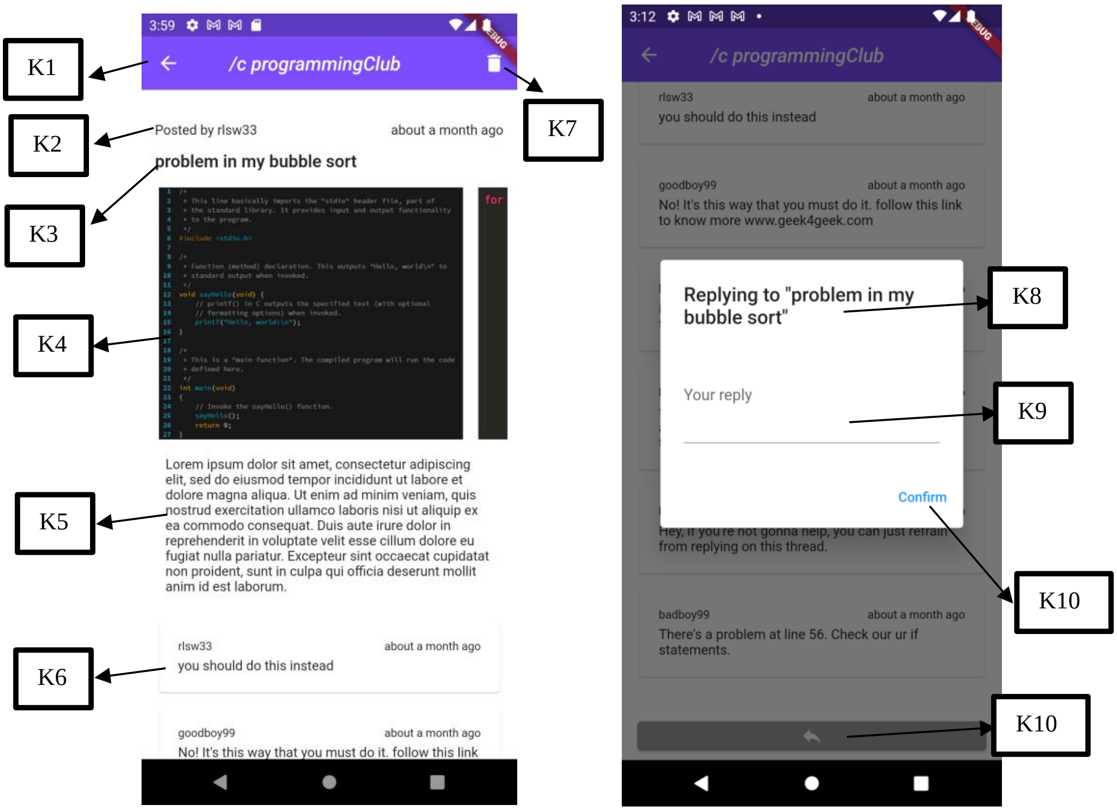
/u rlsw33
problem in my bubble sort
Lorem ipsum dolor sit amet, consectetur adipiscing...

Figure 4.10 Community Thread, Thread Creation

FUNCTION	**DESCRIPTION**
J1	Navigate back to community index page.
J2	Search for threads within the community.

J3	Navigate to selected thread details.
J4	Navigate to thread creation page.
J5	Navigate to community settings menu.
J6	Input new thread name.
J7	Input new thread description
J8	Display uploaded images.
J9	Uploaded images from system file browser.
J10	Submit new community creation.

4.11 Thread Details, Thread Reply



Posted by rlsw33 about a month ago

problem in my bubble sort

```
/*
 * This line basically imports the "stdio" header file, part of
 * the standard library. It provides input and output functionality
 * to the program.
 */
#include <stdio.h>

/*
 * Function (method) declaration. This outputs "Hello, world\n" to
 * standard output when invoked.
 */
void sayHello(void) {
    // printf() in C outputs the specified text (with optional
    // formatting options) when invoked.
    printf("Hello, world!\n");
}

/*
 * This is a "main function". The compiled program will run the code
 * defined here.
 */
int main(void)
{
    // Invoke the sayHello() function.
    sayHello();
    return 0;
}
```

```
for
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

rlsw33 about a month ago
you should do this instead

goodboy99 about a month ago
No! It's this way that you must do it. follow this link to know more www.geek4geek.com

Replying to "problem in my bubble sort"

Your reply

Hey, if you're not gonna help, you can just refrain from replying on this thread.

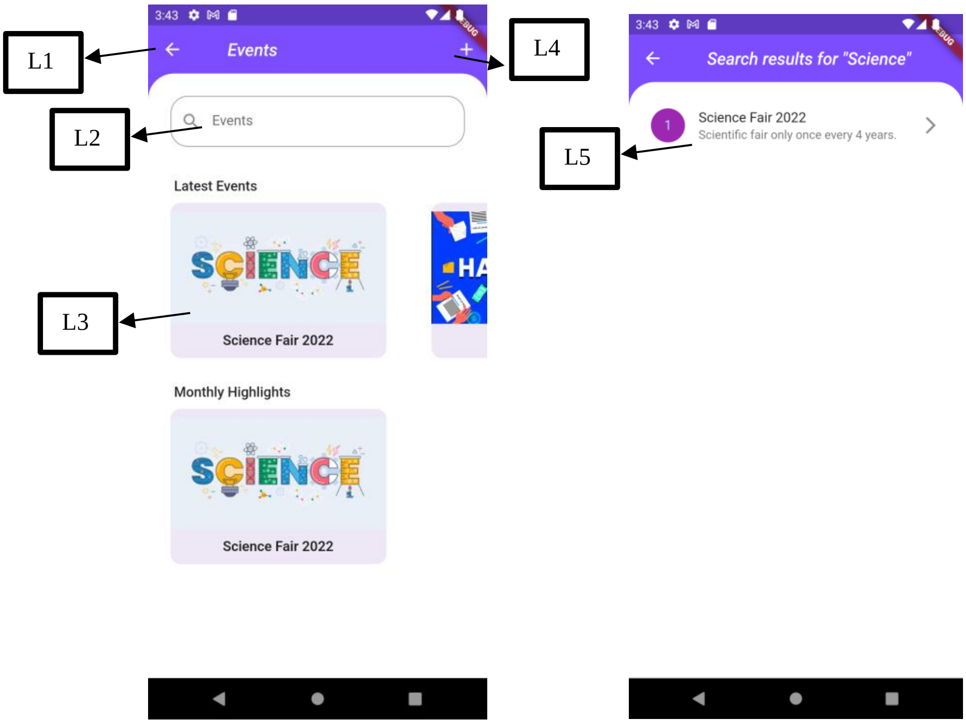
badboy99 about a month ago
There's a problem at line 56. Check our ur if statements.

Figure 4.11 Thread Details, Thread Reply

FUNCTION	DESCRIPTION
K1	Navigate back to thread list.
K2	Display original poster’s name.
K3	Display thread name
K4	Display thread photos.
K5	Display thread description.
K6	Display thread replies.

K7	Delete current thread.
K8	Display current thread that the user is replying to.
K9	Input reply message to thread.
K10	Prompt thread reply pop up modal box.

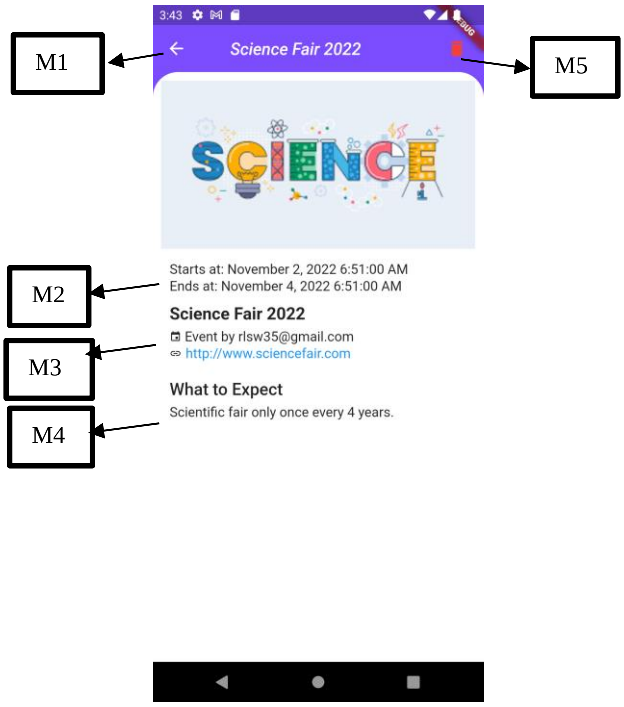
4.12 Event Index, Event Search



4.12 Event Index, Event Search

FUNCTION	DESCRIPTION
L1	Navigate back to home page.
L2	Search for specific events.
L3	Navigate to event details page.
L4	Navigate to add new event page.
L5	Display search results based on entered keywords.

4.13 Event Details



Science Fair 2022

Starts at: November 2, 2022 6:51:00 AM
Ends at: November 4, 2022 6:51:00 AM

Science Fair 2022

Event by rlsw35@gmail.com
http://www.sciencefair.com

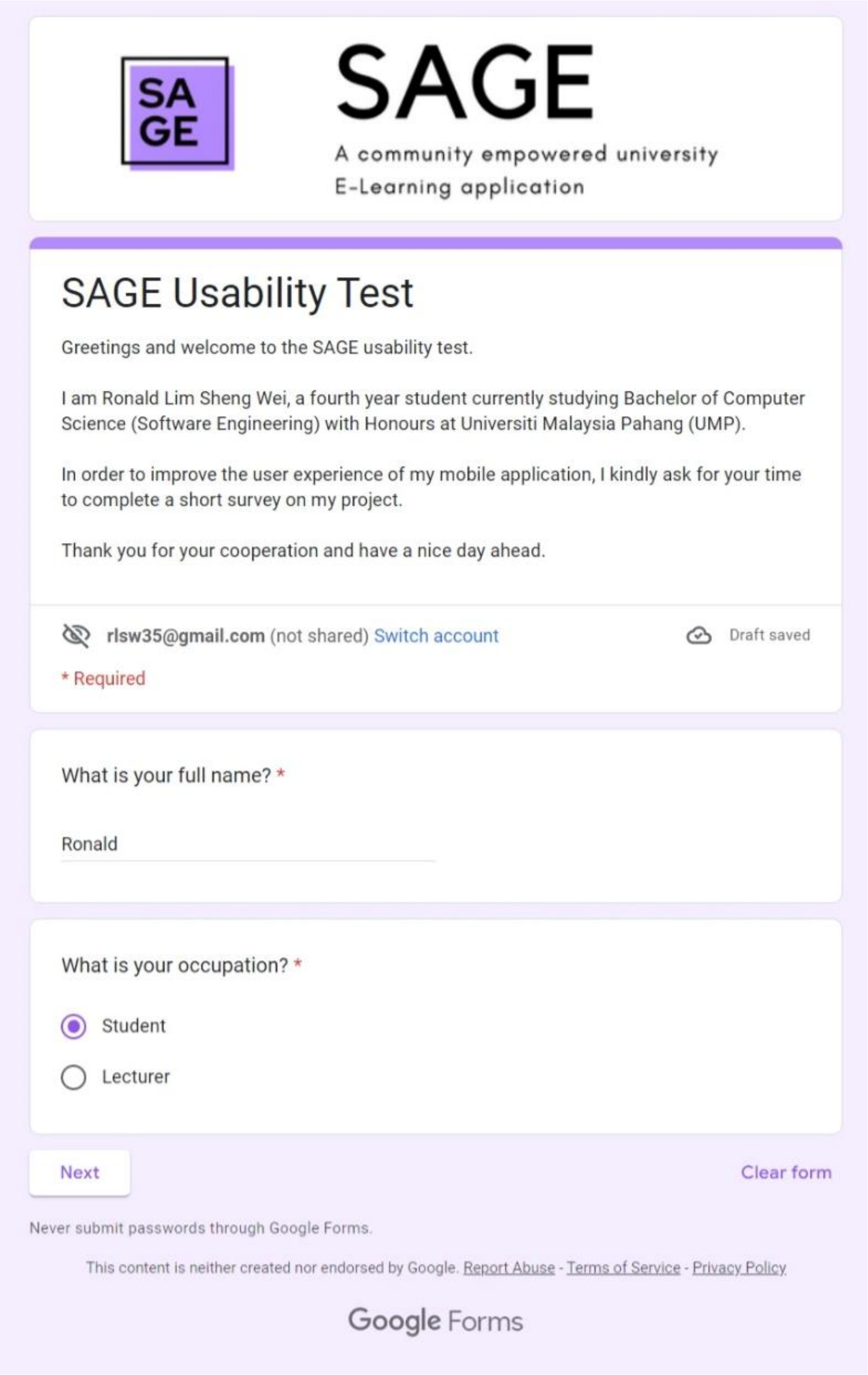
What to Expect

Scientific fair only once every 4 years.

4.13 Event Details

FUNCTION	**DESCRIPTION**
M1	Navigate back to event page.
M2	Display start and end date of event.
M3	Display original’s poster and website link.
M4	Display event description.

APPENDIX B
USABILITY TEST FORM



SAGE Usability Test

Greetings and welcome to the SAGE usability test.

I am Ronald Lim Sheng Wei, a fourth year student currently studying Bachelor of Computer Science (Software Engineering) with Honours at Universiti Malaysia Pahang (UMP).

In order to improve the user experience of my mobile application, I kindly ask for your time to complete a short survey on my project.

Thank you for your cooperation and have a nice day ahead.

rlsw35@gmail.com (not shared) Switch account

Draft saved

* Required

What is your full name? *

Ronald

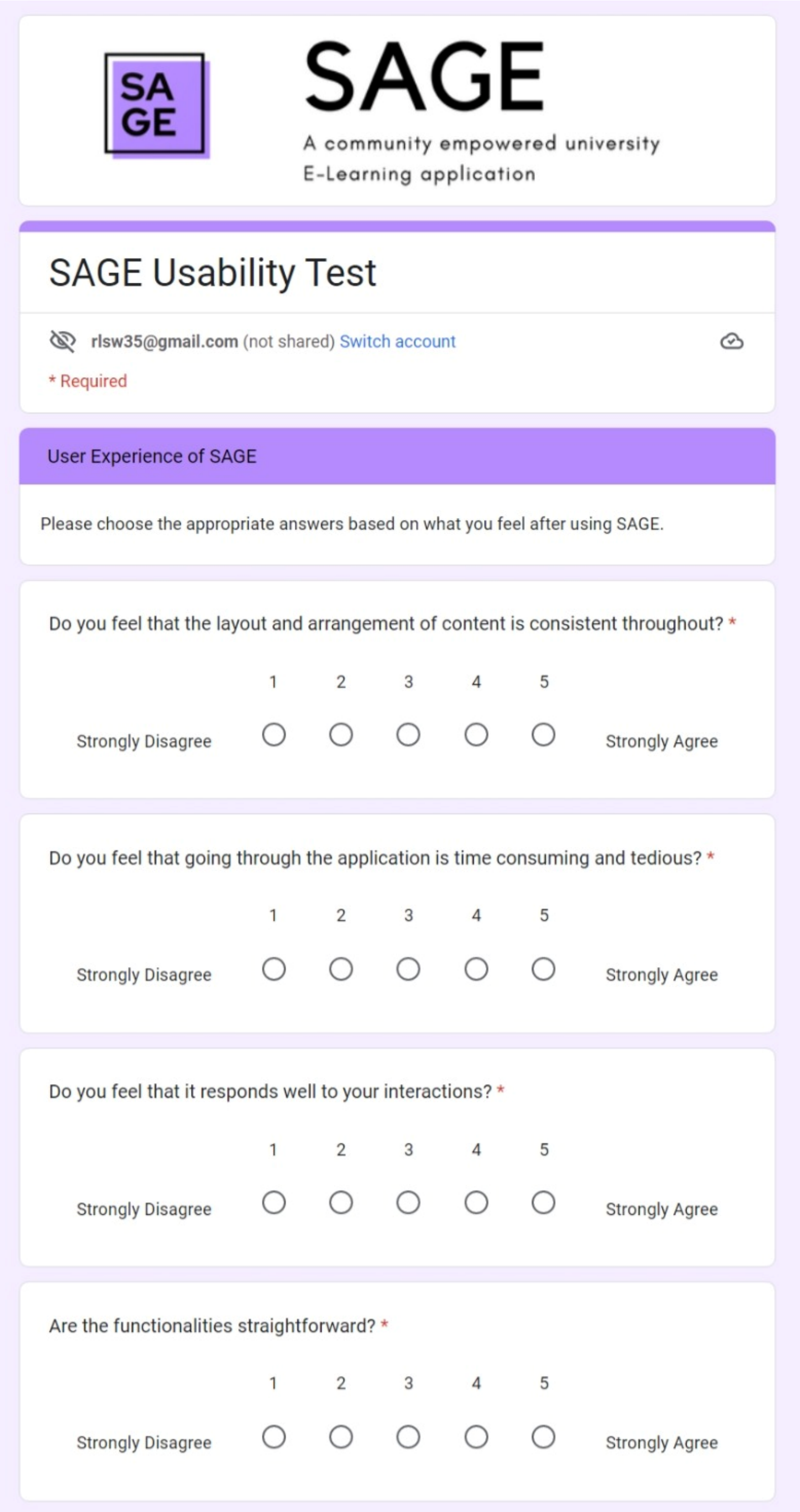
What is your occupation? *

● Student

○ Lecturer

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy



SA GE

SAGE

A community empowered university
E-Learning application

SAGE Usability Test

rlsw35@gmail.com (not shared) Switch account

* Required

User Experience of SAGE

Please choose the appropriate answers based on what you feel after using SAGE.

Do you feel that the layout and arrangement of content is consistent throughout? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Do you feel that going through the application is time consuming and tedious? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

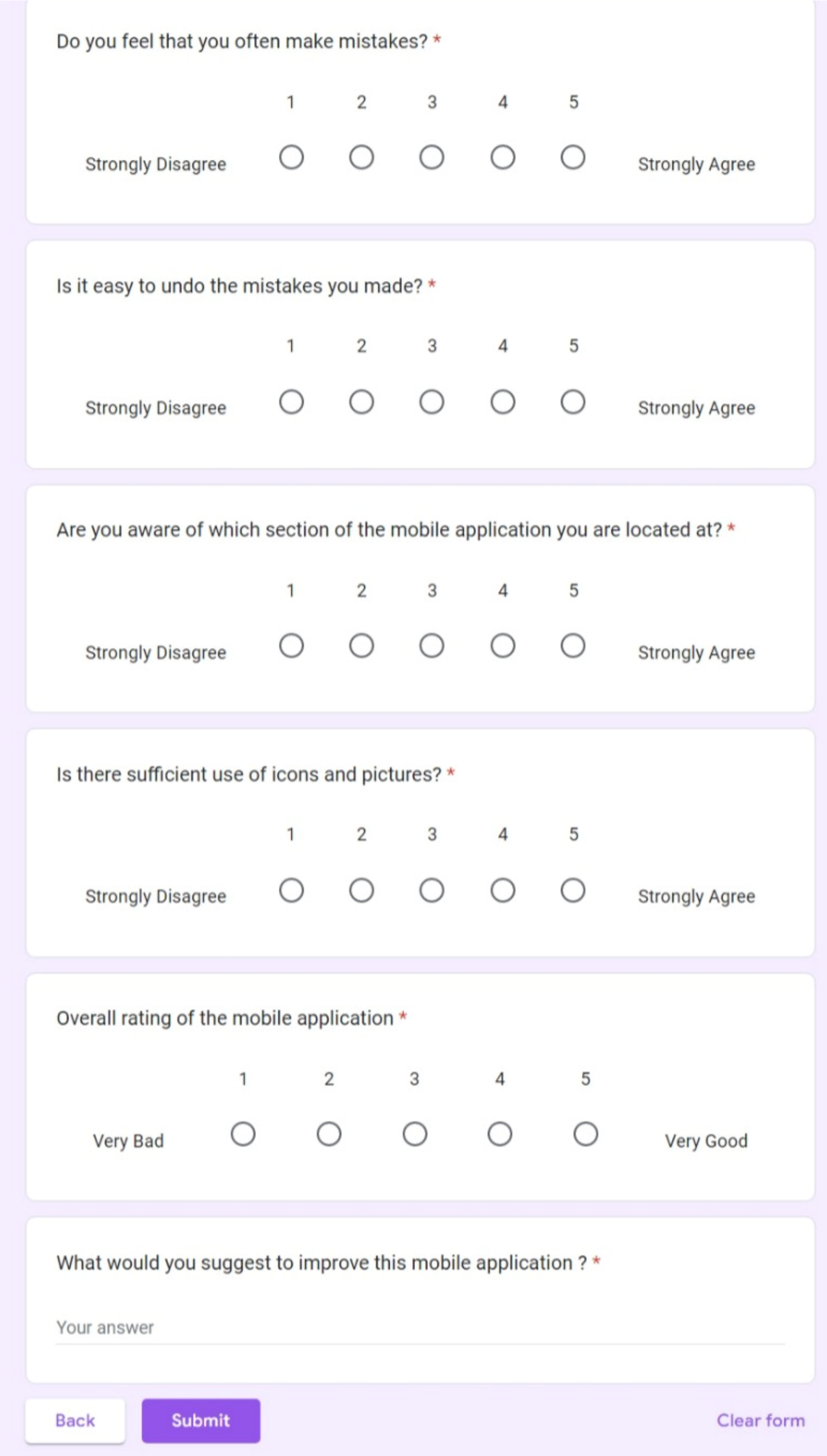
Do you feel that it responds well to your interactions? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Are the functionalities straightforward? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure 1: Usability Test Form



Do you feel that you often make mistakes? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Is it easy to undo the mistakes you made? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Are you aware of which section of the mobile application you are located at? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Is there sufficient use of icons and pictures? *

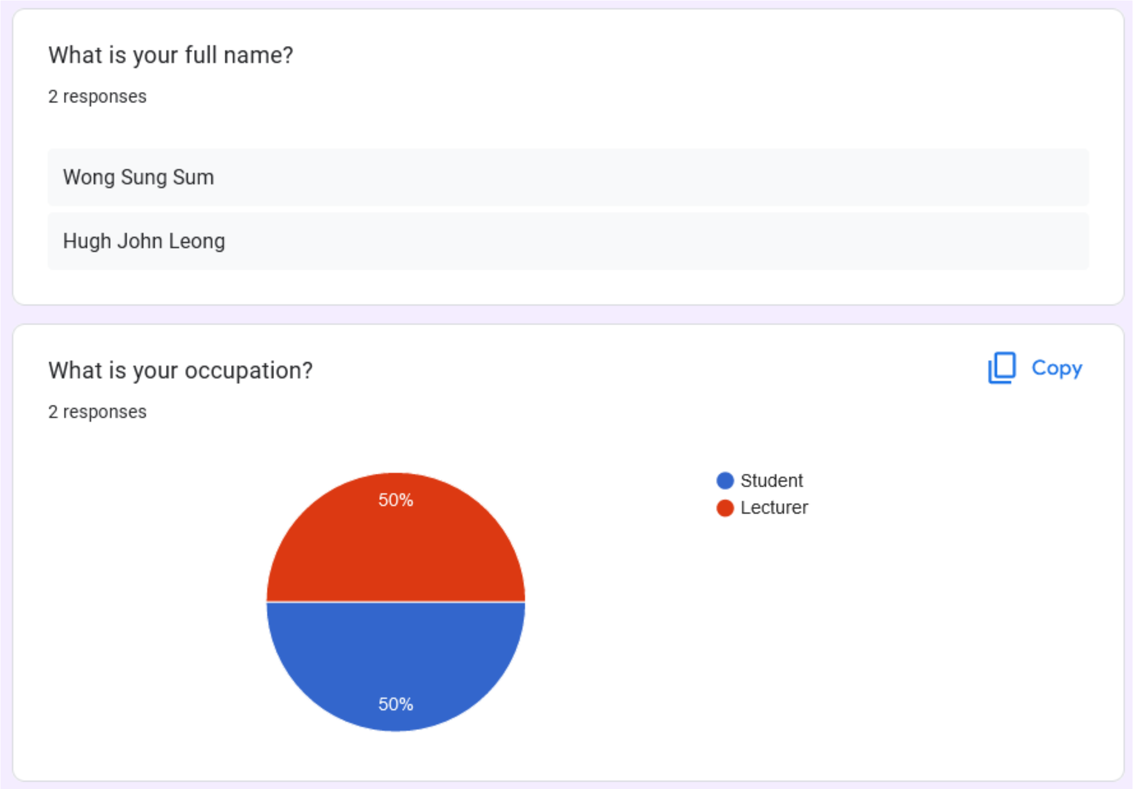
	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Overall rating of the mobile application *

	1	2	3	4	5	
Very Bad	○	○	○	○	○	Very Good

What would you suggest to improve this mobile application ? *

Your answer



What is your full name?

