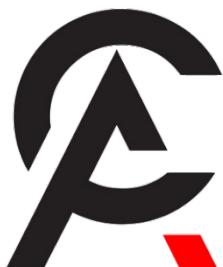




Faculty of Computing and Informatics

ALUMNI CENTER



Student Name	ID
Alex Tay Mao Xiang	1171101755
Lim Wei Xiang	1171101805
Ong Wen Hao	1181301414
Teo Sheng Pu	1171101665

Company Name: TOLA Software Sdn Bhd

Section: TC08 TT08

Submission Date:

Tables of Content

1. Introduction & Acknowledgement	3
2. Analysis Model of ALUMNI CENTER (part 1)	4 - 28
i. Introduction	4
ii. Work Distribution	4
iii. Problem Statement	5 – 7
iv. User Requirement	8 – 12
v. System Description	13
vi. Functional & Non-Functional Requirement	13 – 14
vii. Use Case Diagram	15 – 16
viii. Sequence Diagram	17 – 23
ix. Class Diagram	24 – 25
x. Class Responsibility Collaborator (CRC) Diagram	26 – 28
3. Design Model of ALUMNI CENTER (Part 2)	29 – 55
i. Refactoring Elements	29 – 31
ii. Data Design	32
iii. Entity Relationship Diagram (ERD)	33
iv. Data Dictionary	34 – 36
v. Architecture Design	37
vi. Interface Design	38 – 53
a. User Interface	38 – 51
b. External Interface Design	52
c. Internal Interface Design	53
vii. Component Design & Deployment Design	54
viii. 3-Tier Architecture Design (Additional Diagram)	55
4. Prototype Model of ALUMNI CENTER (Part 3)	56 – 103
i. Introduction	56
ii. Unit Testing with Test Cases	57 – 77
iii. Software Tools	78
iv. System Screen Shots	79 – 95
a. Public User	79 – 82
b. Logged-in User	83 – 87
c. Admin	88 – 95
v. Sample Codes and Database	96 - 103
5. Conclusion and Future Plans	104
6. References	105

Project Introduction & Acknowledgement

Alumni Management System (AMS) is a medium for the university to communicate and keep track of its graduates. Today, universities have their own alumni management systems (AMS) to manage data and details of their alumni. However, most of the systems used by these institutions are outdated. Some are inconvenient to use and view on mobile devices as the systems were built using obsolete technologies.

Moreover, the system developed by some of these universities are not integrated (all-in-one) as different components or functions are hosted on multiple domains, making it hard to be used as users have to navigate through different website for different functions. At the same time, the security issue had risen when the alumni management system of some universities allows an outsider to register as their alumni. This will create chaos for the management and is a horrible tragedy for the whole system as the invaders can easily access to the AMS.

Other than communication between university and its graduates, the AMS should allow communication between graduates themselves and display their personal portfolio to the public. Information like achievements in their studies as well as experiences in their career should be displayed in their profile. There are many benefits for being an alumni member of a university or institution such as providing information about the events that are organized in the campus.

To tackle the problems above, the team decided to create an integrated AMS application for android and iOS platform, with all functionalities all bundle in a centralized platform. This project focuses on the usability and comfortable user interface to facilitate navigation in the application.

The AMS includes functions such as registering, searching and managing the alumni information for sharing their expertise and finding job opportunities. With a feedback and suggestion dashboard, this alumni management system is a step towards creating interactions between current and past students as well as the staff in the university. In order to protect the system from outsiders, student ID is required during registration and the alumni account need to be verified by admin before the account can be used to access the application.

We would like to express our gratitude to our lecturer, Ts. Dr. Nor'Ain Binti Mohd Yusoff, and Mr. Khairi Shazwan Bin Dollmat who is our tutor for giving us the opportunity to carry out our very first software engineering project as well as the guidance which truly helped us along the development of the prototype. With this experience, it will help in our future work in conducting software development project in the future.

Analysis Model of ALUMNI CENTER (Part 1)

Introduction

There are three objectives of our systems:

1. Acts as a medium for universities or institutions to communicate with their graduated alumni
2. Provides a platform to facilitate the communication between graduates themselves
3. Delivers an organized management system which impose restrictions for different user groups

Work Distribution

Alex Tay	Lim Wei Xiang	Ong Wen Hao	Teo Sheng Pu
Join Discussion	Join Discussion	Join Discussion	Join Discussion
Study Existing System and Define Problem Statement			
Describe Scenario	Use Case Diagram	Describe Scenario	Use Case Diagram
Use Case Description Table	Conduct Survey	Use case Description Table	Sequence Diagram
Compile Survey Result	Class Diagram	Conduct Survey	Class Diagram
Finalize Report	Finalize Report	Finalize Report	Finalize Report

Problem Statement

Prior to designing the software, an analysis study had been conducted to define our problem statement. We studied several existing alumni management system (AMS) to discover the underlying problems which weren't solved yet. These AMS are developed by universities, namely Universiti Teknologi Malaysia (UTM), Universiti Tunku Abdul Rahman (UTAR), Taylor's University, Multimedia University (MMU), Universiti Sains Malaysia (USM). Tables below included details of each system along with elaborations and proofs.

Existing Software	Advantages	Disadvantages
UTAR (DARP) http://www.utar.edu.my/darp/	<ul style="list-style-type: none">• Able to organize events relevant to the alumni• Receive e-Newspaper• Job Discovery	<ul style="list-style-type: none">• Public unable to see existing alumni• Low maintenance of website (QR code provided on the website is outdated)• Low count of registered users.



Exiting Software	Advantages	Disadvantages
UTM (Alumnoid) http://alumnoid.utm.my/ http://alumni.utm.my/	<ul style="list-style-type: none"> • Able to search alumni by public • Receive email regarding latest events 	<ul style="list-style-type: none"> • Low security measurement (NO. I/C of students are visible during registration) • Not integrated (It has two websites for AMS)

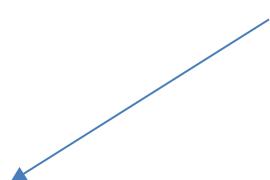
Create New Account

Register now to:

- update your details
- find old friends
- access to our worldwide UTM alumni community
- receive emails about UTM alumni events for you

Join the fun. Register now.

Low Security Measurement



Personal information

MyKad or ID International Student

940310105589
MOHAMAD HAFIZAT ZAINAL ABIDIN
Full name *
Full name (Full name without 'bin' or 'binti')

Exiting Software	Advantages	Disadvantages
Taylor's University Alumni Management System https://www.taylorsalumni.org/	<ul style="list-style-type: none"> • Logged-in users can search for other alumni • Logged-in users can find jobs via career centre. • Public users can view the upcoming events and announcements. 	<ul style="list-style-type: none"> • Public users unable to search for alumni • Security issue arises when the outsider can pass through the validation process and register as an alumni user.

The screenshot shows a user profile for Wei Xiang Lim, Class of 2018. The profile includes a photo, location (Alor Setar, Kedah, Malaysia), and birth date (29 March, 1995). The website has a navigation bar with links like Welcome, Network, Events, Contribute, Library, Directory, and Wei. A map of Southeast Asia is visible in the background.

The validation process is weak as one of the software team members can register as an alumnus

Existing Software	Advantages	Disadvantages
USM (SOM) http://www.som.usm.my/	<ul style="list-style-type: none"> • Alumni able to discover various job opportunities • Provide a platform which connects alumnus and lecturers 	<ul style="list-style-type: none"> • User registration requires a relative longer time to wait for its approval (users have to send the completed form to the administrator himself) • It is not accessible by public (visitor unable to see existing alumni)

Register as an alumni by filling in the Alumni Form ([download form](#))

Kindly email completed forms to: shankar@usm.my

For any queries, kindly contact:
Associate Professor Dr. Shankar Cheelliah
Research and Alumni Manager
School of Management
Universiti Sains Malaysia
11800, Penang
Universiti Sains Malaysia
Office no: +604 - 653 3426
Fax no: +604 - 657 7448
Email: shankar@usm.my

User must send his registration form himself and wait for reply which is not instantaneous

Existing Software	Advantages	Disadvantages
Multimedia University Alumni Management System	<ul style="list-style-type: none"> • Public users can view the announcement and the upcoming events. • Public users need to undergo verification process (IC) before they can register as an alumnus 	<ul style="list-style-type: none"> • Public users cannot search for alumni • Layout of the webpage is not user-friendly as the elements in the footer of the site is quite messy arranged.

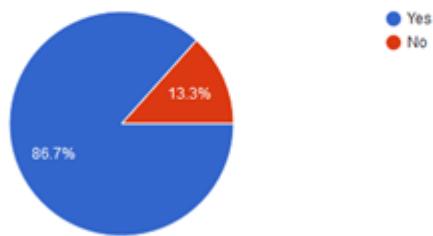
User Requirements

The team has decided to conduct a survey using random sampling among 30 persons from Multimedia University (MMU).

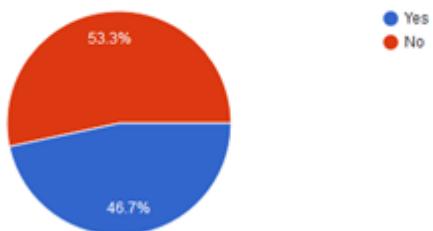
The purpose of doing this is to determine the problems encountered by the users of in MMU alumni website. By doing this survey, the team is able to obtain feedbacks from users about the features they expect to have in this application. These feedbacks are crucial for the team to generate new ideas for a better quality AMS application.

The results of the survey are summarized as follows:

Does your institute have an alumni management system? 
30 responses

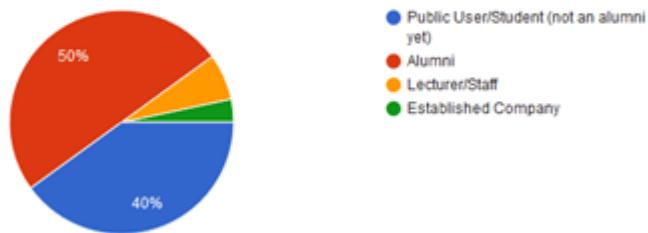


Do you recognize the necessity of having alumni management system? 
30 responses



Which user category you fall into ?

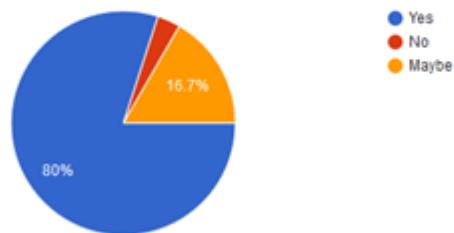
30 responses



There are quite a number of public users or current students who are interested in the proposed application. Therefore, it will be available for general public use through their portable interface. However, the team decides to limit public access to some of the features.

Should this AMS has a search function which allows you to find other registered alumni ?

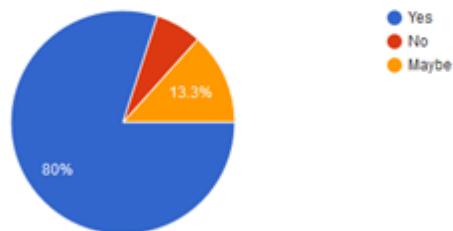
30 responses



Most of the respondents prefer to find their seniors through the alumni system. Therefore, a non-registered visitor will have the features to look at the list of graduates and their profiles.

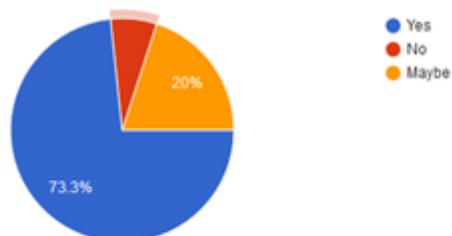
Should this AMS has a event management features which allows you to plan and organize an event as an alumni ?

30 responses



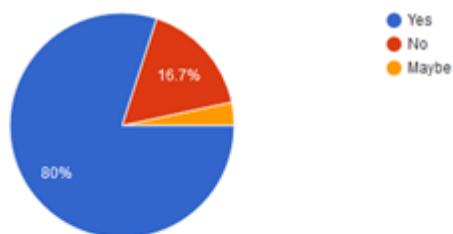
Should this AMS allows you to seek for available job opportunities in your field of study ?

30 responses



Should this AMS allows you to feedback to the institute in order to improve its environment ?

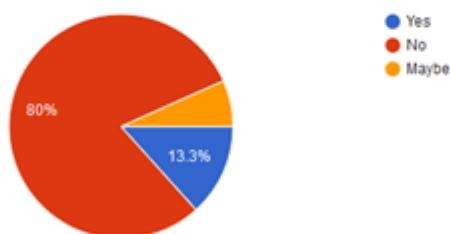
30 responses



Should this AMS enables you to fundraise for any legitimate and reasonable events?

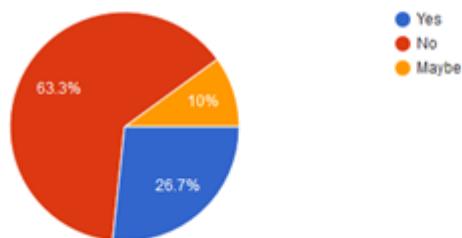


30 responses



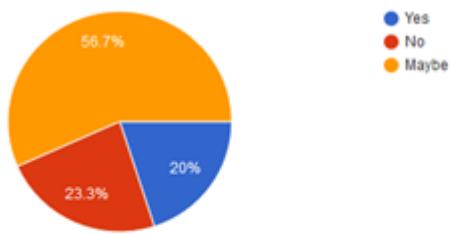
Should this AMS enables you to submit your resume to apply for certain job positions ?

30 responses



Should this AMS has a mailing feature which facilitate the communication between users ?

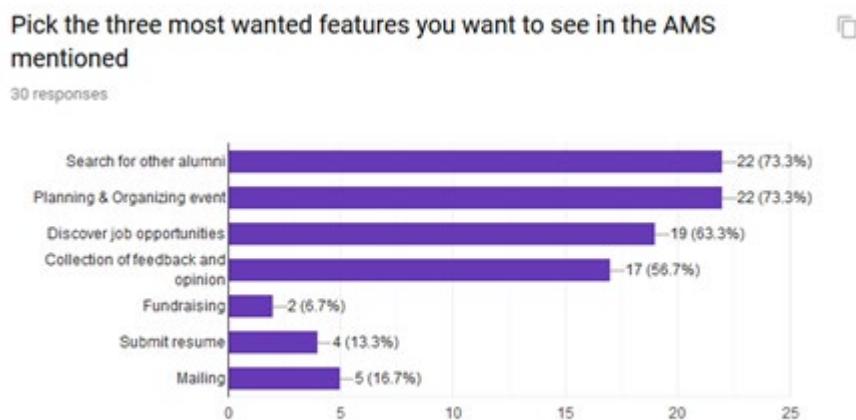
30 responses



From the pie charts above, we can conclude that majority of the respondents agree that the application should have alumni searching as well as event organizing function.

In short, most respondents recognize the fact that they are responsible to provide anonymous feedback through the alumni system. At the same time, they believe that an event management feature can help them to stay connected to the institution. In contrast, mailing, fundraising and resume submission features were the least favored where they don't recognize the importance of their existence.

The respondents were enquired to select their top three preferences in the application. The results are as follows:



It can therefore be concluded that most respondents prefer to have searching function (73.3%), event organizing feature (73.3%), job seeking platform (63.3%) and a medium to provide feedback to the institution (56.7%).

Hence, the team will implement the aforementioned features in the application and develop the models and designs required.

System Description

Based on the survey results, we proposed solutions as listed below (Requirements):

1. The public, alumni and admin must be able to search for personal details about all the alumni.
2. The public users are allowed to view the upcoming events and the announcement in the homepage of the application.
3. The public can choose to register or login to access all of the features in the application.
4. The system should only allow alumni with valid student ID to register as users.
5. The alumni should be able to view the list of jobs.
6. The alumni can organize an event through the application; The event has to get approval from admin to create.
7. The alumni are able to submit a feedback regarding their opinions towards the universities.
8. The alumni are given the right to modify their own user details as well as changing their login password.
9. The admins must be able to generate reports of the overview data of the system (number of users), list of events and the feedback from alumni.
10. The admins should be granted the right to post announcement
11. The admins are able to create an event.
12. The admins have the right to approve or decline the event that is created by alumni.
13. The admins must be able to add a job to the job lists.

Functional Requirement

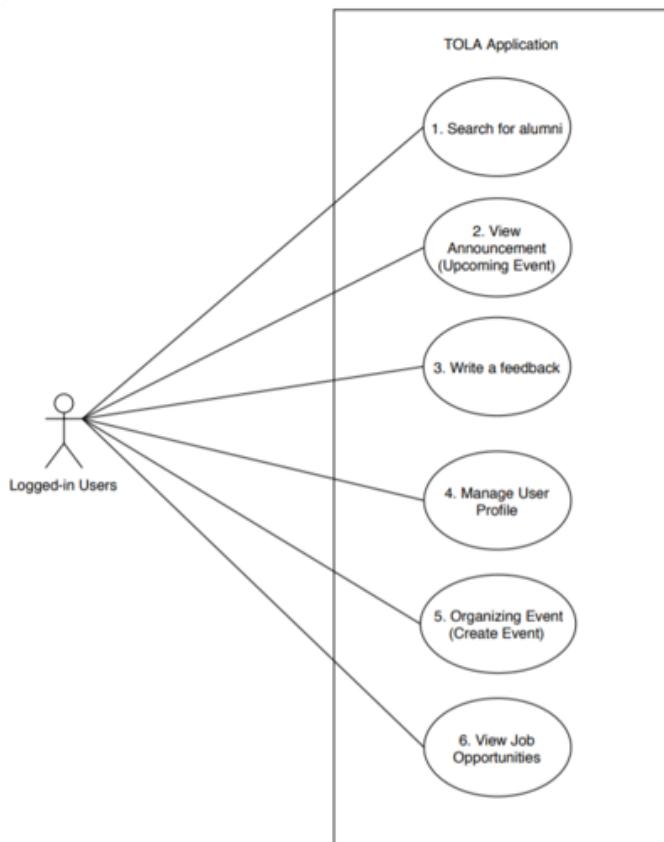
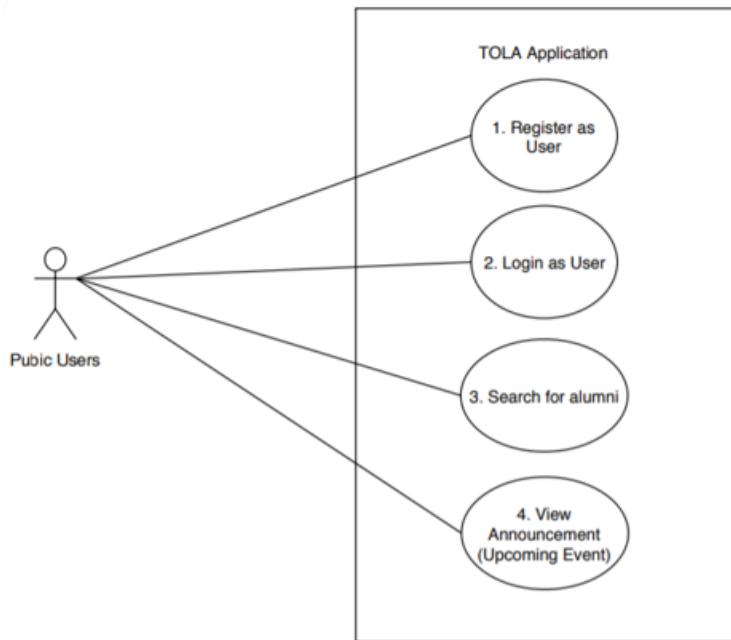
No.	Requirement Statement	Description
1.	System must be able to register user	Add new user's credential
2.	System must have login function	Validate user's ID and password
3.	Public users must be able to search for alumni	User's name
4.	Public users must be able to view announcement	
5.	Alumni must be able to write a feedback	User's comment
6.	Alumni must be able to manage profile	Update user's credential
7.	Alumni and admin must be able to organize an event	Add event's details
8.	Alumni must be able to view job opportunity	
9.	Admin must be able to manage events	Update events' details
10.	Admin must be able to manage announcement	Update announcement
11.	Admin must be able to generate report	Retrieve feedback information, user information, event information
12.	Admin must be able to manage an account	Update user's details
13.	Admin must be able to post job opportunities	Add new job opportunities

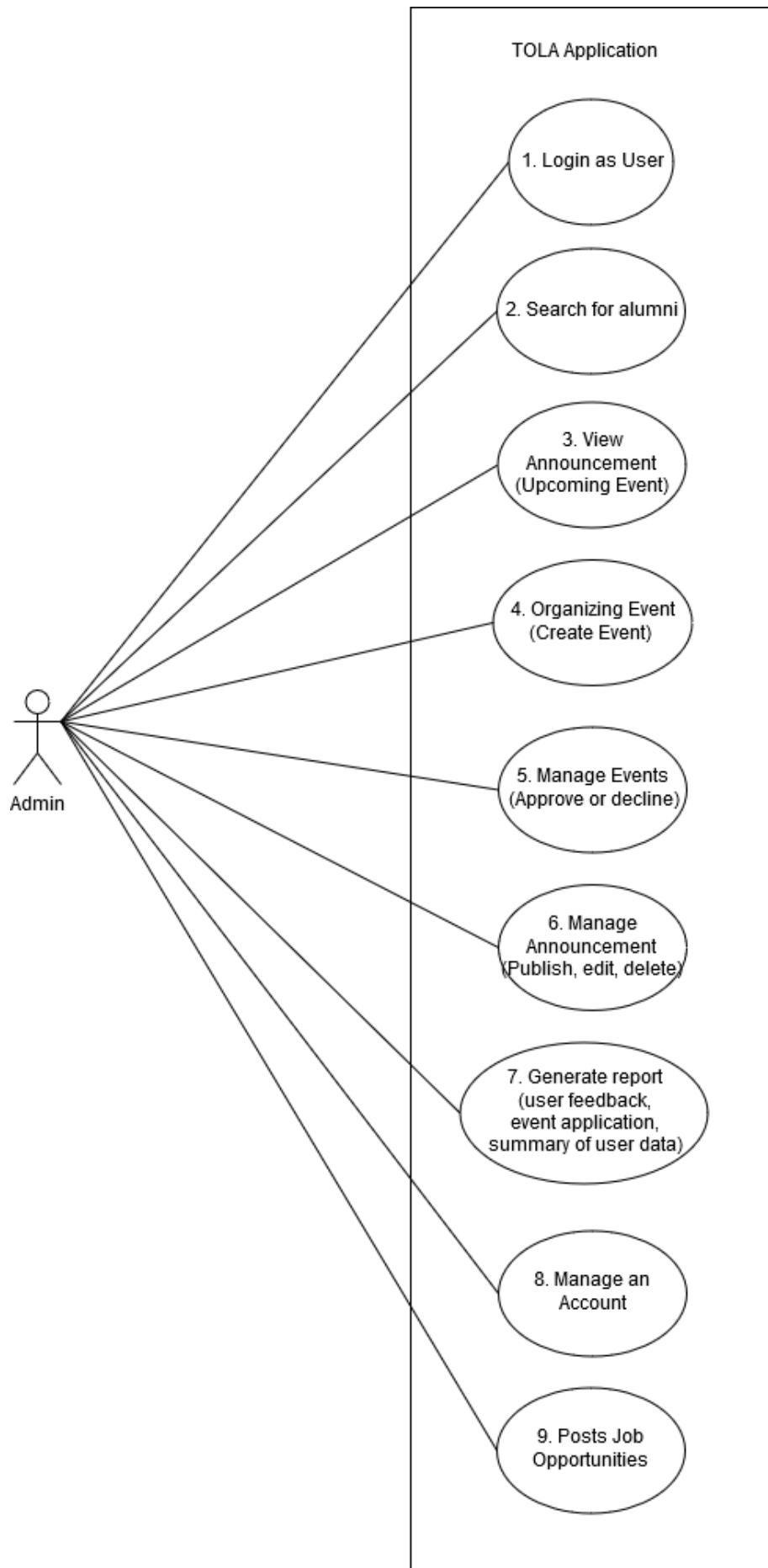
Non-Functional Requirement

No.	Requirement Statement	Description
1.	Security	<ul style="list-style-type: none"> User account will be locked for a period of time after several times of failed login attempts. The users must register with valid student ID and undergo verification process that is carried out by admin. The password registered must contain combinations of alphabet, number and at least one special symbol. Only alumni with valid user account are allowed to create event or submit feedback in order to prevent unwanted activities carried out by outsiders.
2.	Usability	<ul style="list-style-type: none"> A board which contains announcements and upcoming events are placed in the homepage to ease the use of this application. Each process such as event creation or feedback submission will have instruction given to navigate the users.
3.	Data Integrity	<ul style="list-style-type: none"> The data is well categorized into several tables such as Users, Events, Feedback, Report and Announcements to ease the process of insert, update or delete the data.
4.	Transparency	<ul style="list-style-type: none"> The report generated are true based on the real information of the database as the admin cannot interfere with the data that is being generated or extracted out from the database.
5.	Maintainability	<ul style="list-style-type: none"> The system of the application will be well documented and written in a readable format for others to examine it.
6.	Extensibility	<ul style="list-style-type: none"> The system is created in such a way that follows object oriented concept and therefore, it is easy for other programmers to add extra features as only one object/entity is required to extend the functionality of the application.

Use Case Diagram

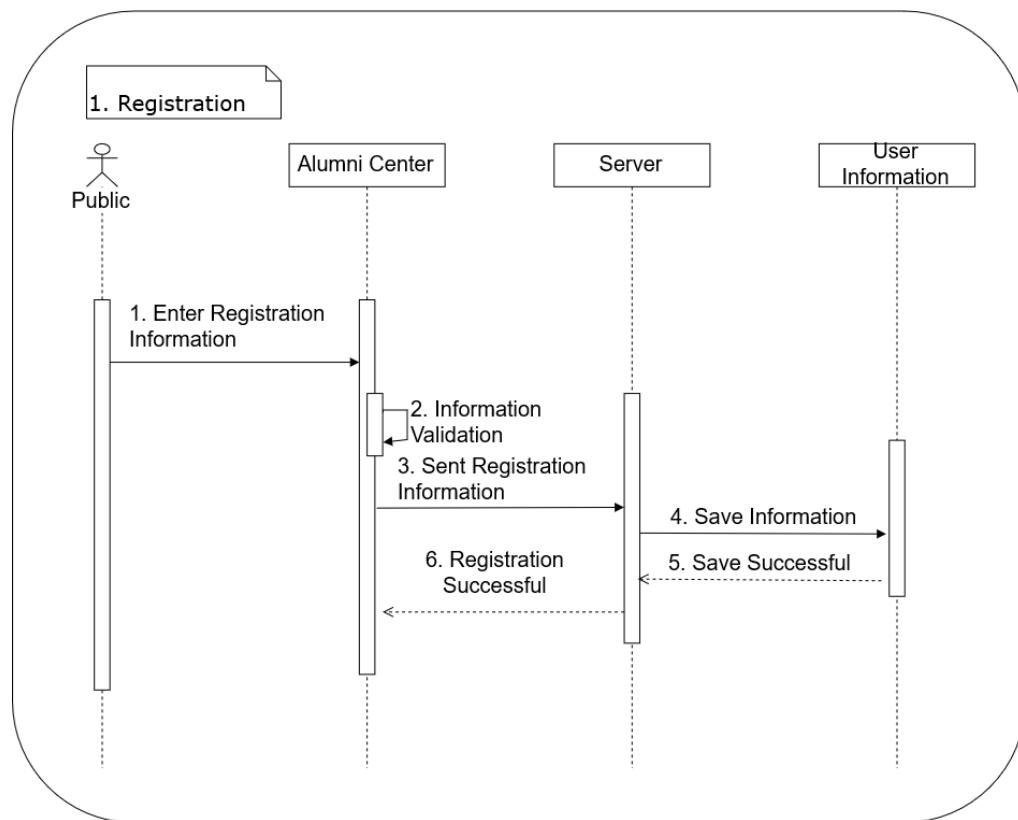
The software team identified use cases of the system which are the functions user can perform (Brandenburg). There are three categories of users, which are public users, logged-in users, and admin. Below are the use case diagrams for each category of user.

