University Finder

By

Wazir Afrasiyab Hussain	13-SE-40
Tayyab Muzamal	13-SE-32
Asif Ahmad sheikh	13-SE-41

Project Supervisor

Engr. Afzal Ahmad

Jr. Lecturer

Department of Software Engineering

Mirpur University of Science and Technology

August, 2017

CERTIFICATION

It is certified that the following students have completed their project on

“**University Finder**”

Submitted By:

Name	**Roll No**
Tayyab Muzamal	13-SE-32
Wazir Afrasiyab Hussain	13-SE-40
Asif Ahmad sheikh	13-SE-41

Session: 2013-17

This report is submitted in partial fulfillment of the requirements of the award of degree of **Bachelors of Software Engineering**. Mirpur University of Science and Technology (MUST)

External Examiner

Project Supervisor
(Engr. Afzal Ahmad)

Chairman

Department of Software Engineering

MUST Mirpur (A.K

Declaration

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed in this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

ABSTRACT

Currently students are facing many problems for searching of universities by reading newspapers, search on internet and visit each university Site to find courses offered, their criteria and schedule of admission. This is very difficult, time consuming and costly task. Some time students are unable to get admission in their desire University due to lack of information about degree program and time schedule. To overcome this problem, we will design an application to facilitate students which will help users to find the suitable university as well as that will match to their academic abilities. Our target audience for this kind of tool is primarily students. Best time when students aren't explicitly connected to an institution where they are able to obtain guidance and can easily compare different universities from every aspect. We are looking forward for ways to fill up the gap between students and institutes with the help of latest technology which will be definitely improve the patterns of polishing someone's skills by choosing the right platform without wasting the precious time of their life, instead investing it in a right direction.

Our application allows users to find the university by field of study, degree programs, location, ranking and name.

System will show the list of top universities that having information about their offered courses, duration, admission schedule, pre-requisites, fee information and address of every universities to the end users. Result will be filtered to the users for their best match.

UNDERTAKING

We certify that research work titled "*University Finder*" is our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged/ referred.

Tayyab Muzamal
13-SE-32

Wazir Afrasiyab Hussain
13-SE-40

Asif Ahmad Sheikh
13-SE-41

Acknowledgements

We thank all who in one way or another contributed in the completion of this thesis. First, we give thanks to Allah almighty for protection and ability to do work. We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them.

We would like to express my gratitude towards my parents & family members for their kind co-operation and encouragement which help me in completion of this project.

We would like to express my special gratitude and thanks to FYP committee for giving us such attention and time.

Our thanks and appreciations also go to our class fellows and seniors who have willingly helped us out with their abilities.

Contents

Figures

Tables

Chapter 1 Introduction

University Finder application can help students to find the platform of their dreams based on the criteria that have the most significance to them. Students can easily find a university that is in the desired location. University Finder application let users to choose from variety of options. This application will be a great place to search for degree programs.

1.1 Goals

This Android application will provide a modern UI Application with clean simplified design with text oriented navigation and simple button navigation which is suitable touch user interface. A touch sensitive application for Android tablet and Smart Phones that are used now-a-days. Easy access to all university information. Students can easily search about universities in a specific location. Students can search universities according to their field of study. Registered students get notification about admissions in universities. Guest user can also search about universities.

1.2 Motivation

Finding the right university is not easy and it is one of the most important decisions anyone will make in their lifetime. University Finder application can help students to find the platform of their dreams based on the criteria that have the most significance to them. Students can easily find a university that is in the desired location. University Finder application let users to choose from variety of options. This application will be a great place to search for degree programs.

1.3 Method

We will design an application to facilitate students which will help users to find the suitable university as well as that will match to their academic abilities. Our target audience for this kind of tool is primarily students. Best time when students aren’t explicitly connected to an institution where they are able to obtain guidance and can easily compare different universities from each aspect. We are looking forward for ways to fill up the gap between students and institutes with the help of latest technology which will be improve the patterns of polishing someone’s skills by choosing the right platform without wasting the precious time of their life, instead investing it in a right direction. [2]

1.4 Overview

In daily life selecting or finding the university is the most difficult job for a student who is just out of college. Reading different newspapers, viewing ads and searching on different resources for selecting the university takes a lot of time.

To succeed in our lifelong educational journey, we will need accurate, dependable and practical tools with reliable resources. University Finder is an android application which is used to find the universities. This application is very reliable for student to find courses that universities offers, pre-requisites and fee schedules. By using this application, the user will save a lot of their time.