2 responses

Wong Sung Sum

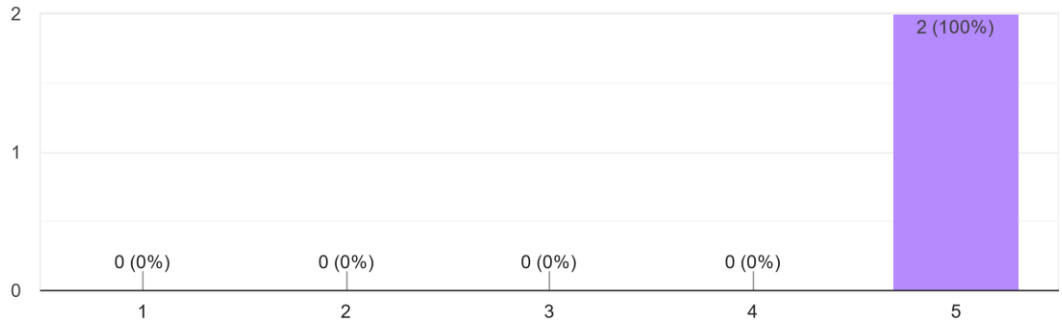
Hugh John Leong

What is your occupation?

2 responses

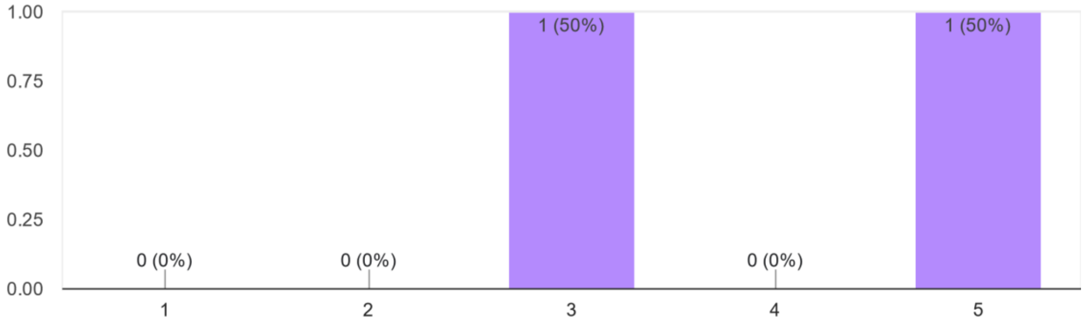
Do you feel that the layout and arrangement of content is consistent throughout?

2 responses



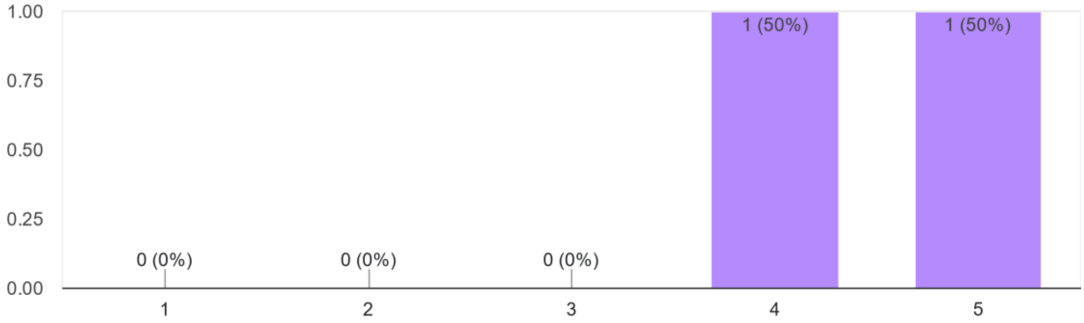
Do you feel that going through the application is time consuming and tedious?

2 responses



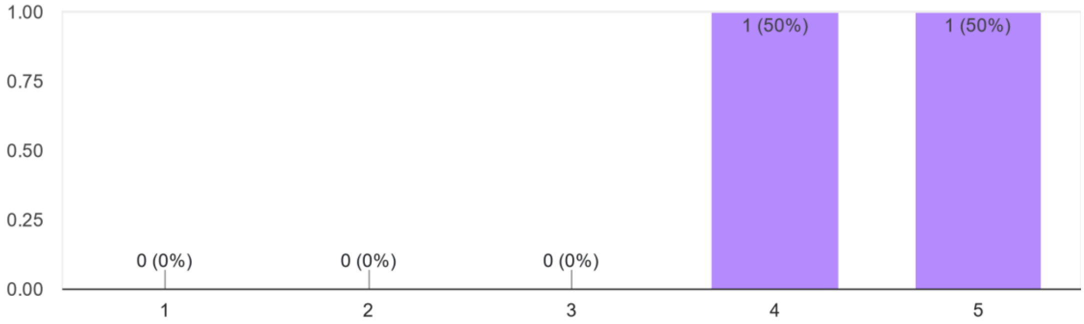
Do you feel that it responds well to your interactions?

2 responses



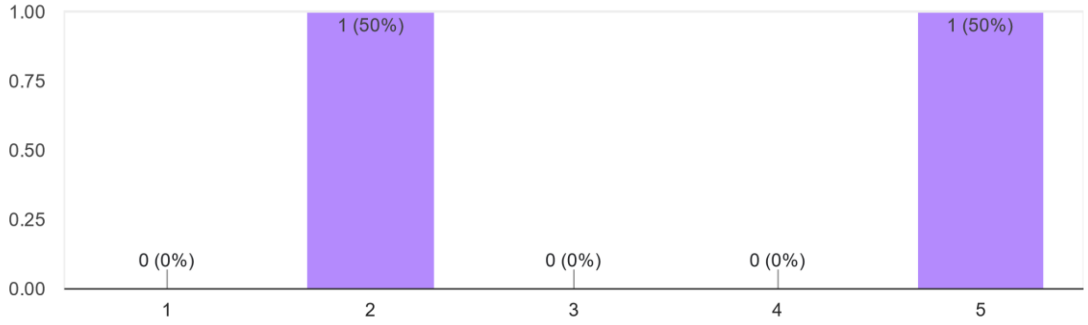
Are the functionalities straightforward?

2 responses



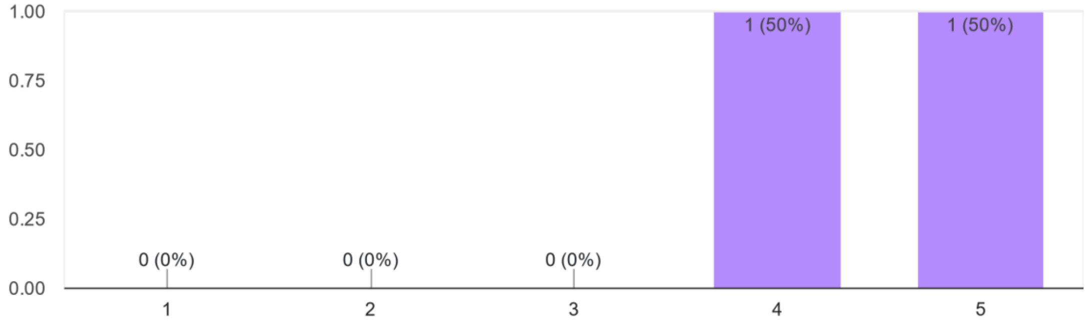
Do you feel that you often make mistakes?

2 responses



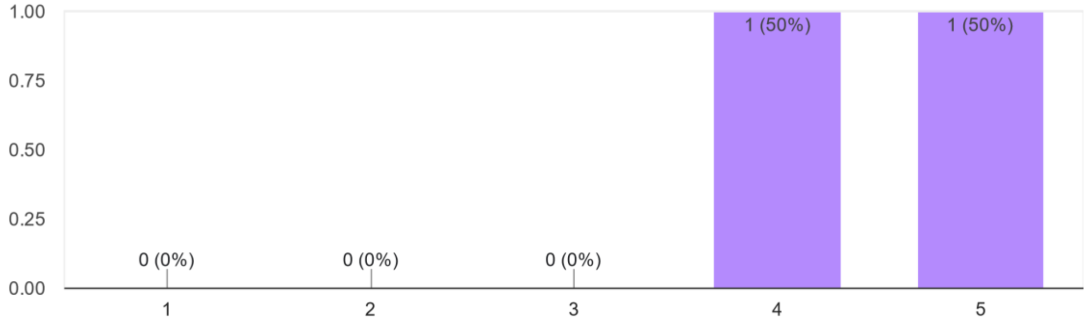
Is it easy to undo the mistakes you made?

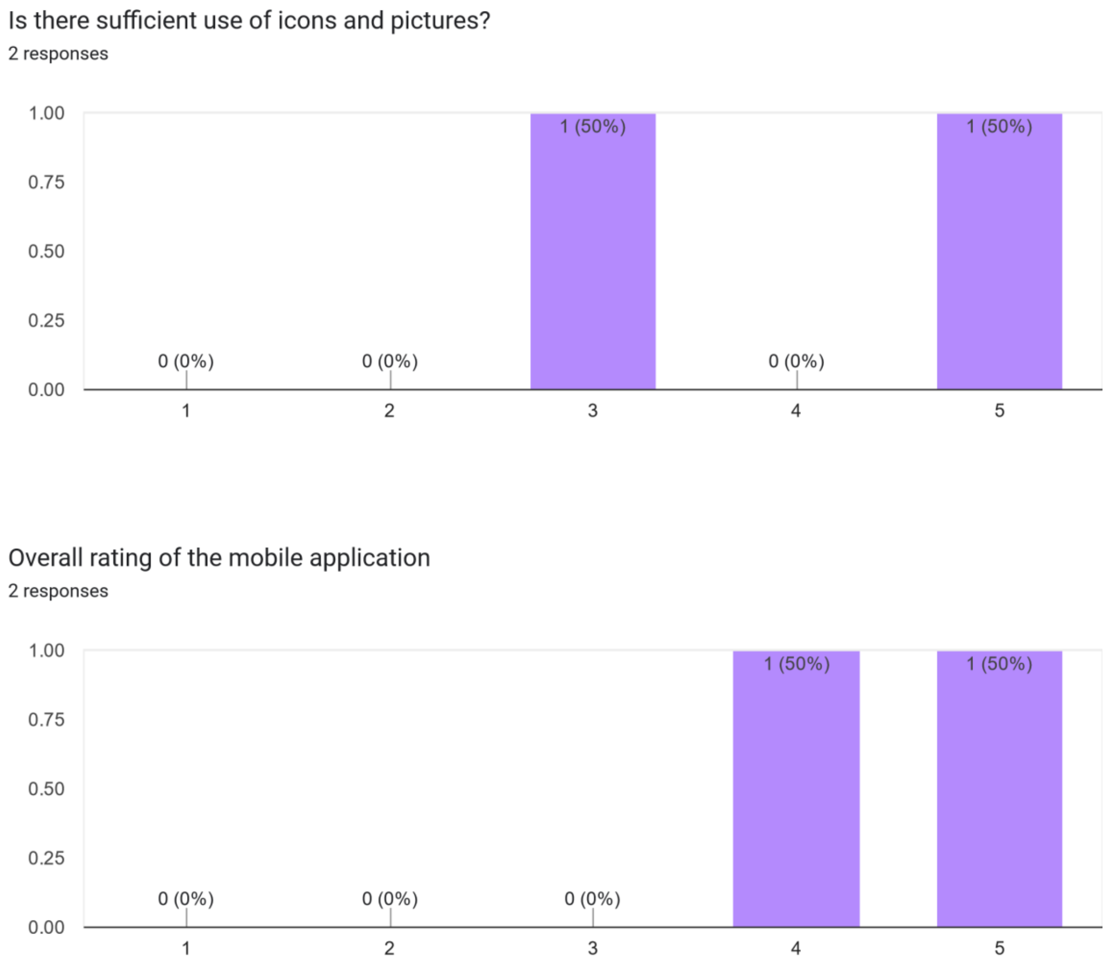
2 responses

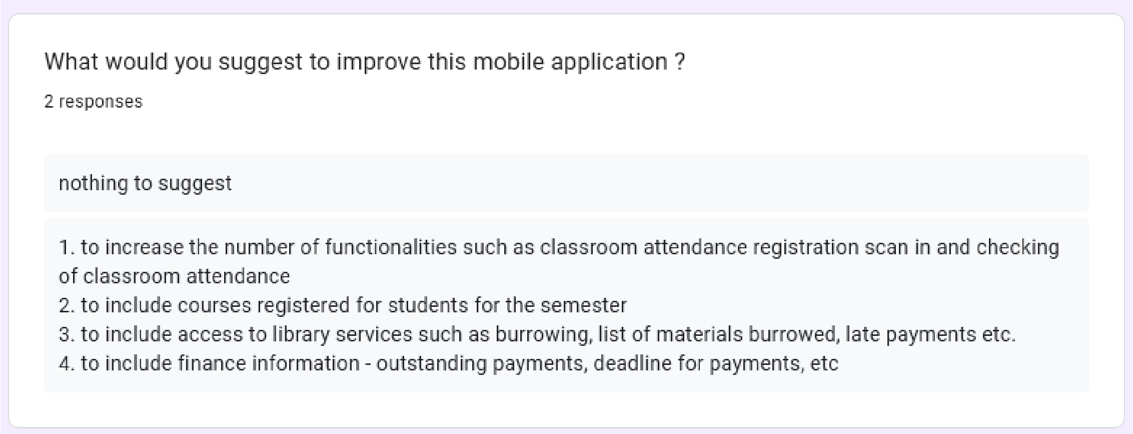


Are you aware of which section of the mobile application you are located at?

2 responses






What would you suggest to improve this mobile application ?

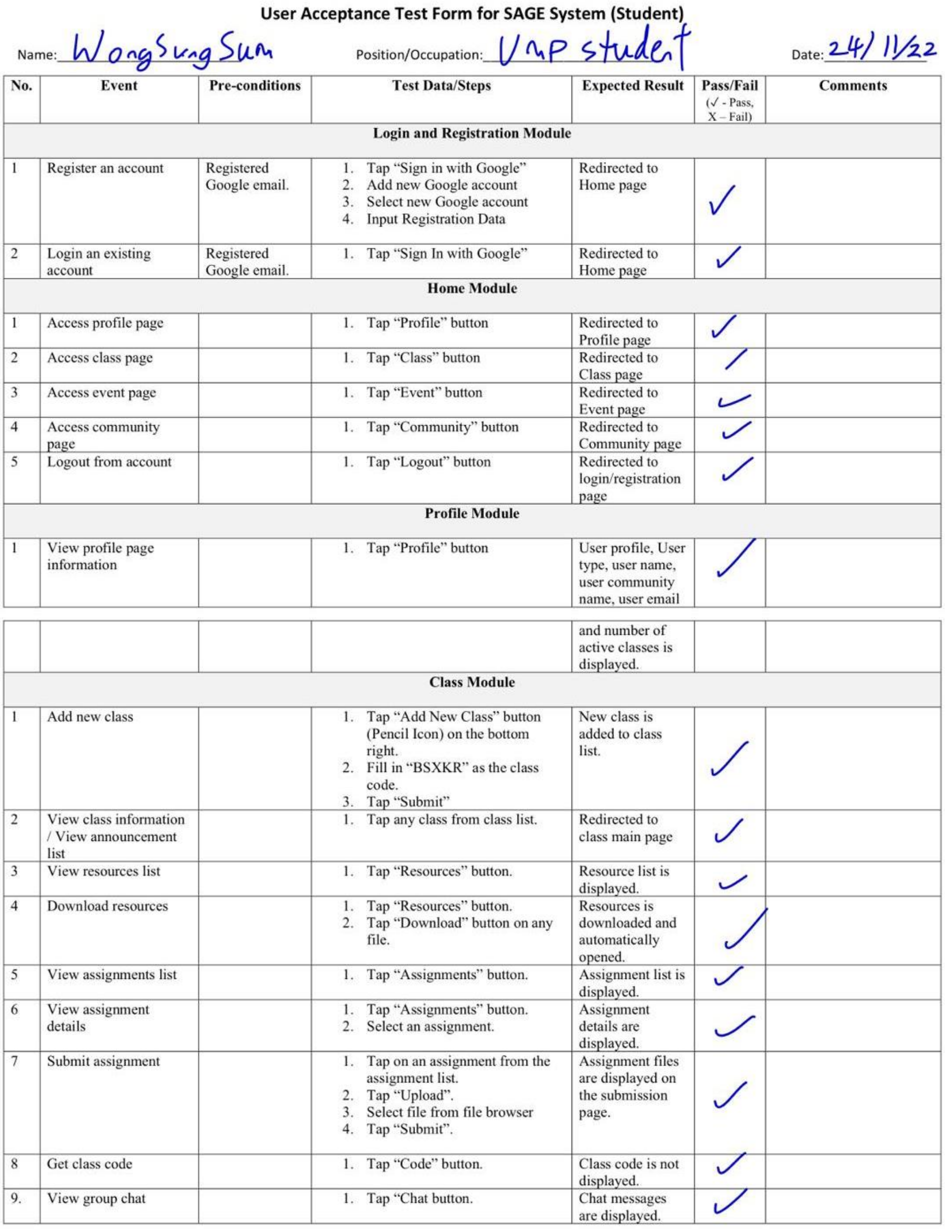
2 responses

nothing to suggest

1. to increase the number of functionalities such as classroom attendance registration scan in and checking of classroom attendance
2. to include courses registered for students for the semester
3. to include access to library services such as burrowing, list of materials burrowed, late payments etc.