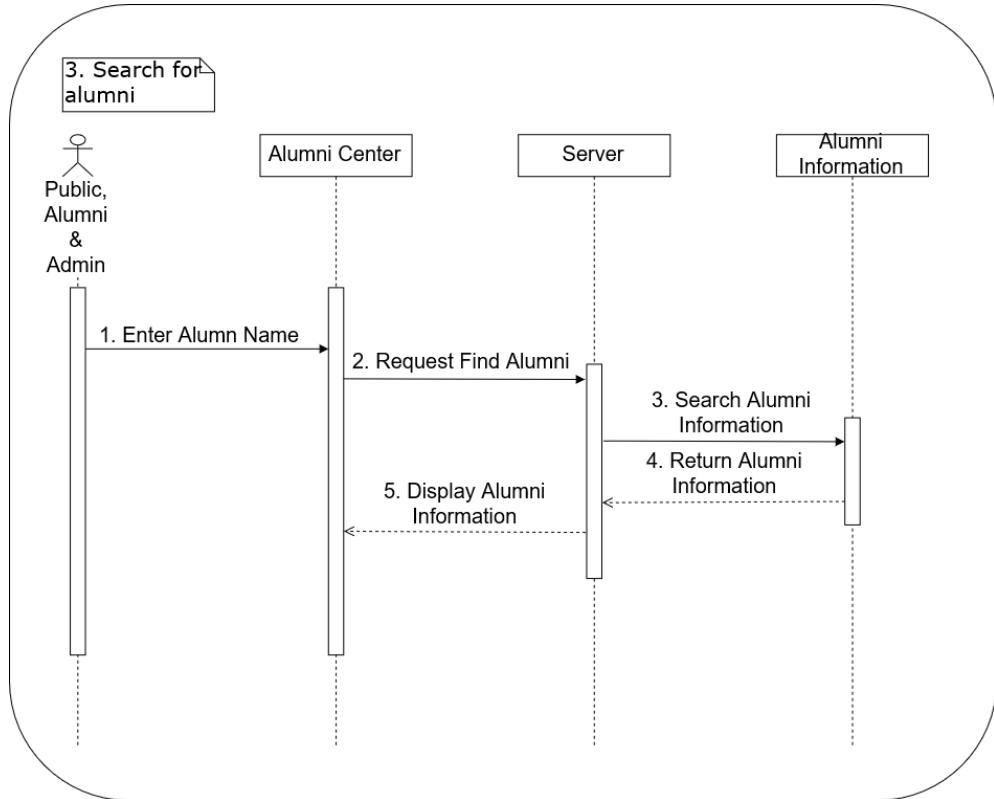
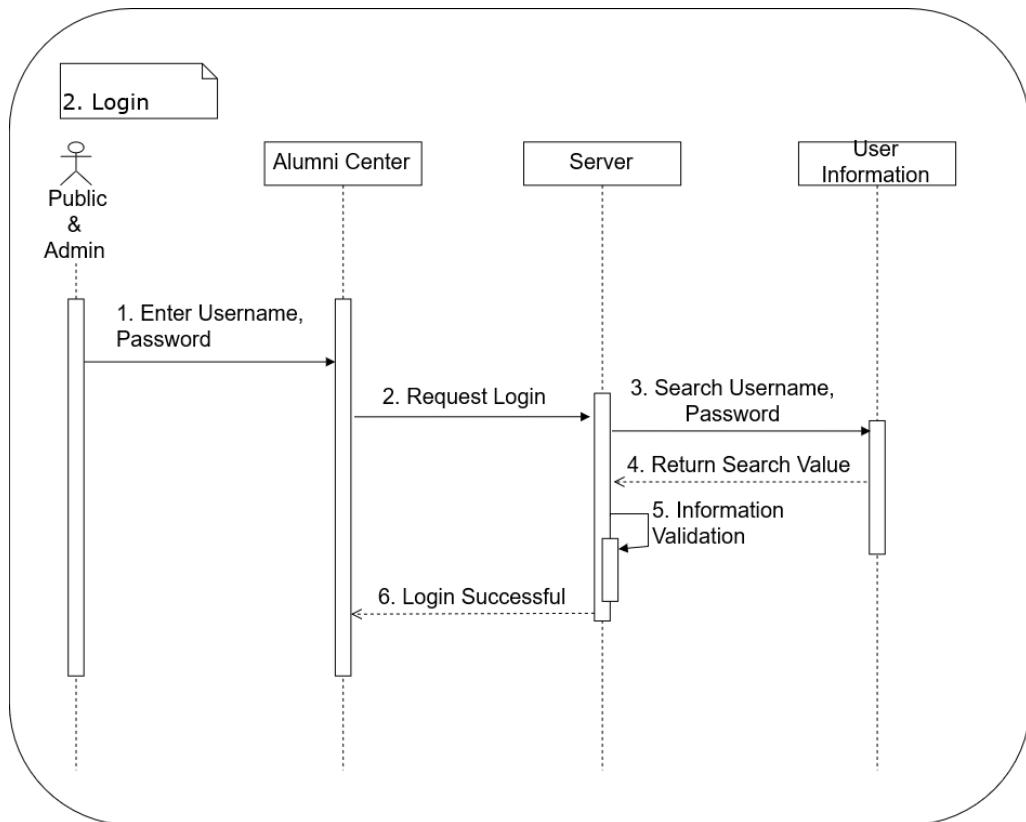


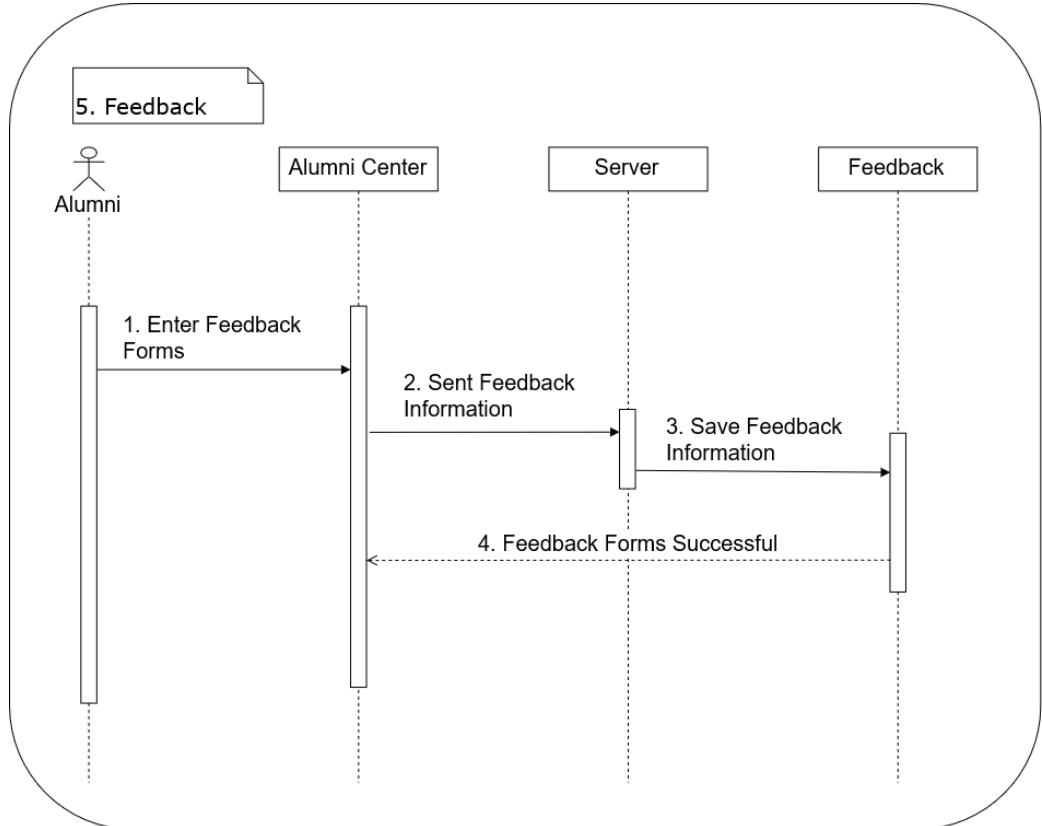
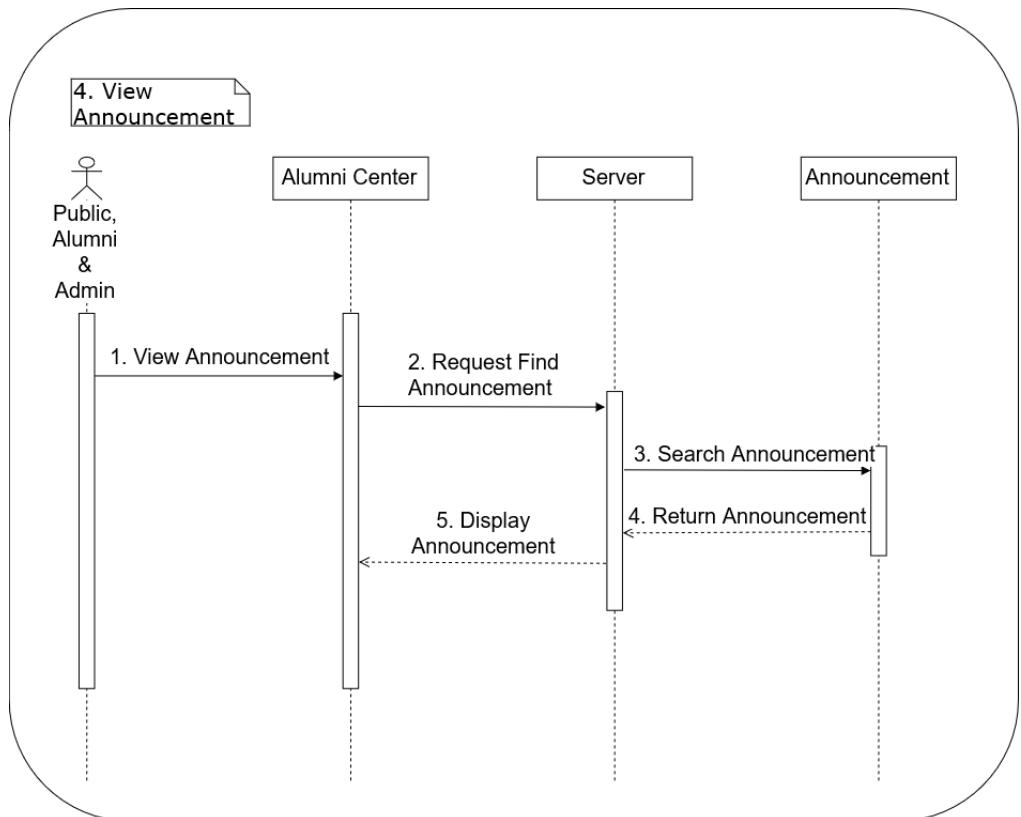


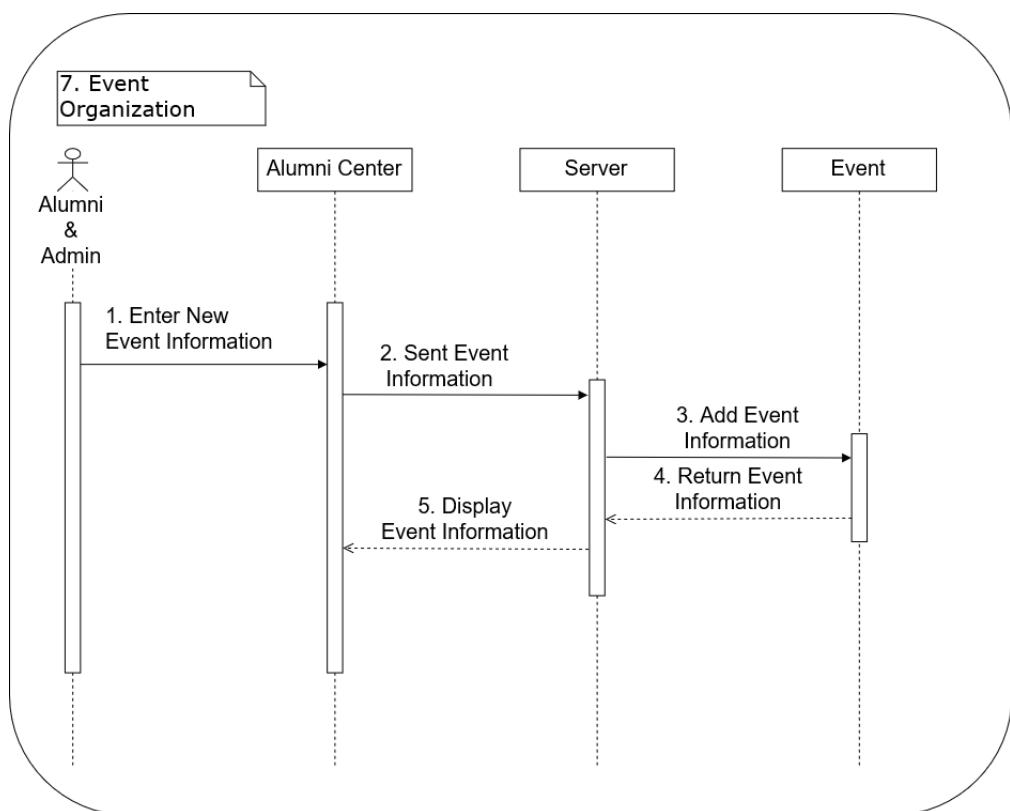
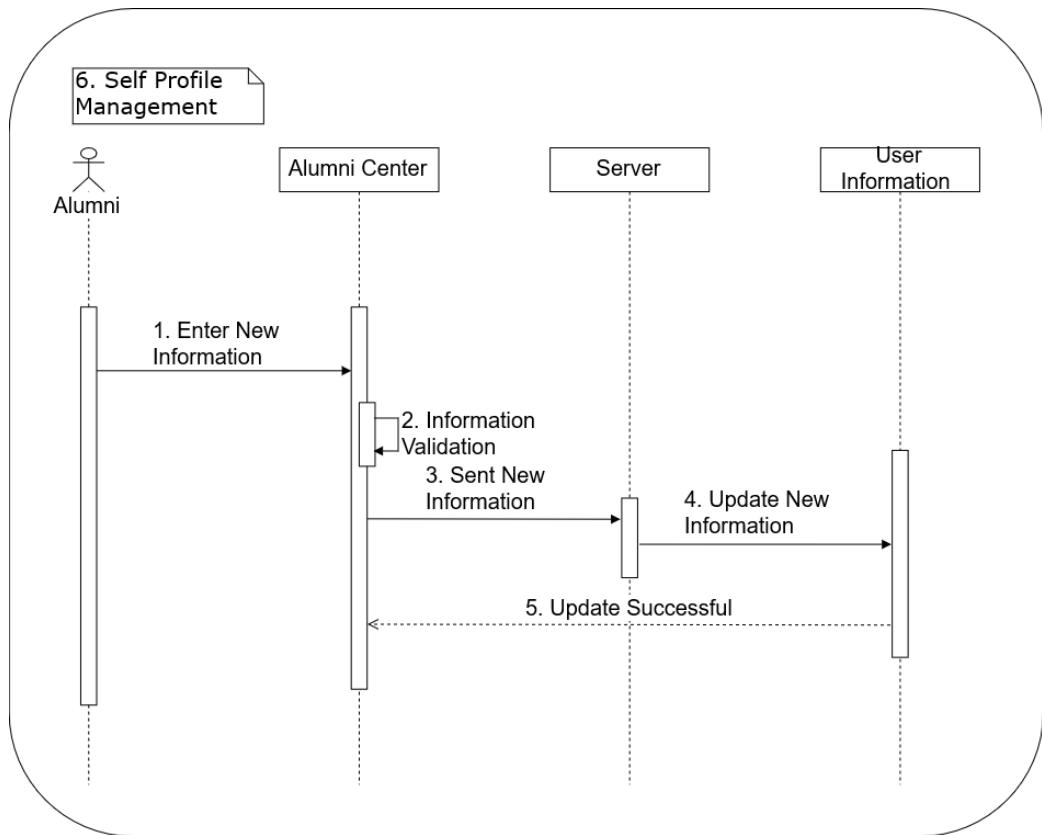
Sequence Diagram

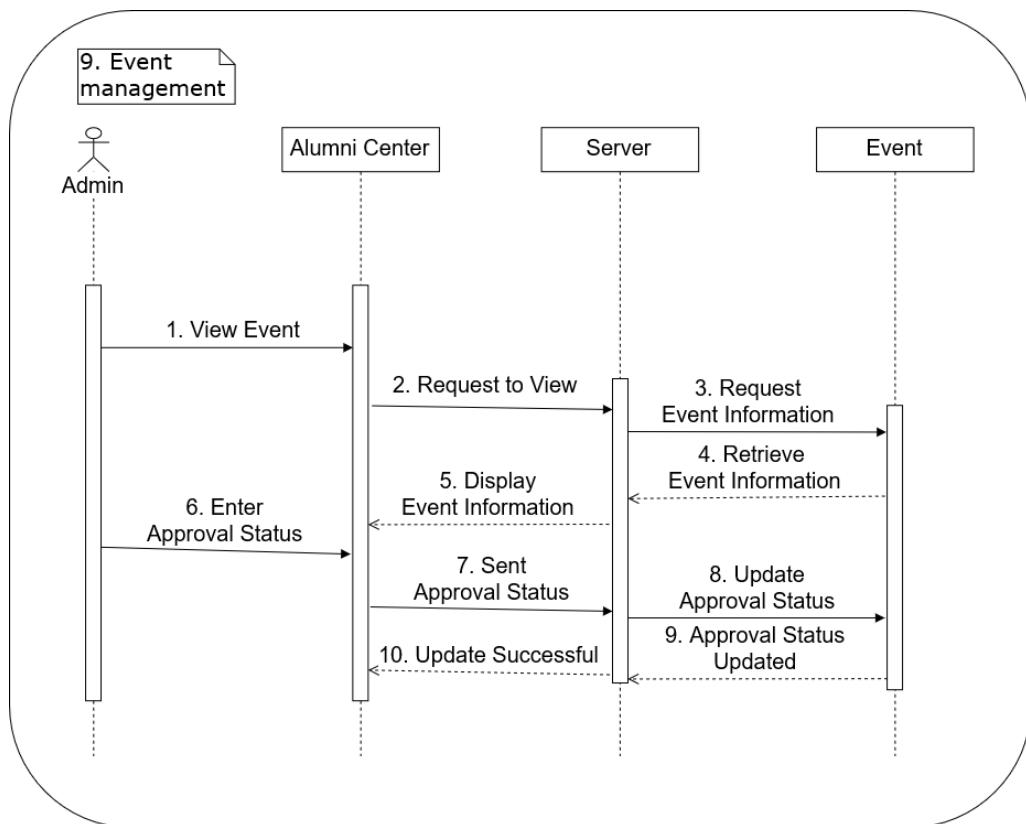
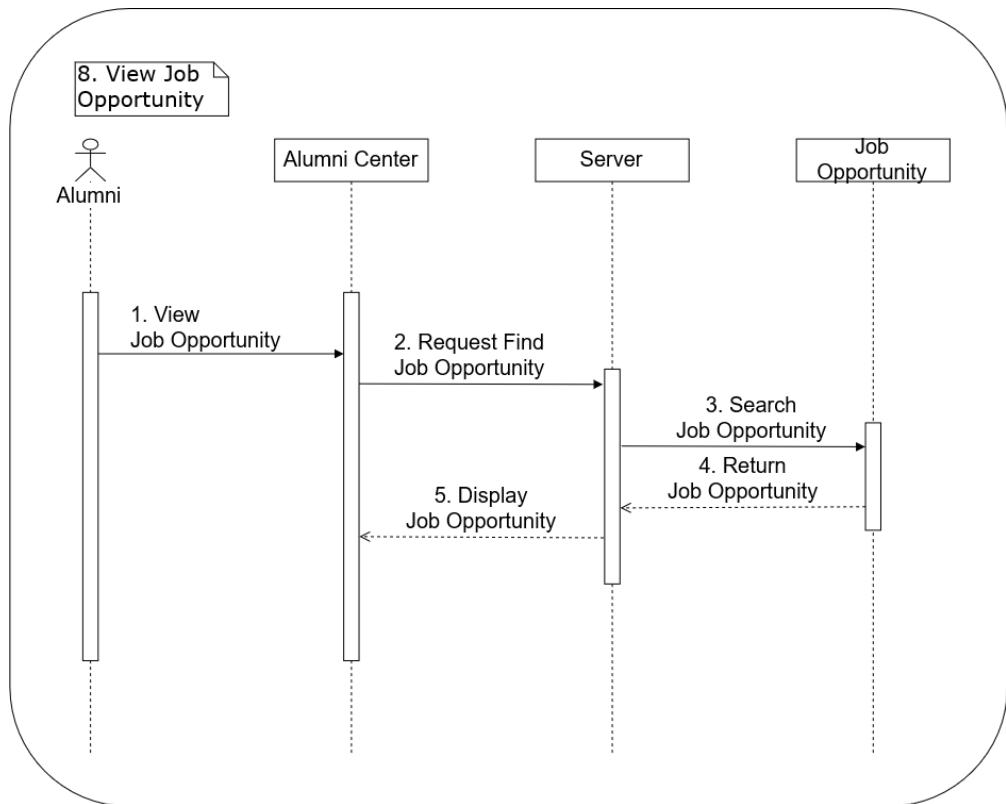
Below are the sequence diagrams correspond to use cases of the users. They illustrate the procedures of functions to be performed.

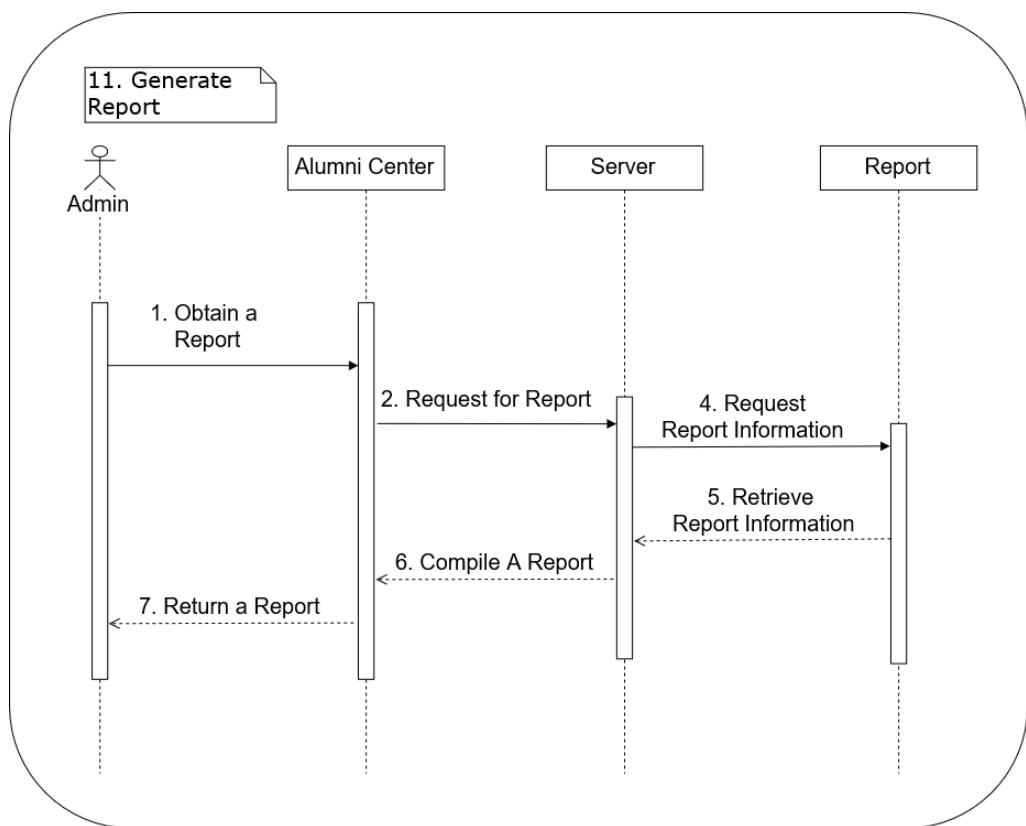
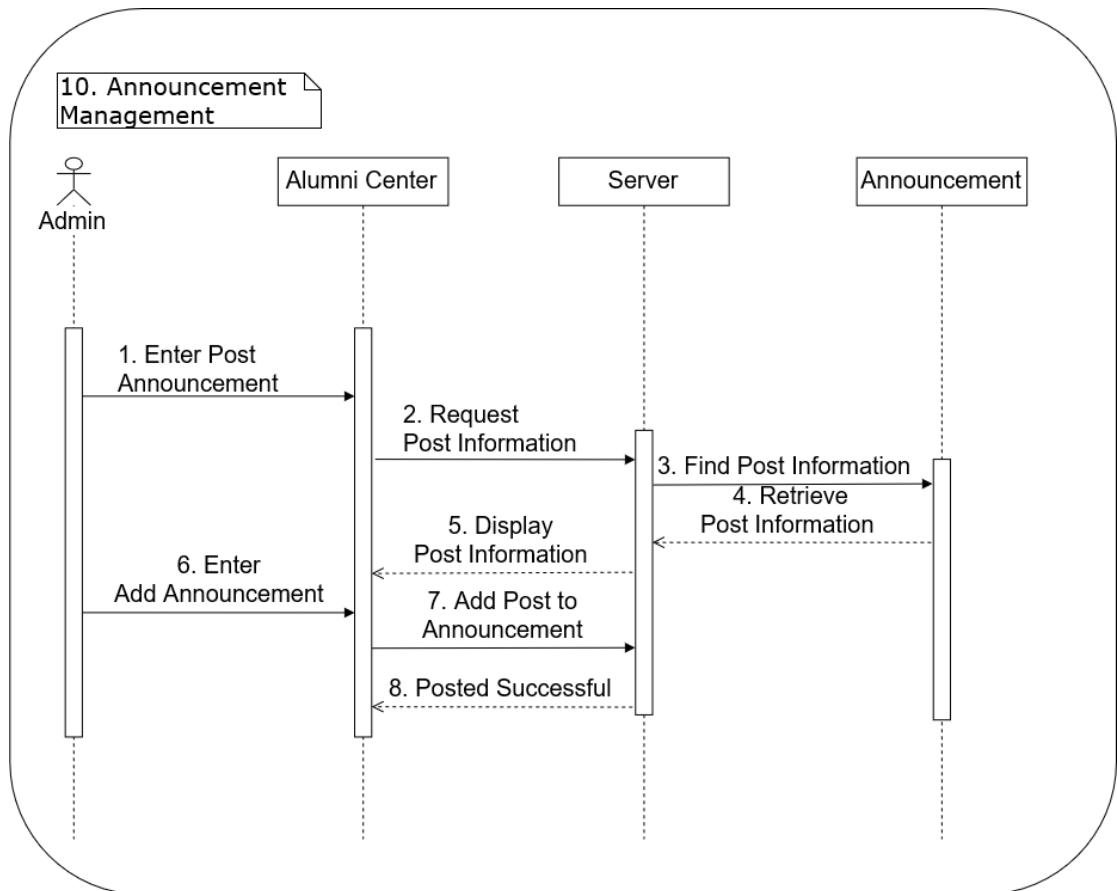


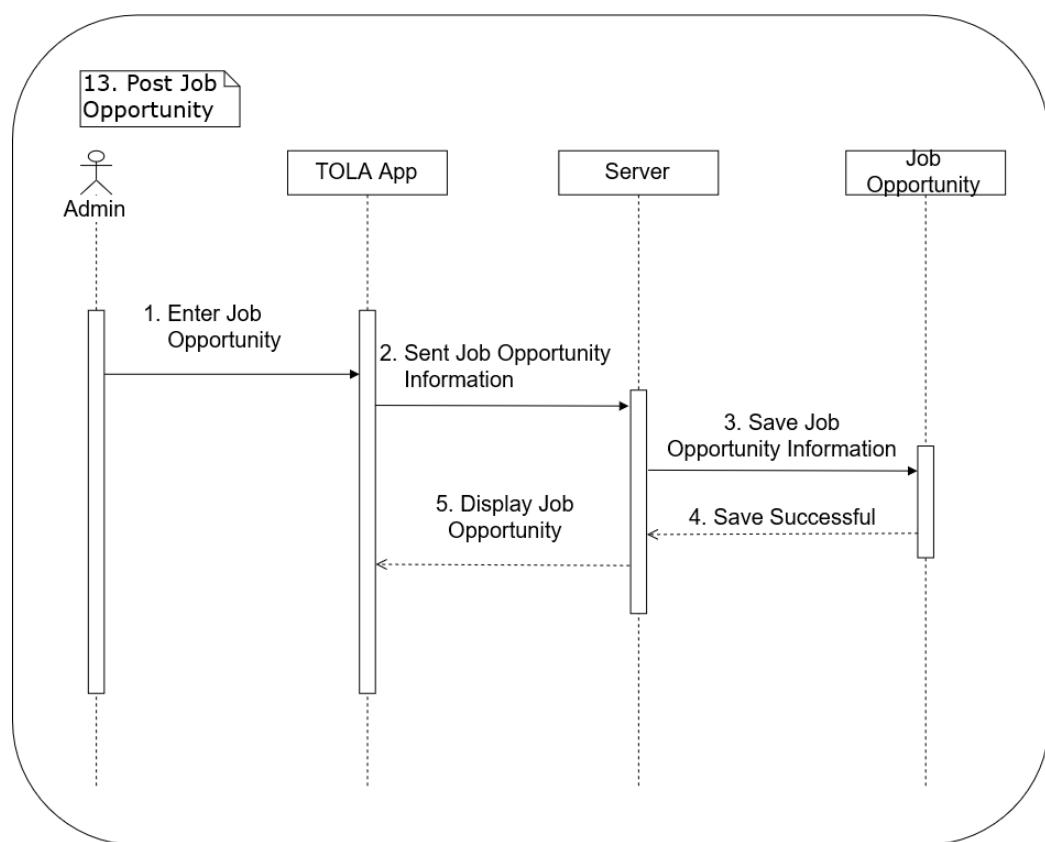
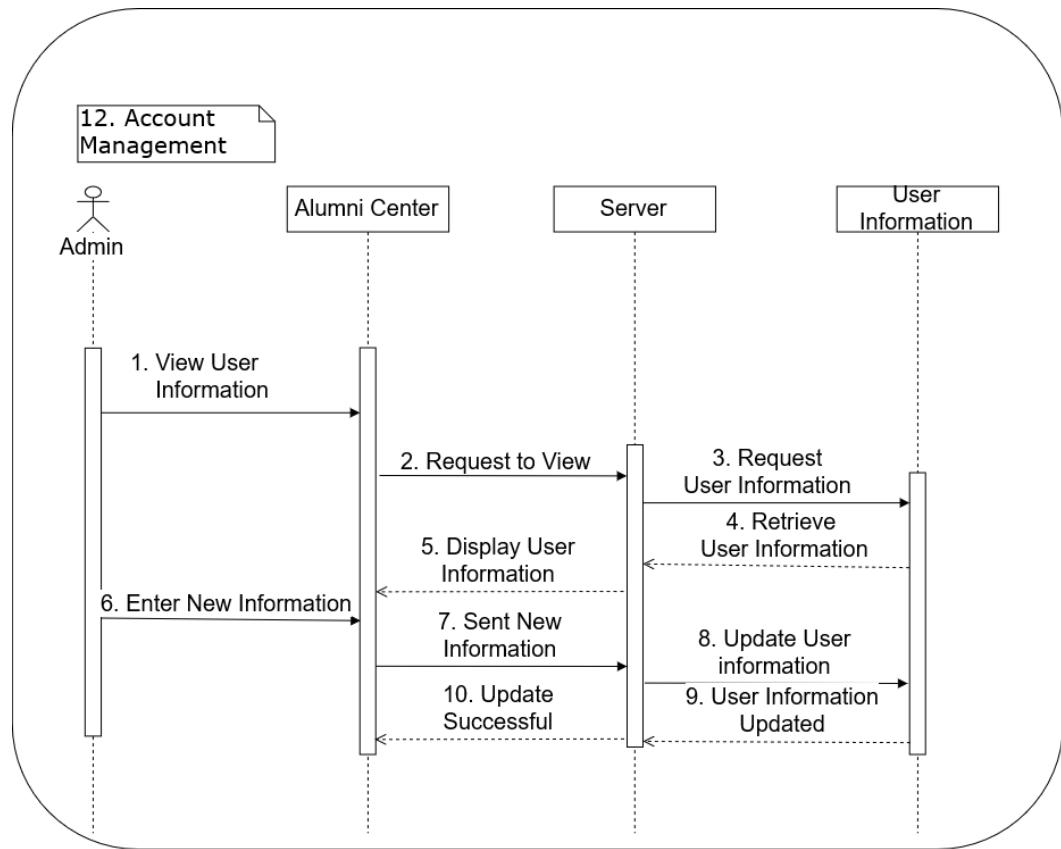




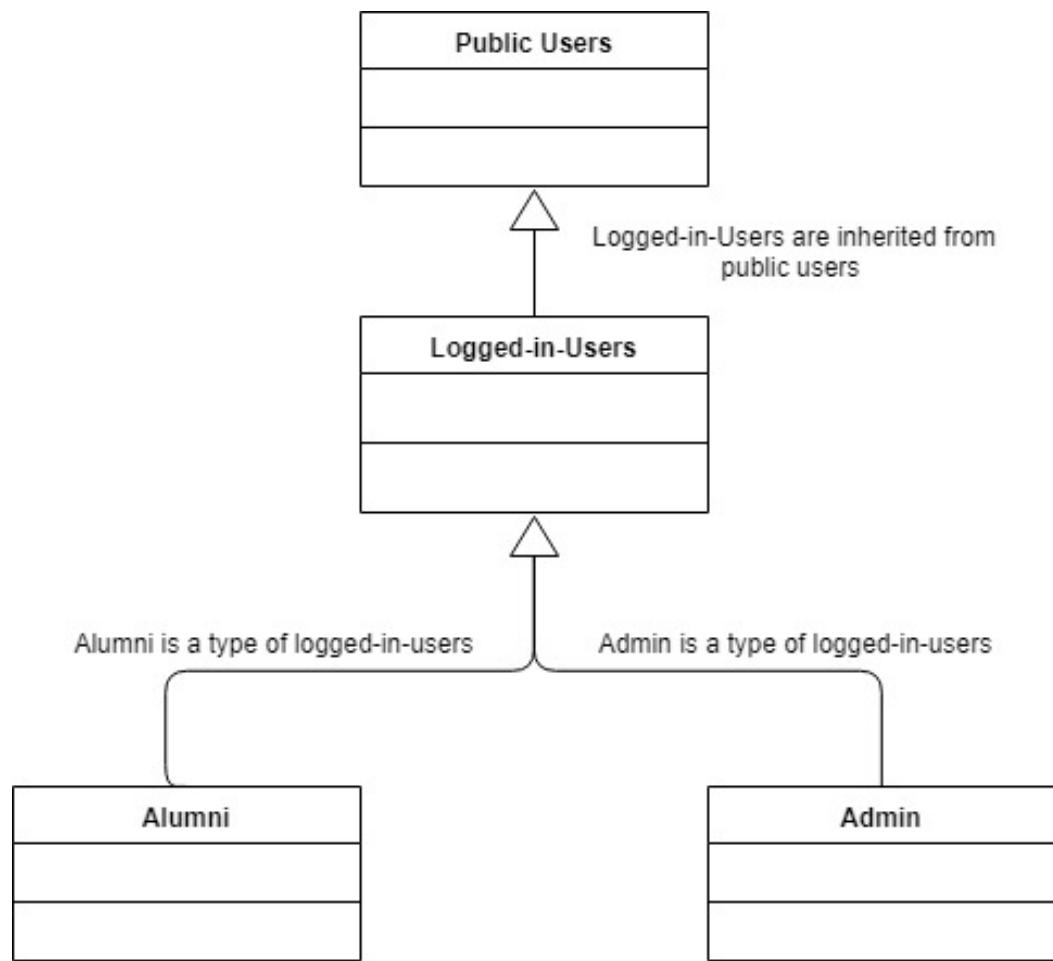




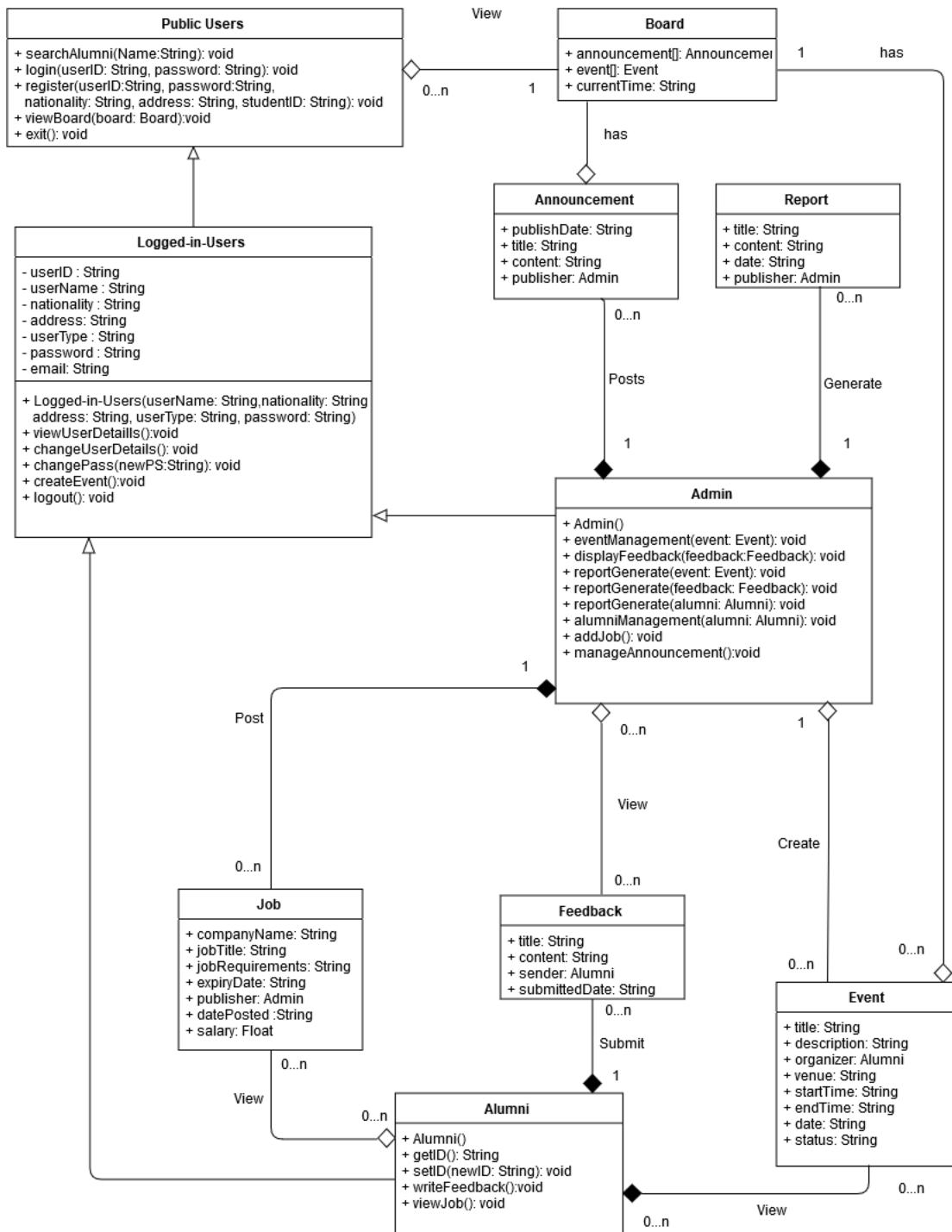




Class Diagram



Overview



Full Class Diagram

Class Responsibility Collaborator (CRC) Diagram

As an additional diagram, the software team also defined an CRC diagram to elaborate and eliminate ambiguity.

Class: Public Users	
Description: Public user is a class that describes the operation that can perform by guests or non-logged-in users.	
Responsibility	Collaborator
1. Search for alumni	Alumni
2. Login	
3. Register	
4. View board of announcement	Board

Class: Logged-in-Users	
Description: Logged-in User is a class that is used to describe the attributes of a group of users that have an account in the system and the operation that they can perform in the system.	
<i>*this class is Inherited from public users class</i>	
Responsibility	Collaborator
1. Define the userID and its type	
2. Define the userName and its type	
3. Define the nationality and its type	
4. Define the address and its type	
5. Define the email and its type	
6. Define the userType and its type	
7. Define the password and its type	
8. View profile	
9. Edit profile	
10. Change password	
11. Create events	Event
12. View events	Event
13. Logout	

Class: Alumni	
Description: Alumni is a class that is used to describe the attribute of a group of main targeted users of this system, which are the alumni of a university and the operations that can perform by them.	
<i>*this class is Inherited from public users class and logged-in users class</i>	
Responsibility	Collaborator
1. Write feedback and submit to the system.	Feedback
2. View the jobs list.	Job

Class: Admin	
Description: Admin is a class that is used to describe the attributes of a group of users which are the people in charge of the whole system and the functionality of them.	
<i>*this class is Inherited from public users class and logged-in users class</i>	
Responsibility	Collaborator
1. Manage the events by approving or declining the events that are created by alumni	Event
2. View feedback submitted by alumni	Feedback
3. Generate report regarding to the system statistics (list of alumni, list of events, and list of feedback).	Alumni, Event, Announcement
4. Manage the alumni account by changing their account details such as password.	Alumni
5. Create job in the job lists	Job
6. Manage the announcement by posting, editing, and deleting it.	Announcement

Class: Event	
Description: Event is a class that is used to describe the attributes of an event object that is created by either alumni or admin.	
Responsibility	Collaborator
1. Define the title of event and its type	
2. Define the description of event and its type	
3. Define the organizer of event and its type	Logged-in users
4. Define the venue of event and its type	
5. Define the start time of event and its type	
6. Define the end time of event and its type	
7. Define the date of event and its type	
8. Define the status of event and its type	

Class: Job	
Description: Job is a class that is used for defining the attributes of job object. Job object will aid the alumni in finding job.	
Responsibility	Collaborator
1. Define the company name of job and its type.	
2. Define the job title and its type.	
3. Define the job responsibility and its type	
4. Define the expiry date of the job and its type.	
5. Define the date of job posted and its type.	
6. Define the salary of the job and its type.	
7. Define the publisher of the job and its type.	Admin

Class: Feedback	
Description: Feedback is a class that is used to describe the attributes of feedback submitted by alumni.	
Responsibility	Collaborator
1. Define the title of feedback and its type.	
2. Define the content of feedback and its type.	
3. Define the sender of feedback and its type.	Alumni
4. Define the date submitted of feedback and its type.	

Class: Report	
Description: Report is a class that is used to define the attribute of the system generated report.	
Responsibility	Collaborator
1. Define the title of the report and its type.	
2. Define the content of the report and its type.	
3. Define the date of report generated and its type.	
4. Define the publisher and its type	Admin

Class: Board	
Description: Board is a class that is used to store the information of announcement or upcoming events and is used by all users for display purpose.	
Responsibility	Collaborator
1. Define a list of announcements that are created by admin and its type.	Announcement
2. Define a list of upcoming events that are created by either admin or alumni and its type.	Event
3. Define the current time and its type.	

Class: Announcement	
Description: Announcement is a class that is used to define the attributes of the announcement created by admin.	
Responsibility	Collaborator
1. Define the title of the announcement and its type.	
2. Define the content of the announcement and its type.	
3. Define the publish date of the announcement and its type.	
4. Define the publisher and its type	Admin

Design Model of ALUMNI CENTER (Part 2)

Refactoring Elements

In this part, our main focus is to **improve the design** of the foundation of the project, which is the **UML class diagram**. The UML class diagram describe what objects should exist and which function to be performed by the module, in other words, the class diagram is a simple representation of the application. By reading the class diagram, the developer should understand the behavior of the system and estimate the flow of program.

The main changes that have made to the class diagram is that we apply the principle of “**single class responsibility (SCR)**” in our design. According to the article “Principles of Object Oriented Design” that is written by Robert C.Martin, an American Software Engineering and one of the authors of “Agile Manifesto”, SCR means that for every single modules or function or class, they should serve only one functionality of the application. (C.Martin)

- In previous class diagram, both the admin and user classes need to carry out many **unrelated tasks** such as displayBoard(), viewEvent(), addJob() and etc.
- Obviously, the unrelated tasks that are implemented in the class will create **confusion** and make the next stage of development tougher. Therefore, a good design of class should be carried out and we decided to follow the popular rule of “single class single responsibility”.
- All the unrelated operations inside the alumni and admin class are removed and those operations are moved and carried out by other related classes.
- Changes made to improve the design is represented by the table below.
- The following table can be read as “**Function**” is moved from “**Originated class**” to “**Destination class**”.

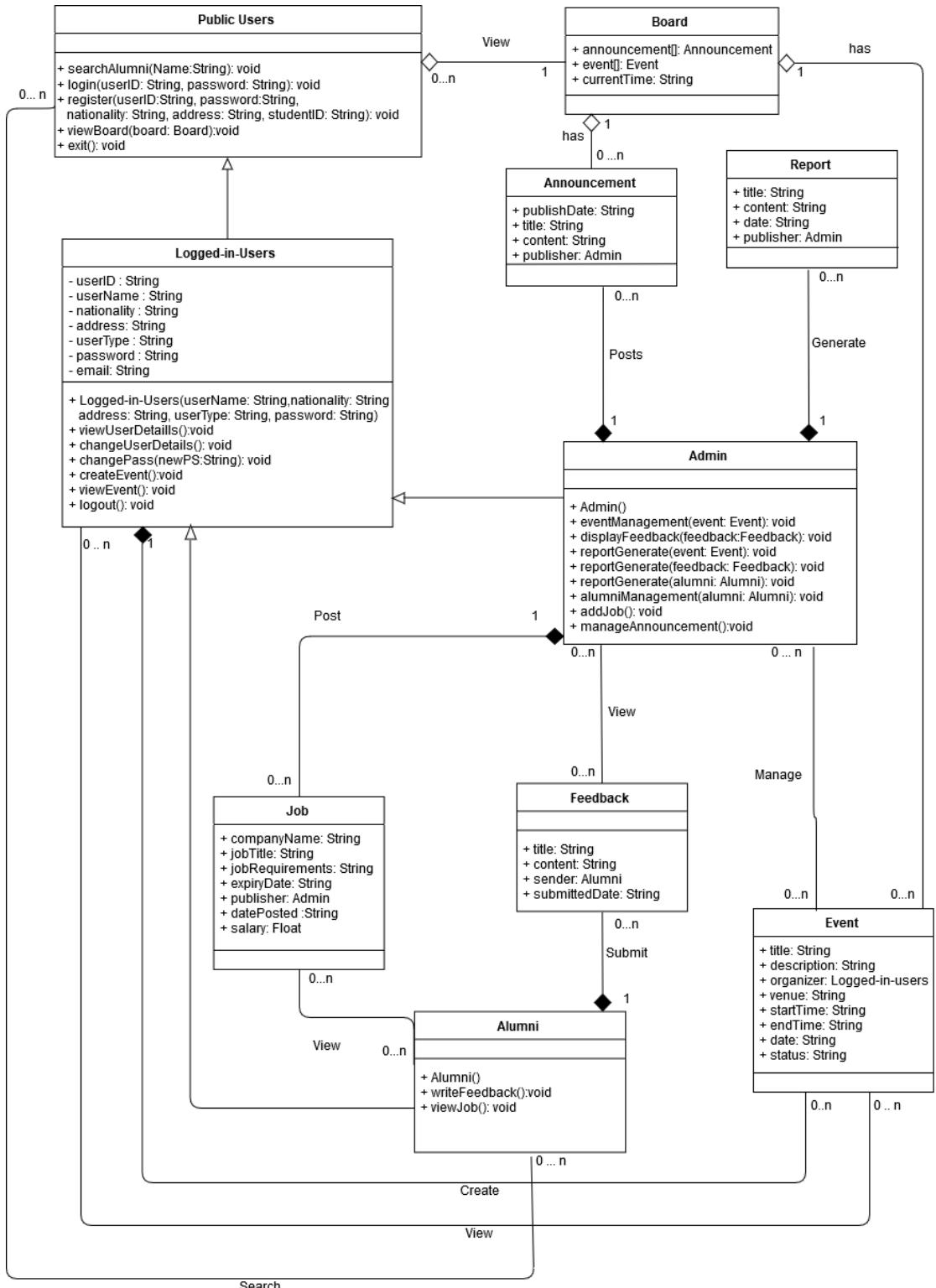
Function	Originated Class	Destination Class
viewBoard()	Public-Users	Board
createEvent()	Logged-in-Users	Event
viewEvent()	Logged-in-Users	Event
writeFeedback()	Alumni	Feedback
viewJob()	Alumni	Job
reportGenerate()	Admin	Report
(Three overloaded functions)		
displayFeedback()	Admin	Feedback
addJob()	Admin	Job
manageAnnouncement()	Admin	Announcement
eventManagement()	Admin	Event

On the other hand, other minor changes had taken place in order to improve the class design:

1. The function changePass() inside the class “Logged-in-Users” is removed as it will become a redundant when the function changeUserDetails() can perform the same operation.
2. New attributes are added.
 - The following table can be read as “**Attribute**” is added into the “**Class**” (Fakhroutdinov).

Attribute	Class
profilePic : Image	Alumni
description : String	Alumni
mobileNum : String	Alumni
address : String	Alumni
course : String	Alumni
yearOfGraduation: String	Alumni
jobID : String	Job
companyIcon : Image	Job
eventID : String	Event
eventPoster : Image	Event
birthday : String	Logged-in-Users
feedbackID : String	Feedback
reportID : String	Report
announcementID : String	Announcement

- Class diagram after refactoring:



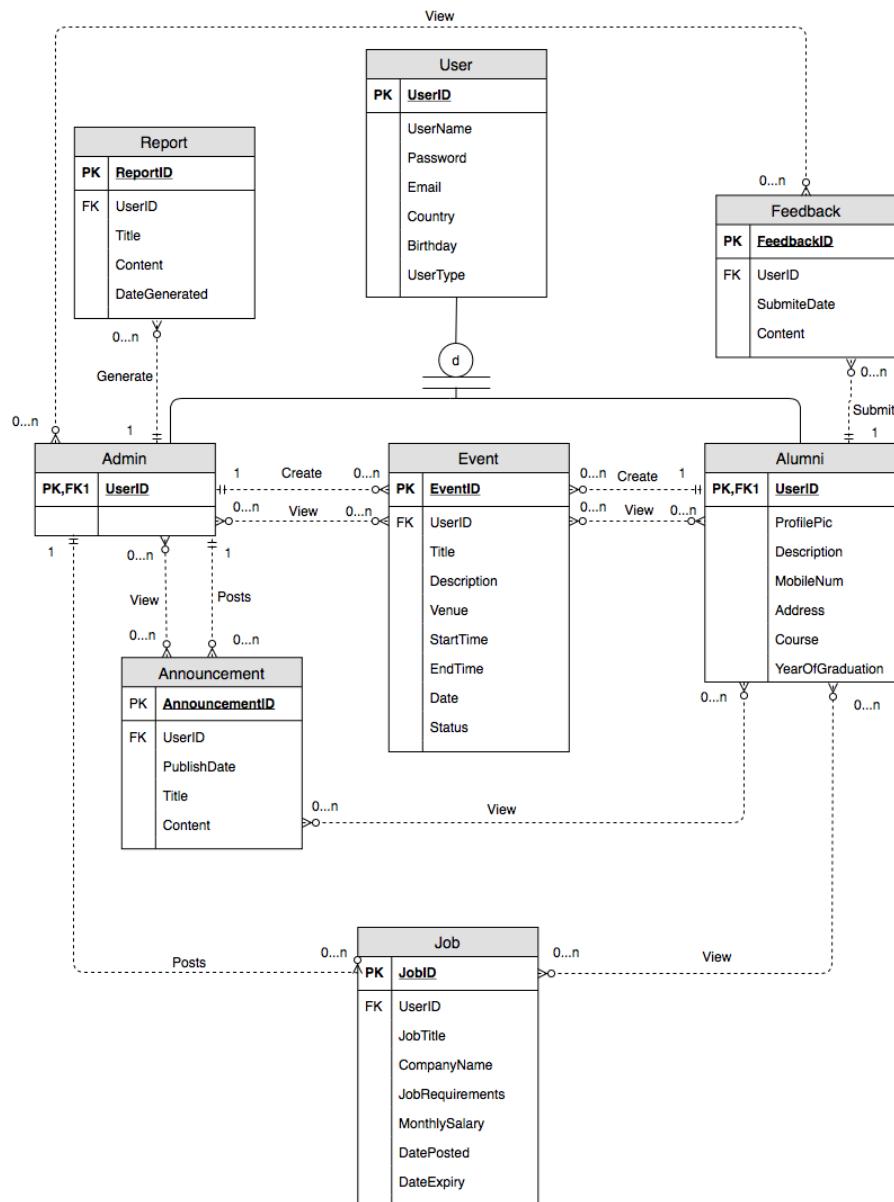
Data Design

The software team defined the data design using Entity-Relationship Diagram (ERD) and data dictionary with business rules. The business rules act as constraints and describe the relationships between the entities. With ERD, basic structure of the database system and format of storing data will be defined. Purpose of having data dictionary is to explain the entities and attributes in ERD.

The business rules are stated as follow:

1. A user must be either an alumnus or an admin.
2. An alumnus can view many jobs and a job can be viewed by many alumni.
3. An alumnus can view many announcements and an announcement can be viewed by many alumni.
4. An alumnus can create many events but an event can only be created by an alumnus.
5. An alumnus can submit many feedbacks but a feedback can only be submitted by one alumnus.
6. An admin can view many announcements and an announcement can be viewed by many admins.
7. An admin can post many announcements but an announcement can only be posted by one admin.
8. An admin can create many events but an event can only be created by one admin.
9. An admin can generate many reports while one report can only be generated by one admin.
10. An admin can post many jobs but a job can only be posted by one admin.
11. An admin can view many events and an event can be viewed by many admins.
12. An admin can view many feedbacks and a feedback can be viewed by many admins.

ERD



“M:M” relationships exist in the ERD is to convey the relationship between entity in a clearer way so that all developers can have the same understanding of the system

Data Dictionary

Table Name	Attribute Name	Contents	Data Type	Format	Constraint	Primary Key or Foreign Key	FK referenced table
User	UserID	Student ID that is possessed by the alumni while they haven't graduate from university and is used for login into the system OR admin ID that is used for login purpose.	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key	
	UserName	Name of Alumni as registered in school system OR name of Admin.	Varchar(255)	XXXXX XXXXX	NOT NULL		
	Password	User password that is used to login into the system	Varchar(255)	XXXXX XXXXX	NOT NULL		
	Email	Email address of users.	Varchar(255)	XXXXX @ XXXXX. XXX	NOT NULL UNIQUE		
	Country	Nationality of the users.	Varchar(255)	XXXXX XXXXX	NOT NULL		
	Birthday	Birthday of users.	Date	YYYY-MM-DD	NOT NULL		
	UserType	Type of users, it can be either normal users or admin.	Char	X (Either "N" for normal users or "A" for admin)	NOT NULL		
Admin	UserID	ID that is used for login purpose by an admin.	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key Foreign Key	Users

Table Name	Attribute Name	Contents	Data Type	Format	Constraint	Primary Key or Foreign Key	FK referenced table
Alumni	UserID	Student ID that is possessed by the alumni while they haven't graduate from university and is used for login into the system	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key Foreign Key	Users
	ProfilePic	Image that is used as profile picture of the alumni account.	Mediumblob (A data type that can store a large amount of data as binary)	XXXXX XXXXX			
	Description	Background information of the alumni that will display in their profile.	Varchar(255)	XXXXX XXXXX			
	MobileNum	Mobile number of alumni	Varchar(11)	XXXXX XXXXX			
	Address	Address of the location where alumni stay.	Varchar(255)	XXXXX XXXXX			
	Course	Course taken by alumni during university school life.	Varchar(100)	XXXXX XXXXX			
	YearOfGraduation	Year of graduation of alumni.	Year	YYYY (values allowed are from 1901 to 2155, and 0000)			

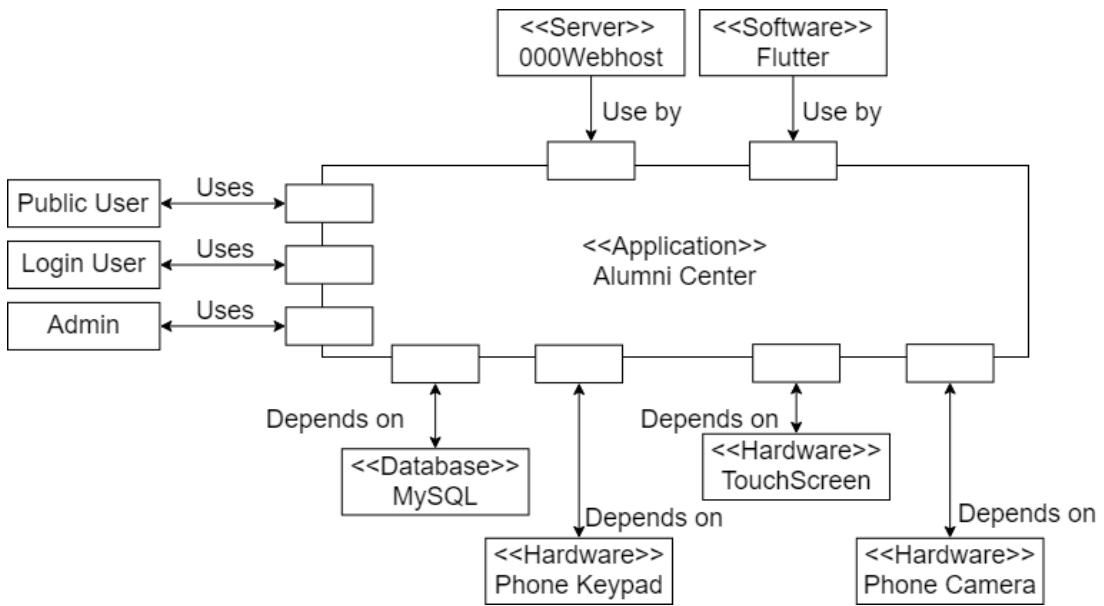
Table Name	Attribute Name	Contents	Data Type	Format	Constraint	Primary Key or Foreign Key	FK referenced table
Event	EventID	Unique ID that is used to identify a specific event.	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key	
	Title	Title of the event.	Varchar(50)	XXXXX XXXXX	NOT NULL		
	Description	Description of the event.	Text(5000)	XXXXX XXXXX	NOT NULL		
	UserID	ID of the alumni or admin that created the event (Event Organizer)	Varchar(10)	XXXXX XXXXX	NOT NULL	Foreign Key	Users
	Venue	Venue where the event will be organized	Varchar(255)	XXXXX XXXXX	NOT NULL		
	StartTime	The start time of the event	Time	HH:MM:SS	NOT NULL		
	EndTime	The end time of the event	Time	HH:MM:SS	NOT NULL		
	Date	The date when the event will occur.	Date	YYYY-MM-DD	NOT NULL		
	Status	Status of the event (It can be either "Pending", "Approved" or "Declined".)	Char	X (Either "P" for pending, "A" for approval or "D" for declined.)	NOT NULL		
	EventPoster	Image that describe the event.	Mediumblob	XXXXX XXXXX	NOT NULL		

Table Name	Attribute Name	Contents	Data Type	Format	Constraint	Primary Key or Foreign Key	FK referenced table
Feedback	FeedbackID	ID that is used to identify a specific feedback.	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key	
	Content	Content of the feedback.	Text(5000)	XXXXX XXXXX	NOT NULL		
	UserID	ID of alumni that send the feedback (Feedback Sender)	Varchar(10)	XXXXX XXXXX	NOT NULL	Foreign Key	Alumni
	SubmittedDate	Date of feedback submission.	Date	YYYY-MM-DD	NOT NULL		

Table Name	Attribute Name	Contents	Data Type	Format	Constraint	Primary Key or Foreign Key	FK referenced table
Job	JobID	ID that is used to identity a specific job.	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key	
	CompanyName	Company that offers the job opportunity.	Varchar(50)	XXXXX XXXXX	NOT NULL		
	JobTitle	Title of the job.	Varchar(50)	XXXXX XXXXX	NOT NULL		
	JobRequirements	Requirements of the job.	Varchar(255)	XXXXX XXXXX	NOT NULL		
	UserID	ID of admin that publishes the job information (Job Publisher)	Varchar(10)	XXXXX XXXXX	NOT NULL	Foreign Key	Admin
	MonthlySalary	Monthly salary offers by the job.	Decimal(8,2)	99999.99			
	DatePosted	The date of job posted by the admin.	Date	YYYY-MM-DD	NOT NULL		
	DateExpiry	The expiry date of the job.	Date	YYYY-MM-DD	NOT NULL		
	CompanyIcon	Image that symbolizes the company	Mediumblob	XXXXX XXXXX	NOT NULL		

Table Name	Attribute Name	Contents	Data Type	Format	Constraint	Primary Key or Foreign Key	FK referenced table
Announcement	AnnouncementID	ID that is used to identity a specific announcement.	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key	
	PublishDate	The date of announcement posted by admin.	Date	YYYY-MM-DD	NOT NULL		
	Title	Title of the announcement	Varchar(50)	XXXXX XXXXX	NOT NULL		
	Content	Content of the announcement	Text(5000)	XXXXX XXXXX	NOT NULL		
	UserID	Admin ID that posted the announcement. (Announcement publisher)	Varchar(10)	XXXXX XXXXX	NOT NULL	Foreign Key	Admin
Report	ReportID	ID that is used to identity a specific report.	Varchar(10)	XXXXX XXXXX	NOT NULL	Primary Key	
	Title	Title of the report generated.	Varchar(50)	XXXXX XXXXX	NOT NULL		
	Content	Content of the report.	Text(5000)	XXXXX XXXXX	NOT NULL		
	DateGenerated	Date in which the report is generated.	Date	YYYY-MM-DD	NOT NULL		
	UserID	Admin ID that created the report. (Report generator)	Varchar(10)	XXXXX XXXXX	NOT NULL	Foreign Key	Admin

Architecture Design



Architecture Context Diagram

Elaborations (Pressman, 2010) (S.Pressman, 2014) (Sommerville, 2015):