4. to include finance information - outstanding payments, deadline for payments, etc

Figure 2: Usability Test Results

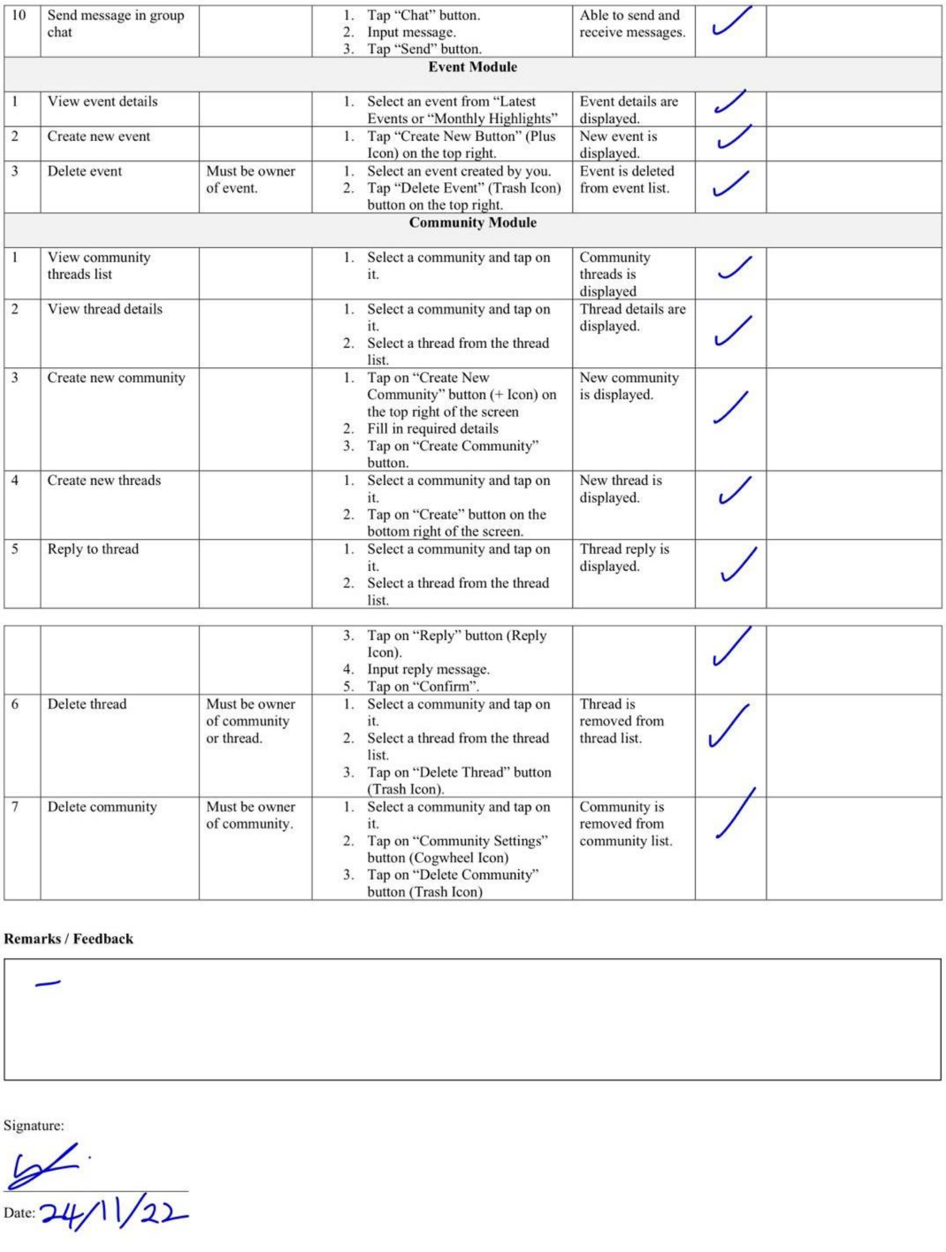
APPENDIX C
USER ACCEPTANCE TEST FORM



User Acceptance Test Form for SAGE System (Student)

Name: Wong Sung Sum Position/Occupation: Ump student Date: 24/11/22

No.	Event	Pre-conditions	Test Data/Steps	Expected Result	Pass/Fail (✓ - Pass, X – Fail)	Comments
Login and Registration Module						
1	Register an account	Registered Google email.	1. Tap "Sign in with Google" 2. Add new Google account 3. Select new Google account 4. Input Registration Data	Redirected to Home page	✓	
2	Login an existing account	Registered Google email.	1. Tap "Sign In with Google"	Redirected to Home page	✓	
Home Module						
1	Access profile page		1. Tap "Profile" button	Redirected to Profile page	✓	
2	Access class page		1. Tap "Class" button	Redirected to Class page	✓	
3	Access event page		1. Tap "Event" button	Redirected to Event page	✓	
4	Access community page		1. Tap "Community" button	Redirected to Community page	✓	
5	Logout from account		1. Tap "Logout" button	Redirected to login/registration page	✓	
Profile Module						
1	View profile page information		1. Tap "Profile" button	User profile, User type, user name, user community name, user email and number of active classes is displayed.	✓	
Class Module						
1	Add new class		1. Tap "Add New Class" button (Pencil Icon) on the bottom right. 2. Fill in "BSXKR" as the class code. 3. Tap "Submit"	New class is added to class list.	✓	
2	View class information / View announcement list		1. Tap any class from class list.	Redirected to class main page	✓	
3	View resources list		1. Tap "Resources" button.	Resource list is displayed.	✓	
4	Download resources		1. Tap "Resources" button. 2. Tap "Download" button on any file.	Resources is downloaded and automatically opened.	✓	
5	View assignments list		1. Tap "Assignments" button.	Assignment list is displayed.	✓	
6	View assignment details		1. Tap "Assignments" button. 2. Select an assignment.	Assignment details are displayed.	✓	
7	Submit assignment		1. Tap on an assignment from the assignment list. 2. Tap "Upload". 3. Select file from file browser 4. Tap "Submit".	Assignment files are displayed on the submission page.	✓	
8	Get class code		1. Tap "Code" button.	Class code is not displayed.	✓	
9.	View group chat		1. Tap "Chat button.	Chat messages are displayed.	✓	



10	Send message in group chat		1. Tap "Chat" button. 2. Input message. 3. Tap "Send" button.	Able to send and receive messages.	✓	
Event Module						
1	View event details		1. Select an event from "Latest Events or "Monthly Highlights"	Event details are displayed.	✓	
2	Create new event		1. Tap "Create New Button" (Plus Icon) on the top right.	New event is displayed.	✓	
3	Delete event	Must be owner of event.	1. Select an event created by you. 2. Tap "Delete Event" (Trash Icon) button on the top right.	Event is deleted from event list.	✓	
Community Module						
1	View community threads list		1. Select a community and tap on it.	Community threads is displayed	✓	
2	View thread details		1. Select a community and tap on it. 2. Select a thread from the thread list.	Thread details are displayed.	✓	
3	Create new community		1. Tap on "Create New Community" button (+ Icon) on the top right of the screen 2. Fill in required details 3. Tap on "Create Community" button.	New community is displayed.	✓	
4	Create new threads		1. Select a community and tap on it. 2. Tap on "Create" button on the bottom right of the screen.	New thread is displayed.	✓	
5	Reply to thread		1. Select a community and tap on it. 2. Select a thread from the thread list.	Thread reply is displayed.	✓	
			3. Tap on "Reply" button (Reply Icon). 4. Input reply message. 5. Tap on "Confirm".		✓	
6	Delete thread	Must be owner of community or thread.	1. Select a community and tap on it. 2. Select a thread from the thread list. 3. Tap on "Delete Thread" button (Trash Icon).	Thread is removed from thread list.	✓	
7	Delete community	Must be owner of community.	1. Select a community and tap on it. 2. Tap on "Community Settings" button (Cogwheel Icon) 3. Tap on "Delete Community" button (Trash Icon)	Community is removed from community list.	✓	

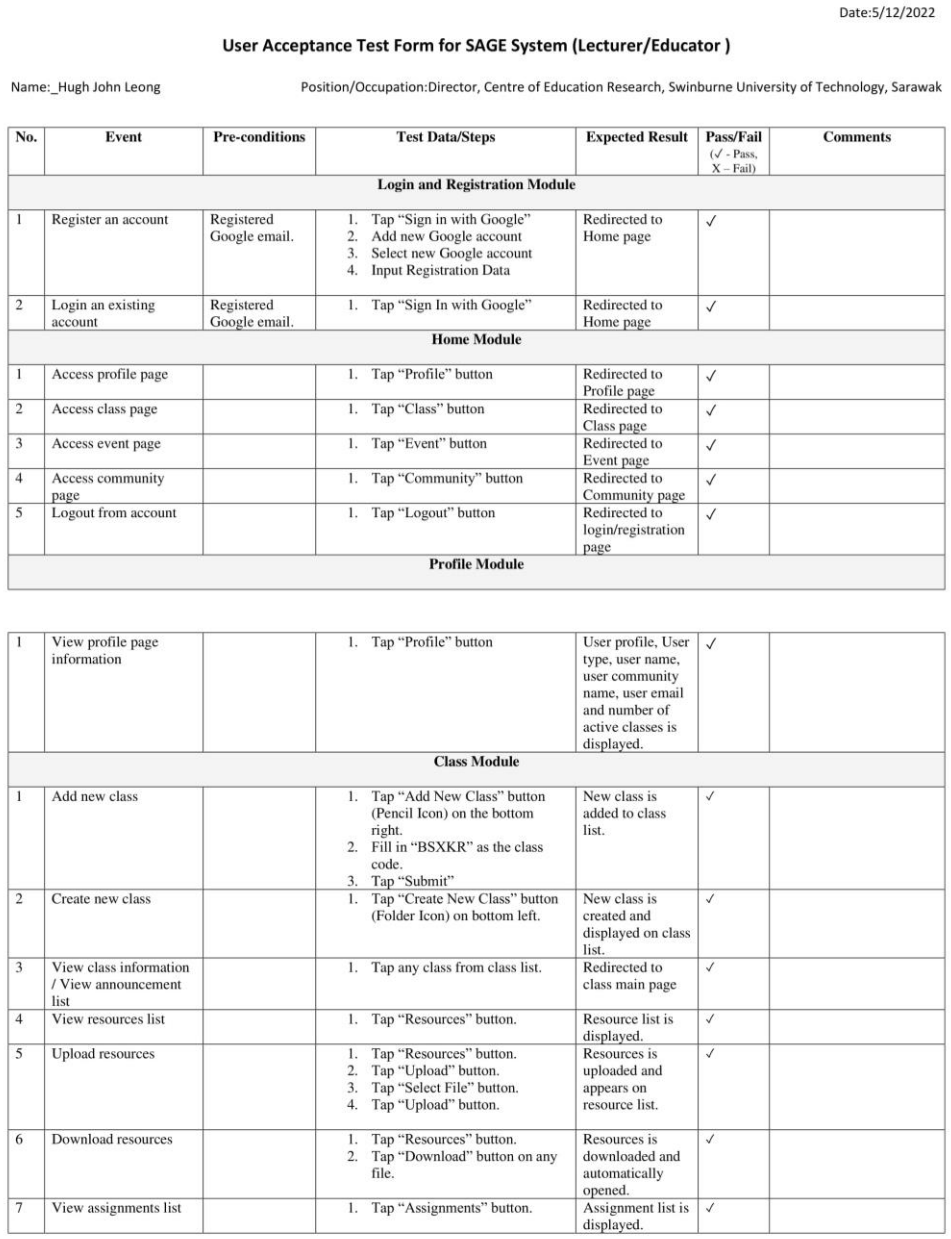
Remarks / Feedback

—

Signature:

Date: 24/11/22

Figure 1: User Acceptance Test Form



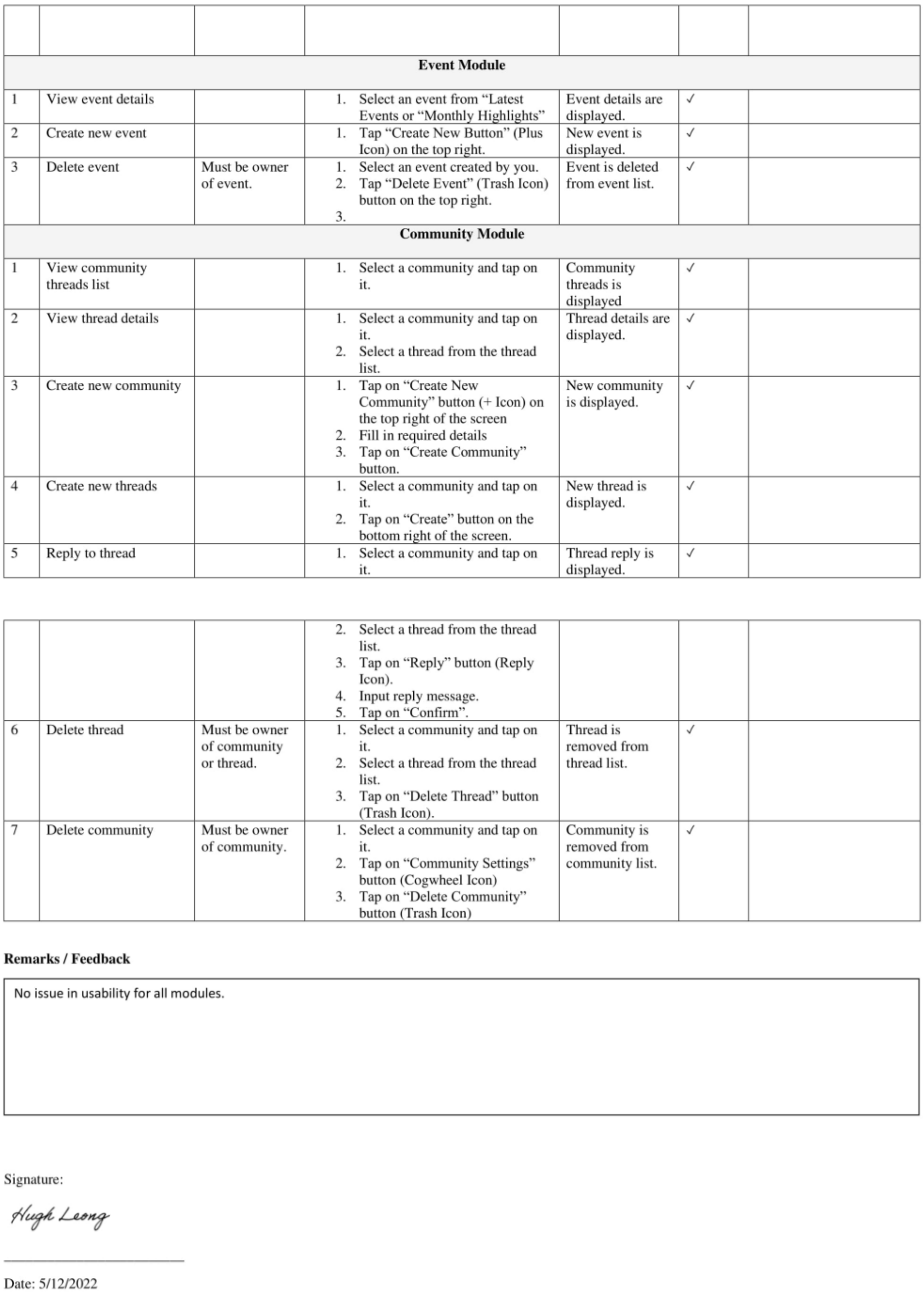
Date:5/12/2022

User Acceptance Test Form for SAGE System (Lecturer/Educator)

Name:_Hugh John Leong Position/Occupation:Director, Centre of Education Research, Swinburne University of Technology, Sarawak

No.	Event	Pre-conditions	Test Data/Steps	Expected Result	Pass/Fail (✓ - Pass, X – Fail)	Comments
Login and Registration Module						
1	Register an account	Registered Google email.	1. Tap "Sign in with Google" 2. Add new Google account 3. Select new Google account 4. Input Registration Data	Redirected to Home page	✓	
2	Login an existing account	Registered Google email.	1. Tap "Sign In with Google"	Redirected to Home page	✓	
Home Module						
1	Access profile page		1. Tap "Profile" button	Redirected to Profile page	✓	
2	Access class page		1. Tap "Class" button	Redirected to Class page	✓	
3	Access event page		1. Tap "Event" button	Redirected to Event page	✓	
4	Access community page		1. Tap "Community" button	Redirected to Community page	✓	
5	Logout from account		1. Tap "Logout" button	Redirected to login/registration page	✓	
Profile Module						
1	View profile page information		1. Tap "Profile" button	User profile, User type, user name, user community name, user email and number of active classes is displayed.	✓	
Class Module						
1	Add new class		1. Tap "Add New Class" button (Pencil Icon) on the bottom right. 2. Fill in "BSXKR" as the class code. 3. Tap "Submit"	New class is added to class list.	✓	
2	Create new class		1. Tap "Create New Class" button (Folder Icon) on bottom left.	New class is created and displayed on class list.	✓	
3	View class information / View announcement list		1. Tap any class from class list.	Redirected to class main page	✓	
4	View resources list		1. Tap "Resources" button.	Resource list is displayed.	✓	
5	Upload resources		1. Tap "Resources" button. 2. Tap "Upload" button. 3. Tap "Select File" button. 4. Tap "Upload" button.	Resources is uploaded and appears on resource list.	✓	
6	Download resources		1. Tap "Resources" button. 2. Tap "Download" button on any file.	Resources is downloaded and automatically opened.	✓	
7	View assignments list		1. Tap "Assignments" button.	Assignment list is displayed.	✓	

8	View assignment details		1. Tap “Assignments” button. 2. Select an assignment.	Assignment details are displayed.	✓	
9	Create new assignment		1. Tap “Assignments” button. 2. Tap “Create New Assignment” (Pencil Icon) on the bottom right. 3. Fill in required information 4. Tap “Submit”.	New assignment is created and displayed on assignment list.	✓	
10	View and grade submissions.		1. Tap “Assignments” button. 2. Select an assignment. 3. Select a submission 4. Tap “Grade” 5. Input marks 6. Tap “Confirm”	Assignment grades are displayed in student submission details.	✓	
11	Get class code		1. Tap “Code” button.	Class code is displayed.	✓	
12.	View group chat		1. Tap “Chat button.	Chat messages are displayed.	✓	
13	Send message in group chat		1. Tap “Chat” button. 2. Input message. 3. Tap “Send” button.	Able to send and receive messages.	✓	
14	View class settings		1. Tap “Settings” button.	Settings is displayed.	✓	
15	Delete class		1. Tap “Settings” button. 2. Tap “Delete Class” button.	Class is deleted from class list.	✓	
16	Create new announcement		1. Select class from class list. 2. Tap “Create New Announcement” button from bottom right of screen.	New announcement is displayed on class index.	✓	
17	Delete announcement		1. Tap “Delete Announcement” (Trash Icon) button.	Announcement is deleted from announcement list.	✓	



Event Module						
1	View event details		1. Select an event from "Latest Events or "Monthly Highlights"	Event details are displayed.	✓	
2	Create new event		1. Tap "Create New Button" (Plus Icon) on the top right.	New event is displayed.	✓	
3	Delete event	Must be owner of event.	1. Select an event created by you. 2. Tap "Delete Event" (Trash Icon) button on the top right. 3.	Event is deleted from event list.	✓	
Community Module						
1	View community threads list		1. Select a community and tap on it.	Community threads is displayed	✓	
2	View thread details		1. Select a community and tap on it. 2. Select a thread from the thread list.	Thread details are displayed.	✓	
3	Create new community		1. Tap on "Create New Community" button (+ Icon) on the top right of the screen 2. Fill in required details 3. Tap on "Create Community" button.	New community is displayed.	✓	
4	Create new threads		1. Select a community and tap on it. 2. Tap on "Create" button on the bottom right of the screen.	New thread is displayed.	✓	
5	Reply to thread		1. Select a community and tap on it. 2. Select a thread from the thread list. 3. Tap on "Reply" button (Reply Icon). 4. Input reply message. 5. Tap on "Confirm".	Thread reply is displayed.	✓	
6	Delete thread	Must be owner of community or thread.	1. Select a community and tap on it. 2. Select a thread from the thread list. 3. Tap on "Delete Thread" button (Trash Icon).	Thread is removed from thread list.	✓	
7	Delete community	Must be owner of community.	1. Select a community and tap on it. 2. Tap on "Community Settings" button (Cogwheel Icon) 3. Tap on "Delete Community" button (Trash Icon)	Community is removed from community list.	✓	

Remarks / Feedback

No issue in usability for all modules.

Signature:

Hugh Leong

Date: 5/12/2022

Figure 2: User Acceptance Test Form

APPENDIX D
SOFTWARE REQUIREMENT SPECIFICATION (SRS)

2022

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

[SAGE: A COMMUNITY EMPOWERED UNIVERISITY E-LEARNING APPLICATION]

DOCUMENT APPROVAL

	Name	Date
Authenticated by: _________________________ Ronald Lim Sheng Wei	Ronald Lim Sheng Wei	27/5/2022
Approved by: _________________________ Client		

Software :

Archiving Place :

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

SAGE aims to provide tertiary level students and lecturers alike, an application for peer-to peer communication and academic discussion as well as a tool to manage their classes. The SAGE application includes four main modules. The systems are as follows:

a. Manage Class

The Manage Class allows students to join class via a generated class code. From within a class, they can retrieve uploaded educational resources, chat with peers from the same class, view announcements by lecturers and submit their assignments. For lecturers, they can create classes, assignments as well as announcement. Assignments can be graded by the lecturers and returned to the students. They can participate in the chat room of the class as well.

b. Manage Events

The Manage Events module allows the user to create, delete, edit their hosted events. Users wanting to explore the campus for more events to attend can do so on the module as well.

c. Manage Profile

Manage Profile module allows user to edit their profile picture as well as personal information.

d. Manage Community

Manage Community allows users to create, edit, delete as well as update their communities. From within the communities, users can create, edit or delete their threads. Users can reply to their threads with words and images.

1.1 SYSTEM IDENTIFICATION

System Title: Community Empowered University E-Learning Application

System Abbreviation: SAGE

System Identification Number: SRS-SAGE-V01-23

1.2 CONTEXT DIAGRAM

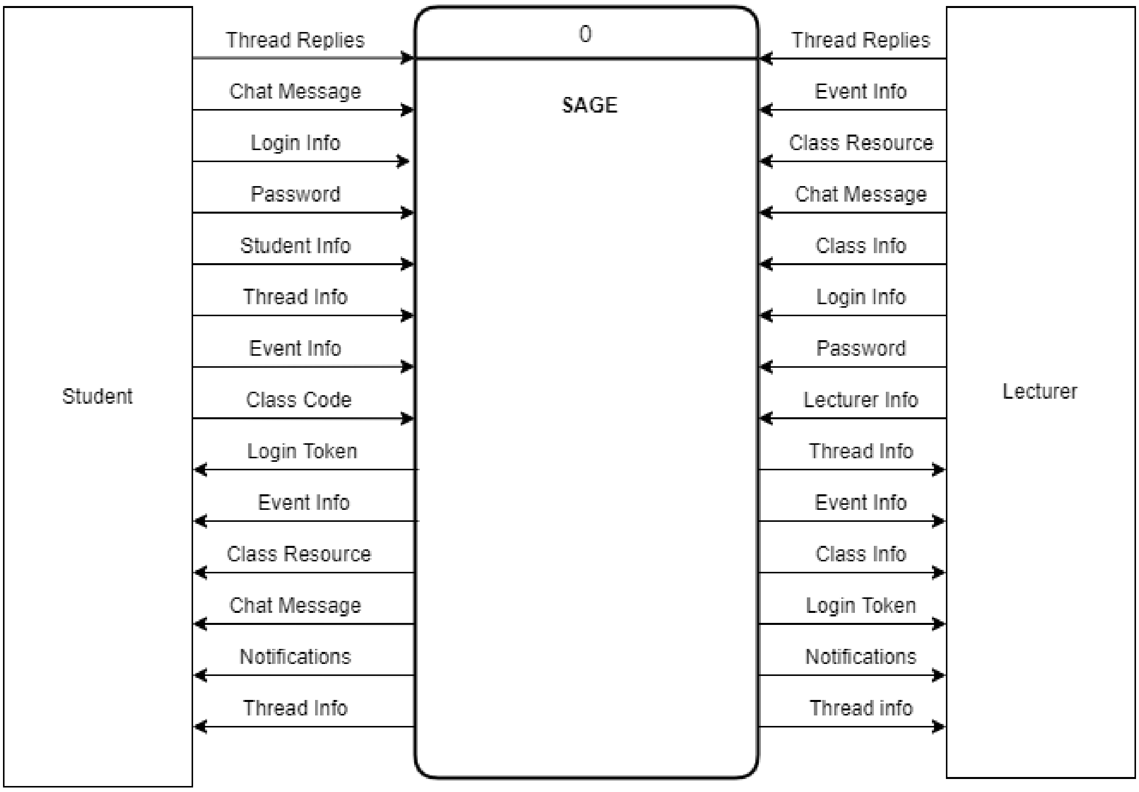


Figure 1.1 Context Diagram

The context diagram contains two entities which are the student and the lecturer who will interact with the system. From the two entities to the system, there will be data flowing from multiple modules from within the system. For the login function of the system, user credentials or login information will be sent to the system while a login token will be returned to the user. Next, for the Manage Class Module, there will be class information and class code and also class messages going into the system and out from the system to the students. Class code will be used to join new classes while class information for students includes the submission of class assignments. For the lecturers, class information indicates the upload of educational resources, creating announcements, as well as creating and marking of announcements. Besides that, for the Manage Event module, both entities will be required to provide event information to the system for the creation, editing, and deleting of events. The system will also provide users with the ability to browse events as seen as the event information flowing from the system to the entities. After that, for the Manage Community module, both entities will be expected to send and receive community information, thread info and also thread replies from the system. This allows the managing of communities, threads and also thread replies.