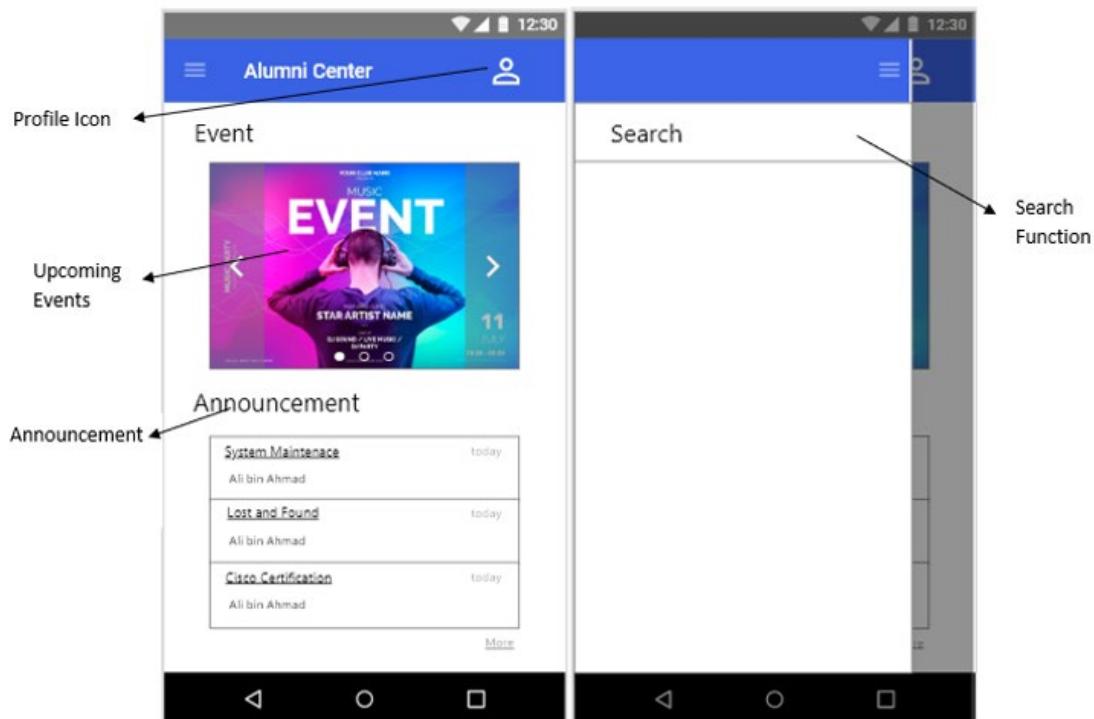
- **Target system** (centre) = the main system
- **Superordinate system** (top) = those system that use the target system as part of some higher level processing scheme
- **Subordinate system** (bottom) = those system that used by the target system and provide data or processing that are necessary to complete target system functionality
- **Peer-level system** (right) = those systems that produce or consume information from the target system
- **Actor** (left) = entities that interact with the system by producing or consuming information that is necessary for requisite processing

Interface Design

There are three crucial interface elements in interface design model (Software Engineering Design, 2017) (Pressman, 2010) (S.Pressman, 2014) (Sommerville, 2015), namely:

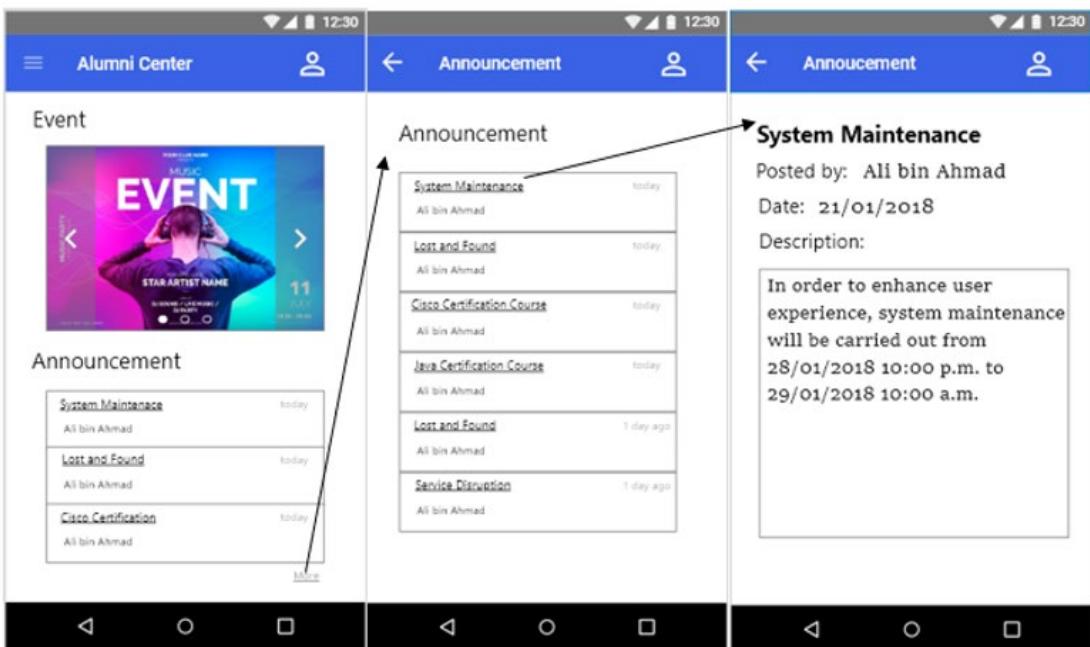
- User Interface (UI)
- External interfaces to other systems, devices, networks or other producers or consumers of information
- Internal interfaces between various design components

UI Design



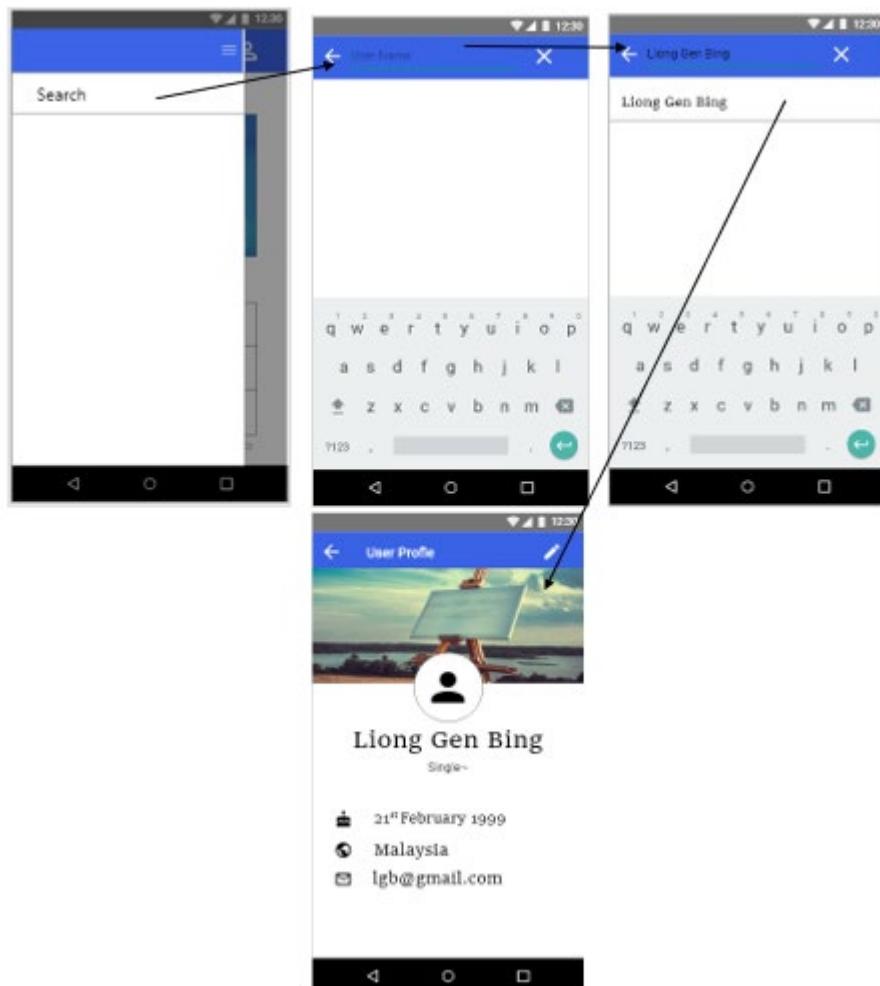
Main Interface (Public User)

Screenshots above shows the main interface of the application. In this interface, can perform several operations, such as view upcoming events, see announcement, log in and search for other users.

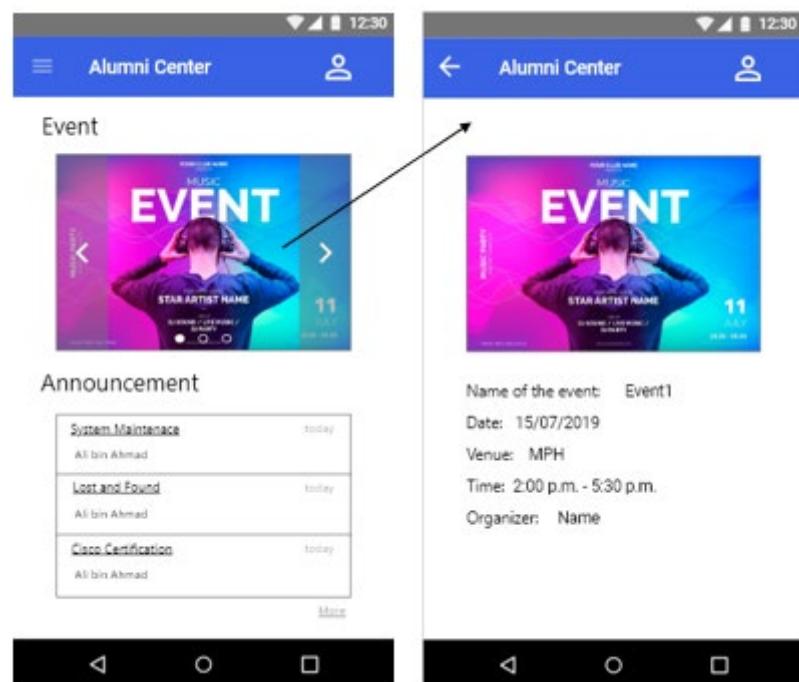


View Announcement (Public User)

Screenshots above are the interfaces expected to be seen by the public users. After clicking “More” in the main interface, it will lead user to announcement board. Here, each announcement’s title is a clickable link which allows user to see the details of the announcement.

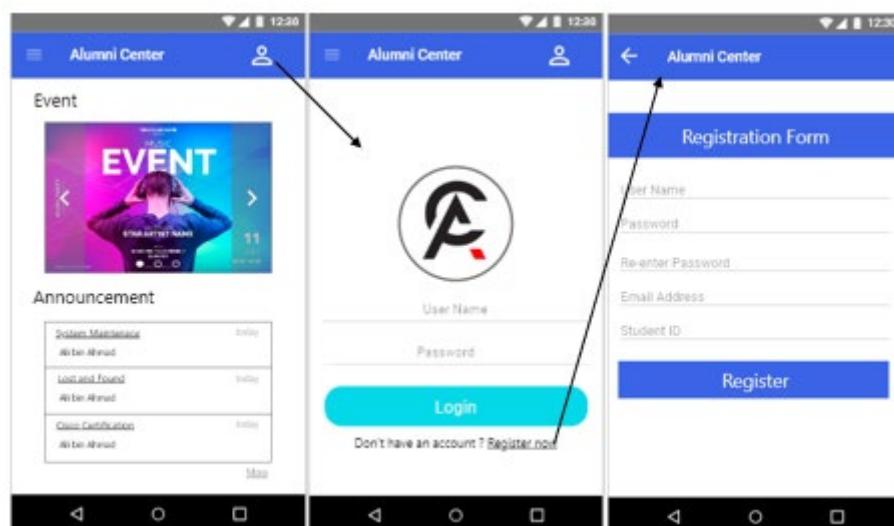


Screenshots above illustrate search operation is carried out by public user. Public user will have to click “search” and enter keyword. The result will be shown and user able to see the user profile.



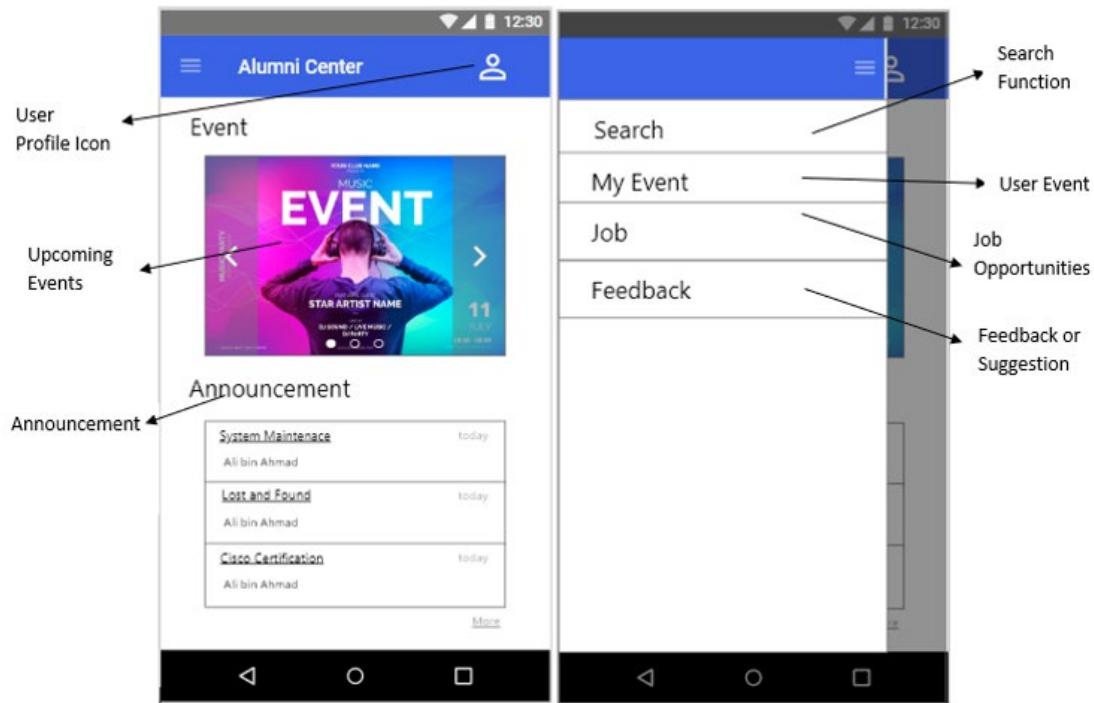
View Event (Public User)

Public user can also view the details of event by clicking it in the main interface



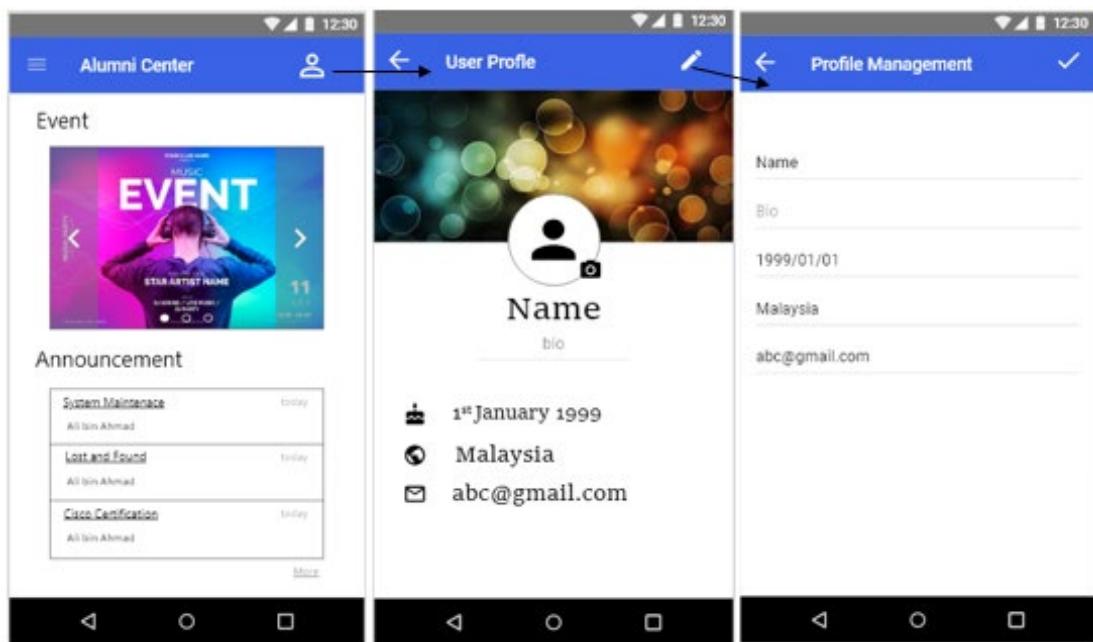
Login/Registration (Public User)

If a public user has an account, he will need to profile icon and the application will lead him to login page. Here, credentials are required in order to enter the application as a user. In contrast, he will need to proceed to registration in order to create an account if he does not have one.



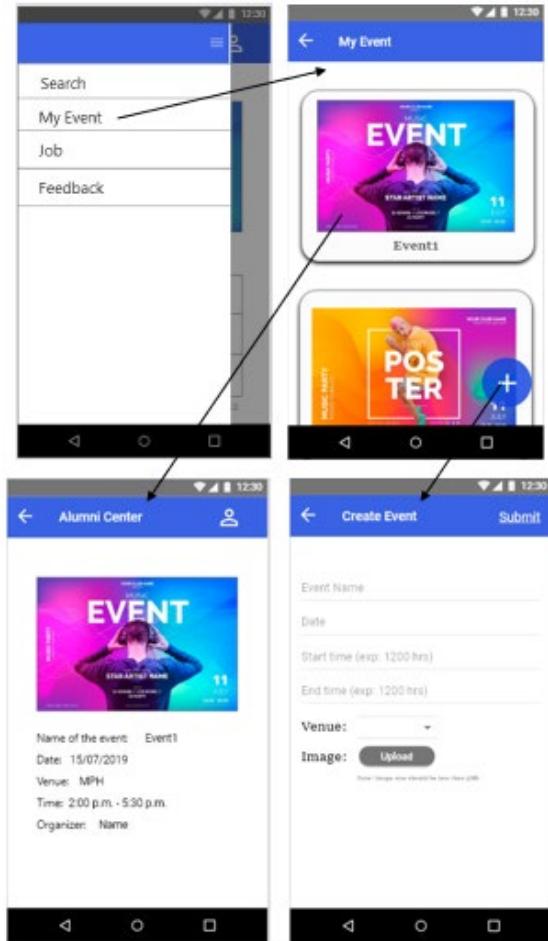
Main Interface (Logged-in User)

Screenshot above is the main interface of a logged-in user. Logged-in user is an extend of public user which means he can perform the same operations. In addition, he is granted the privilege to manage his own profile. Besides, he also able to view his events as well as propose one. Finally, he can view the job opportunities posted by admins and submit feedback to the institution.



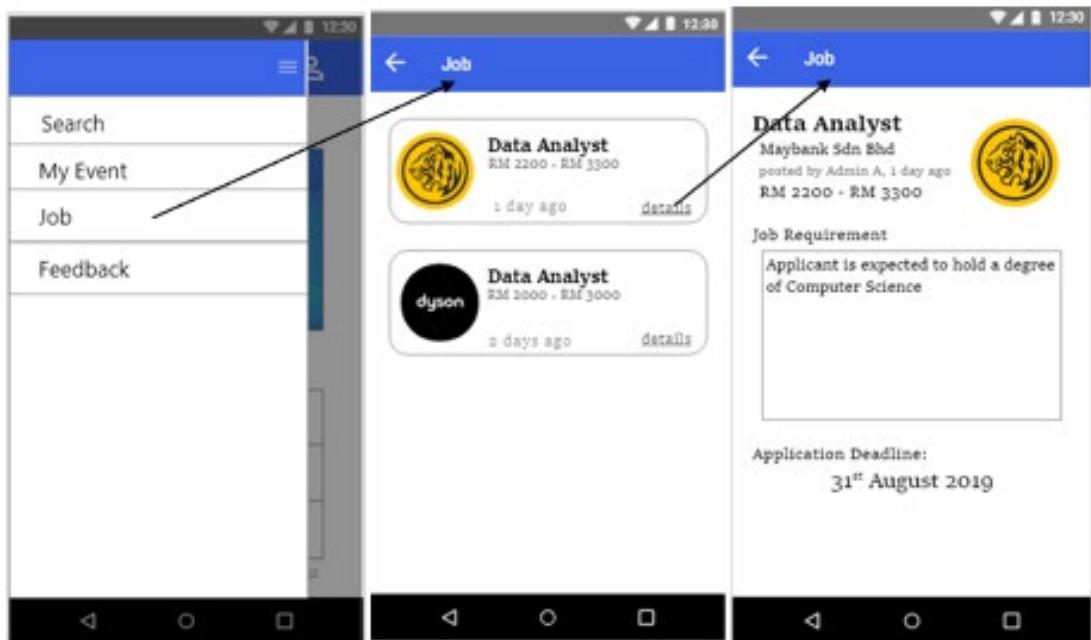
Profile Management (Logged-in User)

Logged-in user may manage his profile by click the profile icon on the top-right corner. It will then lead to his profile interface which is shown to public. He can edit his personal details by clicking the pencil icon at top-right corner of the interface.



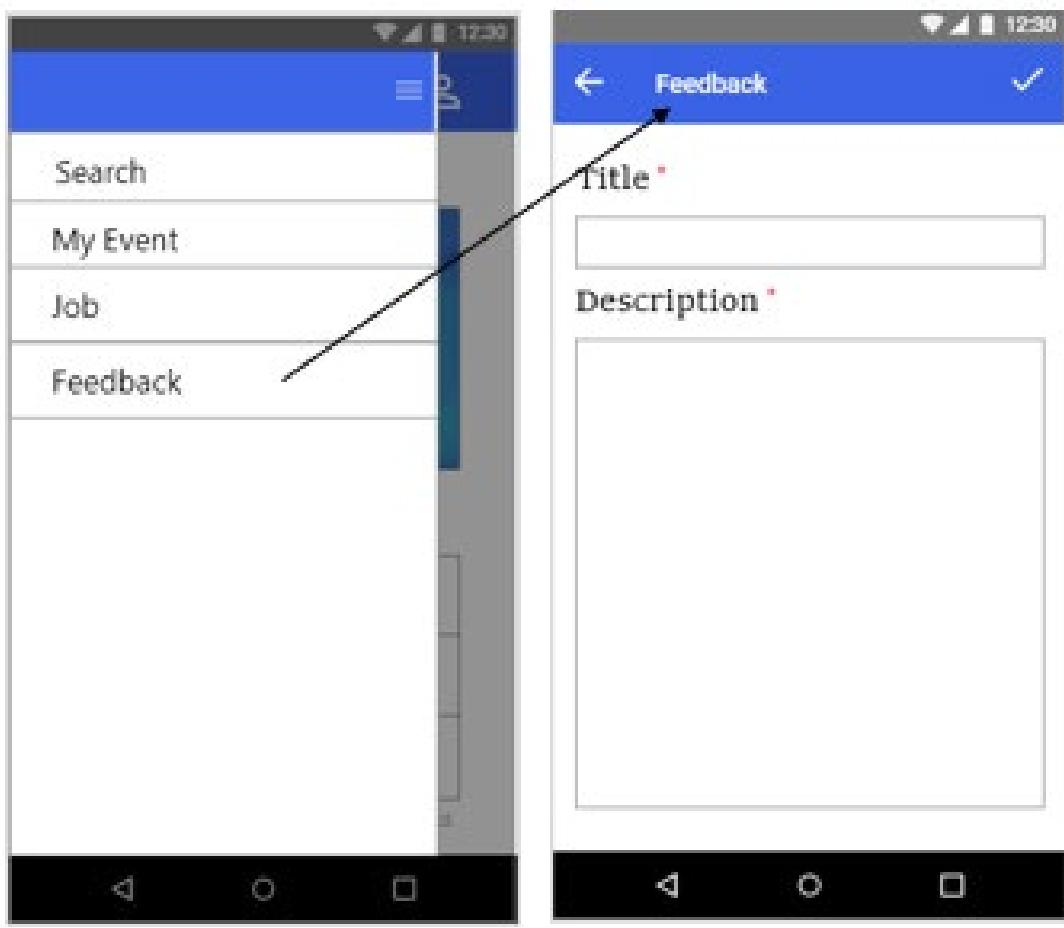
View User Event (Logged-in User)

A logged-in can view his own event by clicking “My Event”. It leads the user to the list of events he created. By clicking them, it shows the details of that particular event. Other-than-that, he can submit a new event application through the icon at the bottom-right corner.



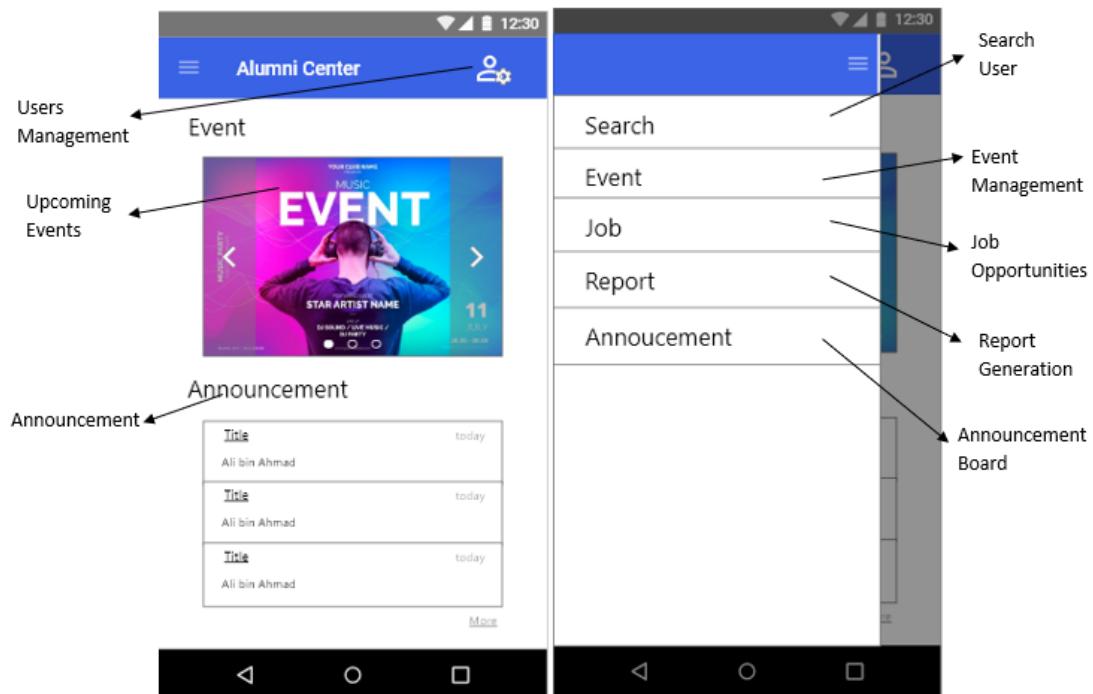
View Job Opportunities (Logged-in User)

Logged-in user can see the posted job opportunities through “Job”. He may click the view the detailed information of a particular job by clicking its “details” button.



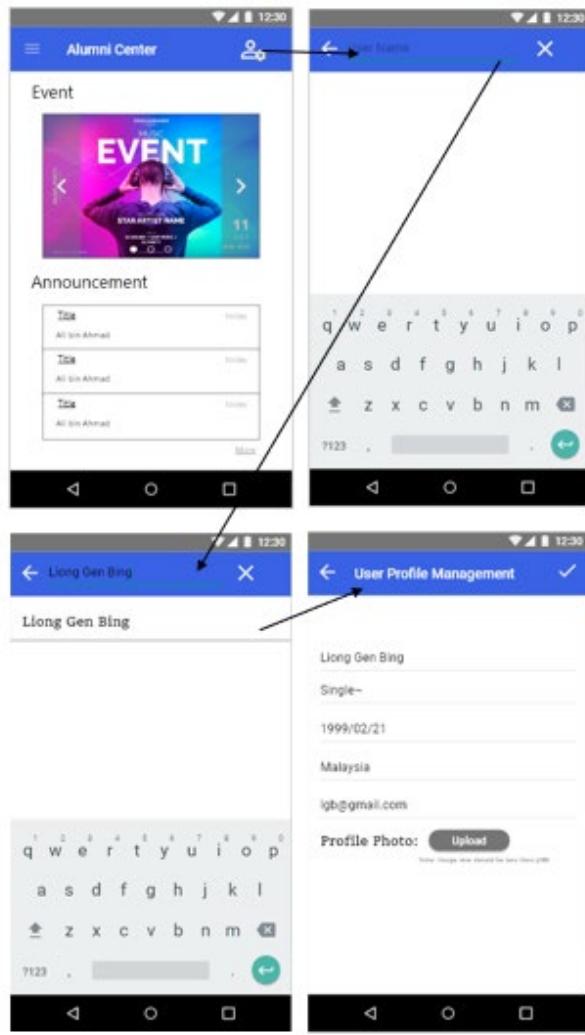
Submit Feedback or Suggestion (Logged-in User)

Logged-in user able to voice out his relevant opinion to the university through “Feedback” function. It requires the user provide the title and detailed explanation regarding his feedback.



Main Interface (Admin)

Admin is also an extend of public user. In contrast to logged-in user, an admin has higher privilege in terms of function granted. The additional operations are users profile management, approving event applications, view system-generated reports, post new announcement.



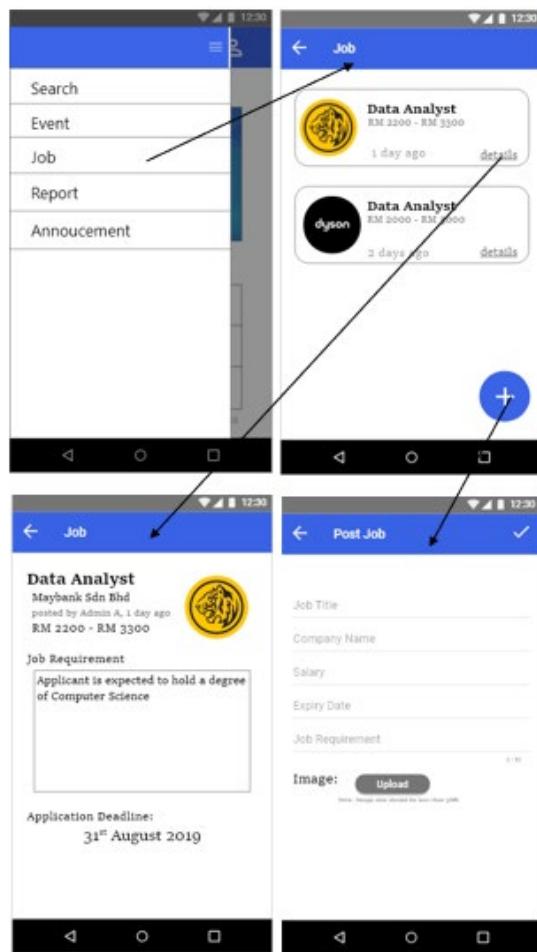
Users Profile Management (Admin)

In contrast to logged-in user, the top-right corner of main interface is replaced with user management icon. It allows admin to change the details of a specific user.



Event Management (Admin)

In addition of viewing approved events, admin also granted the privilege of approving an event through “Pending” page.



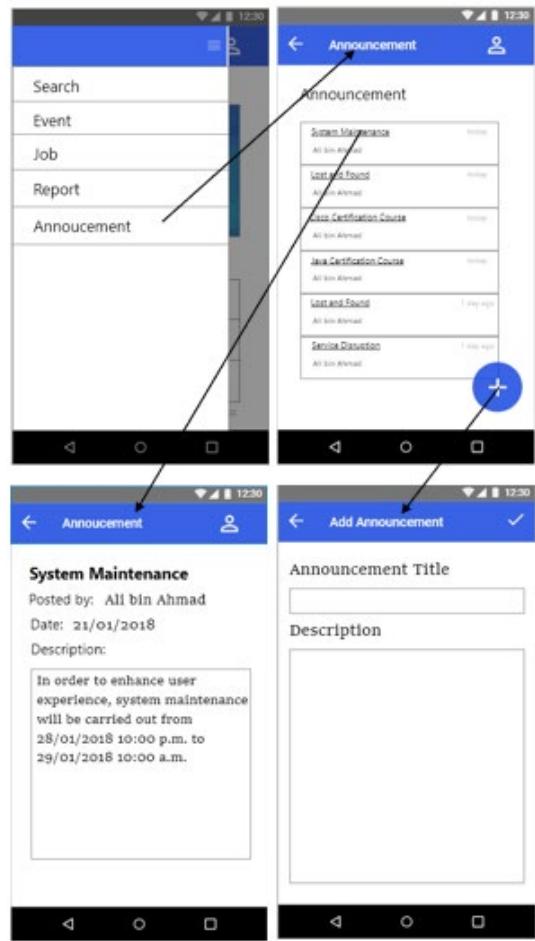
Post Job Opportunities (Admin)

Similar to logged-in user, admin can also view the list of job opportunities posted and their details. However, he can add a new job opportunity to the list through the add icon at bottom-right corner.



View System-Generated Reports (Admin)

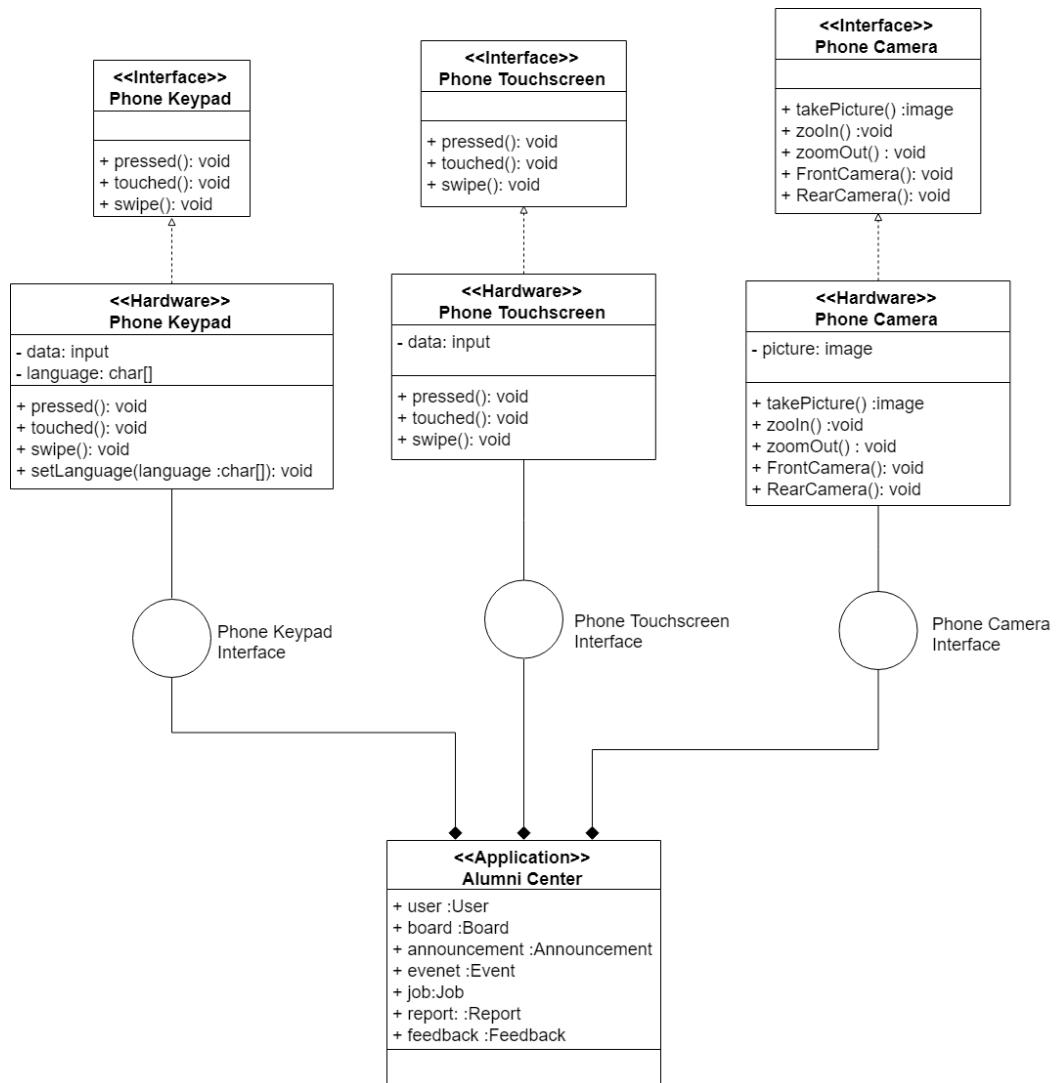
As an admin, he can view the reports that summaries the data collected by system and stored in the database. Three types of reports can be generated, namely user, job, and event which are separated in tabs.



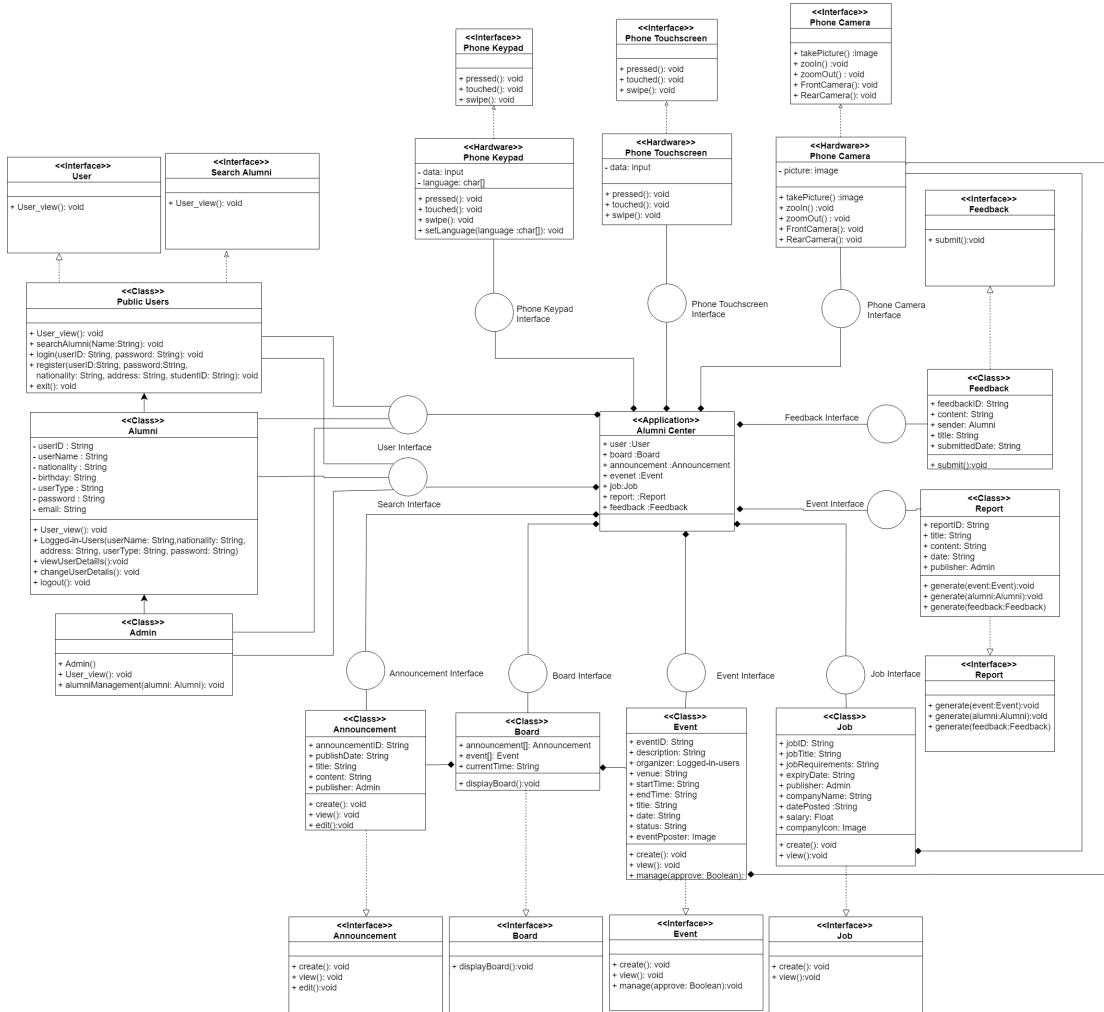
Announcement Management (Admin)

Admin can access the announcement board same as logged-in user. On top of that, admin can post a new one to the existing list of announcements through the add icon appears at bottom-right corner.

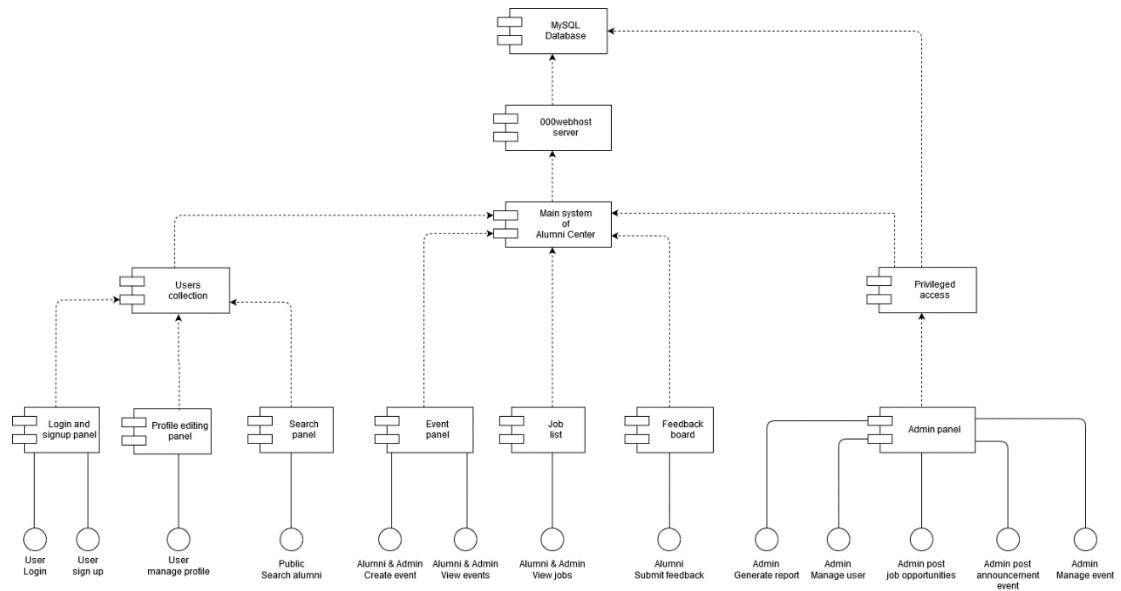
External Interface Design



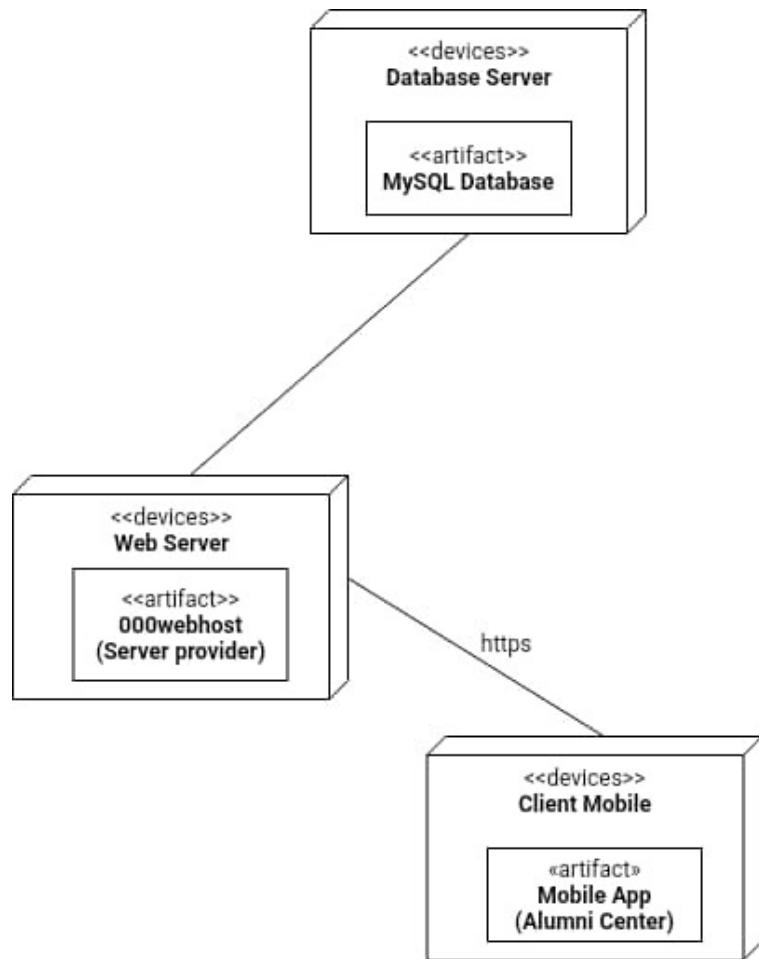
Internal Interface Design



Component Design

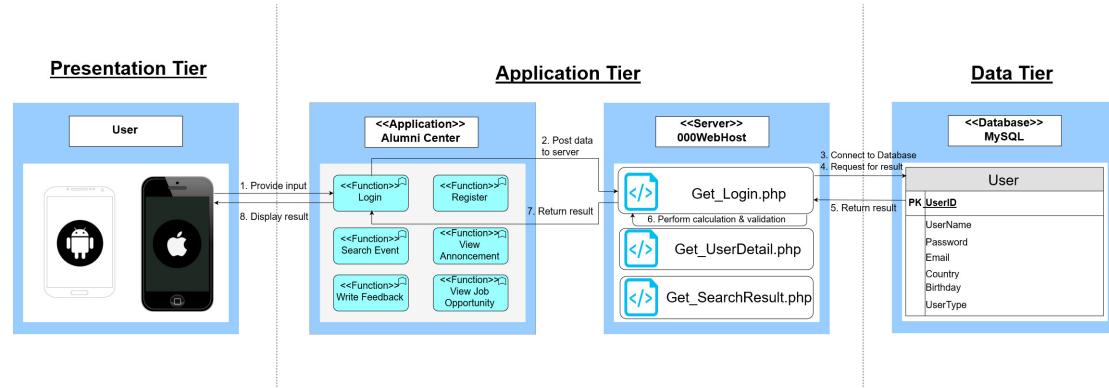


Deployment Design



3-Tier Architecture Design (Additional Diagram)

Diagram below illustrates one of the functions (Multitier architecture: Three-tier architecture).



Elaborations:

- 3-tier application architecture = a modular client-server architecture that consist of presentation tier, application tier and data tier.
- Presentation tier = graphical user interface (GUI) that communicate with other two tier.
- Application tier = handles logic and supports the application core functions
- Data tier = managing read and write access to a database.

Prototype Model of ALUMNI CENTER (Part 3)

Introduction

In this stage, the team starts with the construction of application. We used a software called Flutter with the help of a server named xiaosha007.hostingerapp.com and MySQL for database. Instead of using Firebase, we used our own server. This allow us to stimulate asynchronized processing of data and we are able to lessen the CPU processing.

Besides, we need to test our application to check the functionality of the application is working normally. The team decided to implements unit testing on our application. Hence, we created some unit testing test cases (Sample Test Case Template With Test Case Examples, 2019).

In this stage also, the team will provide some screenshot of the application in this documentation. On the other hand, the coding of the application will be provided in another file because it will make the documentation too lengthy if it is included.

Public User Test Cases

Project Name: ALUMNI CENTER

Test Case

Test Case ID:	Test_Public_01	Test Designed by:	Ong Wen Hao
Test Priority (Low/Medium/High):	Low	Test Designed date:	26/09/2019
Module Name:	PublicViewEvent	Test Executed by:	Teo Sheng Pu
Test Title:	View Event by Public User	Test Execution date:	29/9/2019
Description: Public user views the details of an event at home page			

Pre-conditions: -

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	User able to see homepage	Pass	
2	Show upcoming events		Valid upcoming event	Valid	Pass	
3	Click on event wished to see		Show event details	Shown	Pass	

Post-conditions:

-

Test Case

Test Case ID: Test_Public_02

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Medium

Test Designed date: 26/09/2019

Module Name: PublicSearch

Test Executed by: Teo Sheng Pu

Test Title: Public User search

Test Execution date: 29/9/2019

Description:

Public user attempt to search for alumnus

Pre-conditions: -

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Search		Directed to search page	Directed	Pass	
4	Enter keyword	Available username in database	Show the names of related user	Shown	Pass	
5	Click on one user		Show the information of user	Shown	Pass	

Post-conditions:

Public user should be able to see the search result and select the user in the list of names to view his profile. It will show all the usernames containing the keyword.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_Public_03

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Med

Test Designed date: 26/09/2019

Module Name: PublicRegister

Test Executed by: Teo Sheng Pu

Test Title: Public user register (Successful)

Test Execution date: 29/9/2019

Description:

Public user attempts to register as an alumnus with valid credentials

Pre-conditions: Public user has the correct credentials

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Login icon		Directed to login page	Directed	Pass	
4	Click on Register		Directed to register page	Directed	Pass	
5	Provide valid credentials	Username = lim wen pu Password = 123456 Email Address = lwp@outlook.com Country = Malaysia Birthday = 1999-01-01 Student ID = 1112223333	Register successful	Registered	Pass	

Post-conditions:

User should be redirected and navigate to Login page again.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_Public_04

Test Priority (Low/Medium/High): Med

Module Name: PublicRegister

Test Title: Public user register (Failure)

Description:

Public user attempts to register as an alumnus with invalid credentials

Test Designed by: Ong Wen Hao

Test Designed date: 26/09/2019

Test Executed by: Teo Sheng Pu

Test Execution date: 29/9/2019

Pre-conditions: Public user has the correct credentials but enter wrong/invalid credentials

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Login icon		Directed to login page	Directed	Pass	
4	Click on Register		Directed to register page	Directed	Pass	
5	Provide invalid credentials	Username (<10 characters) Password (<6 characters) Email Address (doesn't end with @domain.com) Country (no country name) Birthday (born after 2005) Student ID (<8 characters)	Prompt error for user to try again and fill the correct information	Prompted	Pass	

Post-conditions:

User will remain at register page until they enter valid credentials or navigate back to homepage

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_Public_05

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Low

Test Designed date: 26/09/2019

Module Name: PublicAnnouncement

Test Executed by: Teo Sheng Pu

Test Title: Public user view announcement

Test Execution date: 29/9/2019

Description:

Public user attempts to view announcement posted by admins

Pre-conditions: -

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	pass	There are only 4 announcements in the board
2	Show 6 latest announcement		Show 6 latest announcement	Shown	Pass	
3	Click on more announcement		Show more announcement	Shown	Pass	
4						

Post-conditions:

User should be able to see the announcement list and announcement's details.

Log in User Test Cases

Project Name: **ALUMNI CENTER**

Test Case

Test Case ID: Test_Login_01	Test Designed by: Ong Wen Hao
Test Priority (Low/Medium/High): High	Test Designed date: 26/09/2019
Module Name: UserLogin	Test Executed by: Teo Sheng Pu
Test Title: User Logging in (Successful)	Test Execution date: 29/9/2019
Description: Alumnus attempt to log in to the application	

Pre-conditions: User has the correct credentials to log in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see the home page	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Login icon		Direct to login page	Directed	Pass	
4	Provide valid credentials	StudentID= 1112223333 Password = 123456	Log in successful Direct to homepage	Logged-in	Pass	

Post-conditions:

A message will be prompted indicating user successfully logged in. User should be able to be redirect and navigate to homepage again. The status of user switch from public user to logged-in user with more accessible functions.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_Login_02

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): High

Test Designed date: 26/09/2019

Module Name: UserLogin

Test Executed by: Teo Sheng Pu

Test Title: User Logging (Failed)

Test Execution date: 29/9/2019

Description:

Alumnus attempt to log in to the application

Pre-conditions: User login with the wrong credentials

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Login icon		Direct to login page	Directed	Pass	
4	Provide invalid credentials	StudentId=1234567890 Password=123456	Log in failed Prompt user to try again	Failed	Pass	

Post-conditions:

A message will be prompted indicating user log in unsuccessfully. User should be redirected and navigate to login page again. The status of user does not switch from public user to logged-in user with more accessible functions.

Test Case

Test Case ID: Test_Login_03

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Low

Test Designed date: 26/09/2019

Module Name: UserAnnouncement

Test Executed by: Teo Sheng Pu

Test Title: Logged-in user view announcement

Test Execution date: 29/9/2019

Description:

Logged-in user attempts to view announcement posted by admins

Pre-conditions: User must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	Only 4 announcements in the board
2	Show 6 latest announcement		Show 6 latest announcement	Shown	Pass	
3	Click on more announcement		Show more announcement	Shown	Pass	

Post-conditions:

User should be able to see the announcement list and announcement's details.

Test Case

Test Case ID: Test_Login_04

Test Priority (Low/Medium/High): Medium

Module Name: UserProfileMangement

Test Title: Logged-in User manage profile

Description:

Logged-in user attempt to edit his profile

Test Designed by: Ong Wen Hao

Test Designed date: 26/09/2019

Test Executed by: Teo Sheng Pu

Test Execution date: 29/9/2019

Pre-conditions: User must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Profile icon		Direct to profile page	Directed	Pass	
3	Click on Edit icon		Direct to edit profile page	Directed	Pass	
4	Enter correct information	User Description = Healthy Mobile number = 0111111111 Address = 17,jalan hela 2 Course = Cs Graduation Year = 2011				
5	Confirm edit		Changes successfully done Information updated	Changed	Pass	

Post-conditions:

Logged-in user should be able to change the profile and view the correct version of his profile.

Test Case

Test Case ID: Test_Login_05

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Medium

Test Designed date: 26/09/2019

Module Name: UserSearch

Test Executed by: Teo Sheng Pu

Test Title: Logged-in User search

Test Execution date: 29/9/2019

Description:

Logged-in user attempt to search for alumnus

Pre-conditions: User must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Search		Directed to search page	Directed	Pass	
4	Enter keyword	Keyword = lim	Show the names of related user	Shown	Pass	
5	Click on one user		Show the information of user	Shown	Pass	Self Profile appeared

Post-conditions:

Logged-in user should be able to see the search result and select the user in the list of names to view his profile. It will show all the usernames containing the keyword.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_Login_06

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Low

Test Designed date: 26/09/2019

Module Name: UserEventListPage

Test Executed by: Teo Sheng Pu

Test Title: Logged-in User organize events

Test Execution date: 29/9/2019

Description:

Logged-in user attempt to view existing events and even create one

Pre-conditions: User must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Events		Direct to event page	Directed	Pass	
4	View events		Show all events	Shown	Pass	
5	Click create Event button		Direct to create event page	Directed	Pass	
6	Fill in the form	Title = hackathon Venue= Mph Start time =12:00:00 End Time = 18:00:00 Date = 2019-Nov-1 Description = casual activity	Able to fill the form	Filled	Pass	
7	Click on Submit		Save the form	Submitted	Pass	

Post-conditions:

Logged-in user should be able to view events and create one if wish to.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_Login_07

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Low

Test Designed date: 26/09/2019

Module Name: UserFeedBack

Test Executed by: Teo Sheng Pu

Test Title: Logged-in User submit feedback

Test Execution date: 29/9/2019

Description:

Logged-in user attempt to submit his feedback or opinion to the institution

Pre-conditions: User must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Feedback		Direct to feedback	Directed	Pass	
4	Fills the form	Text = Aircon not working Description =CQBR1007	Able to fill the form	Filled	Pass	
5	Click Submit		Save the form	Submitted	Pass	

Post-conditions:

Logged-in user should be able to submit his feedback or opinion.

Test Case

Test Case ID: Test_Login_08

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Medium

Test Designed date: 26/09/2019

Module Name: userJobListPage

Test Executed by: Teo Sheng Pu

Test Title: Logged-in User view job opportunities

Test Execution date: 29/9/2019

Description:

Logged-in user attempt to view the enlisted job opportunities added by admin

Pre-conditions: User must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Job		Direct to job page	Directed	Pass	
4	View enlisted jobs		Show available jobs	Shown	Pass	
5	Click on "details" to see more information		Show details of a job	Shown	Pass	

Post-conditions:

Logged-in user should be able to see the job list and details.

Admin Test Cases

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_01	Test Designed by: Ong Wen Hao
Test Priority (Low/Medium/High): High	Test Designed date: 26/09/2019
Module Name: AdminLogin	Test Executed by: Teo Sheng Pu
Test Title: Admin Logging in	Test Execution date: 29/9/2019
Description: Admin attempt to log in to the application	

Pre-conditions: Admin has the correct credentials

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Login icon		Direct to login page	Directed	Pass	
4	Provide valid credentials	StudentID = 1171101755 Password = 123456	Log in successful Direct to homepage	Logged-in	Pass	

Post-conditions:

A message will be prompted indicating user successfully logged in. Admin should be able to be redirect and navigate to homepage again. The status of user switch from public user to admin with the most number of accessible functions.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_02

Test Priority (Low/Medium/High): High

Module Name: AdminLogin

Test Title: Admin logging in falsefully

Description:

Admin attempt to log in to the application

Test Designed by: Ong Wen Hao

Test Designed date: 26/09/2019

Test Executed by: Teo Sheng Pu

Test Execution date: 29/9/2019

Pre-conditions: Admin has the wrong credentials

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Login icon		Direct to login page	Directed	Pass	
4	Provide invalid credentials	StudentID=1171101755 Password=abcdefg	Log in failed Prompt user to try again	Prompted	Pass	

Post-conditions:

A message will be prompted indicating admin log in unsuccessfully. Admin should be redirected and navigate to login page again. The status of user does not switch from public user to admin with more accessible functions.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_03	Test Designed by: Ong Wen Hao
Test Priority (Low/Medium/High): Low	Test Designed date: 26/09/2019
Module Name: AdminEventListPage	Test Executed by: Teo Sheng Pu
Test Title: Admin view and organize events	Test Execution date: 29/9/2019
Description: Admin attempt to view existing events, and create new event	

Pre-conditions: Admin must be logged-in
Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Event		Direct to event page	Directed	Pass	
4	View events		Show all events	Shown	Pass	
5	Click create Event button		Direct to create event page	Directed	Pass	
6	Fill in the form	Title = hack Venue = mph Start time = 00:00:00 End time = 15:00:00 Date = 2019 oct 1 Description = wow	Able to fill the form	Filled	Pass	
7	Click on Submit		Save the form Return to Event page	Submitted	Pass	

Post-conditions:

Admin should be able to view events and create one if wish to.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_04

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Low

Test Designed date: 26/09/2019

Module Name: AdminManageEventListPage

Test Executed by: Teo Sheng Pu

Test Title: Admin manage events

Test Execution date: 29/9/2019

Description:

Admin attempt to manage events

Pre-conditions: Admin must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Events		Direct to event page	Directed	Pass	
4	View events		Show all events	Shown	Pass	
5	Click pending		Direct to pending page	Directed	Pass	
6	Click on an event		Direct to the event page with the particulars	Directed	Pass	
7	Click on Approve or decline		If clicked approved ->Event will be approved else ->Event will be declined	Clicked	Pass	Approved hackathon

Post-conditions:

Admin user should be able to manage events.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_05

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Low

Test Designed date: 26/09/2019

Module Name: AdminFeedBack

Test Executed by: Teo Sheng Pu

Test Title: Admin view feedback

Test Execution date: 29/9/2019

Description:

Admin attempt to view his feedback or opinion to the institution

Pre-conditions: Admin must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Feedback		Direct to feedback	Directed	Pass	
4	View Feedback		Able to view feedback	Viewed	Pass	
5	Click a Feedback		Show information of feedback	Shown	Pass	

Post-conditions:

Admin should be able to view his feedback or opinion.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_06

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Medium

Test Designed date: 26/09/2019

Module Name: AdminJobListPage

Test Executed by: Teo Sheng Pu

Test Title: Admin view and add job opportunities

Test Execution date: 29/9/2019

Description:

Admin attempt to view or add the enlisted job opportunities

Pre-conditions: Admin must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Job		Direct to job page	Directed	Pass	
4	View enlisted jobs		Show available jobs	Shown	Pass	
5	Add Job	Job Title = data analyst Company Name = Mayback Salary = RM3100 Expiry Date = 2020 Jan 1 Job Requirement =Passionate	Save the job successfully	Saved	Pass	

Post-conditions:

Admin should be able to add job to job list.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_07

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Medium

Test Designed date: 26/09/2019

Module Name: AdminReportPage

Test Executed by: Teo Sheng Pu

Test Title: Admin view reports

Test Execution date: 29/9/2019

Description:

Admin attempt to view reports

Pre-conditions: Admin must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Report		Direct to report page	Directed	Pass	
4	Click on User		Direct to user report page	Directed	Pass	
5	Click on Job		Direct to job report page	Directed	Pass	
6	Click on Event		Direct to event report page	Directed	Pass	

Post-conditions:

Admin should be able to view different kinds of reports.