1.3 DATA FLOW DIAGRAM

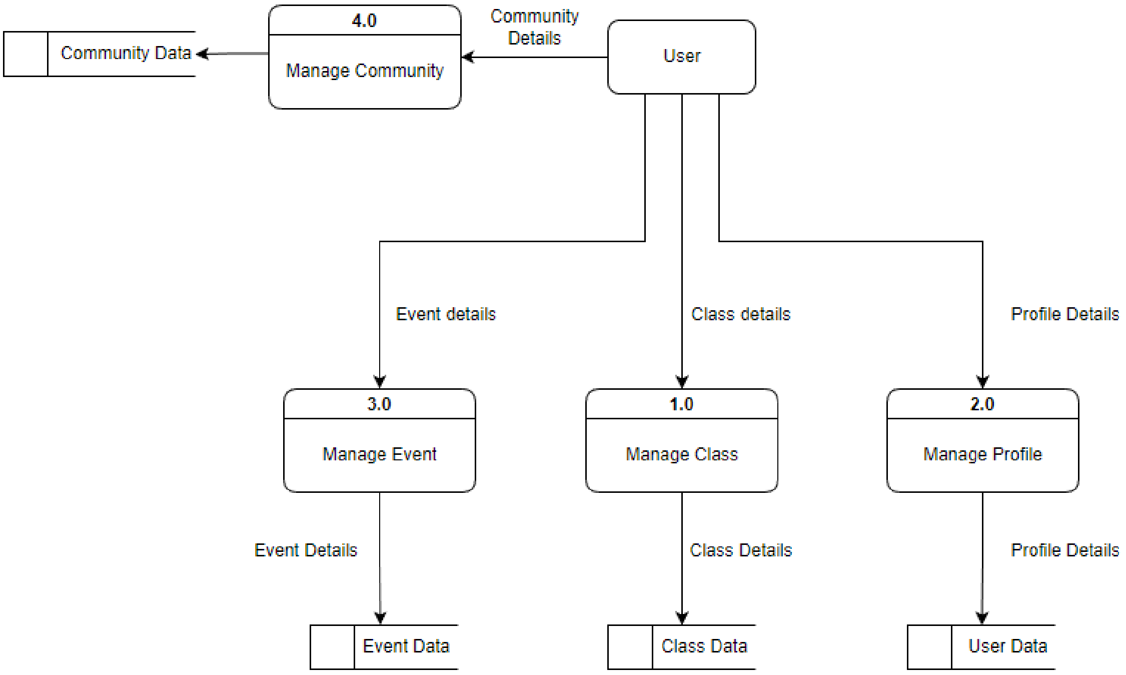


Figure 1.2 Data Flow Diagram

Figure 1.2 depicts the data flow diagram of the system. There are a total of 4 data stores, 4 processes, and one external entity. For the Manage Class process, class details data will flow into the process. The resulting class details data will flow into Class data store. Next, for the Manage Event process, it will receive event details from the user. Eventually, the output from the process will have data flowing into the Event data store. For the Manage Profile, it will receive profile details from the external entity, user and subsequently output profile details to be stored in the user data store. Finally, the Manage Community process receives community details and subsequently outputs data to be stored in the Community data store.

CHAPTER 2

1.1 USE CASE DIAGRAM AND DESCRIPTION

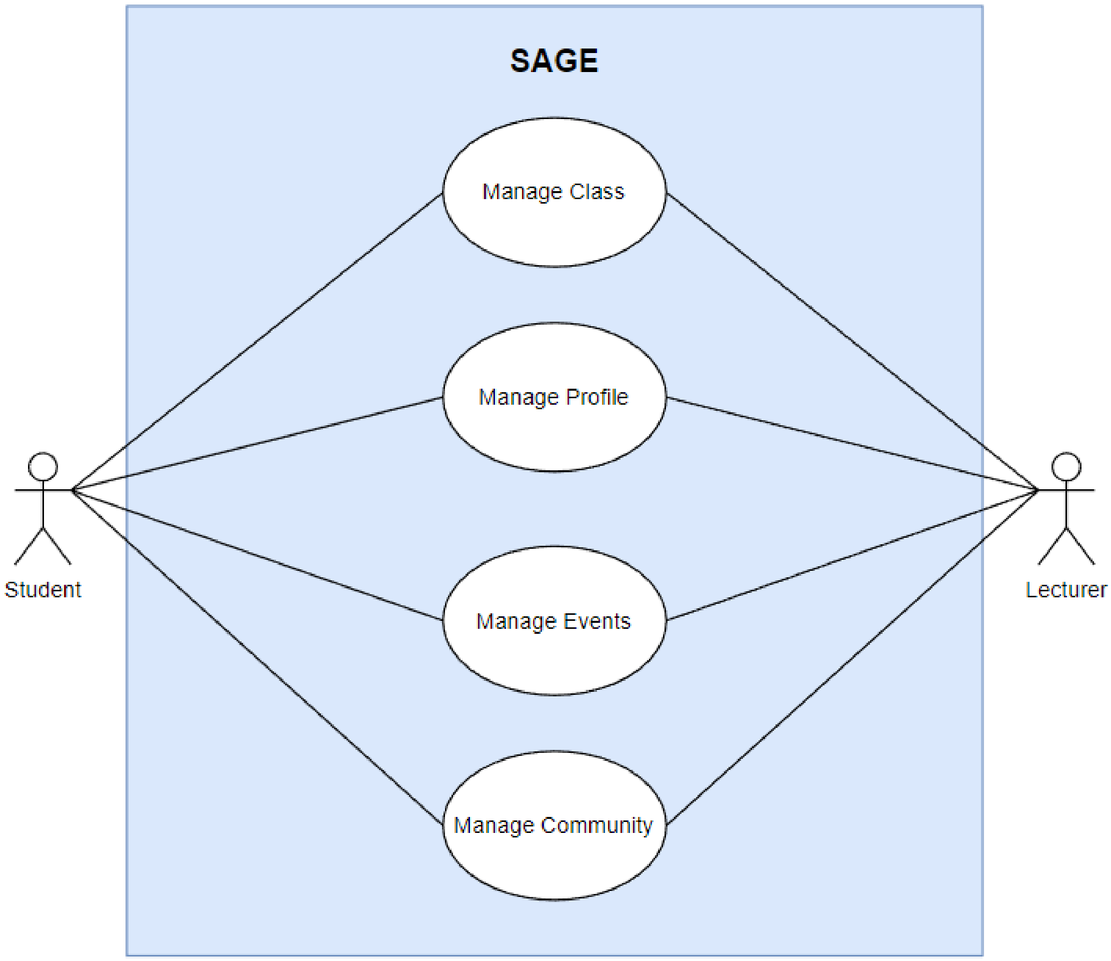


Figure 2.1 Use Case Diagram

Module	Function	Actor
Manage Class	• Lecturer can view, create, and delete class details. • Lecturer can view, create, and delete resource details.	• Lecturer • Student

	• Lecturer can view, create, and delete announcement details. • Lecturer can view, create, and delete assignment details. • Lecturer can view, and grade submitted assignments. • Students can join classes via class joining code. • Students can view and exit classes. • Students can view and download resources. • Students can view and upload assignments. • Students can view announcements. • Users can send and receive messages in the class chat room.	
Manage Profile	• Users can edit user profile details.	• Lecturer • Student
Manage Events	• Users can view, create, delete events. • Users can join events. • Users can remove event participants.	• Lecturer • Student
Manage Community	• Users can view, create, and delete communities.	• Lecturer

	- Users can view, create, and delete threads. - Users can view, create, delete thread replies.	- Student

2.1.1 Manage Class

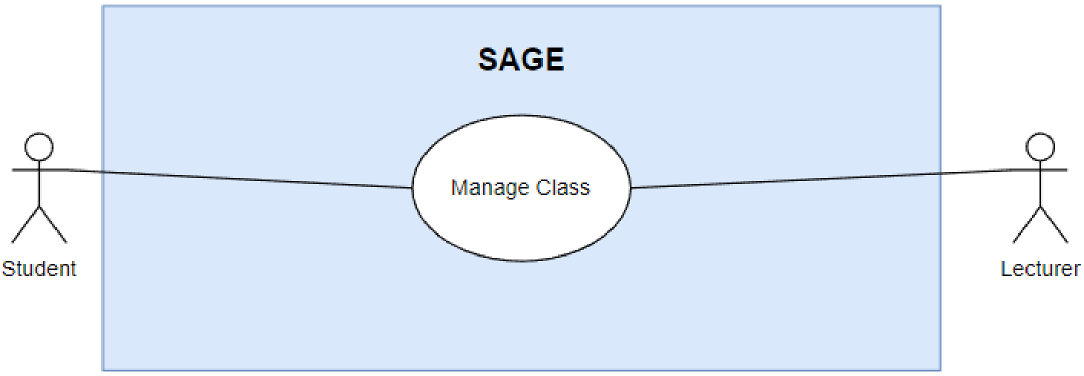


Figure 2.2 Manage Class Use Case Diagram

Table 2.1 Manage Class Use Case Description

Use Case ID	SAGE-SRS-UC001
Brief Description	This use case describes how the lecturer and students manage their classes.
Actor	Lecturer, Student
Pre-Conditions	1. User must be logged into the system. 2. User must have the correct authorizations. 3. User must be in the Manage Class module.
Basic Flow	**Lecturer:**

	[B1: Create Class]. 1. Lecturer presses the <<+>> button. 2. System display form for class creation. 3. Lecturer fills in the class details. [A1: Create Missing Required Information] 4. Lecturer presses <<Save>>. 5. System validates the data. **[A1: Create Class Missing Required Information]** 6. System inserts class information into the database. **[B2: Delete Class]** 1. Lecturer selects a class. 2. System retrieves class details from database. 3. Lecturer presses <<Settings>>. 4. System displays settings menu. 5. Lecturer presses <<Delete Class>>. **[B3: Create Announcement]** 1. Lecturer selects a class. 2. System retrieves class details from database. 3. Lecturer presses the announcement tab. 4. System retrieves announcement details from database. 5. Lecturer presses the <<+>> button. 6. System updates view. 7. Lecturer fills in the required details. 8. Lecturer presses <<Post>>. 9. System validates data. 10. System inserts data in database.

[B4: Delete Announcement]

1. Lecturer selects a class.
2. System retrieves class details from database.
3. Lecturer goes to the announcement tab.
4. System retrieves announcement details from database and displays it.
5. Lecturer presses the <<Delete>> button.
6. Lecturer fills in the required details.
7. Lecturer presses <<Post>>.
8. System validates data.
9. System deletes data from database.

[B5: Create Resources]

1. Lecturer selects a class.
2. System retrieves class details from database.
3. Lecturer navigates to the resources tab.
4. Lecturer presses the <<+>> button.
5. System displays add resource view.
6. Lecturer fills in the required details.
7. Lecturer uploads files. **[A2: Resource Exceeded File Size Limit]**
8. Lecturer presses <<Post>>.
9. System validates data.
10. System inserts data into database.

[B6: Delete Resource]

1. Lecturer selects a class.
2. System retrieves class details from database.

	3. Lecturer goes to the resources tab. 4. Lecturer selects a resource. 5. Lecturer presses <<Delete>>. **[B7: Create Assignments]** 1. Lecturer selects a class. 2. System retrieves class details from database. 3. Lecturer goes to the Assignments tab. 4. System retrieves assignment details from database. 5. Lecturer presses <<+>> button. 6. System displays add assignment view. 7. Lecturer fills in the details. 8. Lecturer uploads assignment files. 9. Lecturer presses <<Save>>. 10. System validates data. **[A3: Lecturer Assignment Exceeded File Size Limit]** 11. System inserts data into database. **[B8: Delete Assignments]** 1. Lecturer selects a class. 2. System retrieves class details from database. 3. Lecturer goes to the Assignments tab. 4. Lecturer selects an assignment. 5. Lecturer presses <<Delete>>. **[B9: Grade Assignments]** 1. Lecturer selects a class. 2. System retrieves class details from database.

3. Lecturer goes to the Assignments tab.
4. Lecturer selects an assignment.
5. Lecturer presses Grade.
6. System retrieves and display student list.
7. Lecturer select student.
8. Lecturer downloads uploaded files.
9. Lecturer fill in marks.
10. Lecturer presses <<Save>>.
11. System updates assignment info in database.

Students:

[B10: Join Class]

1. Student press <<+>> button.
2. System displays window to enter code.
3. Student fills in class code. [E1: Invalid Class Code]
4. System validates data.
5. System adds student to the class.

[B11: Upload Assignments]

1. Student selects a class.
2. System retrieves class information from database.
3. Student selects Assignments tab.
4. System retrieves assignments information from database.
5. Students select an assignment to view.

	6. System displays assignment information. 7. Student presses <<Upload Files>> 8. Student selects a file from their phone file manager. 9. Student presses <<Submit>>. 10. System validates data. [A4: Student Assignment Exceeded File Size Limit] 11. System inserts data into database. **Students and Lecturers:** **[B12: View Announcements]** 1. User selects a class. 2. System retrieves class information from database. 3. User selects announcement tab. 4. User selects an announcement to view. 5. System displays announcement information. **[B13: View Resources]** 1. User selects a class. 2. System retrieves class information from database. 3. User selects Resources tab. 4. System retrieves resource information from database. 5. User selects resource to view. 6. System displays resource information. 7. User presses uploaded file to download. 8. System retrieves file from database.

	[B14: View Assignments] 1. User selects a class. 2. System retrieves class information from database. 3. User selects Assignments tab. 4. System retrieves assignment information from database. 5. User selects assignment to view. 6. System displays assignment information. **[B15: Chat]** 1. User selects a class. 2. System retrieves class information from database. 3. User selects Chat tab. 4. System retrieves chat log from database. 5. User types a message. 6. User presses <<Send>> button. 7. System inserts chat log into the database. 8. System displays new chat log in interface.
Alternative Flow	**[A1: Create Class Missing Required Information]** 1. System displays error message. 2. Lecturer fills in required information. 3. Lecturer is returned to step 6 of B1. **[A2: Resource Exceeded File Size Limit]**

	1. System displays error message. 2. Lecturer reuploads file. 3. Lecturer is returned to step 9 of B5. **[A3: Lecturer Assignment Exceeded File Size Limit]** 1. System displays error message. 2. Student reuploads file. 3. Student presses <<Save>> 4. System validates data. 5. Student is returned to step 9 of B7. **[A4: Student Assignment Exceeded File Size Limit]** 6. System displays error message. 7. Student reuploads file. 8. Student is returned to step 11 of B11.
Exception Flow	**[E1: Invalid Class Code]** 1. System displays error message. 2. Student reenters correct class code. 3. Student is returned to step 5 of B10.
Post-Conditions	None
Rules	None
Constraints	None

2.1.2 Manage Event

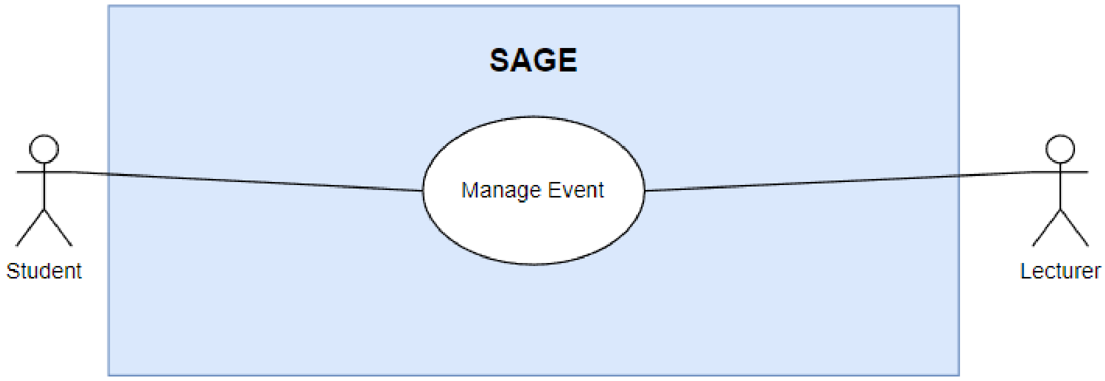


Figure 2.3 Manage Event Use Case Diagram

Table 2.2 Manage Event Use Case Description

Use Case ID	SAGE-SRS-UC002
Brief Description	This use case describes how the lecturers and students manage their events.
Actor	Lecturer, Student
Pre-Conditions	1. User must be logged into the system. 2. User must be in the Manage Event module.
Basic Flow	**[B1: Create Event]** 1. User presses <<+>> button. 2. System displays event creation form. 3. User fills in information. 4. User presses <<Create>>. 5. System validates data. **[A1: Create Missing Information]** 6. System inserts data into database. **[B2: Delete Event]:**

	1. User presses <<My Events>> button. 2. System retrieves user hosted events. 3. User presses <<Delete>> button. 4. System prompts for deletion confirmation. 5. User confirms deletion. 6. System deletes event from database. **[B3: View Participants]** 1. User presses <<My Events>> button. 2. System retrieves user hosted events. 3. User presses <<Participants>> button. 4. System retrieves participants info from database. 5. System displays event participants. **[B4: Search for Events]** 1. User types in keywords for events. 2. System queries database for keywords related to events. 3. System displays data.
Alternative Flow	**[A1: Create Event Missing Information]** 1. System displays error message. 2. User reenters correct information. 3. System validates data. 4. User is returned to step 6 of B1.
Exception Flow	None

Post-Conditions	None
Rules	None
Constraints	None

2.1.3 Manage Profile

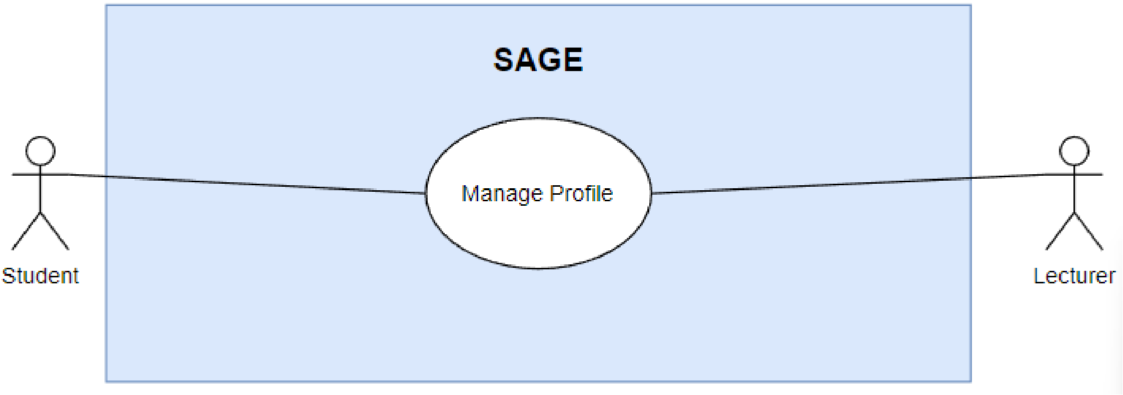


Figure 2.4 Manage Profile Use Case Diagram

Table 2.3 Manage Profile Use Case Description

Use Case ID	SAGE-SRS-UC003
Brief Description	This use case describes how the lecturers and students manage their profiles.
Actor	Lecturer, Student
Pre-Conditions	1. User must be logged into the system. 2. User must be in the Manage Profile module.
Basic Flow	**[B1: View Profile]**

	1. User presses <<Profile Image>> button. 2. System retrieves user information. 3. System displays user information. **[B2: Edit Profile]** 1. User presses <<Profile Image>> button. 2. System retrieves user information. 3. System displays user information. 4. User presses <<Edit Profile>> button. 5. System displays editing form. 6. User fills in information. 7. User presses <<Update>>. 8. System validates data. **[A1: Required Information Missing]** 9. System updates data in database.
Alternative Flow	**[A1: Required Information Missing]** 1. System displays error message. 2. User reenters correct information. 3. User presses <<Update>>. 4. System validates data. 5. User is returned to step 9 of B2.
Exception Flow	None
Post-Conditions	None
Rules	None

Constraints	None

2.1.4 Manage Community

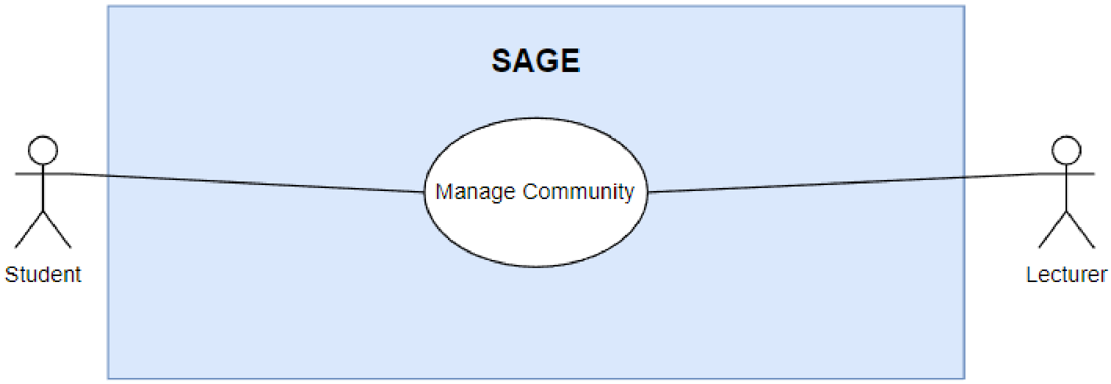


Figure 2.5 Manage Community Use Case Diagram

Table 2.4 Manage Community Use Case Description

Use Case ID	SAGE-SRS-UC004
Brief Description	This use case describes how the lecturers and students manage their communities.
Actor	Lecturer, Student
Pre-Conditions	1. User must be logged into the system. 2. User must be in the Manage Community module.
Basic Flow	**[B1: Create Community]** 1. User presses <<+>> button. 2. System displays creation form. 3. User fills in information.

	4. User presses <<Create>> button. 5. System validates data. **[A1: Create Community Missing Information]** 6. System inserts data into database. **[B2: Search Community]** 1. User clicks the search bar. 2. User types in keywords in search bar. 3. System queries database for written keywords. 4. System displays list of communities according to keywords. **[E1: No Communities Found]** **[B3: Create Thread]** 1. User selects a desired community. 2. System displays community page. 3. User presses <<+>> button. 4. System displays thread creation form. 5. User fills in required information. 6. User presses <<Create>> button. 7. System validates data. **[A2: Create Thread Missing Information]** 8. System inserts data into database. **[B4: Reply to Thread]** 1. User selects a desired community. 2. System displays community page. 3. User selects a desired thread. 4. System retrieves and display thread replies. 5. User selects <<Add Comment>> button.

	6. System displays comment form. 7. User fills in required information. 8. User presses <<Done>> button. 9. System validates data. **[A3: Create Reply Missing Information]** 10. System inserts data into database. **[B5: Search for Thread]** 1. User selects a desired community. 2. System displays community page. 3. User type keywords into search bar of community page. 4. System queries the database for thread data related to the keywords. 5. System displays results. **[E2: No Threads Found]**
Alternative Flow	**[A1: Create Community Missing Information]** 1. System displays error message. 2. User reenters correct information. 3. System validates data. 4. User is returned to step 6 of B1. **[A2: Create Thread Missing Information]** 1. System displays error message. 2. User reenters correct information. 3. System validates data.

	4. User is returned to step 8 of B2. **[A3: Create Reply Missing Information]** 1. System displays error message. 2. User reenters correct information. 3. System validates data. 4. User is returned to step 9 of B3.