Project Name: ALUMNI CENTER

Test Case

Test Case ID: Test_admin_08

Test Designed by: Ong Wen Hao

Test Priority (Low/Medium/High): Medium

Test Designed date: 26/09/2019

Module Name: AdminJobListPage

Test Executed by: Teo Sheng Pu

Test Title: Admin create announcement

Test Execution date: 29/9/2019

Description:

Admin attempt to create announcement

Pre-conditions: Admin must be logged-in

Dependencies: -

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to homepage		User should be able to see homepage	Seen	Pass	
2	Click on Side Navigation icon		Show all function	Shown	Pass	
3	Click on Announcement		Direct to announce page	Directed	Pass	
4	Show Announcement		Able to show appropriate announcement	Shown	Pass	
5	Add Announcement	Title = Done Unit Testing Content = Ive done all the unit tests	Save the announcement successfully	Saved	Pass	

Post-conditions:

Admin should be able to add announcement

Software Tools

Flutter

The software team chose to use Flutter to build and construct the front end of the prototype. This is because Flutter is a cross-platform mobile application development framework which means the same code can be used for iOS and Android app. Besides, this open-source framework also exhibits high performance because it has its own widget that decrease mediated communication between app and platform. Last but not least, using flutter saves a lot of time as it has “hot reload” which does not require emulators, simulators and hardware to restart after every change.

MySQL

MySQL is used as the backend tool which is to manage the database of ALUMNI CENTER. One of the reasons is because it shows high performance because of its distinct storage-engine framework. Secondly, MySQL also has been the one of the most secured and reliable database management system across the globe. Its scalability attracts the team to customize the database with unique database server requirements.

PHP

The software team used PHP to act as a bridge to link the software with database. The reason is because PHP is cross-platform same as Flutter. This allows compatibility during development process of the software. It also eases the connection to link the database as one of the reasons it is widely used by many programmers. Finally, it is open-source and easy to use like the other two software tools. This is very important due to the limited resources and time given to the software team.

Note:

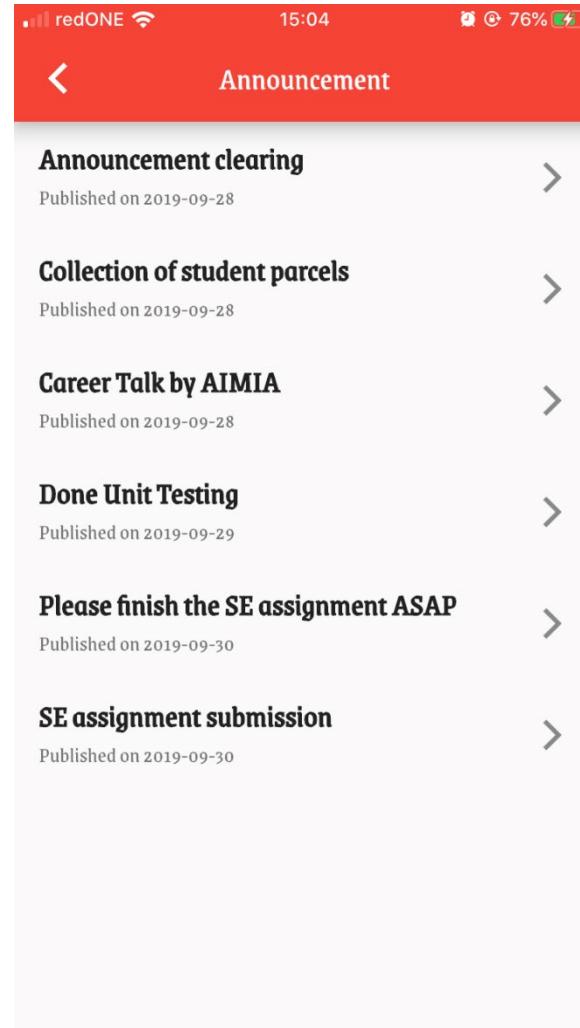
Firebase was considered but not chosen because MySQL delivers more beneficial functions such as it allows the software team to acquire full control access in terms of security and authorization. Besides it enables asynchronous processing that is essential for a real-time application so that hosting server and the application can be programmed at the same time. Finally, Firebase does not ensure data integrity although it is flexible because addition of user data is relatively easier than MySQL.

System Screen Shots

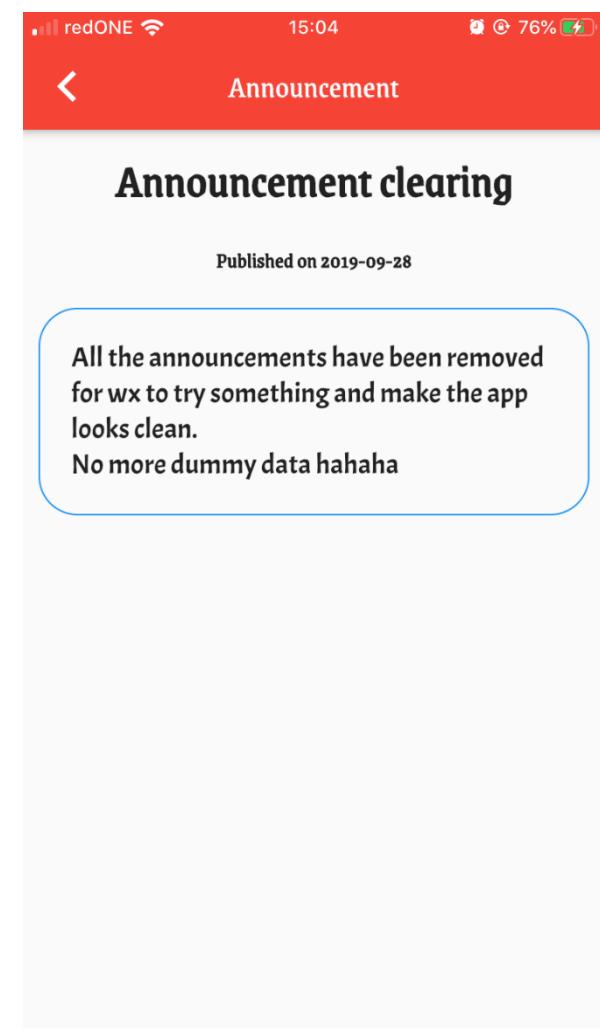
Public user



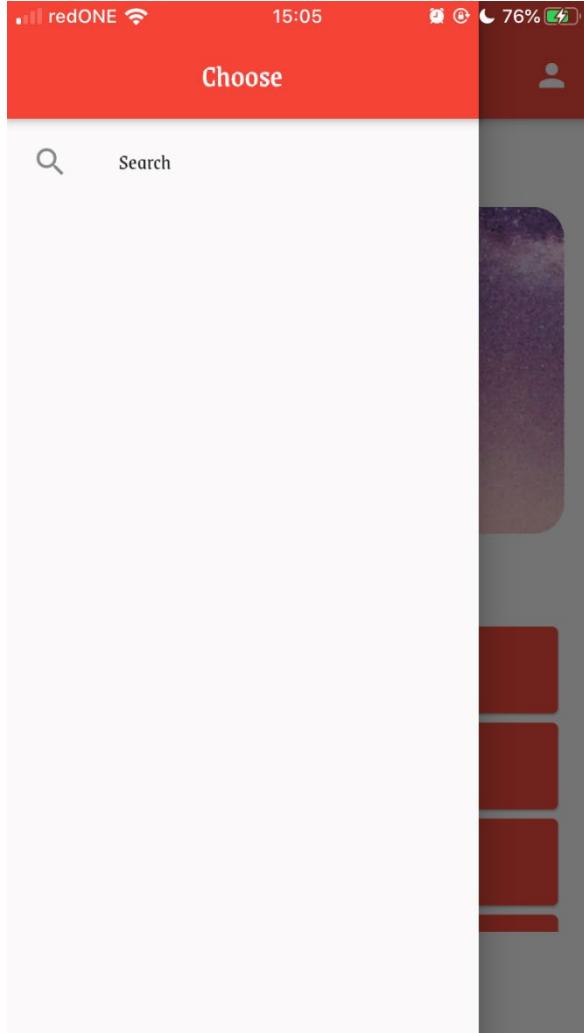
Home Page of Public User



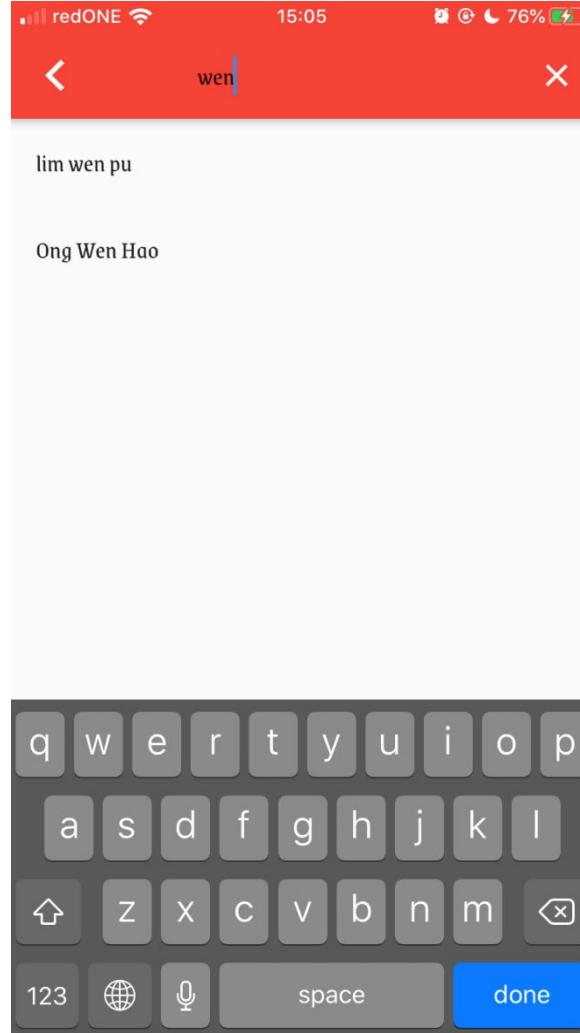
Announcement List



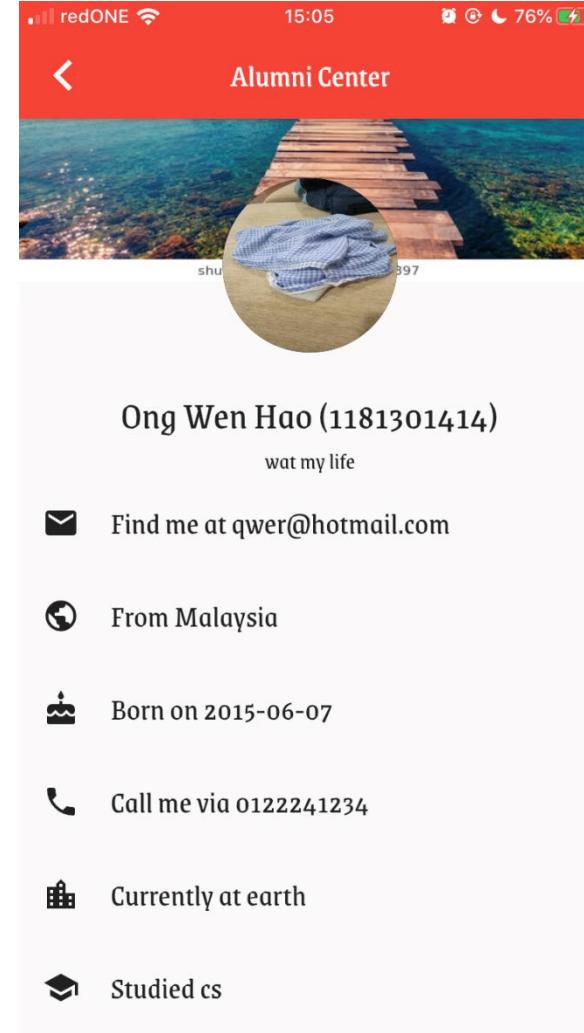
Announcement Details



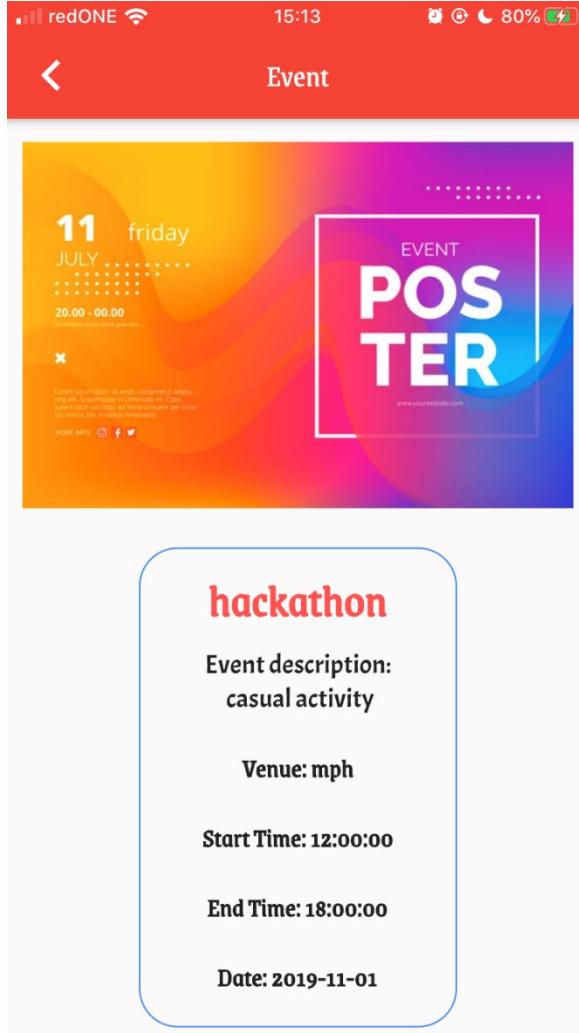
Side Menu Bar



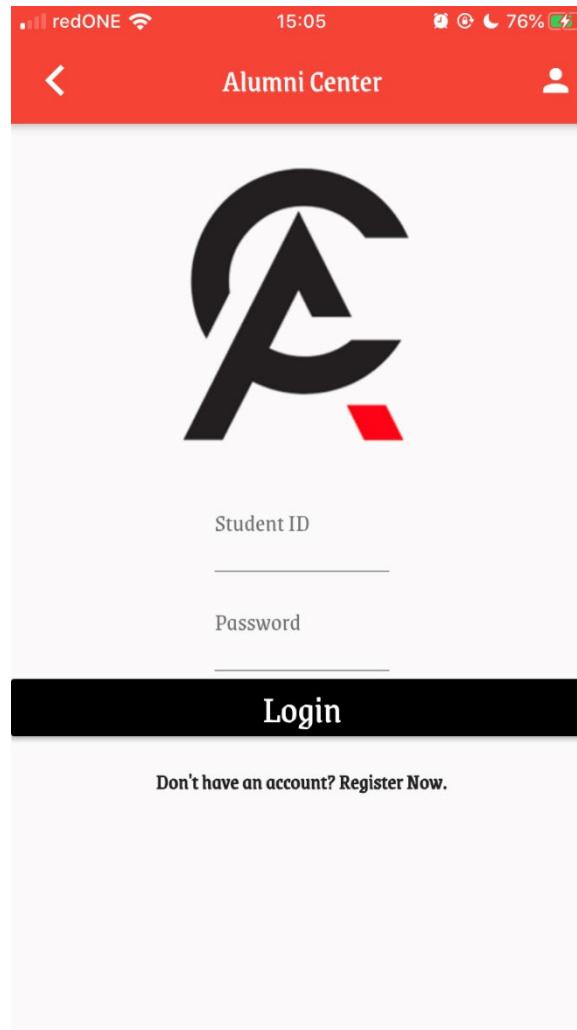
Searched User Profile



Searched User



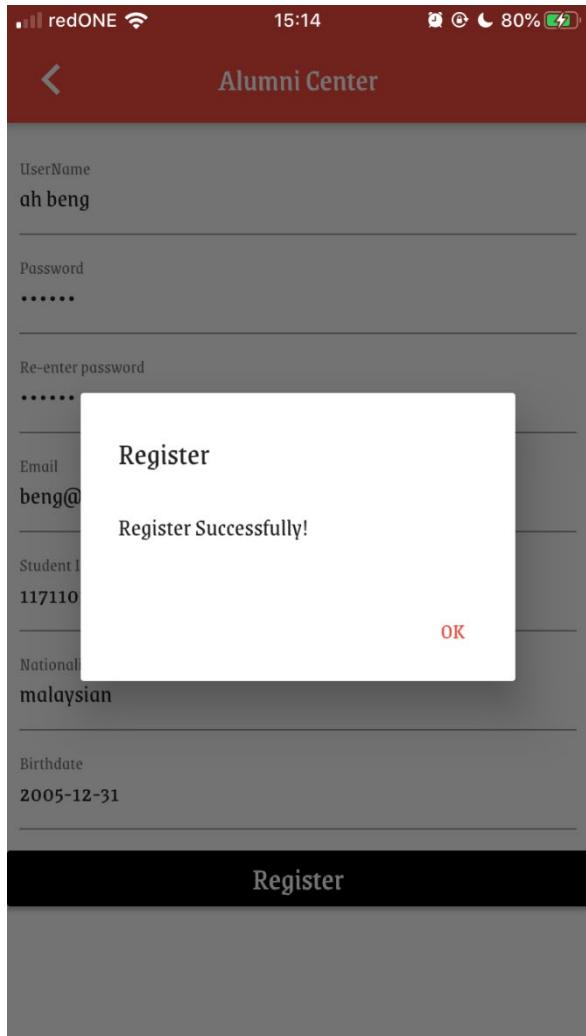
Event Details



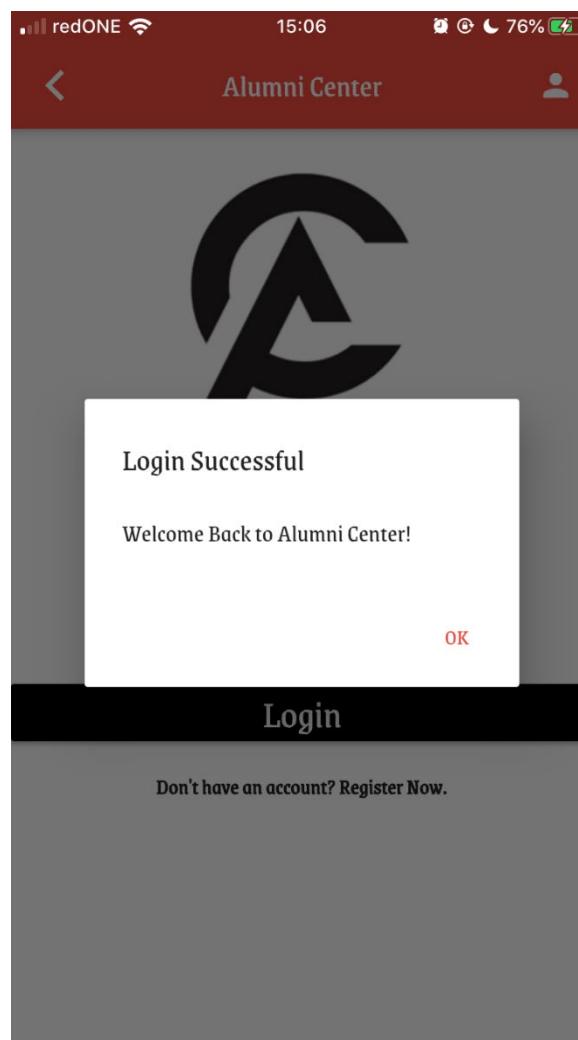
Login Page

The screen shows a mobile application interface for an "Alumni Center". At the top, there's a red header bar with the words "Alumni Center". To the right of the header are two input fields: "UserName" and "Password". Below these are two more input fields: "Re-enter password" and "Email". To the right of the "Email" field is another input field labeled "Student ID". Further down are fields for "Nationality" and "Birthdate". At the bottom right is a large black button labeled "Register".

Registration Page



Register Successful



Login Successful

Logged-in User



14:45 68% redONE

Alumni Center

Upcoming Events

11 JULY 20.00 - 00.00 EVENT POS TER www.yourwebsite.com

Announcement

Announcement clearing

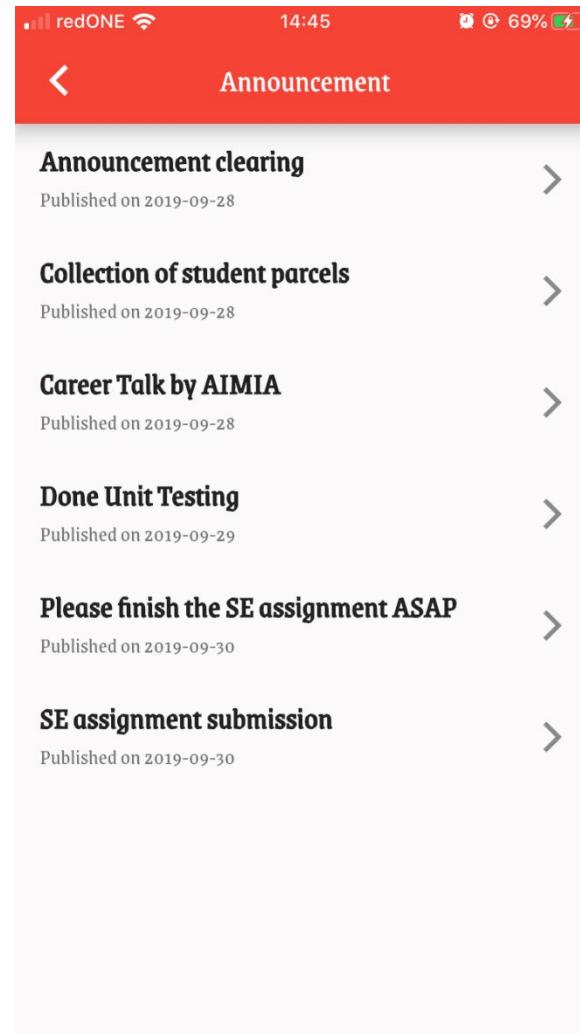
Collection of student parcels

Career Talk by AIMIA

More announcement

This screen shows the home page for a logged-in user. It features a header with signal strength, time (14:45), battery level (68%), and network (redONE). Below the header is the 'Alumni Center' navigation bar. The main content area includes a section for 'Upcoming Events' with a placeholder card for an event on July 11th. Below this is a large 'Announcement' section containing three items: 'Announcement clearing', 'Collection of student parcels', and 'Career Talk by AIMIA'. At the bottom is a link to 'More announcement'.

Home Page of Logged-in User



14:45 69% redONE

Announcement

Announcement clearing
Published on 2019-09-28

Collection of student parcels
Published on 2019-09-28

Career Talk by AIMIA
Published on 2019-09-28

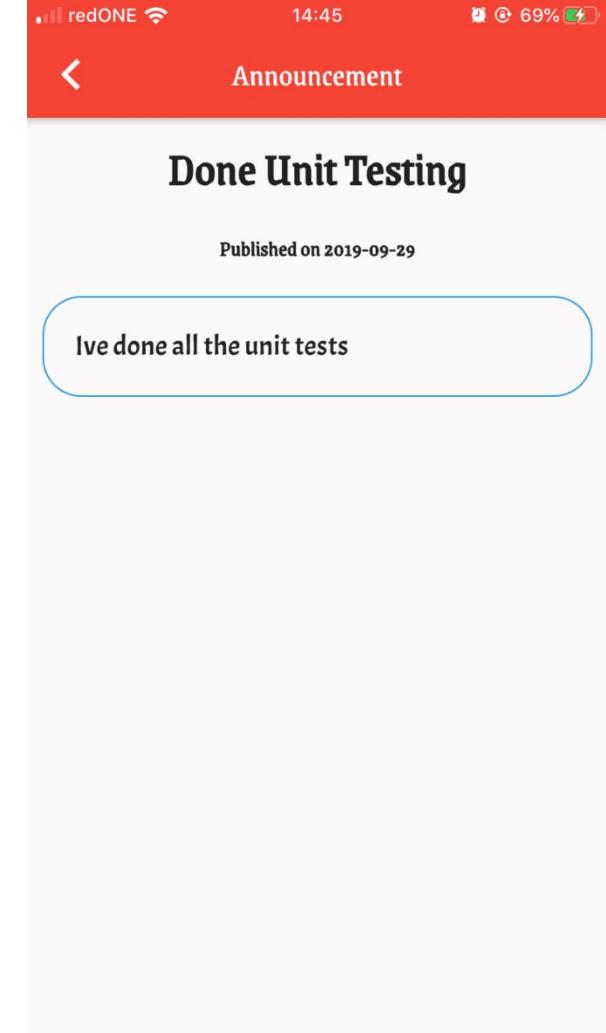
Done Unit Testing
Published on 2019-09-29

Please finish the SE assignment ASAP
Published on 2019-09-30

SE assignment submission
Published on 2019-09-30

This screen displays a list of announcements. It has a header with signal strength, time (14:45), battery level (69%), and network (redONE). Below the header is the 'Announcement' title. The list contains six items, each with a title and a 'Published on' date. The first five items are standard announcements, while the last one is titled 'Done Unit Testing' and includes a blue-outlined box with the text 'Ive done all the unit tests'.

Announcement Lists



14:45 69% redONE

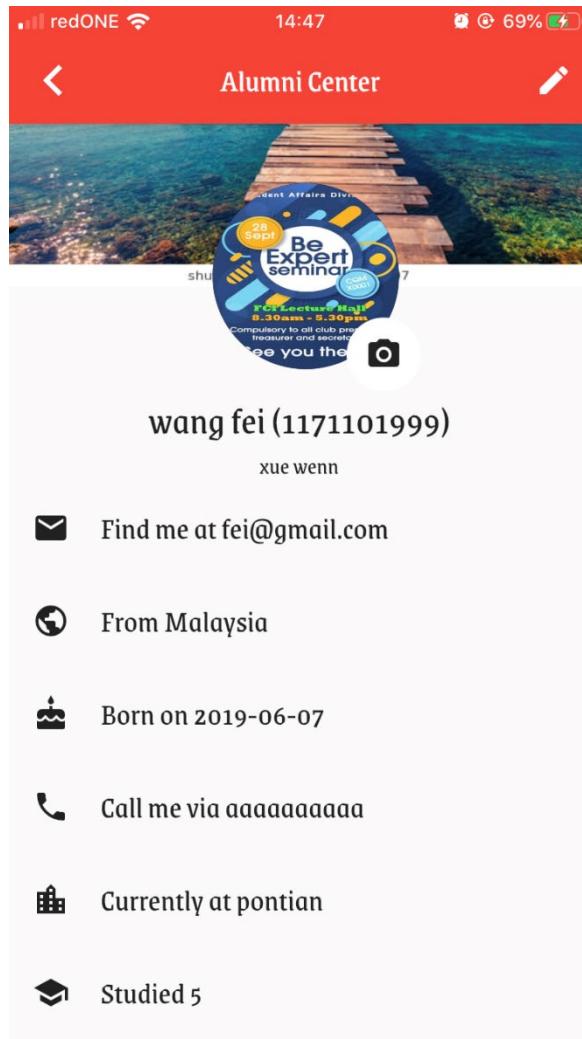
Announcement

Done Unit Testing
Published on 2019-09-29

Ive done all the unit tests

This screen shows the details of a specific announcement titled 'Done Unit Testing'. It has a header with signal strength, time (14:45), battery level (69%), and network (redONE). Below the header is the 'Announcement' title. The main content is a single item: 'Done Unit Testing' published on 2019-09-29, which includes a blue-outlined box containing the text 'Ive done all the unit tests'.

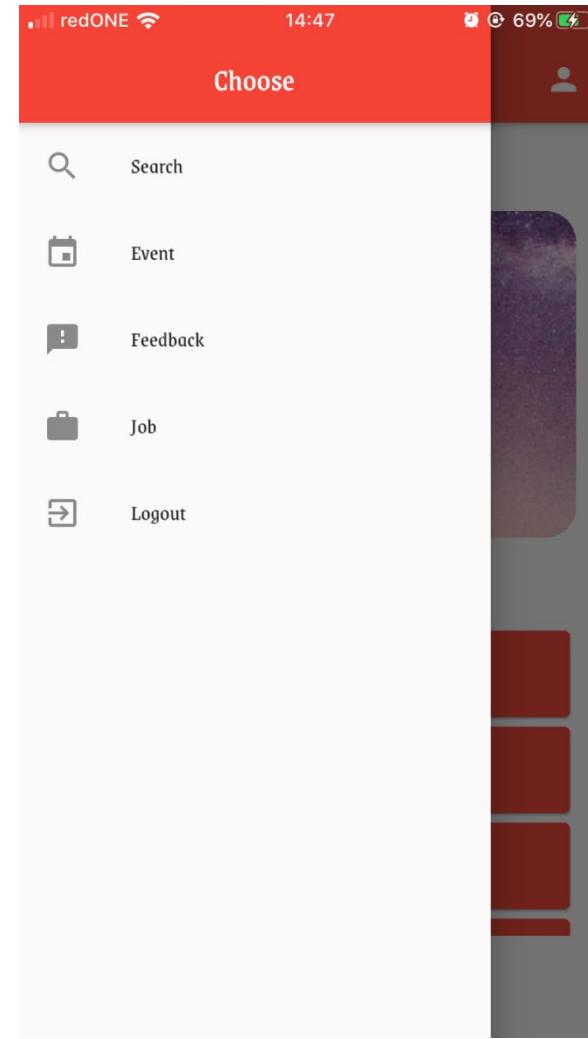
Announcement Details



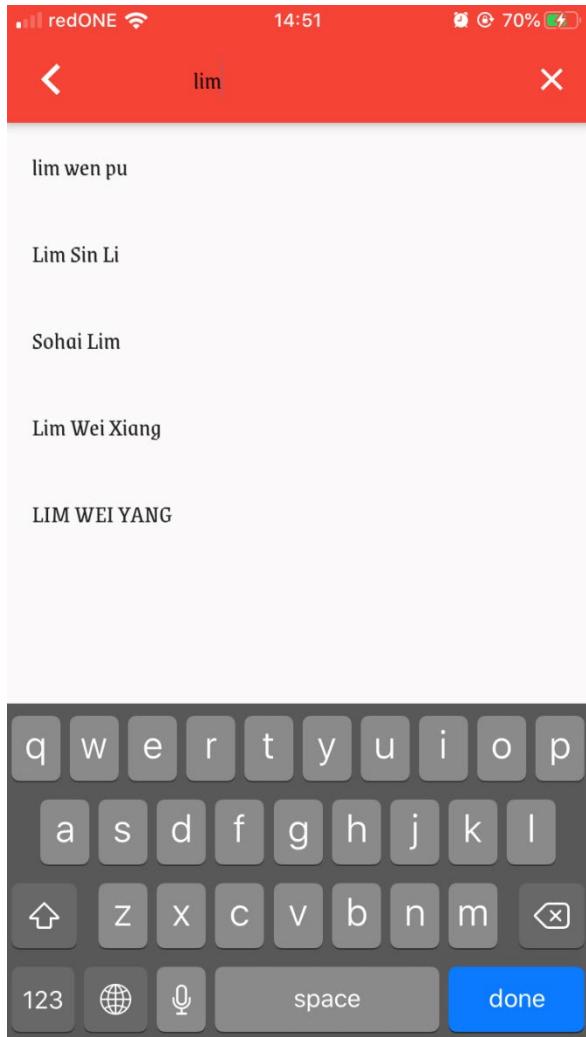
Profile

This screen is titled "Edit Profile" with a back arrow and a checkmark icon. It contains several input fields: "User Description", "Mobile Number", "Address", "Course Taken During University Time", and "Year of Graduation". Each field has a corresponding placeholder text below it.