Exception Flow	**[E1: No Communities Found]** 1. System displays error message. 2. User is returned to step 1 of B2. **[E2: No Threads Found]** 1. System displays error message. 2. User is returned to step 1 of B4.
Post-Conditions	None
Rules	None
Constraints	None

CHAPTER 3

3.1 INTERFACE DESIGN

3.1.1 Manage Class

3.1.1.1 Lecturer View

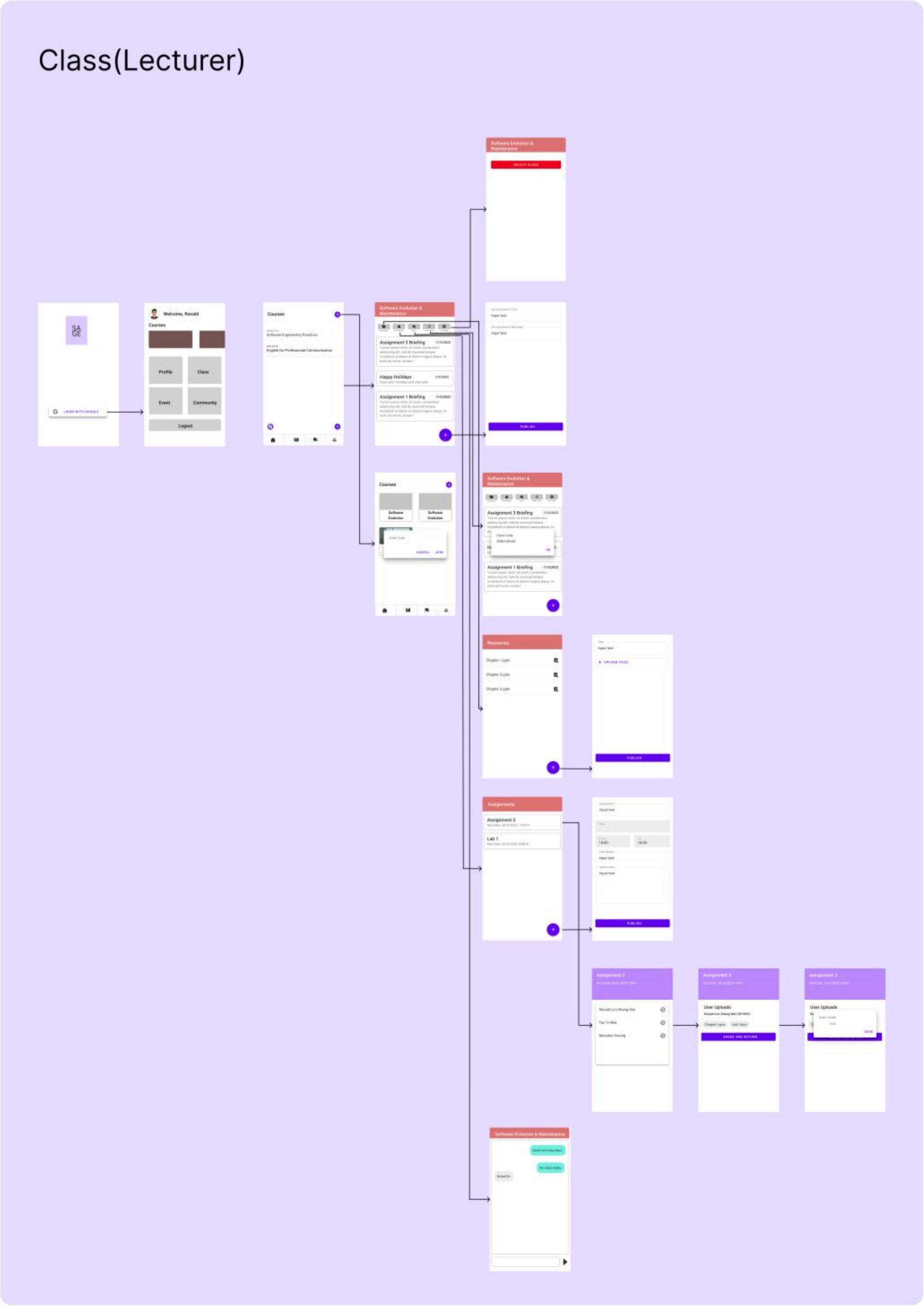


Figure 3.1 Lecturer’s View of Manage Class Module

Figure 3.1 depicts a lecturer’s view of the Manage Class. Lecturer will have full access to the administrative functions across the Manage Class module. Lecturers are able to create, edit, delete announcement, assignments, resources, participate in the chatroom, as well as manage the classroom.

3.1.1.2 *Student View*

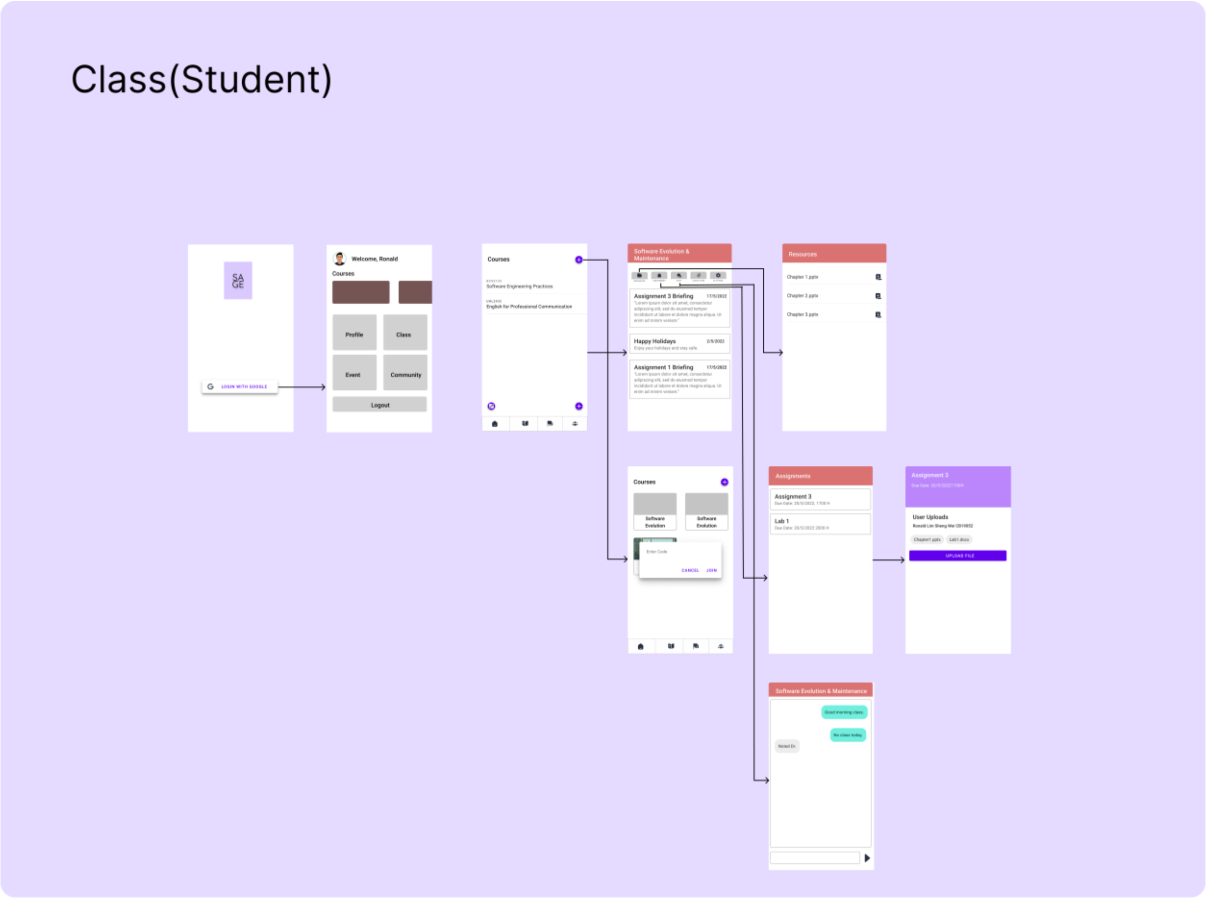
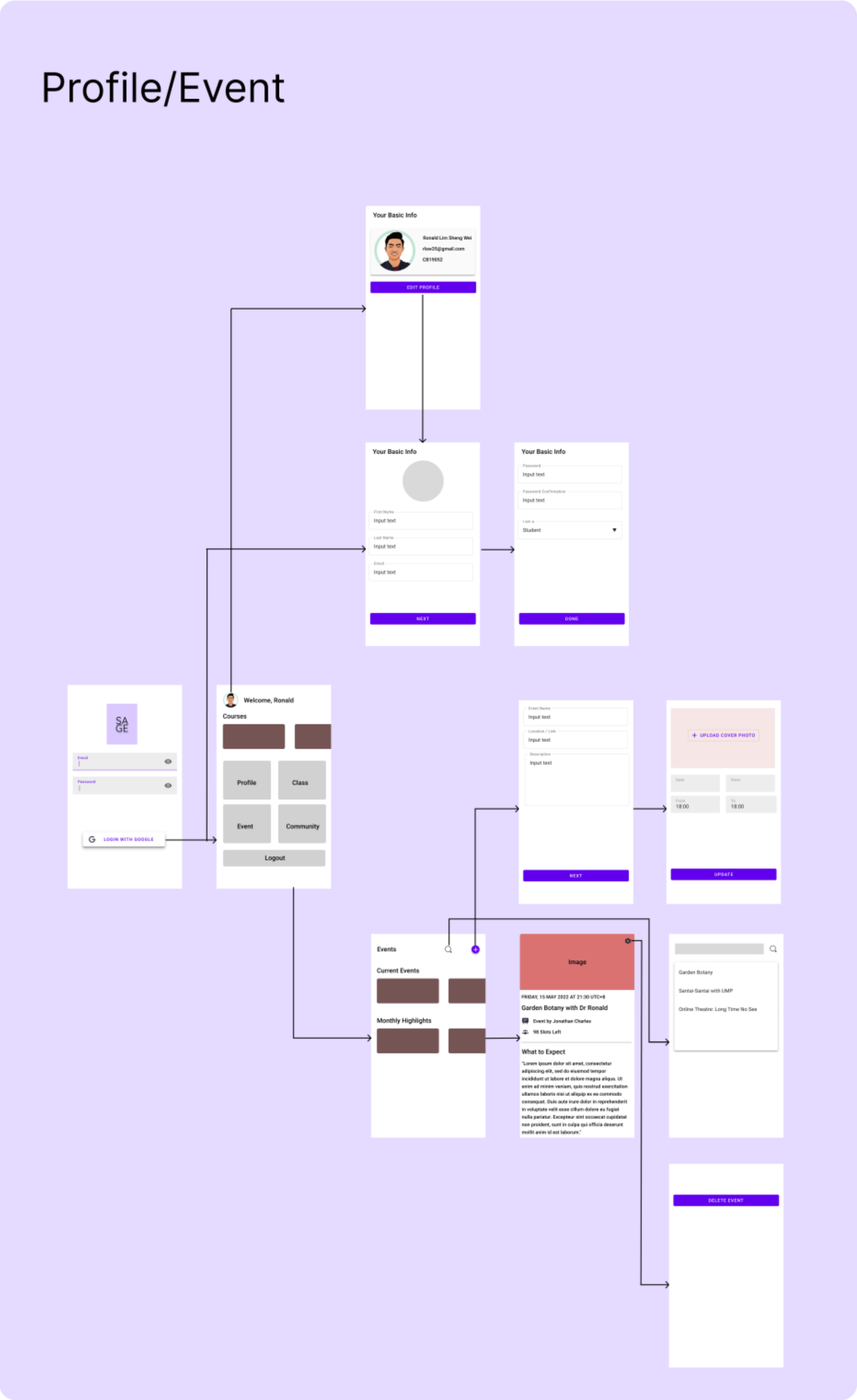


Figure 3.2 Manage Class – Student View

Figure 3.2 depicts a student's view of the Manage Class module. In this view, many of the administrative functions are stripped off as they are only limited to the lecturer or creator of the class. Meanwhile, users can still view announcements, chat with their classmates and educator in the group chat. There is also an interface to input the class code to join a class. The resources tab can be accessed to read the instructions and download the required files. Finally, students can submit their assignments in the assignments tab. Once the files are uploaded, they can press submit for the lecturer's evaluation.

3.1.2 Manage Profile and Manage Event



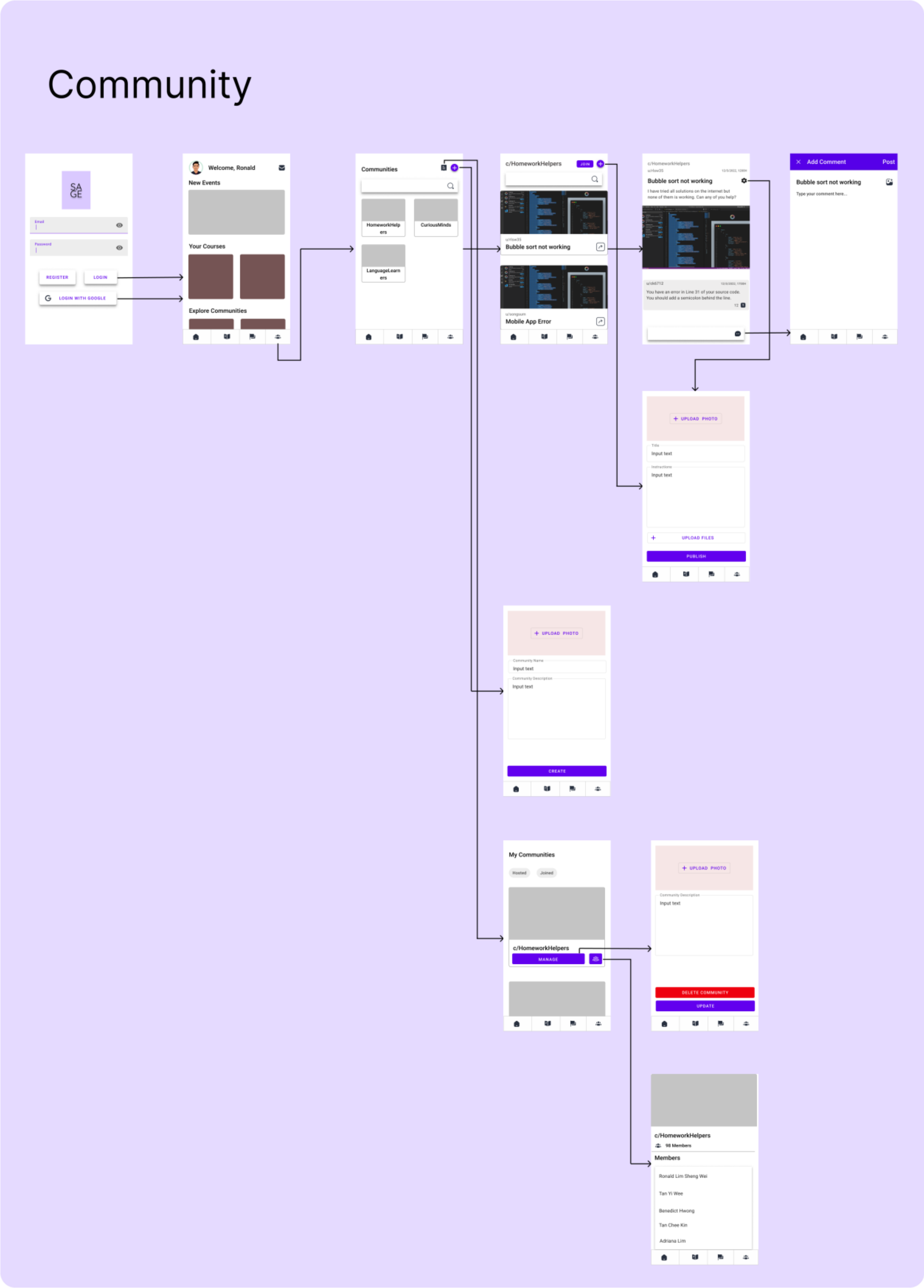
"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Figure 3.3 Manage Profile and Event Interfaces

Figure 3.3 depicts the flow of the Manage Profile and Manage Event modules. Starting with the Manage Profile, users can access this module from the home page of the system. From there, users can choose to edit their details at the edit interface.

For the Manage Event module, users can explore the latest events on the index page. Next, users can also choose to host their own events. By filling the event creation form, a new event will be created. Once created, the user hosted event can be managed as well. Users can view information regarding the participants, edit the details of the event as well as delete it.

3.1.3 Manage Community



I have tried all solutions on the internet but none of them is working. Can any of you help?

You have an error in Line 31 of your source code. You should add a semicolon behind the line.

Figure 3.4 Manage Community Interfaces

Figure 3.4 depicts the interface of Manage Community module. Starting with the login screen, the flow of the interfaces starts when the user presses on the community icon to access the module. Users will be first introduced to a few communities that the user has joined or new communities. Users can create their own communities by using the provided community creation form. As an admin of the community, users can manage the community by pressing the My Communities button. There, the community can be edited or deleted. The members of the community can be viewed as well.

Upon accessing an existing community, the user can view many threads that have been submitted by other users. The thread list contains a picture of the problem and the preview of the title and description of the problem. After accessing a thread, users can view the full content of the thread which includes the replies. Users can then submit a reply to an existing thread by using the reply bar on the bottom of the interface. Replies can be attached with images as desired. Next, users can also create a new thread within a community by using the provided thread form. Users will need to include the title, description and image of the problem.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

Table 3.1 Specification of Hardware and Software

Name	Type	Description	Purpose
Lenovo IdeaPad Gaming Gen 6	Laptop	A consumer laptop running Windows 11.	For word processing, documentation, and development of project.
Google Firebase	Cloud Server	A cloud-based NoSQL database tool.	To create, update, retrieve, delete system data of the project.
Android Studio	Software	An integrated development environment for the Android operating system.	To write the source code and run the simulation of the project.

APPENDIX E
SOFTWARE DESIGN DESCRIPTION (SDD)

2022

SOFTWARE DESIGN DESCRIPTION (SDD)

[SAGE: A Community Empowered University E-Learning Application]

DOCUMENT APPROVAL

	Name	Date
Authenticated by: ________________________ Ronald Lim Sheng Wei	Ronald Lim Sheng Wei	1/6/2022
Approved by: ________________________ Client		

Software :

Archiving Place :

TABLE OF CONTENT

CONTENT **PAGE**

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

SAGE aims to provide tertiary level students and lecturers alike, an application for peer-to peer communication and academic discussion as well as a tool to manage their classes. The SAGE application includes four main modules. The systems are as follows:

e. Manage Class

The Manage Class allows students to join class via a generated class code. From within a class, they can retrieve uploaded educational resources, chat with peers from the same class, view announcements by lecturers and submit their assignments. For lecturers, they can create classes, assignments as well as announcement. Assignments can be graded by the lecturers and returned to the students. They can participate in the chat room of the class as well.

f. Manage Events

The Manage Events module allows the user to create, delete, edit their hosted events. Users wanting to explore the campus for more events to attend can do so on the module as well.

g. Manage Profile

Manage Profile module allows user to edit their profile picture as well as personal information.

h. Manage Community

Manage Community allows users to create, edit, delete as well as update their communities. From within the communities, users can create, edit or delete their threads. Users can reply to their threads with words and images.