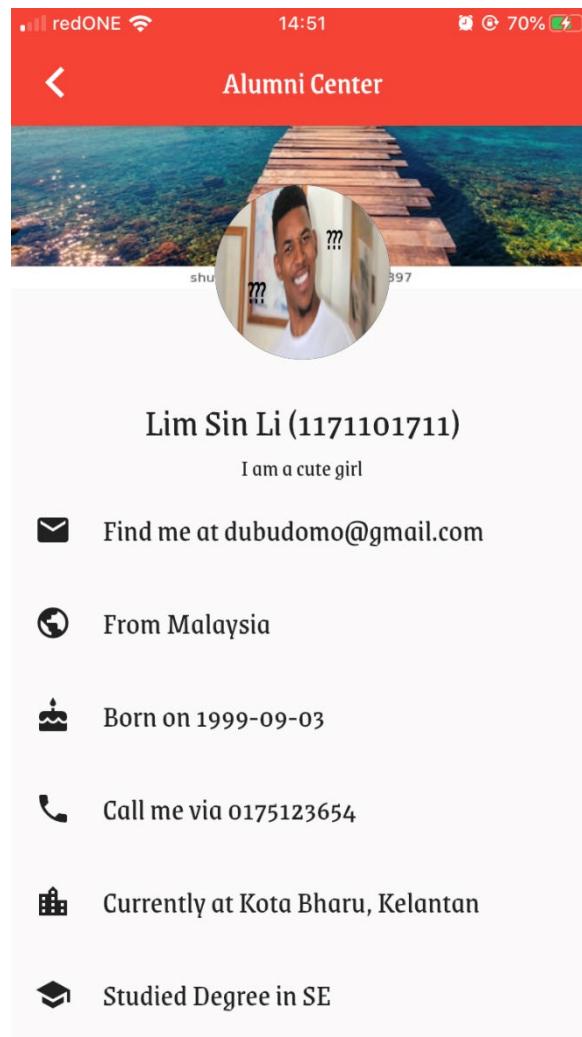
Edit Profile



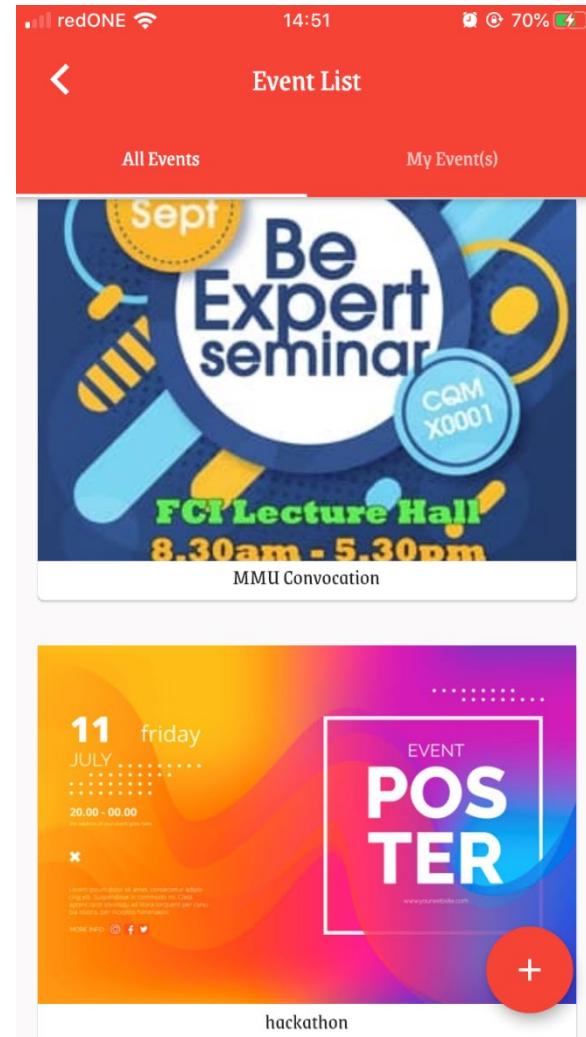
Side Menu Bar



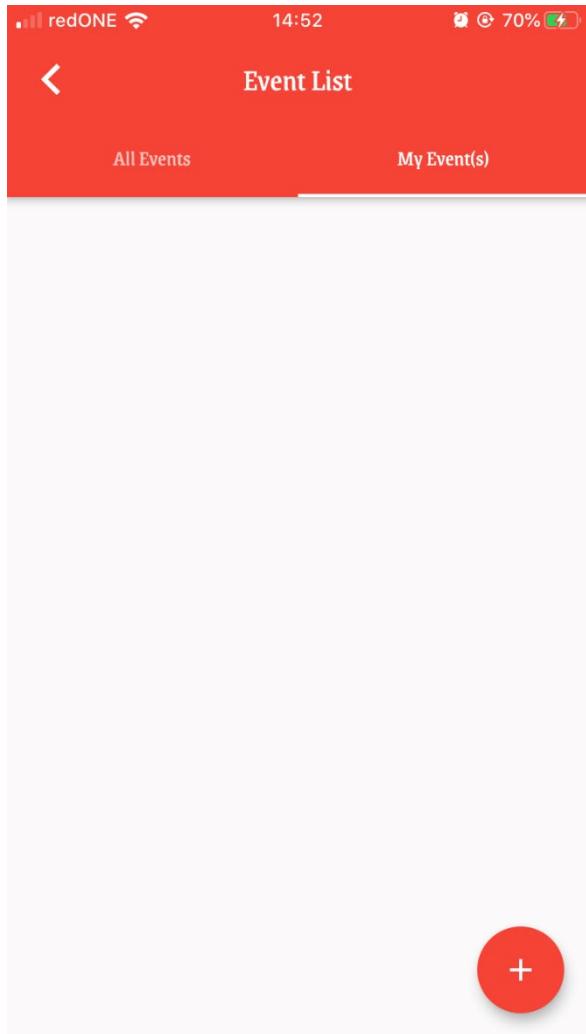
Search User



Searched User



Event List (All Event)



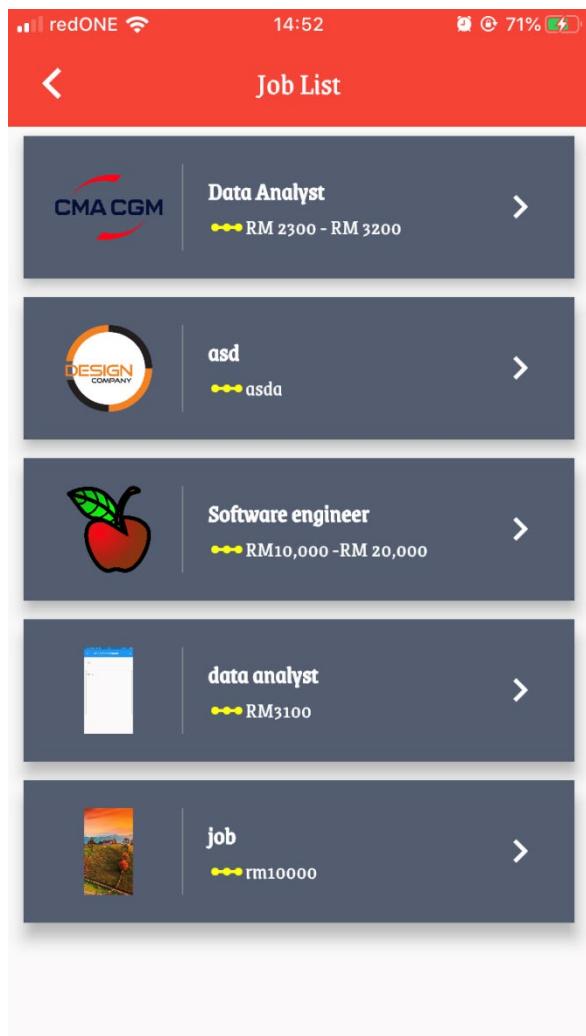
Event List (My Event)

A screenshot of a mobile application interface titled "Submit Feedback". The top status bar shows signal strength, battery level at 71%, and the time 14:52. The header "Submit Feedback" is centered above two navigation icons: a back arrow on the left and a checkmark icon on the right. Below the header is a "Title" input field with a horizontal line underneath. A larger input field labeled "Description" is centered below it.

Feedback form

A screenshot of a mobile application interface titled "Add an event". The top status bar shows signal strength, battery level at 70%, and the time 14:52. The header "Add an event" is centered above two navigation icons: a back arrow on the left and a checkmark icon on the right. Below the header are five input fields: "Title", "Venue", "Start Time", "End Time", and "Date", each with a horizontal line underneath. A larger input field labeled "Event Description" is centered below these fields.

Add event



View Job Lists

A screenshot of a mobile application showing the details of a specific job posting. The screen has a red header with a back arrow and the title 'Job Details'. Below the header is the job title 'Data Analyst' and the company logo 'CMA CGM'. The job was posted by 'Maybank Sdn Bhd' on '2019-09-26'. The salary range is 'RM 2300 - RM 3200'. A large blue box contains the 'Job Requirements' section, which includes instructions for saving the app and a note about a circular progress indicator. Below this box is the 'Application Deadline' information: '2019-12-31'.

Job Details

A screenshot of a mobile application showing an event poster. The screen has a red header with a back arrow and the title 'Event'. The poster features a colorful background with a yellow-to-blue gradient. It displays the date '11 JULY', the time '20.00 - 00.00', and the event name 'EVENT POS TER'. Below the poster is a blue rounded rectangle containing event details: 'hackathon', 'Event description: casual activity', 'Venue: mph', 'Start Time: 12:00:00', 'End Time: 18:00:00', and 'Date: 2019-11-01'.

hackathon

Event description:
casual activity

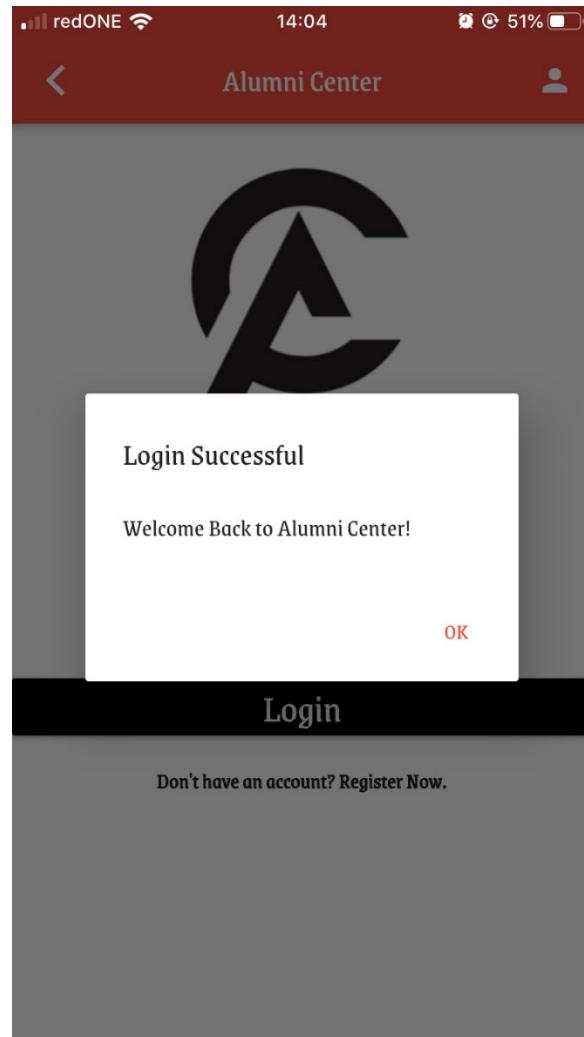
Venue: mph

Start Time: 12:00:00

End Time: 18:00:00

Date: 2019-11-01

Admin



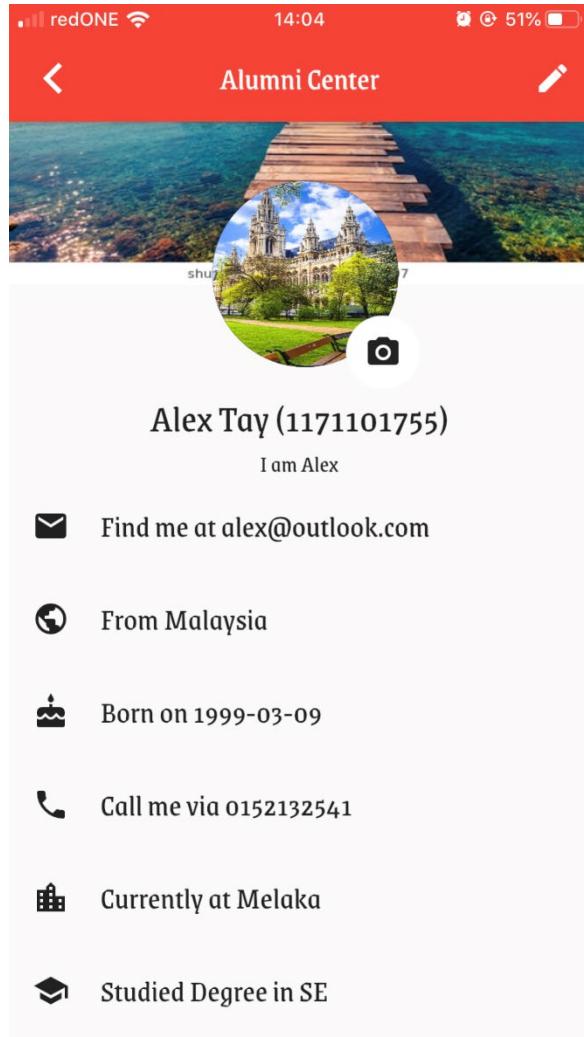
Logged-in as Admin

This screenshot shows the main home page of the Alumni Center. At the top, there is a red header bar with the text "Alumni Center" and a user icon. Below the header is a section titled "Upcoming Events" featuring a large graphic with a yellow-to-blue gradient and a wavy pattern. It displays an event card for "11 JULY" with the title "EVENT POSTER" and a link "www.yourwebsite.com". Below this, there is a section titled "Announcement" containing three red rectangular buttons with white text: "Announcement clearing", "Collection of student parcels", and "Career Talk by AIMIA". At the bottom of this section is a blue link "More announcement".

Home Page of Admin

This screenshot shows the details of an event titled "hackathon". The event information is displayed in a blue-bordered box: "Event description: casual activity", "Venue: mph", "Start Time: 12:00:00", "End Time: 18:00:00", and "Date: 2019-11-01". Above this box, there is a large graphic with a yellow-to-blue gradient and a wavy pattern, similar to the one on the home page. On the right side of the graphic, there is a white box with the text "EVENT POSTER" and a link "www.yourwebsite.com".

View Event Details

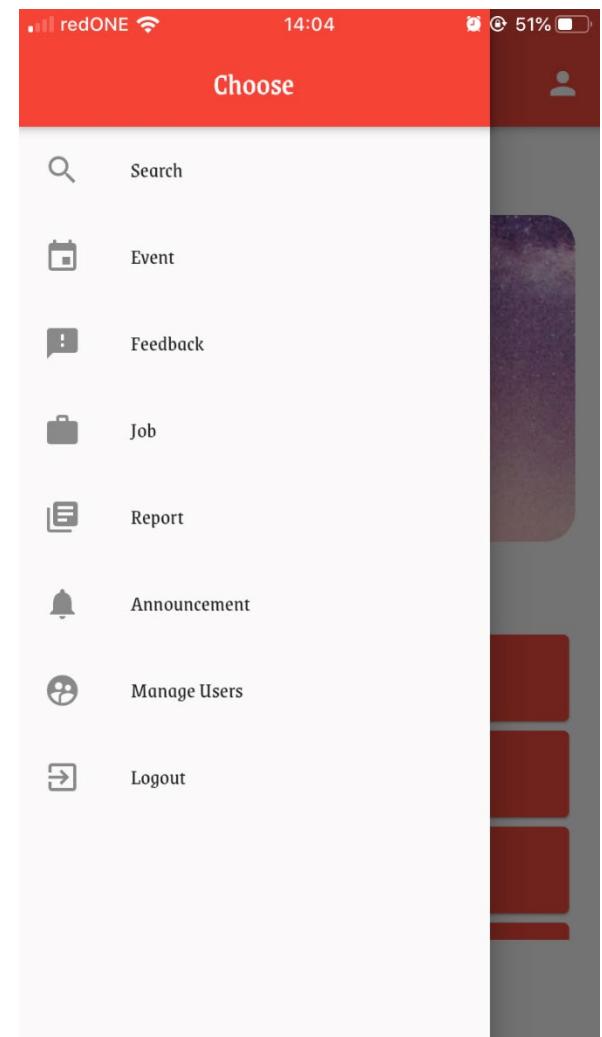


Admin Profile

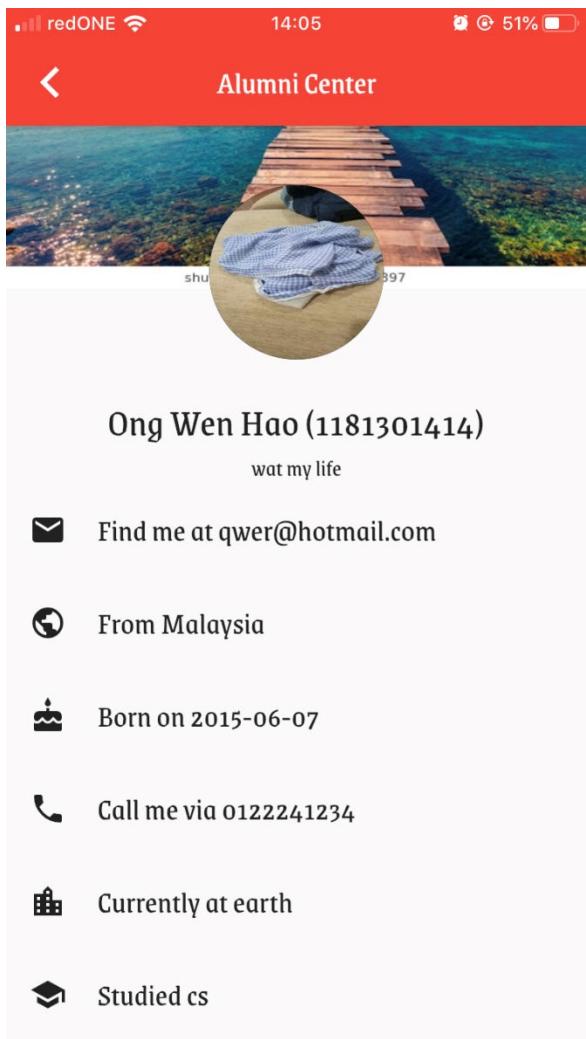
The screenshot shows the 'Edit Profile' screen. At the top, there's a red header bar with the text 'Edit Profile' and a checkmark icon. Below the header are several input fields with placeholder text:

- User Description
- Mobile Number
- Address
- Course Taken During University Time
- Year of Graduation

Edit Admin Profile



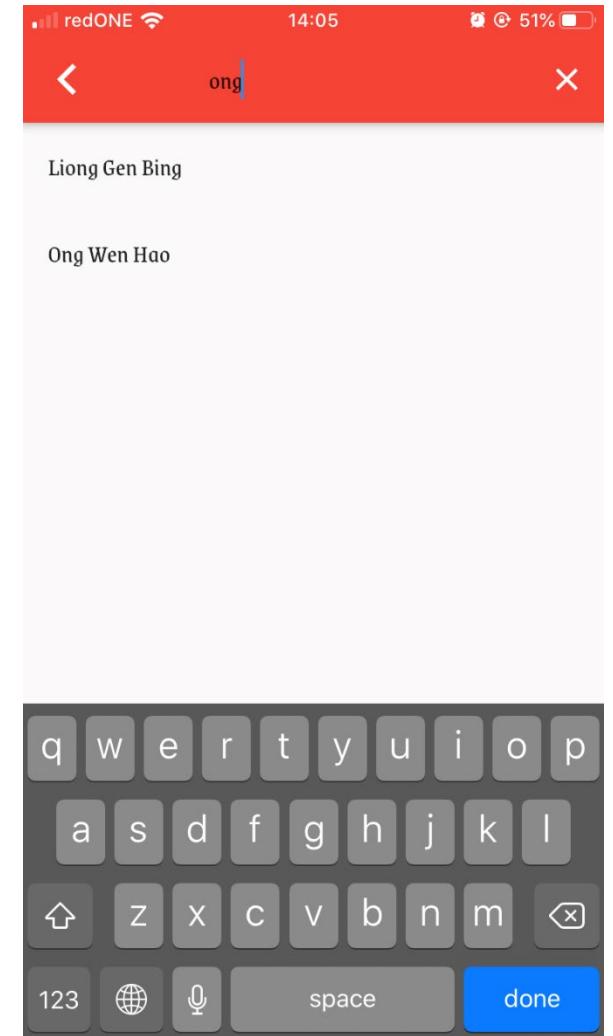
Side Menu Bar



Search User

-
- This screenshot shows a list of announcements. The title "Announcement" is at the top. There are six items, each with a right-pointing arrow icon to its right:
- Announcement clearing**
Published on 2019-09-28
 - Collection of student parcels**
Published on 2019-09-28
 - Career Talk by AIMIA**
Published on 2019-09-28
 - Done Unit Testing**
Published on 2019-09-29
 - Please finish the SE assignment ASAP**
Published on 2019-09-30
 - SE assignment submission**
Published on 2019-09-30
- A red circular button with a white plus sign is located at the bottom right of the list.

Announcement List



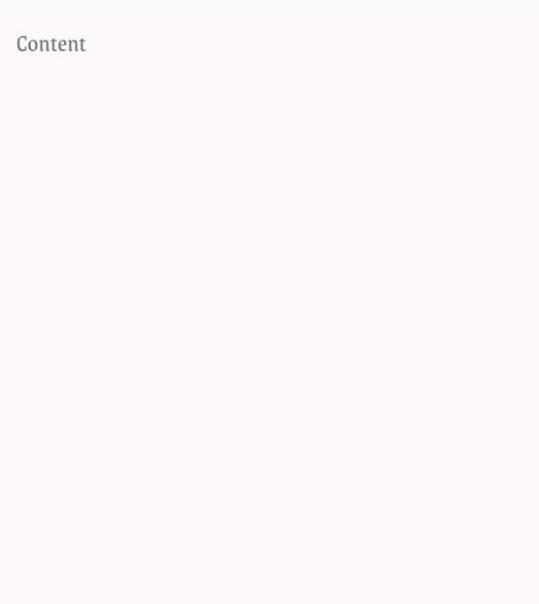
Search Users

redONE 14:05 51%

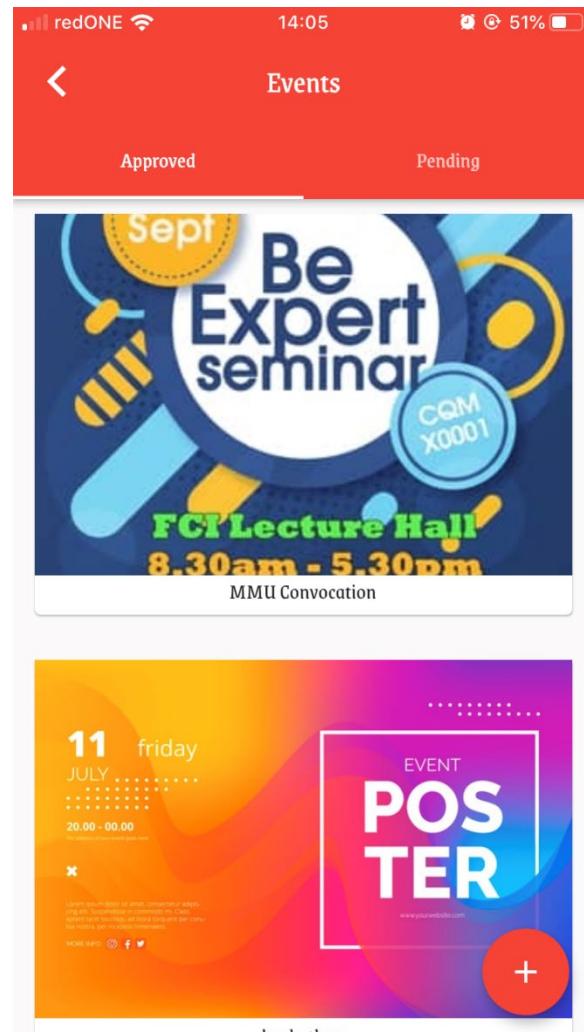
Add Announcement

Title

Content

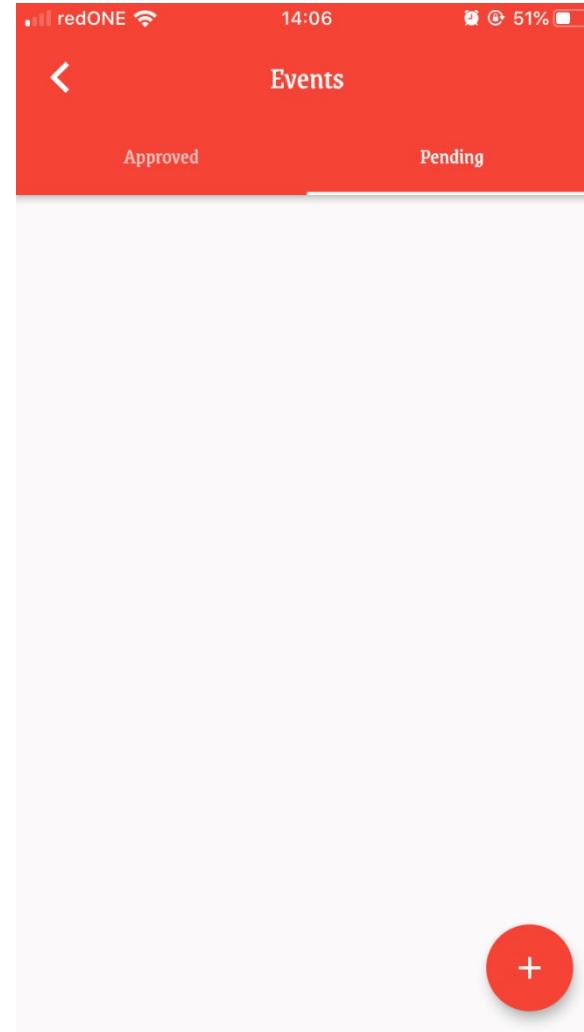


A large, empty rectangular box intended for users to input their announcement content.



Add Announcement

View Event List (Approved)



View Event List (Pending)

redONE 14:06 51%

Add an event

Title _____

Venue _____

Start Time _____

End Time _____

Date _____

Event Description

-
- redONE 14:10 53%
- Feedback
- Complain** >
Published on 2015-02-02
- asdasd** >
Published on 2019-09-27
- I am Liong Gen Bing** >
Published on 2019-09-27
- Alex stupid** >
Published on 2019-09-27
- 12e** >
Published on 2019-09-28
- Hello** >
Published on 2019-09-28
- air con not working** >
Published on 2019-09-29

redONE 14:10 53%

Feedback

air con not working

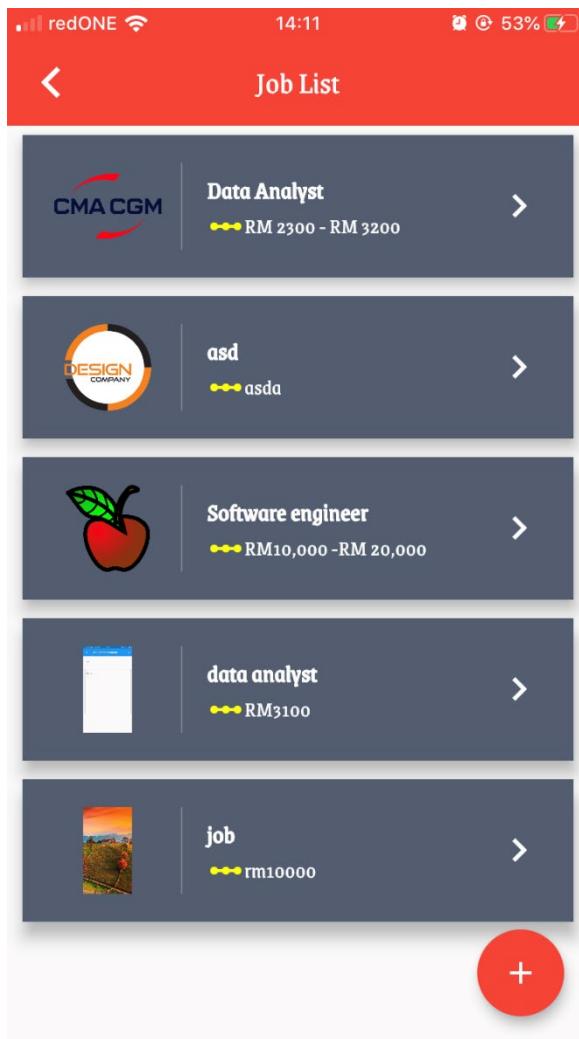
Published on 2019-09-29

cqbr1007

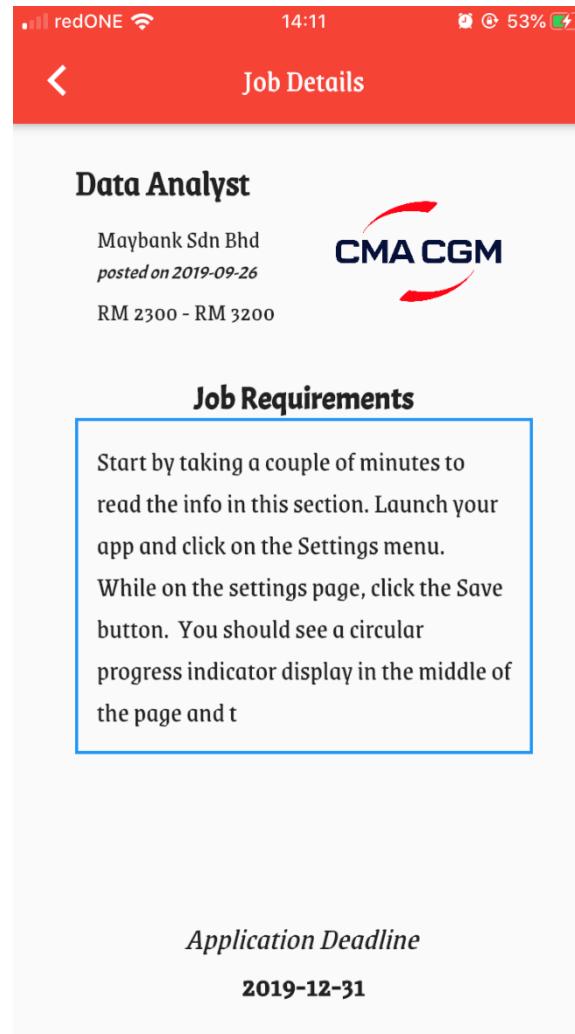
Add Event

View Feedback List

View Feedback Details



View Job List



View Job Details

A screenshot of a mobile application showing a form to add a new job listing. The screen has a red header bar with the title "Add Job". It contains fields for "Job Title", "Company Name", "Salary", and "Date". Below these is a large text area for "Description/Requirements". A blue button at the bottom right says "Click here to upload Company Logo.".

Add Job to Job List

redONE 14:11 53%

Report

User Job Event

UserID	UserName	Email
1101151232	Mandy	mandy@outlook.com
1112223333	lim wen pu	lwp@outlook.com
1121101512	Low Zi Shi	zishi@outlook.com
1141101805	Ian Chai Mei Wei	ianchai@outlook.com
1151101805	Ng Jin Bin	jinbin@outlook.com
1161101805	Liong Gen Bing	genbing@gmail.com
1171101234	Leow Yen Siang	yensiang@outlook.com
1171101333	Ang yan kai	Kai@Gmail.com
1171101444	Chan Wai Kok	cisco@mmu.edu.my

Report (User)

redONE 14:11 53%

Report

User Job Event

JobTitle	Company	DatePosted
Data Analyst	Maybank Sdn Bhd	2019-09-26
asd	asd	2019-09-26
Software engineer	Microsoft	2019-09-27
data analyst	may back	2019-09-29
job	apple	2019-09-29

Report (Job)

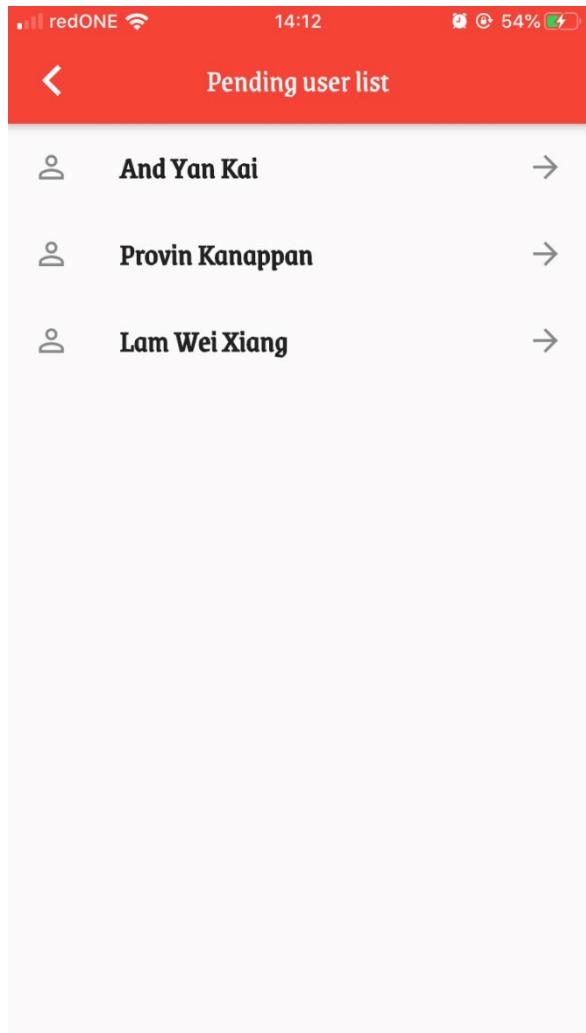
redONE 14:12 53%

Report

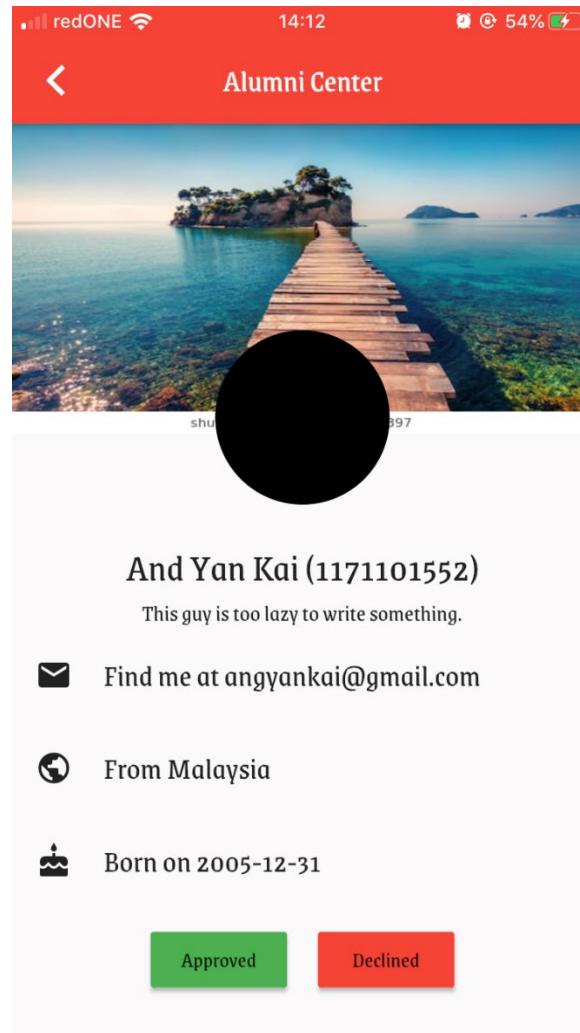
User Job Event

Title	Venue	Date
Event Clearing		2019-09-28
Picture testing		2019-09-28
MMU Convocation	MMU Grand hall	2019-09-28
hackathon	mph	2019-11-01
hack	mph	2019-10-01
I am the boss	MMU Cyberjaya	2019-11-30
MMU Final Exam	MMU Grand Hall	2019-10-12

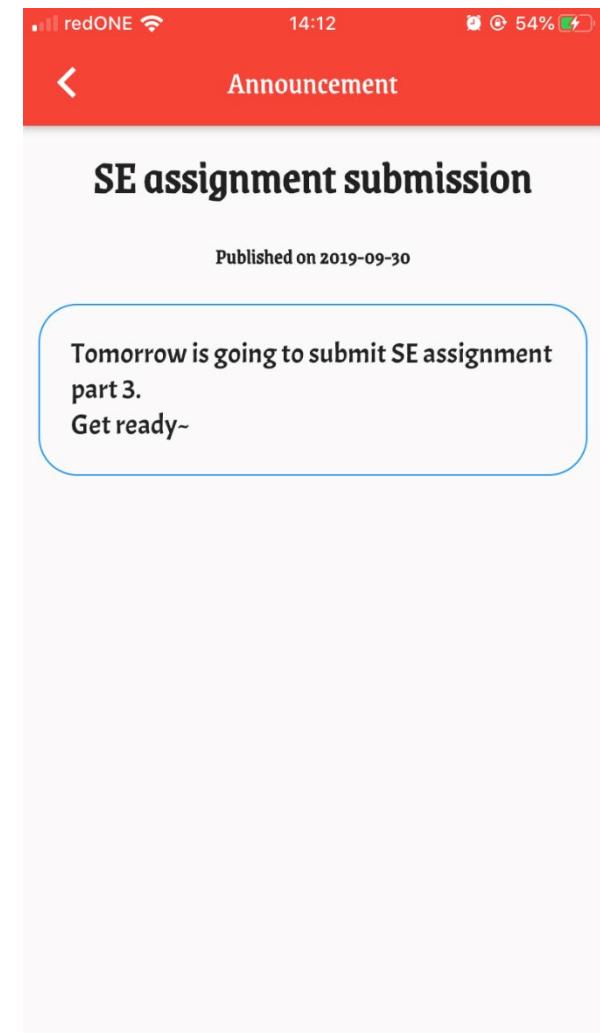
Report (Event)



Pending user List



Pending User



View Announcement Details

Sample Codes and Database

As there are too many lines of code that we used to develop the program, we can only show part of the code in the document and explain part of it. The codes shown below are the php code sample which we used to connect our app with the MySQL database.

1. Register Alumni

```
1 <?php
2     include 'check_connection.php';
3     $sqlCommand = "SELECT UserID,Email FROM Users;";
4     $data = $connection->query($sqlCommand);
5
6     $studentID = $_POST["studentID"];
7     $userName = addslashes($_POST["userName"]);
8     $password = addslashes($_POST["password"]);
9     $email = addslashes($_POST["email"]);
10    $country = addslashes($_POST["country"]);
11    $userType = $_POST["userType"];
12    $birthday = $_POST["birthday"];
13
14    $result = array();
15    $register_pass = 1;
16    if($data->num_rows >0){
17        while($searchResult = $data->fetch_assoc()){
18            $result[] = $searchResult;
19            if($searchResult["UserID"]==$studentID || $searchResult["Email"]==$email){
20                $register_pass = 0;
21                break;
22            }
23        }
24    }
25    if($register_pass==1){
26        $sqlCommand = "INSERT INTO Users(UserID,UserName,UserPassword,Email,Country,UserType,Birthday)
27                     VALUES('" . $studentID . "','" . $userName . "','" . $password . ',
28                           "' . $email . "','" . $country . "','" . $userType . ',
29                           "' . $birthday . "')";
30        if($connection->query($sqlCommand)==true){
31            $sqlCommand = "INSERT INTO Alumni(UserID,Description,MobileNum,Address,Course,YearOfGraduation)
32                         VALUES ('" . $studentID . "','default','default','default','default','2022')";
33            if($connection->query($sqlCommand)==true){}
34            else{
35                echo $sqlCommand."<br>".$connection->error;
36            }
37        }
38        else{
39            echo $sqlCommand."<br>".$connection->error;
40        }
41    }
42    echo $register_pass
43
44 ?>
```

Pseudocode

1. Get the information input by the user (Student ID, username, password, email, country, user type and birthday)
2. Extract all the email and studentID from the database. The data is used to make sure there is no identical email or studentID being registered by other users.
3. A loop is created to compare the value (email & studentID) input by users with the existing value (email & studentID) of database. If the loop found out that the email or studentID input by users have been used by other users, the loop will break and will return a 0 to the app.

4. If the email and studentID input are unique, it will then post the information to the database and choose the right table to add (either user table or admin table).

5. Error will be prompted if there is any connection error.

Database View after adding one user

	+ Options	← T →	▼	UserID	Description	MobileNum	Address	Course
<input type="checkbox"/>		Edit		1101151232	default	default	default	default
<input type="checkbox"/>		Edit		1111121328	default	default	default	default
<input type="checkbox"/>		Edit		1111121329	default	default	default	default
<input type="checkbox"/>		Edit		1112223333	Healthy	0111111111	17,Jalan hela	cs
<input type="checkbox"/>		Edit		1121101512	default	default	default	default
<input type="checkbox"/>		Edit		1141101805	I am Dr Ian Chai Mei Wei. Born in USA.	0176875271	Near Cyberjaya	Degree in BlowingWater
<input type="checkbox"/>		Edit		1151101805	I am Jin Bin for	0176786543	Johor Bharu	Degree in Data Science
<input type="checkbox"/>		Edit		1161101805	Im a genius	0175687527	Bangi, Selangor	Degree in Data Science
<input type="checkbox"/>		Edit		1171101444	default	default	default	default
<input type="checkbox"/>		Edit		1171101556	default	default	default	default
<input type="checkbox"/>		Edit		1171101755	I am Alex	0152132541	Melaka	Degree in SE
<input type="checkbox"/>		Edit		1171101805	Life is about coding....	0175817993	17, Jalan Tasik Prima 4/5a	Bachelor Degree In Software Engin
<input type="checkbox"/>		Edit		1171101999	xue wenn	aaaaaaaaaa	pontian	5
<input type="checkbox"/>		Edit		1181301414	wat my life	0122241234	earth	cs
<input type="checkbox"/>		Edit		1201121328	default	default	default	default
<input type="checkbox"/>		Edit		afvsgghjvg	default	default	default	default

2. Search Alumni

```

1 <?php
2     include 'check_connection.php';
3     $searchKey = addslashes($_GET["searchKey"]);
4
5     $sqlCommand = "SELECT UserName,UserID
6                     FROM Users
7                     WHERE LOCATE(''" . $searchKey . "'",UserName)>0 AND UserType='Normal';";
8     $data = $connection->query($sqlCommand);
9     $result = array();
10    if($data->num_rows >0){
11        while($searchResult = $data->fetch_assoc()){
12            $result[] = $searchResult;
13        }
14    }
15    echo json_encode($result);
16 ?>
```

Pseudocode

- Get user's search key input
- The SQL query being executed by the MySQL database will ask the database to return the results that contain the substring which matches the search key input by users.
- The result will then be stored inside an array.

4. The array contains the search result will be returned to the app.

3. Add Event

```
1 <?php
2     include 'check_connection.php';
3
4     $userID      = $_POST["userID"];
5     $title       = addslashes($_POST["title"]);
6     $description = addslashes($_POST["description"]);
7     $venue        = addslashes($_POST["venue"]);
8     $startTime   = $_POST["startTime"];
9     $endTime     = $_POST["endTime"];
10    $date        = $_POST["date"];
11    $status      = $_POST["status"];
12    $image        = $_POST["image"];
13    $insertEvent  = 1;
14
15    $sqlCommand = "INSERT INTO Events(UserID, Title, Description, Venue, StartTime, EndTime, Date, Status)
16    VALUES('$userID','$title','$description','$venue','$startTime','$endTime','$date','$status')";
17
18    if($connection->query($sqlCommand) === true){
19        $last_id    = $connection->insert_id;
20        echo $last_id;
21        $fileName   = "./EventPoster/" . $last_id;
22        $realImage  = base64_decode($image);
23        file_put_contents($fileName,$realImage);
24    }
25    else{
26        $insertEvent = 0;
27        echo $sqlCommand."<br>$connection->error";
28    }
29    echo $insertEvent;
30
31 ?>
```

Pseudocode

1. Get the information of the event
2. Insert the event information to event table in the database
3. Make sure the connection between server and database is connected
4. Else prompt error

View of the event table in database

+ Options										
	EventID	UserID	Title	Description	Venue	StartTime	EndTime	Date	Status	
<input type="checkbox"/>	40	1171101755	Event Clearing	I've clear all the events, please add it again. ha...		04:07:20	05:07:22	2019-09-28	A	
<input type="checkbox"/>	41	1171101755	Picture testing	Im testing the new compressed image function		04:26:43	04:26:45	2019-09-28	A	
<input type="checkbox"/>	43	1171101711	MMU Convocation	MMU Convocation is nearby! Please come!	MMU Grand hall	20:00:00	21:00:00	2019-09-28	A	
<input type="checkbox"/>	44	1112223333	hackathon	casual activity	mph	12:00:00	18:00:00	2019-11-01	A	
<input type="checkbox"/>	45	1171101755	hack	wow	mph	00:00:00	15:00:00	2019-10-01	A	
<input type="checkbox"/>	46	1171101805	I am the boss	hello world!	MMU Cyberjaya	17:51:54	17:59:56	2019-11-30	P	

4. Get Event Detail

```
1 <?php
2     include 'check_connection.php';
3     $input_ID = $_GET["eventID"];
4     $user_ID = null;
5     $upcoming = $_GET["upcoming"];
6
7     if($_GET["userID"]!=null) {
8         $user_ID = $_GET["userID"];
9         $sqlCommand = "SELECT * FROM Events WHERE UserID = '$user_ID'";
10    }
11   else if($input_ID=="all" && $upcoming == "true") {
12       $sqlCommand = "SELECT * FROM Events WHERE Date>=CURRENT_DATE";
13   }
14   else if($input_ID=="all" && $upcoming == "false") {
15       $sqlCommand = "SELECT * FROM Events";
16   }
17   else {
18       $sqlCommand = "SELECT UserID, Title, Description, Venue, StartTime, EndTime, Date, Status
19                   FROM Events
20                   WHERE eventID= '$input_ID' ;";
21   }
22   $data = $connection->query($sqlCommand);
23   $result = array();
24   if($data->num_rows >0) {
25       while($searchResult = $data->fetch_assoc()) {
26           $result[] = $searchResult;
27       }
28   }
29   echo json_encode($result);
30 ?>
```

Pseudocode

1. Get eventID, upcoming and userID from users.
2. If userID is not null then it will show all the events that has been created by that particular user.
3. Else If the eventID is equal to “all” and upcoming is true, then show all the upcoming events
4. Else If the eventID is equal to “all” and upcoming is fail, then show all events including the events that were organized before.
5. Else show the event details of a specific event
6. Then show all results that matches the correct condition

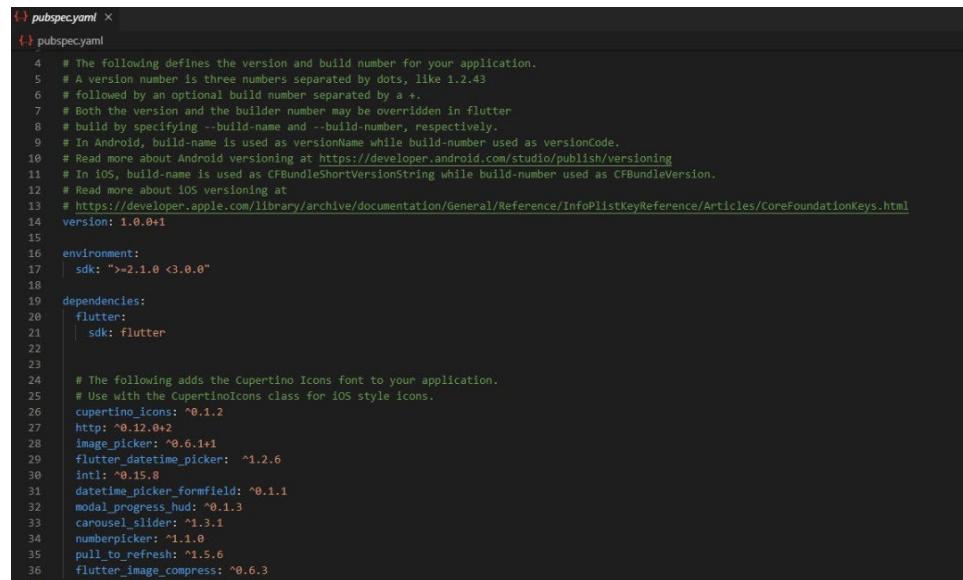
5. Update profile picture

```
1 <?php
2     include 'check_connection.php';
3     $image = $_POST["image"];
4     $fileName = "./userProfilePic/" . $_POST["fileName"];
5     $realImage = base64_decode($image);
6     file_put_contents($fileName, $realImage);
7
8 ?>
9
10
11
12
13
```

Explanation:

1. The ways we upload the pictures from our app to server is:
 - a. Firstly, in the app, we encode the picture into Base64 format.
 - b. As we can see, when the picture in base64 format is sent to the server, the server will accept and store it inside a variable named \$image.
 - c. At the same time, the file name will be posted together with the base64 image to the server and will be used to name the picture.
 - d. After that, the base64 image will be decoded by the server and store inside the variable named \$realImage.
 - e. The server will then output a file which stores the information of the base64 formatted picture. The file is now saved successfully in the server.

The codes shown below are **dart** programming language and **flutter** framework which we used to develop the main body and function of the application. Since there are too many lines of code, we will only display one of it because the other files are designed in a similar way. In our case, we choose to display and explain “home.dart” as it encompasses all the similar functions that are used in other dart files and therefore, it can be used to represent the other files in our explanation.

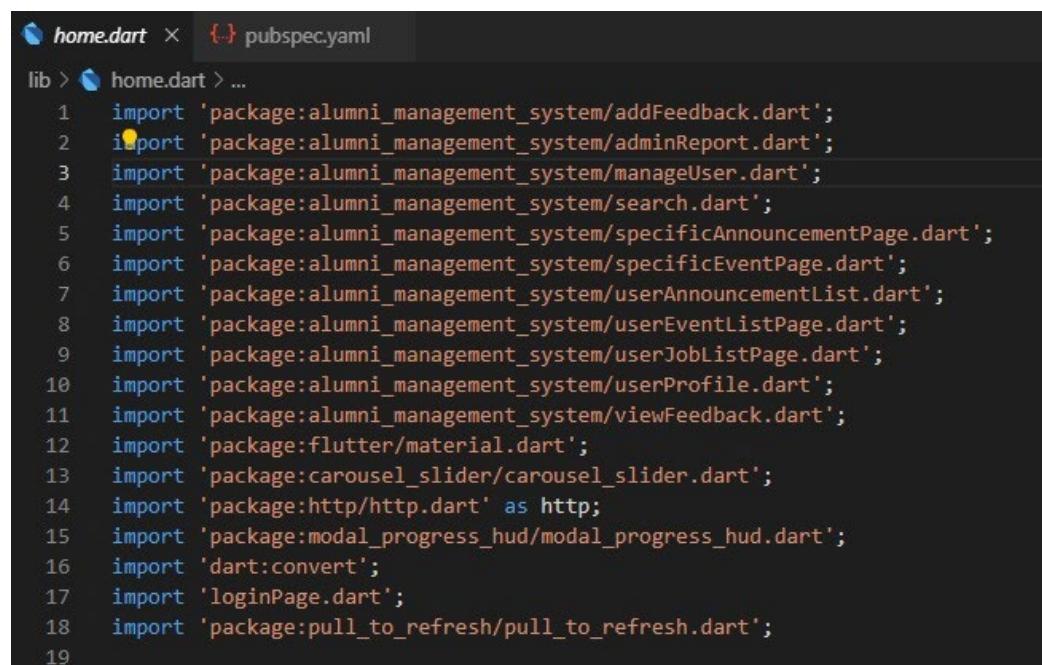


```

4 # The following defines the version and build number for your application.
5 # A version number is three numbers separated by dots, like 1.2.43
6 # followed by an optional build number separated by a +.
7 # Both the version and the builder number may be overridden in flutter
8 # build by specifying --build-name and --build-number, respectively.
9 # In Android, build-name is used as versionName while build-number is used as versionCode.
10 # Read more about Android versioning at https://developer.android.com/studio/publish/versioning
11 # In iOS, build-name is used as CFBundleShortVersionString while build-number is used as CFBundleVersion.
12 # Read more about iOS versioning at
13 # https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
14 version: 1.0.0+1
15
16 environment:
17   | sdk: ">=2.1.0 <3.0.0"
18
19 dependencies:
20   flutter:
21     | sdk: flutter
22
23
24   # The following adds the Cupertino Icons font to your application.
25   # Use with the CupertinoIcons class for iOS style icons.
26  /cupertino_icons: ^0.1.2
27   http: ^0.12.0+2
28   image_picker: ^0.6.1+1
29   flutter_datetime_picker: ^1.2.6
30   intl: ^0.15.8
31   datetime_picker_formfield: ^0.1.1
32   modal_progress_hud: ^0.1.3
33   carousel_slider: ^1.3.1
34   numberpicker: ^1.1.0
35   pull_to_refresh: ^1.5.6
36   flutter_image_compress: ^0.6.3

```

1. The diagram shows the “pubspec.yaml” file which contains all the dependencies used by the flutter framework. In this file, we can simply add in a line of code and import the libraries from the Internet.



```

lib > home.dart > ...
1 import 'package:alumni_management_system/addFeedback.dart';
2 import 'package:alumni_management_system/adminReport.dart';
3 import 'package:alumni_management_system/manageUser.dart';
4 import 'package:alumni_management_system/search.dart';
5 import 'package:alumni_management_system/specificAnnouncementPage.dart';
6 import 'package:alumni_management_system/specificEventPage.dart';
7 import 'package:alumni_management_system/userAnnouncementList.dart';
8 import 'package:alumni_management_system/userEventListPage.dart';
9 import 'package:alumni_management_system/userJobListPage.dart';
10 import 'package:alumni_management_system/userProfile.dart';
11 import 'package:alumni_management_system/viewFeedback.dart';
12 import 'package:flutter/material.dart';
13 import 'package:carousel_slider/carousel_slider.dart';
14 import 'package:http/http.dart' as http;
15 import 'package:modal_progress_hud/modal_progress_hud.dart';
16 import 'dart:convert';
17 import 'loginPage.dart';
18 import 'package:pull_to_refresh/pull_to_refresh.dart';
19

```

1. This is the beginning of every dart code files. The “import” keywords are required to import the libraries from external source in order for the code to function.

```

class HomePage extends StatefulWidget {
    String userID;
    String userType;
    HomePage(this.userID, this.userType);
    @override
    State<StatefulWidget> createState() {
        return _HomePage();
    }
}

```

1. To begin with the construction of the application, we have to apply Object Oriented Programming Principles, which is to create a class that inherits the function of StatefulWidget. StatefulWidget is a widget that allows the application to change state easily without the destruct and construct of the class.
2. For every classes that inherit from StatefulWidget, they must override the createState() function as this method will create the state of the current page, in other words, this method will create the body of this class.

```

class _HomePage extends State<HomePage> {
    bool loadingSpinner = false;

    List<String> eventPosterList = new List();
    List<String> announcementTitleList = new List();
    final RefreshController _refreshController = RefreshController(initialRefresh: false);

    void getEventData() async{
        eventPosterList.clear();
        loadingSpinner = true;
        final response = await http.get("http://xiaosha007.hostingerapp.com/getEventDetails.php?eventID=all&userID=&upcoming=true");
        final eventList = json.decode(response.body);
        print(eventList);
        for(int i=0;i<eventList.length;i++){
            if(eventList[i]["Status"]=="A"){
                eventPosterList.add("http://xiaosha007.hostingerapp.com/EventPoster/"+eventList[i]["EventID"]);
            }
        }
        setState(() {
            loadingSpinner = false;
            _refreshController.refreshCompleted();
        });
    }
}

```

1. This is the class that inherit from State<ClassTYPE> widget.
2. For every createState() method called, they must return the State of the page, which is the body of the page. Which means, this class is to create a body for the classes that inherited StatefulWidget.
3. Therefore, we can implement all the application logic and GUI in this class.

```

@Override
Widget build(BuildContext context) {
    return ModalProgressHUD(
        child: Scaffold(
            //backgroundColor: Colors.grey[600],
            drawer: Drawer(
                child: Column(
                    children: <Widget>[
                        AppBar(
                            automaticallyImplyLeading: false,
                            title: Text("Choose"),
                        ), // AppBar
                        ListTile(
                            leading: Icon(Icons.search),
                            title: Text("Search"),
                            onTap: () {
                                Navigator.push(
                                    context,
                                    MaterialPageRoute(
                                        builder: (BuildContext context) => Search())); // MaterialPageRoute
                            },
                        ), // ListTile
                        widget.userID!=null ? ListTile(
                            leading: Icon(Icons.event),
                            title: Text("Event"),
                            onTap: () {
                                Navigator.push(context, MaterialPageRoute(
                                    builder: (BuildContext context)=>UserEventListPage(widget.userID,widget.userType)
                                )); // MaterialPageRoute
                            },
                        ):Container( // ListTile
                            color: Colors.white,

```

1. Inside the class that inherits State<ClassTYPE>, a method called build() must be overridden. Inside the build method, it will build the body of the app.
2. If there is a need to change the state of the page, a method named setState is called.

Conclusion and Future Plans

To us, the ALUMNI CENTER is beyond an assignment of this Software Engineering Fundamentals course. By hook or by crook, we go through the three months of hard work and effort in planning, modelling, constructing and deploying this application. To develop an application using language and framework that is unfamiliar to us, and to come out with a quality prototype, is indeed a great achievement and something we did not anticipate in the initial stage of the project.

In the process of developing this application, we came across several doubts and also conflicts among the team. Even a mobile application with minimum functionalities turns out to be a challenge as we need to follow the proper standards/coding conventions and eventually come up with efficient code which is maintainable.

We strive hard to find alternative ways to create a better program that runs smoothly with less issues by looking at useful suggestions and resources. Through this learning process, we realize that fulfilling customer requirements is the major concern of software engineers. Comparing to our previous work in Mini IT Project, we found significant improvements in the way we approach to this assignment. Undoubtedly, our application has room for improvements. Compared to the commercial application found in google play store, we are still far away from them in terms of UI design, efficiency and functionality.

Perhaps in the near future, we can improve the application to include more administrative functions to learn about users of this application, and then release it on Google Play Store and also Apple App Store for others to benefit from it. With sufficient budget and solid technical skills, we might touch up the whole project to enhance its core features and provide a better finishing for this project. Till this moment, we have pushed our limits and this is the best we have produced.

References

References

- (n.d.). Retrieved from Multitier architecture: Three-tier architecture:
https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture
- (2017, 7). Retrieved from Software Engineering Design:
<http://softwareengineeringmca.blogspot.com/2017/07/>
- Brandenburg, L. (n.d.). Retrieved from Briding the gap: <https://www.bridging-the-gap.com/what-is-a-use-case/>
- C.Martin, R. (n.d.). The Principles of Object Oriented Design. Retrieved from butunclebob:
<http://www.butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>
- Fakhroutdinov, K. (n.d.). Retrieved from UML Class Diagrams Reference: <https://www.uml-diagrams.org/class-reference.html>
- Pressman, R. S. (2010). Retrieved from Software Engineering: A practitioner approach (8th edition): http://dinus.ac.id/repository/docs/ajar/RPL-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman_.pdf
- S.Pressman, R. (2014). Retrieved from Software Engineering: A practitioner approach (7th edition):
[https://dishantrathi.keybase.pub/Semester%206%20Material/Software%20Engineering%20A%20Practitioner%E2%80%99s%20Approach%20eighth%20edition-\(www.downloadnema.com\).pdf](https://dishantrathi.keybase.pub/Semester%206%20Material/Software%20Engineering%20A%20Practitioner%E2%80%99s%20Approach%20eighth%20edition-(www.downloadnema.com).pdf)
- Sample Test Case Template With Test Case Examples.* (2019, 8 21). Retrieved from Software Tesing Help: <https://www.softwaretestinghelp.com/test-case-template-examples/>
- Sommerville, I. (2015). Retrieved from Software Engineering 10th edition:
<http://dinus.ac.id/repository/docs/ajar/Sommerville-Software-Engineering-10ed.pdf>