1.2 SYSTEM IDENTIFICATION

System Title:	SAGE: Community Empowered University E-Learning Application
System Abbreviation:	SAGE

System Identification Number: SAGE-SDD-V01-23

1.3 ARCHITECTURE / BLUEPRINT

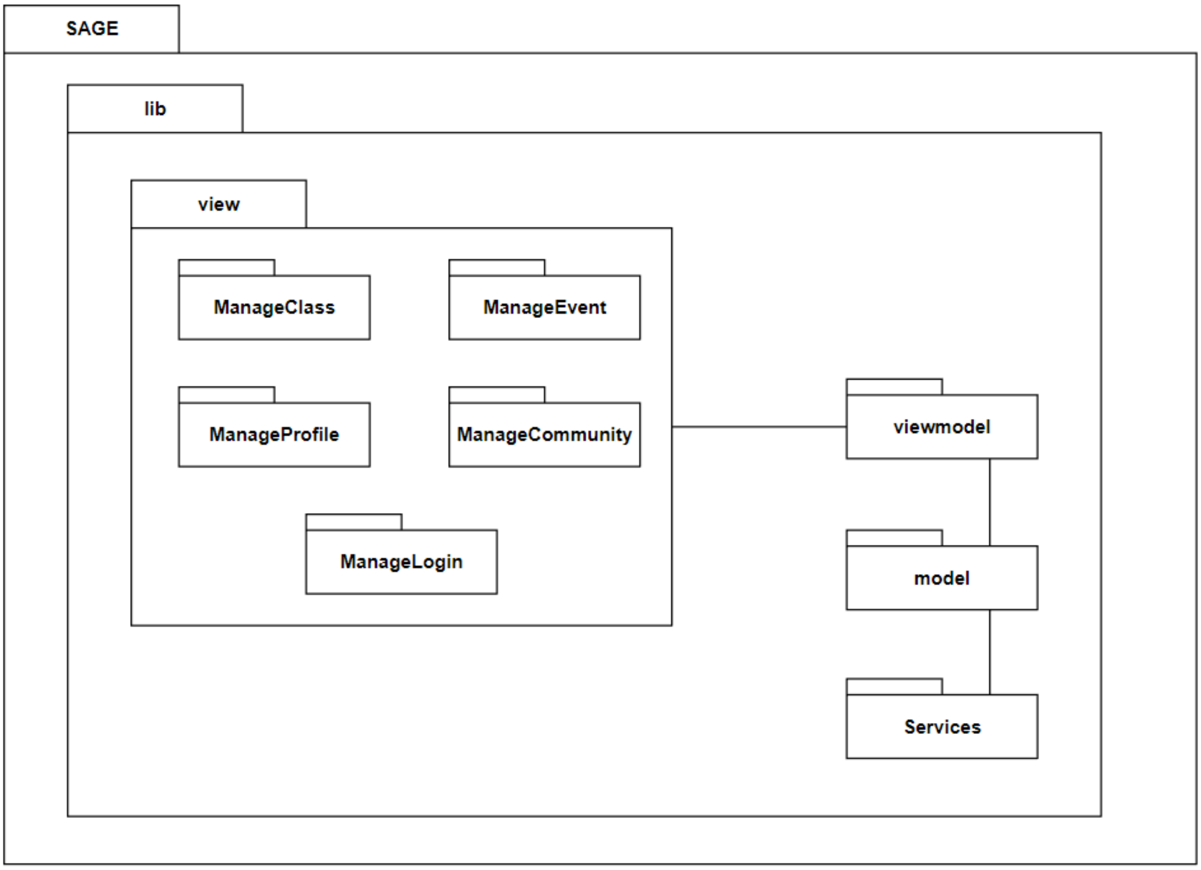


Figure 1.1 General Architecture of SAGE

1.4.1 Application Layer

1.4.1.1 Manage Class

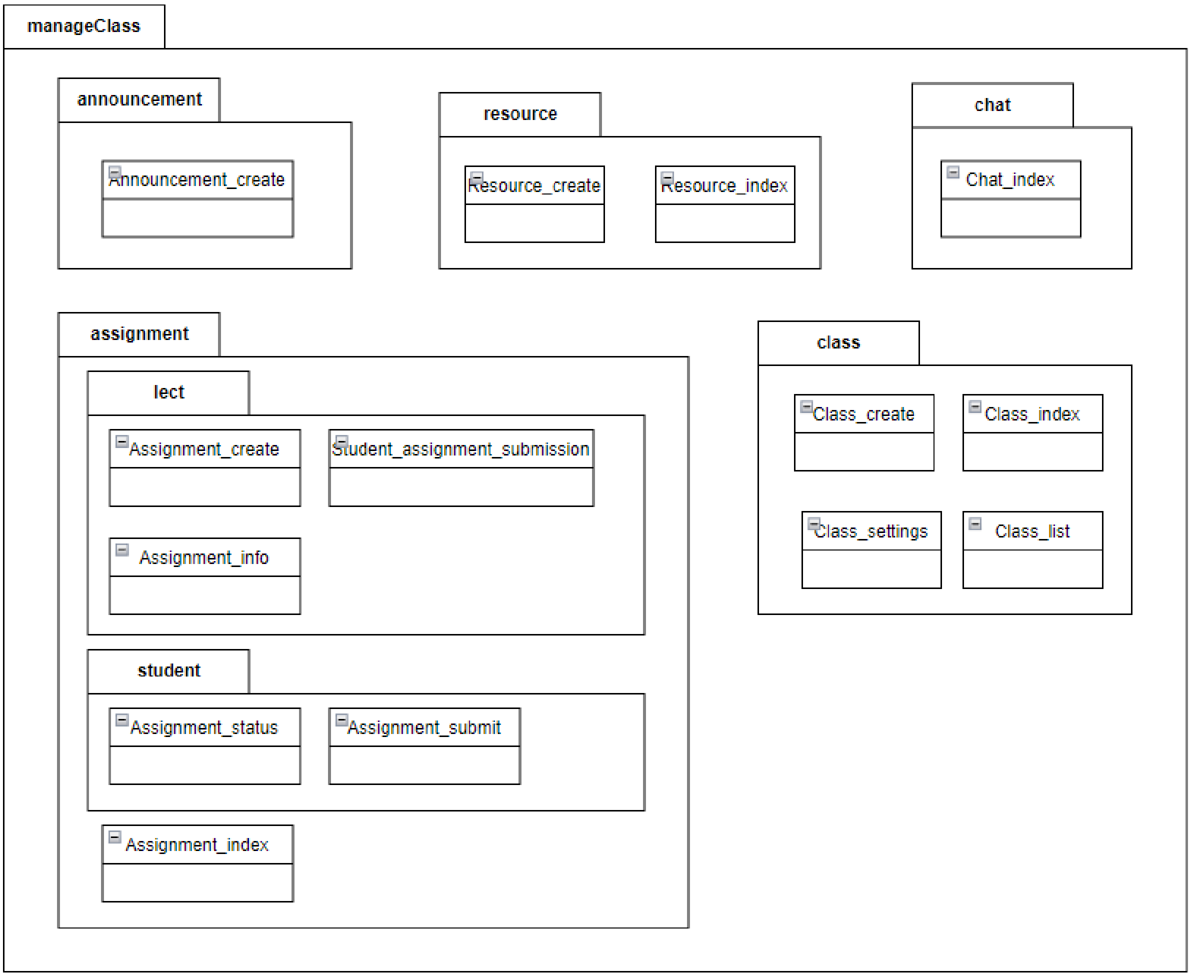


Figure 1.2 Manage Class View

Table 1.1 Manage Class View Description

Class Name	Description
Class_list	Interface that shows the classes joined or hosted by the user.
Class_index	Interface that shows the class details

Class_settings	Interface that shows the class settings
Class_create	Interface that shows the class creation form.
Chat_index	Interface that shows the class chat
Announcement_create	Interface that shows the class announcement creation form.
Assignment_info	Interface that shows the class assignment details
Assignment_create	Interface that shows the class assignment creation form.
Student_assignment_submission	Interface that shows the class assignment submission list.
Assignment_status	Interface that shows the class assignment grade.
Assignment_submit	Interface that shows the class assignment details and submission.
Resource_index	Interface that shows the class resource details.
Resource_create	Interface that shows the class resource creation form.

1.4.1.2 Manage Profile

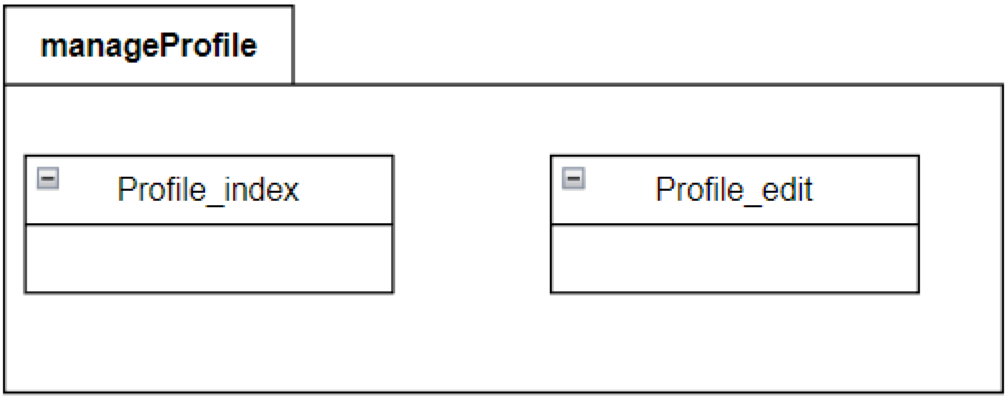


Figure 1.3 Manage Profile View

Table 1.2 Manage Profile View Description

Class Name	Description
Profile_index	Interface that shows the user profile details.
Profile_edit	Interface that shows the user profile details editing form.

1.4.1.3 Manage Event

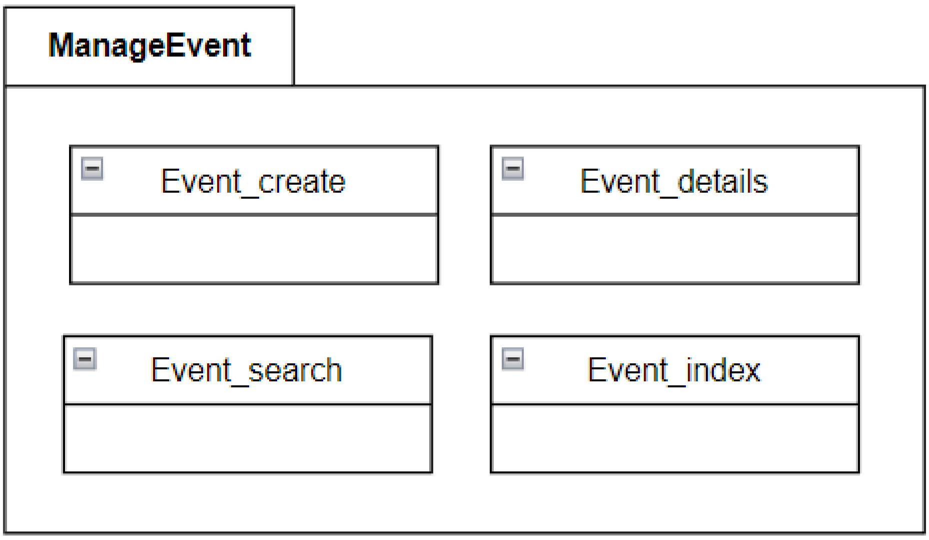


Figure 1.4 Manage Event View

Table 1.3 Manage Event View Description

Class Name	Description
Event_hosted	Interface that shows the events hosted by the user.
Event_index	Interface that shows the available events.
Event_joined	Interface that shows the events joined by the user.
Event_create	Interface that shows the event creation form.
Event_show	Interface that shows the event details.
Event-edit	Interface that shows the events hosted by the user.

Event_manage	Interface that shows the events hosted by the user.
Event_participant	Interface that shows the event participants.

1.4.1.4 Manage Community

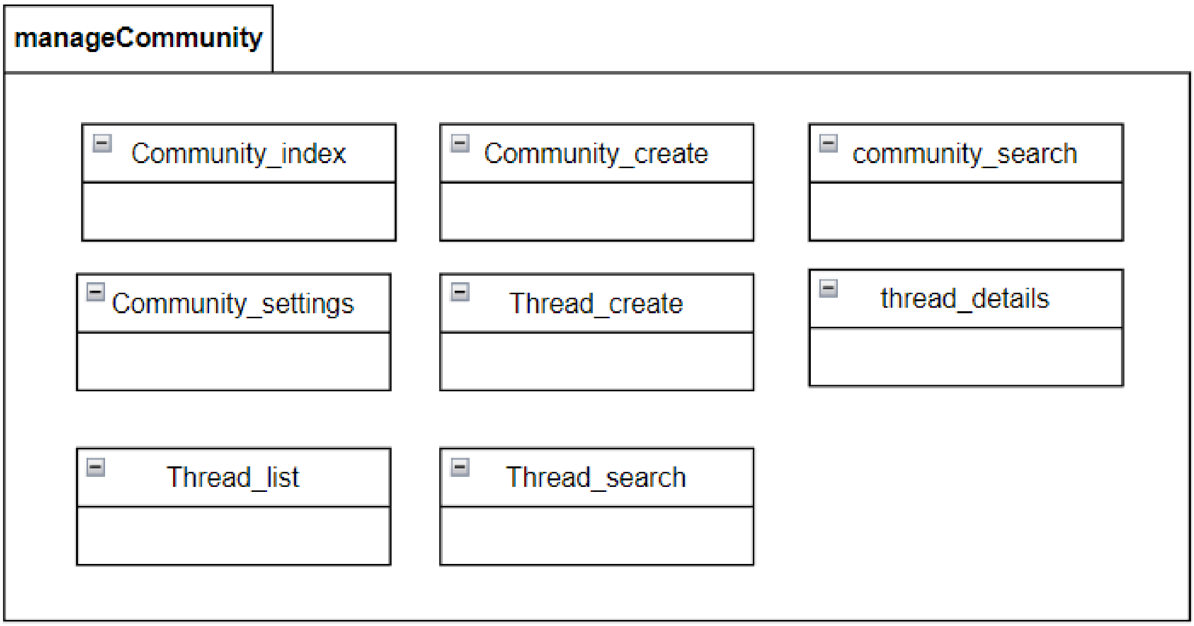


Figure 1.5 Manage Community View

Table 1.4 Manage Community View Description

Class Name	Description
Community_create	Interface that shows the communites creation form.
Community_index	Interface that shows the available communities.

Community_search	Interface that shows the communites creation form.
Community_settings	Interface that shows the community settings page.
Thread_list	Interface that shows the thread list.
Thread_details	Interface that shows the thread details.
Thread_create	Interface that shows the thread creation form.
Thread_search	Interface that shows the thread search page.

1.4.2 Business Layer

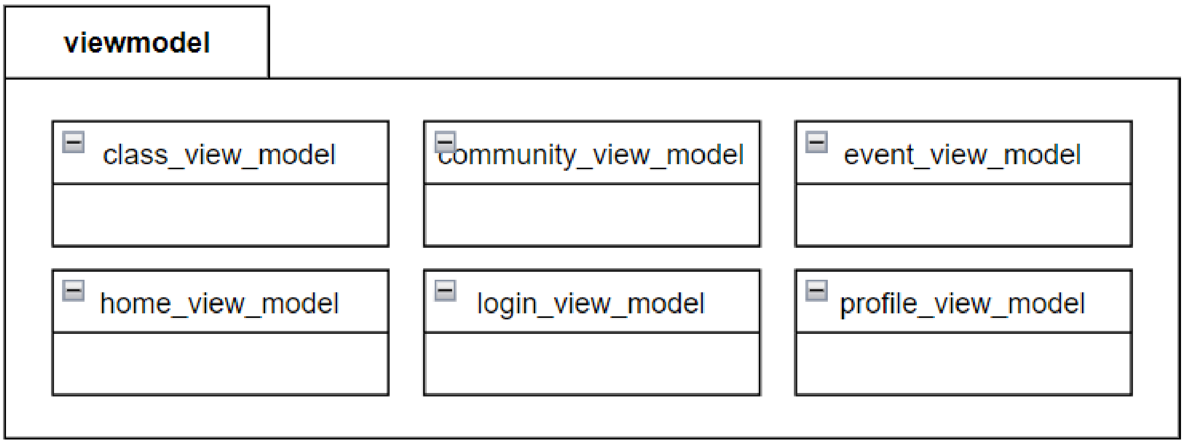


Figure 1.6 ViewModel

Table 1.5 ViewModel Description

Class Name	Description
Class_view_model	ViewModel for the Manage Class module.
Community_view_model	ViewModel for the Manage Community module.
Profile_view_model	ViewModel for the Manage Profile module.
Event_view_model	ViewModel for the Manage Event module.

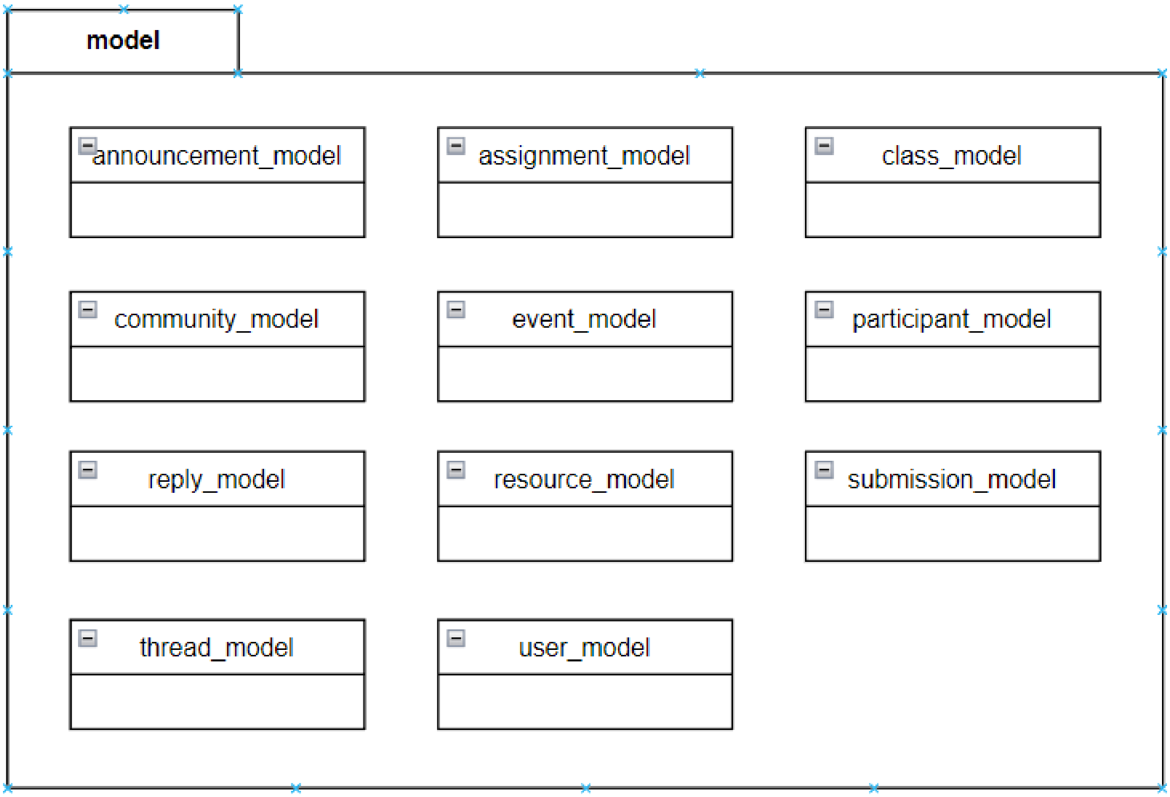


Figure 1.7 Model of SAGE

Table 1.6 Model Description

Class Name	Description
Class	Model for class data.
Users	Model for User data.
Event	Model for Event data.
Community	Model for Community data.
Thread	Model for Thread data.
Reply	Model for Reply data.

Assignment	Model for Assignment data.
Announcement	Model for Announcement data.
Resource	Model for Resource data.
Submission	Model for Submission data.

1.4.3 Middleware Layer

Figure 1.8 Middleware of SAGE

Table 1.7 Middleware Description

Class Name	Description
Firebase	A cloud-based NoSQL service for the system database.

CHAPTER 2

2.1 DETAILED DESCRIPTION

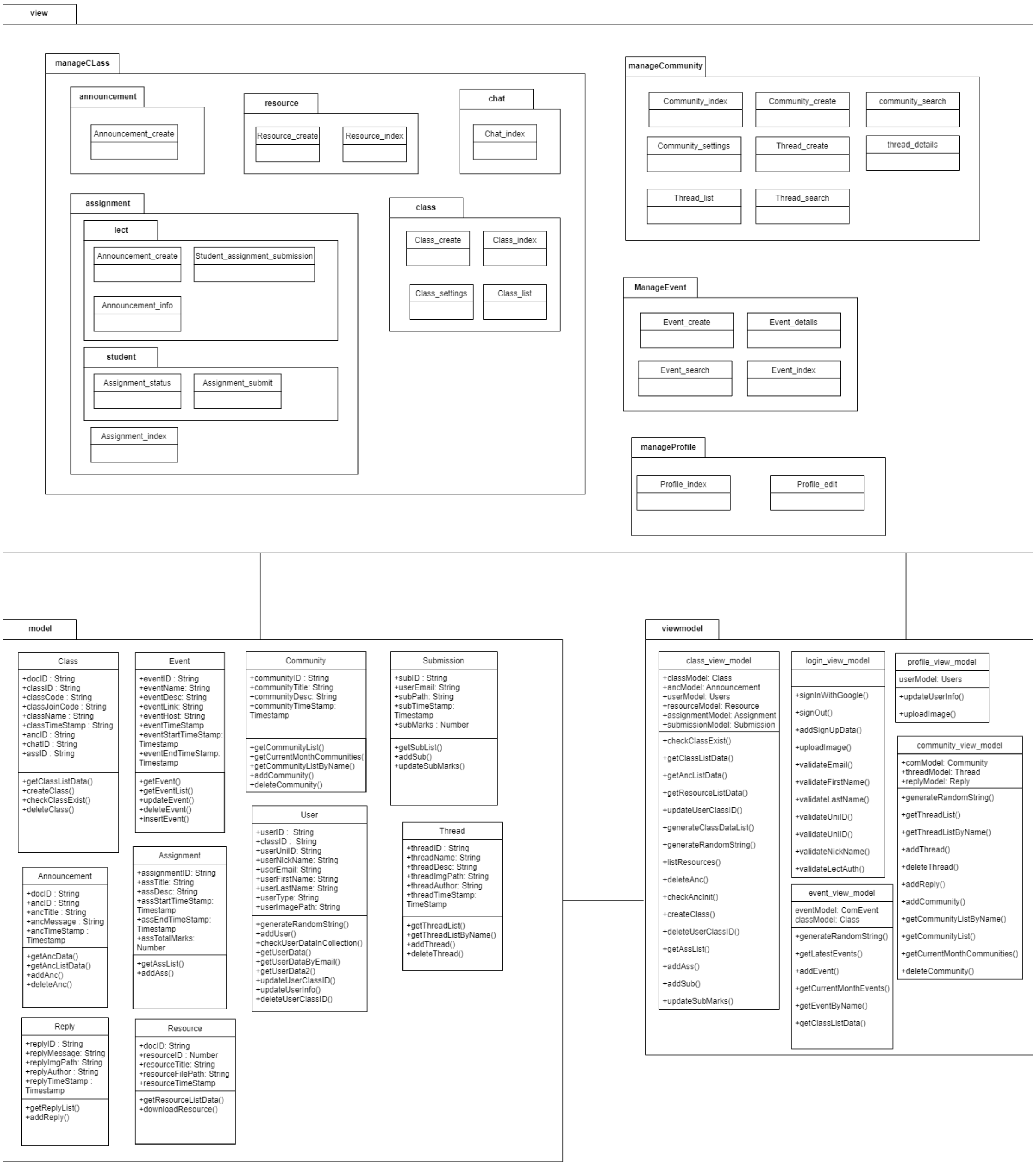


Figure 2.1 General Detailed Class Diagram of SAGE

2.1.1 Manage Class

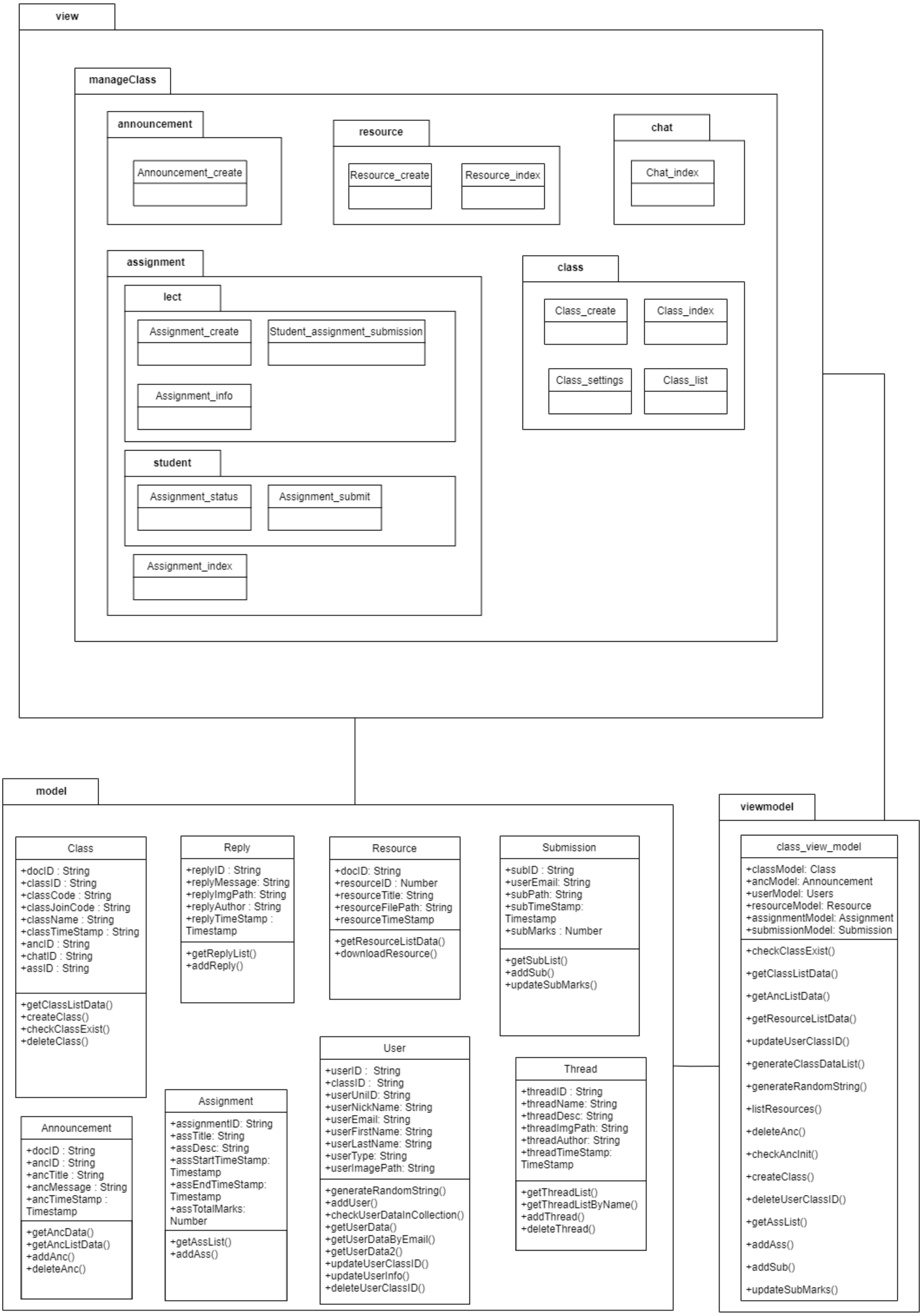


Figure 2.2 Manage Class Class Diagram

2.1.1.1 View

2.1.1.1.1 Class_index

Table 2.1 Class_index Class Description

Class Type	**Boundary Class**	
Responsibility	This class allows the user to view the class module index.	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.2 Class_create

Table 2.2 Class_create Class Description

Class Type	**Boundary Class**	
Responsibility	This class allows the user add new classes.	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**

	N/A	N/A
Algorithm	N/A	

2.1.1.1.3 Class_settings

Table 2.3 Class_settings Class Description

Class Type	**Boundary Class**	
Responsibility	This class allows the user to access class settings.	
Attributes	**Attribute Name**	**Attribute Type**
	classID	Number
Methods	**Method Name**	**Description**
	N/A	**N/A**
Algorithm	**N/A**	

2.1.1.1.4 Class_list

Table 2.4 Class_list Class Description

Class Type	**Boundary Class**
Responsibility	This class allows the user to access their personal registered class list.

Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.5 Chat_index

Table 2.5 Chat_index

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to access the chat room of the class.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.6 Announcement_create

Table 2.6 Announcement_create Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to create new assignments.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.7 Assignment_index

Table 2.7 Assignment_index Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to access the assignments of the class.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**

	N/A	N/A
Algorithm	**N/A**	

2.1.1.1.8 Assignment_info

Table 2.8 Assignment_info Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to view assignment info.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.9 Assignment_create

Table 2.9 Assignment_create Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to create new assignments.**	
Attributes	**Attribute Name**	**Attribute Type**

	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.10 Student_assignment_submission

Table 2.10 Student_assignment_submission Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to view submitted assignment files.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.11 Assignment_status

Table 2.11 Assignment_status Class Description

Class Type	**Boundary Class**
Responsibility	**This class allows the user to grade assignments.**

Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.12 Assignment_submit

Table 2.12 Assignment_submit Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to submit assignments.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.13 Resource_index

Table 2.13 Resource_index Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to view resource list.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.1.14 Resource_create

Table 2.14 Resource_create Class Description

Class Type	**Boundary Class**	
Responsibility	This class allows the user to create new resources.	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A

Algorithm	N/A

2.1.1.1.15 Resource_edit

Table 2.15 Resource_edit Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to edit resources.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.1.2 ViewModel

2.1.1.2.1 Class_view_model

Table 2.16 Class_view_model Class Description

Class Type	**ViewModel Class**	
Responsibility	**This class allows the system to interact with the ManageClass module views and models.**	
Attributes	**Attribute Name**	**Attribute Type**

	classModel	Class
	ancModel	Announcement
	userModel	Users
	resourceModel	Resource
	assignmentModel	Assignment
	submissionModel	Submission
Methods	**Method Name**	**Description**
	checkClassExist()	A function that checks if a class exists
	getClassListData()	A function that retrieves a list of all classes
	getAncListData()	A function that retrieves a list of announcements for a specific class
	getResourceListData()	A function that retrieves a list of resources for a specific class
	updateUserClass|D()	A function that updates the class ID for a specific user
	generateClassDataList()	A function that generates a list of class data from a list of class documents

	generateRandomString ()	A function that generates a random alphanumeric string of a specified length
	listResources()	A function that lists all resources in a specified filepath
	deleteAnc()	A function that deletes a specified announcement
	checkAncInit()	A function that checks if an announcement collection has been initialized for a specific class and initializes it if necessary
	createClass()	A function that creates a new class and updates the user's class ID
	deleteUserClassID()	A function that deletes a user's class ID
	getAssList()	A function that retrieves a list of assignments for a specific class
	addAss()	A function that adds an assignment to a specific class
	addSub()	A function that adds a submission to an assignment for a specific user
	updateSubMarks()	A function that updates the marks for a specific submission of an assignment

Algorithm	**checkClassExist(classJoinCode)**
	START
	RETURN CALL checkClassExist(classJoinCode) from class model
	END	

```
getClassListData()
START
RETURN CALL getClassListData() from class model
END

getAncListData(classID)
START
RETURN CALL getAncListData(classID) from announcement
model
END

getResourceListData(classID)
START
RETURN CALL getResourceListData(classID) from resource
model.
END

updateUserClassID(classID)
START
RETURN CALL updateUserClassID(classID) from user model.
END

generateClassDataList(classData, classDataList)
START
FOR EACH doc in classData
ADD a Class object to classDataList with the following
parameters : doc.id, doc['classID'], doc['className'],
doc['classCode'], doc['classJoinCode'], doc['ancID'],
doc['chatID'],doc['assID']
RETURN classDataList
END

generateRandomString(length)
START
DECLARE _chars as a string containing all the possible
characters
DECLARE _rnd as a new instance of random
RETURN a string composed of random characters of length as
passed in the parameter, selected from _chars
END

listResources(filepath)
START
RETURN Firebase storage list based on filepath.
END

deleteAnc(ancID, docID)
START
RETURN ancModel.deleteAnc(ancID, docID)
```

	END
	
	checkAncInit(classID)
	START
	DECLARE uniqueid as a random string of 20 characters
	RETURN the uniqueid after querying the "ClassAnnouncement" collection for documents with a "classID" field that is equal to the passed in classID, limiting the query to one document. If a document is found, return the uniqueid, otherwise create a new document in the "ClassAnnouncement" collection with "classID" and "ancID" fields, set the "classID" field to the passed in classID and the "ancID" field to the uniqueid. Also update the class document with the same classID in the "Class" collection to include the uniqueid in the "ancID" field.
	END
	
	createClass(classData)
	START
	DECLARE classID as the result of classModel.createClass(classData)
	UPDATE userModel.updateUserClassID(classID)
	END
	
	deleteUserClassID(classID)
	START
	UPDATE CALL deleteClass(classID) from class model.
	UPDATE CALL deleteUserClassID(classID) from user model.
	END
	
	getAssList(assID)
	START
	RETURN CALL getAssList(assID) from assignment model
	END
	
	addAss(assID, assData)
	START
	RETURN CALL addAss(assID, assData) from assignment model
	END
	
	addSub(assID, assignmentID, userEmail)
	START
	RETURN CALL addSub(assID, assignmentID, userEmail) from assignment model
	END
	
	updateSubMarks(assID, assignmentID, subData, marks)
	START
	RETURN CALL updateSubMarks(assID, assignmentID, subData, marks) from submission model

	END

2.1.2 Manage Profile

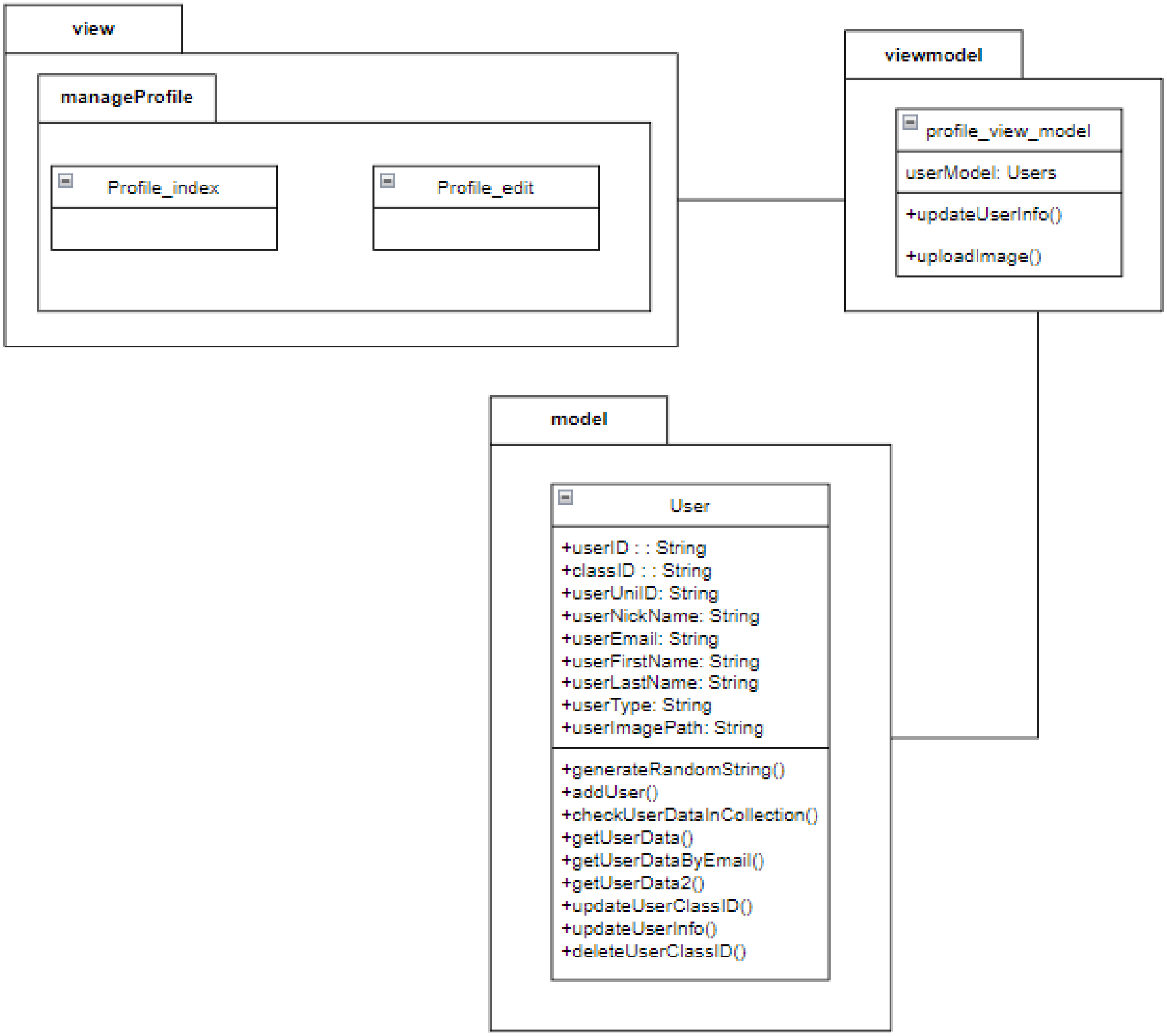


Figure 2.3 Manage Profile Class Diagram

2.1.2.1 View

2.1.2.1.1 Profile_index

Table 2.17 Profile_index Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to view their own user profile.**	
Attributes	**Attribute Name**	**Attribute Type**

	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.2.1.1 Profile_edit

Table 2.18 Profile_edit Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to edit their user profile.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.2.2 ViewModel

2.1.2.2.1 Profile_view_model

Table 2.19 Profile_view_model Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the system to interact with the ManageProfile module views and models.**	
Attributes	**Attribute Name**	**Attribute Type**
	userModel	Users
Methods	**Method Name**	**Description**
	updateUserInfo()	A function to update user information.
	uploadImage()	A function to upload image to the server.
Algorithm	**updateUserInfo(userFirstName, userLastName)** **START** RETURN CALL updateUserInfo(userFirstName, userLastName) from user model **END** **uploadImage(file, uniID)** **START** DECLARE storageRef as a reference to the root of FirebaseStorage DECLARE imagesRef as a reference to the child "profileImage" with uniID as the child of imagesRef TRY UPLOAD file to imagesRef CATCH FirebaseException PRINT "image error" PRINT e **END**	

2.1.3 Manage Event

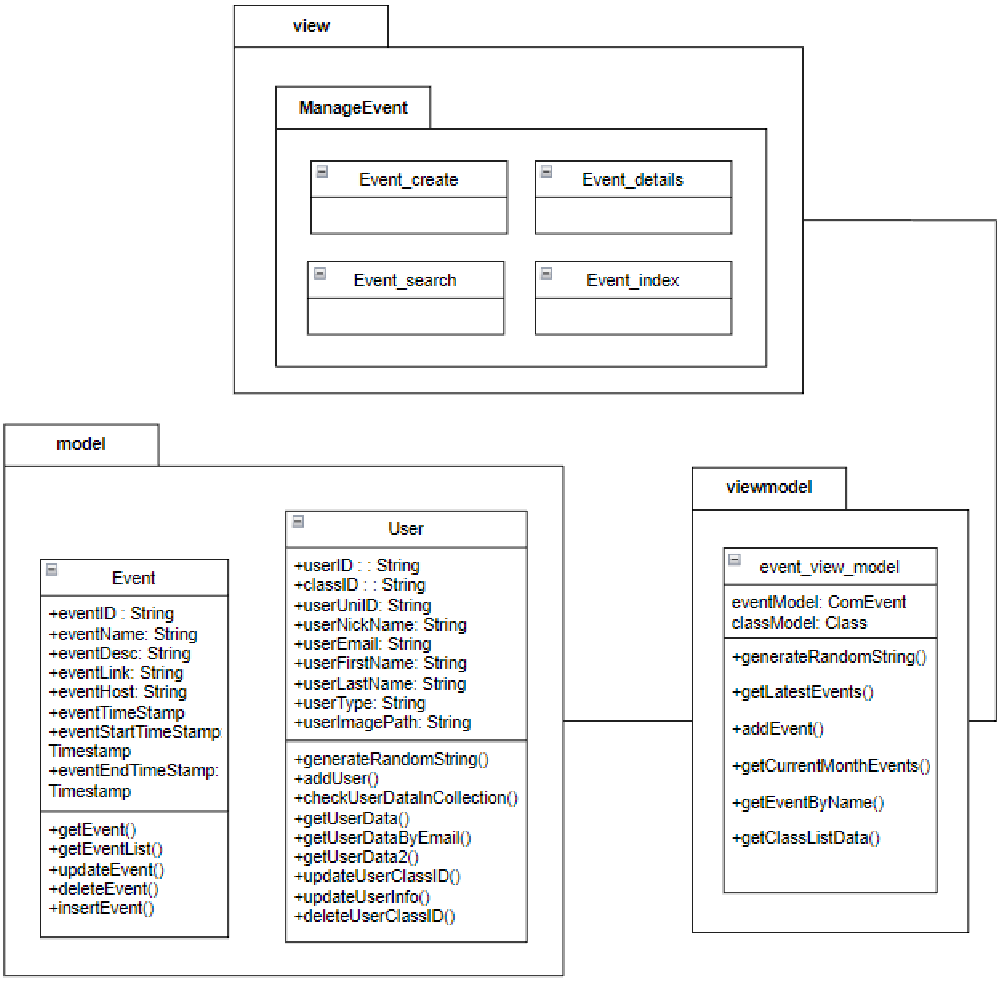


Figure 2.4 Manage Event Class Diagram

2.1.3.1 View

2.1.3.1.1 Event_index

Table 2.20 Event_index Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to view events.**	
Attributes	**Attribute Name**	**Attribute Type**

	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.3.1.2 Event_create

Table 2.21 Event_ create Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to create an event.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.3.1.3 Event_details

Table 2.22 Event_ details Class Description

Class Type	**Boundary Class**

Responsibility	**This class allows the user to edit an event's details.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.3.1.4 Event_search

Table 2.23 Event_ search Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to search for events.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.3.2 ViewModel

2.1.3.2.1 Event_view_model

Table 2.24 Event_view_model Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the system to interact with the ManageEvent module views and models.**	
Attributes	**Attribute Name**	**Attribute Type**
	eventModel classModel	ComEvent Class
Methods	**Method Name**	**Description**
	generateRandomString()	A function that generates a random alphanumeric string of a specified length
	getLatestEvents()	A function that retrieves the latest events
	addEvent()	A function that adds a new event
	getCurrentMonthEvents ()	A function that retrieves a list of events happening in the current month
	getEventByName()	A function that retrieves events with a specified name
	getClassListData()	A function that retrieves a list of all classes

Algorithm	**generateRandomString(length)** **START** DECLARE and INITIALIZE variable _chars as a string of alphanumeric characters DECLARE variable _rnd as a new random object RETURN a new string created by ITERATING through a GENERATED list of code units, where each code unit is DETERMINED by the code unit of a RANDOMLY SELECTED character from the _chars variable, for the specified length **END** **getLatestEvents()** **START** RETURN the result of calling the getLatestEvents() function from the eventModel object **END** **addEvent(eventData)** **START** CALL the addEvent(eventData) function from the eventModel object RETURN the result of the function call **END** **getCurrentMonthEvents()** **START** RETURN the result of calling the getCurrentMonthEvents() function from the eventModel object **END**

	getEventByName(searchQuery) **START** CALL the getEventByName(searchQuery) function from the eventModel object RETURN the result of the function call **END** **getClassListData()** **START** CALL the getClassListData() function from the classModel object RETURN the result of the function call **END** **deleteEvent(eventID)** **START** CALL the deleteEvent(eventID) function from the eventModel object RETURN the result of the function call **END**

2.1.4 Manage Community

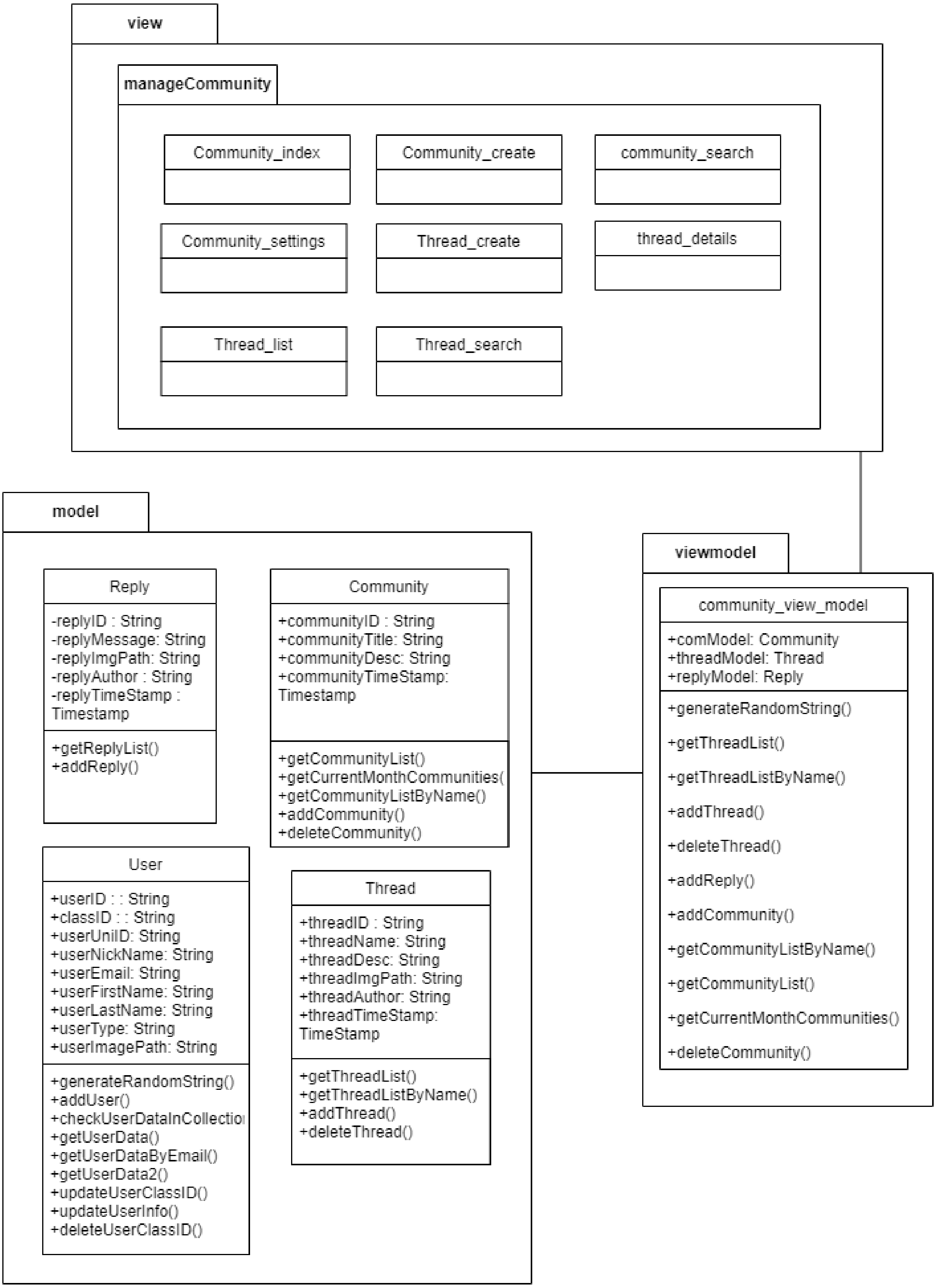


Figure 2.5 Manage Community Class Diagram

2.1.4.1 View

2.1.4.1.1 Community_create

Table 2.25 Community_create Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to create new communities**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.4.1.2 Community_index

Table 2.26 Community_index Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to view communities.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**

	N/A	N/A
Algorithm	N/A	

2.1.4.1.3 Community_search

Table 2.27 Community_ search Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to search for communities.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.4.1.4 Community_edit

Table 2.28 Community_edit Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to edit communities.**	
Attributes	**Attribute Name**	**Attribute Type**

	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.4.1.5 Thread_list

Table 2.29 Thread_list Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to view thread list.**	
Attributes	**Attribute Name**	**Attribute Type**
Methods	**Method Name**	**Description**
	N/A	**N/A**
Algorithm	**N/A**	

2.1.4.1.6 Thread_create

Table 2.30 Thread_ create Class Description

Class Type	**Boundary Class**

Responsibility	**This class allows the user to create threads.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.4.1.7 Thread_search

Table 2.31 Thread_reply Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to create replies to threads.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.4.1.8 Thread_details

Table 2.32 Thread_ details Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the user to create replies to threads.**	
Attributes	**Attribute Name**	**Attribute Type**
	N/A	N/A
Methods	**Method Name**	**Description**
	N/A	N/A
Algorithm	N/A	

2.1.4.2 ViewModel

2.1.4.2.1 Community_view_model

Table 2.33 Community_view_model Class Description

Class Type	**Boundary Class**	
Responsibility	**This class allows the system to interact with the ManageCommunity module views and models.**	
Attributes	**Attribute Name**	**Attribute Type**
	communityID communityTitle	Number String

	communityDesc	String
	communityTimeStamp	Timestamp
	participantID	Number
	organizer	Number
	userID	Number
	userNickName	String
	threadID	Number
	threadName	String
	threadDesc	String
	threadImgPath	String
	replyID	Number
	replyMessage	String
	replyImgPath	String
Methods	**Method Name**	**Description**
	generateRandomString()	A function that generates a random alphanumeric string of a specified length
	getThreadList()	A function that retrieves a list of all threads

	getThreadListByName()	A function that retrieves a list of threads with a specified name
	addThread()	A function that adds a new thread
	deleteThread()	A function that deletes a specified thread from a specified location
	addReply()	A function that adds a new reply to a specified thread
	addCommunity()	A function that adds a new community
	getCommunityListByName()	A function that retrieves a list of communities with a specified name
	getCommunityList()	A function that retrieves a list of all communities
	getCurrentMonthCommunities ()	A function that retrieves a list of communities created in the current month
	deleteCommunity()	A function that deletes a specified community

Algorithm	**checkClassExist(classJoinCode)**
	START
	RETURN CALL checkClassExist(classJoinCode) from class model
	END
	
	getClassListData()
	START
	RETURN CALL getClassListData()from class model
	END	

	getAncListData(classID)
	START
	RETURN CALL getAncListData(classID) from announcement model
	END
	
	getResourceListData(classID)
	START
	RETURN CALL getResourceListData(classID) from resource model
	END
	
	updateUserClassID(classID)
	START
	RETURN CALL updateUserClassID(classID) from user model
	END
	
	generateClassDataList(classData, classDataList)
	START
	FOR EACH doc in classData
	ADD a Class object to classDataList with the following parameters : doc.id, doc['classID'], doc['className'], doc['classCode'], doc['classJoinCode'], doc['ancID'], doc['chatID'],doc['assID']
	RETURN classDataList
	END
	
	generateRandomString(length)
	START
	DECLARE _chars as a string containing all the possible characters
	DECLARE _rnd as a new instance of random
	RETURN a string composed of random characters of length as passed in the parameter, selected from _chars
	END
	
	listResources(filepath)
	START
	RETURN resource from Firebase Storage according to filepath
	END
	
	deleteAnc(ancID, docID)
	START
	RETURN CALL deleteAnc(ancID, docID) from announcement model
	END
	
	checkAncInit(classID)
	START
	DECLARE uniqueid as a random string of 20 characters

	RETURN the uniqueid if the querySnapshot of
	Firebase collection “ClassAnnouncement” where classID equal to classID
	ELSE
	DECLARE data as a map containing the keys "classID" and "ancID" with values classID and uniqueid respectively
	SET data in
	Firebase collection “ClassAnnouncement” with uniqueid as document ID
	UPDATE
	Firebase collection “Class” with classID, SET ancID as uniqueID
	RETURN uniqueid
	END
	
	createClass(classData)
	START
	DECLARE classID as the returned value of classModel.createClass(classData)
	UPDATE userModel.updateUserClassID(classID)
	END
	
	deleteUserClassID(classID)
	START
	CALL deleteClass(classID) from class model
	CALL deleteUserClassID(classID) from user model
	END
	
	getAssList(assID)
	START
	RETURN CALL getAssList(assID) from assignment model
	END
	
	addAss(String assID, Assignment assData)
	START
	RETURN CALL addAss(assID,assData) function from assignment model
	END
	
	addSub(assID, assignmentID,userEmail)
	START
	RETURN addSub function from Submission model
	END
	
	updateSubMarks(assID,assignmentID,Submission subData,marks)
	START
	RETURN updateSubMarks from Submission model
	END

2.1.5 Model

2.1.5.1 User

Table 2.34 User Table

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the user model.**	
Attributes	**Attribute Name**	**Attribute Type**
	userID	String
	classID	String
	userUniID	String
	userNickName	String
	userEmail	String
	userFirstName	String
	userLastName	String
	userType	String
	userTimeStamp	Timestamp
	userImagePath	String
Methods	**Method Name**	**Description**

	generateRandomString()	Generate a random alphanumeric string of a specified length
	addUser()	Add a new user
	checkUserDataInCollection()	Check if user data already exists
	getUserData()	Retrieve user data
	getUserDataByEmail()	Retrieve user data by email
	getUserData2()	Retrieve additional user data
	updateUserClassID()	Update the class ID for a specific user
	updateUserInfo()	Update user's personal information
	deleteUserClassID()	Delete a user's class ID
Algorithm	**generateRandomString()** START Declare and initialize variable _chars as a string of alphanumeric characters Declare variable _rnd as a new random object Return a new string created by iterating through a generated list of code units, where each code unit is determined by the code unit of a randomly selected character from the _chars variable, for the specified length END **addUser(uniID, email, password, nickname, firstName, lastName, userType)** **START** Use Firebase Firestore to add a document to the 'userCollection' collection, using the specified email as the document ID Set the following fields for the document: "userID", "userEmail", "userPassword", "userUniID", "userNickName", "userFirstName", "userLastName", "userType"	

	Print "User Added" if successful, or "Failed to add user: [error]" if not **END** **checkUserDataInCollection()** START Use Firebase Firestore to query the 'User' collection for documents where the 'userEmail' field matches the current user's email If no documents are found, return nothing If documents are found, return "Exist" END **getUserData()** **START** Use Firebase Firestore to query the 'userCollection' collection for documents where the 'userEmail' field matches the current user's email Return the query snapshot **END** **getUserDataByEmail(email)** **START** Use Firebase Firestore to query the 'userCollection' collection for documents where the 'userEmail' field matches the specified email Return the query snapshot **END** **getUserData2()** **START** Declare variable userData Use Firebase Firestore to query the 'userCollection' collection for documents where the 'userEmail' field matches the current user's email For each document in the query snapshot, create a new Users object with the following fields: "userID", "userUniID", "userNickName", "userFirstName", "userLastName", "userType", "", "userEmail" Return the Users object **END** **updateUserClassID(classID)** **START** Declare variable classList as an empty list Add classID to classList Use Firebase Firestore to query the 'userCollection' collection for documents where the 'userEmail' field matches the current user's email Use the ID of the first document in the query snapshot to update the 'classID' field in the 'userCollection' collection, using the

	classList variable with FieldValue.arrayUnion() Print "ClassID Updated" if successful, or "Failed to update user classID: [error]" if not **END** **updateUserInfo(userFirstName, userLastName)** **START** Declare variable userEmail Use Firebase Firestore to query the 'userCollection' collection for documents where the 'userEmail' field matches the current user's email Set userEmail equal to the value of the 'userEmail' field for the first document in the query snapshot Use Firebase Firestore to update the 'userCollection' collection, setting the 'userFirstName' and 'userLastName' fields for the document with the userEmail ID Print "User Info Updated" if successful, or "Failed to update user info: [error]" if not **END**

2.1.5.2 Class

Table 2.35 Class Table

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the class model.**	
Attributes	**Attribute Name**	**Attribute Type**
	docID	String
	classID	String
	classCode	String
	classJoinCode	String

	className	String
	classTimeStamp	String
	ancID	String
	chatID	String
	assID	String
Methods	**Method Name**	**Description**
	getClassListData()	Retrieve a list of all classes
	createClass()	Create a new class
	checkClassExist()	Check if a class with a specified join code already exists
	deleteClass()	Delete a class
Algorithm	**generateRandomString(length)** START Declare variable _chars as a string containing a set of characters Declare variable _rnd as a new instance of the Random class Return a new string created from the code units at the randomly generated index of _chars, for the given length END **getClassListData()** START Declare variable classIDList as an empty list Get user data and for each document retrieved, add the classID value to classIDList Declare variable classRef as a reference to the "Class" collection in Firebase Firestore Declare variable classes as a query of classRef where the classID is in classIDList	

	Return the query snapshot of classes END **createClass(classData)** **START** Declare variable uniqueID as a call to generateRandomString() with a length of 20 Declare variable classJoinCode as a call to generateRandomString() with a length of 5 Declare variable dataMap as a map containing the class information, including uniqueID and classJoinCode Add the dataMap to a document in the "Class" collection in Firebase Firestore using the uniqueID as the document ID Return uniqueID **END** **checkClassExist(classJoinCode)** **START** Declare variable querySnapshot as the query snapshot of the "Class" collection in Firebase Firestore where the classJoinCode matches the input classJoinCode Iterate through the documents in the query snapshot, and return the classID value of the first document found **END** **deleteClass(classID)** **START** Delete the document with the matching classID in the "Class" collection in Firebase Firestore Print a message confirming the deletion, or an error message if the deletion fails **END**

2.1.5.3 Assignment

Table 2.36 Assignment Table

Class Type	**Model Class**

Responsibility	**This class allows the user to access the assignment model.**	
Attributes	**Attribute Name**	**Attribute Type**
	assignmentID: String	String
	assTitle: String	String
	assDesc: String	String
	assStartTimeStamp	Timestamp
	assEndTimeStamp	Timestamp
	assTotalMarks	Number
Methods	**Method Name**	**Description**
	getAssList()	Retrieve assignment list
	addAss()	Add new assignment
Algorithm	**getAssList(assID)** **START** Declare variable assList as an empty list Declare variable querySnapshot as the query snapshot of the collection "Assignments" within a document of the "assCollection" where the document ID is the input assID Iterate through the documents in the query snapshot, creating a new Assignment object for each document with the corresponding data and adding it to assList Return assList **END** **addAss(assID, assData)** **START** Add the assData to a document in the "Assignments" collection within a document of the "assCollection" using the	

	assData.assignmentID as the document ID and the assID as the parent document ID Print a confirmation message if the data was added successfully, or an error message if the addition failed **END**

2.1.5.4 Announcement

Table 2.37 Announcement Table

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the announcement model.**	
Attributes	**Attribute Name**	**Attribute Type**
	docID	String
	ancID	String
	ancTitle	String
	ancMessage	String
	ancTimeStamp	Timestamp
Methods	**Method Name**	**Description**
	getAncData()	Retreive announcement data from annoncement collection

	getAncListData()	Retreive announcement data list from annoncement collection
	addAnc()	Add announcement data to annoncement collection
	deleteAnc()	Delete announcement data from annoncement collection
Algorithm	**generateRandomString(length)** **START** Declare variable _chars as a string containing a set of characters Declare variable _rnd as a new instance of the Random class Return a new string created from the code units at the randomly generated index of _chars, for the given length **END** **getAncListData(classID)** **START** Declare variable ancs as a query of the "ancCollection" where the classID is equal to the input classID and ordered by descending ancTimeStamp Return the query snapshot of ancs **END** **getAncData(ancID)** **START** Declare variable ancs as a query of the "ancCollection" where the ancID is equal to the input ancID Return the query snapshot of ancs **END** **addAnc(ancData, classID)** **START** Add the ancData to a document in the "ancCollection" using ancData.ancID as the document ID, including the classID and a timestamp Print a confirmation message if the data was added successfully, or an error message if the addition failed **END** **deleteAnc(ancID,docID)** **START** Delete the document with the matching ancID in the "Announcements" collection within a document of the "ancCollection"	

	Return after delete **END**

2.1.5.5 Resource

Table 2.38 Resource Table

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the resource model.**	
Attributes	**Attribute Name**	**Attribute Type**
	docID	String
	resourceID : Number	Number
	resourceTitle: String	String
	resourceFilePath: String	String
	resourceTimeStamp	Timestamp
Methods	**Method Name**	**Description**
	getResourceListData()	Retrieve resource data list from Firebase Storage
	downloadResource()	Download resource data list from Firebase Storage
Algorithm	**getResourceListData(classID)** **START**	

	Declare variable resc as a query of the "resourceCollection" where the classID is equal to the input classID and ordered by descending resourceTimeStamp Return the query snapshot of resc **END** **downloadResource(filepath)** **START** Check if the permission to manage external storage is granted, if not, request for permission Declare variable resourceRef as a reference to the file specified by the input filepath in storage Declare variable appDocDir as the application document directory Declare variable file as a new file named "Verification Letter.pdf" in the appDocDir path Declare variable url as the download url of resourceRef Download the file from url to the file location **END**

2.1.5.6 Submission

Table 2.39 Submission Class Description

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the submission model.**	
Attributes	**Attribute Name**	**Attribute Type**
	subID	String
	userID	String
	subPath	String
	subTimeStamp	Timestamp
	subMarks	Number

Methods	Method Name	Description
	getSubList()	Retrieve submission list from submission collection.
	addSub()	Add submission to submission collection.
	updateSubMarks()	Update submission marks at submission collection.
Algorithm	**getSubList(assID, assignmentID)** **START** Declare variable subList as an empty list Declare variable querySnapshot as the query snapshot of the collection "Submissions" within a document of the "Assignments" collection within a document of the "assCollection" where the document ID is the input assID and assignmentID Iterate through the documents in the query snapshot, creating a new Submission object for each document with the corresponding data and adding it to subList Return subList **END** **addSub(assID, assignmentID, subData)** **START** Add the subData to a document in the "Submissions" collection within a document of the "Assignments" collection within a document of the "assCollection" using the subData.subID as the document ID and the assID and assignmentID as the parent document ID Print a confirmation message if the data was added successfully, or an error message if the addition failed **END** **updateSubMarks(assID, assignmentID, subData, marks)** **START** Update the subMarks field in the document of the "Submissions" collection within a document of the "Assignments" collection within a document of the "assCollection" where the document ID is the input subData.subID and the assID and assignmentID as the parent document ID Print a confirmation message if the update was successful, or an error message if the update failed	

	END

2.1.5.7 Event

Table 2.40 Event Class Description

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the event model.**	
Attributes	**Attribute Name**	**Attribute Type**
	eventID	String
	eventName	String
	eventDesc	String
	eventLink	String
	eventHost	String
	eventTimeStamp	Timestamp
	eventStartTimeStamp	Timestamp
	eventEndTimeStamp	Timestamp
Methods	**Method Name**	**Description**
	deleteEvent()	Delete event from event collection
	addEvent()	Add event to event collection

	searchEvent()	Search event from event collection
	getEventByName()	Retrieve event from event collection by name.
	getCurrentMonthEvents()	Retrieve event from event collection by month.
	getLatestEvents()	Retrieve event from event collection by current month.
Algorithm	**getLatestEvents()** **START** Return a stream of snapshots of the "eventCollection" ordered by eventStartTimeStamp **END** **getCurrentMonthEvents()** **START** Declare variable date as the current date Return a stream of snapshots of the "eventCollection" where the eventTimeStamp is greater than or equal to the first day of the current month **END** **getEventByName(searchQuery)** **START** Capitalize the input searchQuery Return a query snapshot of the "eventCollection" where the eventName is greater than or equal to the searchQuery and less than the next character after the last character of the searchQuery **END** **searchEvent(eventName)** **START** Return a query snapshot of the "eventCollection" ordered by eventName, starting at the eventName and ending at the eventName followed by the maximum Unicode character **END** **addEvent(eventData)** **START** Capitalize the eventName field in eventData	

	Declare variable data as a map containing the eventData fields and the current timestamp Add data to a document in the "eventCollection" using the eventData.eventID as the document ID Return after adding **END** **deleteEvent(eventID)** **START** Delete the document in the "eventCollection" where the document ID is the input eventID Print a confirmation message if the deletion was successful, or an error message if the deletion failed **END**

2.1.5.8 Community

Table 2.41 Community Class Description

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the community model.**	
Attributes	**Attribute Name**	**Attribute Type**
	communityID	String
	communityTitle	String
	communityDesc	String
	communityTimeStamp	Timestamp
Methods		**Description**

	getCommunityList()	Retrieve community list from community collection.
	getCurrentMonthCommunities()	Retrieve community list from community collection by current month.
	getCommunityListByName()	Retrieve community list from community collection by name.
	addCommunity()	Add community to community collection.
	deleteCommunity()	Delete community from community collection.
Algorithm	**getCommunityList()** **START** Return a stream of snapshots of the "communityCollection" **END** **getCurrentMonthCommunities()** **START** Declare variable date as the current date Return a stream of snapshots of the "communityCollection" where the communityTimeStamp is greater than or equal to the first day of the current month **END** **getCommunityListByName(searchQuery)** **START** Capitalize the input searchQuery Return a query snapshot of the "communityCollection" where the communityTitle is greater than or equal to the searchQuery and less than the next character after the last character of the searchQuery **END** **addCommunity(data)** **START** Declare variable newdata as a map containing the data fields	

	Add newdata to a document in the "communityCollection" using the data.communityID as the document ID Return after adding **END** **deleteCommunity(comID)** **START** Delete the document in the "communityCollection" where the document ID is the input comID Print a confirmation message if the deletion was successful, or an error message if the deletion failed **END**

2.1.5.9 Thread

Table 2.42 Thread Class Description

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the thread model.**	
Attributes	**Attribute Name**	**Attribute Type**
	threadID	String
	threadName	String
	threadDesc	String
	threadImgPath	String
	threadAuthor	String
	threadTimeStamp	Timestamp
Methods		**Description**

	getThreadList()	Retrieve thread list from thread collection.
	getThreadListByName()	Retrieve thread list from thread collection by name.
	addThread()	Add thread to thread collection.
	deleteThread()	Delete thread from thread collection.

Algorithm	**getThreadList(communityID)**
	START
	RETURN
	“Thread” collection snapshot where “Community” collection document ID is equal to communityID.
	END
	
	getThreadListByName(communityID, searchQuery)
	START
	DECLARE Future variable
	SET variable equal to
	“Thread” collection where threadName is equal to searchQuery where “Community” collection document ID is equal to communityID.
	RETURN variable
	END
	
	addThread(communityID, data)
	START
	DECLARE newdata variable
	SET newdata equal to data
	DECLARE Future variable
	SET variable equal to
	Add newdata to “Thread” collection where document ID is equal to data.threadID where “Community” collection document ID is equal to communityID.
	RETURN variable
	END
	
	deleteThread(communityID, threadID)
	START
	DECLARE Future variable
	SET variable equal to	

	Delete thread from "Thread" collection where document ID is equal to threadID where "Community" collection document ID is equal to communityID.
	RETURN variable
	END

2.1.5.10 Reply

Table 2.43 Reply Class Description

Class Type	**Model Class**	
Responsibility	**This class allows the user to access the reply model.**	
Attributes	**Attribute Name**	**Attribute Type**
	replyID	String
	replyMessage	String
	replyImgPath	String
	replyAuthor	String
	replyTimeStamp	Timestamp
Methods		**Description**
	getReplies()	System retrieves community list details from database.
	deleteReply()	System retrieves community details from database.

	insertReply()	System updates community details from database.

Algorithm	**getReplyList()**
	START
	RETURN “Reply” collection in “Thread” collection in “Community” collection.
	END
	
	addReply(communityID, threadID, data)
	START
	DECLARE newdata as a map containing the keys : "replyID", "replyMessage", "replyImgPath", "replyAuthor", "replyTimeStamp" with values from the data passed
	RETURN
	Add new data to “Thread” collection where document ID is equal to threadID where “Community” collection document ID is equal to communityID an set document id as data.replyID.
	END	

2.2 DATA DICTIONARY

Table 2.44 User Table

Field Name	Data Type	Constraint	Description
userID	String	PK	User Identification Number
classID	Array	FK	Class Identification Number
userUniID	String		User University Identification
userNickName	String		User Community Nickname
userEmail	String		User Email
userFirstName	String		User First Name
userLastName	String		User Last Name
userType	String		User Type
userPassword	String		User Password
userTimeStamp	timestamp		User Timestamp

Table 2.45 Class Table

Field Name	Data Type	Constraint	Description
classID	String	PK	Class Identification Number
ancID	String	FK	Announcement Identification Number
assID	String	FK	Assignment Identification Number
className	String		Class Name
classTimeStamp	Timestamp		Class Created Timestamp

classCode	String		Class Code
classJoinCode	String		Class Joining Code

Table 2.46 ClassAssignment Table

Field Name	Data Type	Constraint	Description
assID	String	PK	Assignment Identification Number
classID	String	FK	Class Identification Number

Table 2.47 Assignments Table

Field Name	Data Type	Constraint	Description
assignmentID	String	PK	Assignment Identification Number
assTitle	String		Assignment Title
assDesc	String		Assignment Description
assStartTimeStamp	String		Assignment Start Timestamp
assEndTimeStamp	String		Assignment End Timestamp
assTotalMarks	Number		Assignment Total Marks

Table 2.48 Submission Table

Field Name	Data Type	Constraint	Description
subID	String	PK	Submission Identification Number

userEmail	String	FK	User Email
subMarks	Number		Submission Returned Marks

Table 2.49 ClassAnnouncement Table

Field Name	Data Type	Constraint	Description
ancID	String	PKFK1	Announcement Identification Number
classID	String	PKFK2	Class Identification Number

Table 2.50 Announcements

Field Name	Data Type	Constraint	Description
ancTimeStamp	Timestamp		Announcement Created Time Stamp
ancTitle	String		Announcement Title
ancMessage	String		Announcement Message

Table 2.51 Community

Field Name	Data Type	Constraint	Description
communityID	String	PK	Community Identification Number
communityTitle	String		Community Title
communityDesc	String		Community Description

communityTimeStamp	Timestamp		Community Created Timestamp
communityAuthor	String		Community Author Name

Table 2.52 Thread

Field Name	Data Type	Constraint	Description
threadID	String	PK	Thread Identification Number
threadAuthor	String		Thread Author Name
threadName	String		Thread Name
threadDesc	String		Thread Description
threadImgPath	String		Thread Image Path
threadID	String	PK	Thread Identification Number

Table 2.53 Reply

Field Name	Data Type	Constraint	Description
replyID	String		Reply Identification Number
replyAuthor	String		Reply Author Name
replyMessage	String		Reply Message
replyTimeStamp	Timestamp		Reply Created Time Stamp

Table 2.54 Event

Field Name	Data Type	Constraint	Description
eventID	String	PK	Event Identification Number
eventName	String		Event Name
eventDesc	String		Event Description
eventHost	String		Event Host Name
eventLink	String		Event Meeting Link
eventTimeStamp	Timestamp		Event Created Timestamp
eventStartTimeStamp	Timestamp		Event Start Timestamp
eventEndTimeStamp	Timestamp		Event End Timestamp