Chapter 2 Background and Problem Statement

2.1 Introduction

Sometime students take admission and after that they came to know other universities offering same courses in good packages while some time students are unable to get admission in their desire University due to lack of information about degree program and time schedule. For that purpose, we made an application through which Students can easily find a university that is in the desired location. University Finder application let users to choose from variety of options. This application will be a great place to search for degree programs. [1]

2.2 Literature Review

Some applications already exist on the Google Play store related to.

2.3 Problem Statement

Currently students must face the following problems.

- Read newspapers
- Search on internet and visit each and every university Site to find courses offered, their criteria and schedule of admission.
- This is very difficult, time consuming and costly task.
- Some time students are unable to get admission in their desire University due to lack of information about degree program and time schedule.
- Research shows that students from lower-income backgrounds tend to select the university which is not well matched to their academic abilities or in term of finance will be a bad choice.
- Sometime students take admission and after that they came to know other universities offering same courses in good packages

Chapter 3 Project Management

3.1 Approach

The agile methodology for mobile application development seeks to provide an alternative to traditional methods of project management such as the waterfall methodology. This approach enables requirements and solutions to evolve through the combined effort of the development team and the customer. It promotes adaptive planning, evolutionary development, early delivery and continuous improvements. This iterative and flexible approach can be used in complex projects where the customer requirements change frequently. [3]

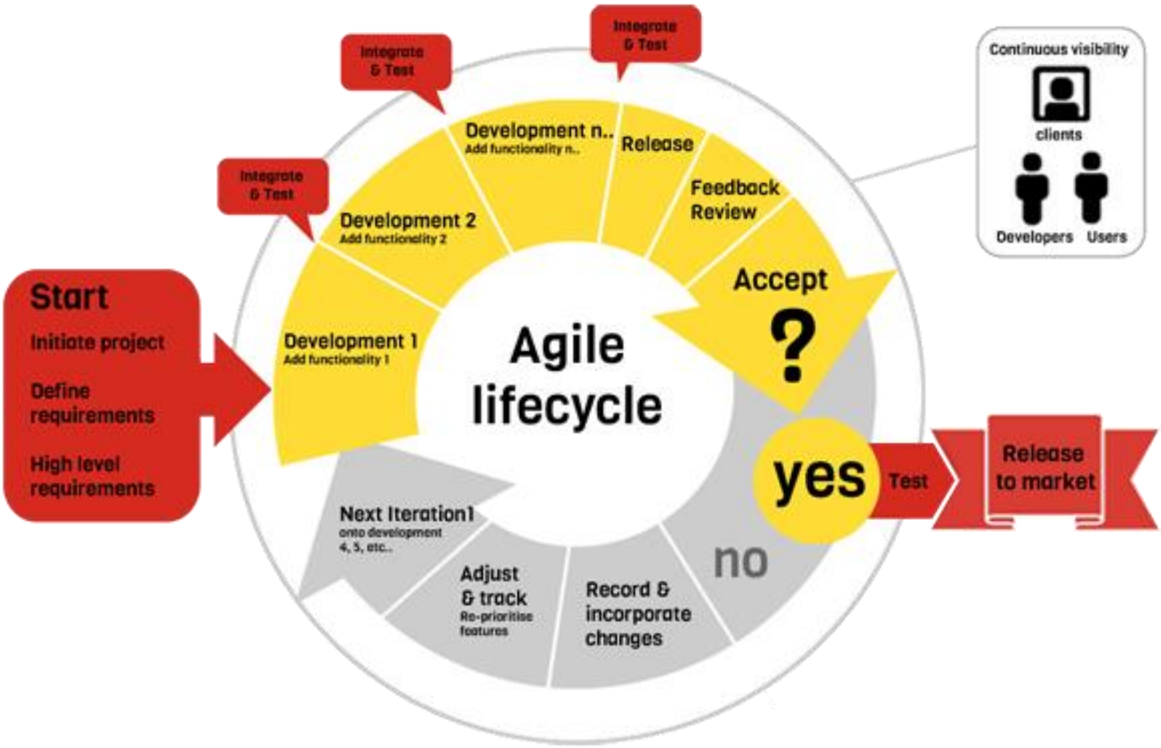


Figure 3-1 Agile Process Modelling Diagram

Chapter 4 Analysis

The following section presents the complete set of functional and non-functional requirements identified for UFA. Functional requirements are listed first, according to their relationship to the overall system, students and administrator. The non-functional requirements that pertain to safety, security, accuracy, reliability, accessibility, usability, maintenance and performance are subsequently presented.

4.1 Problem Modelling

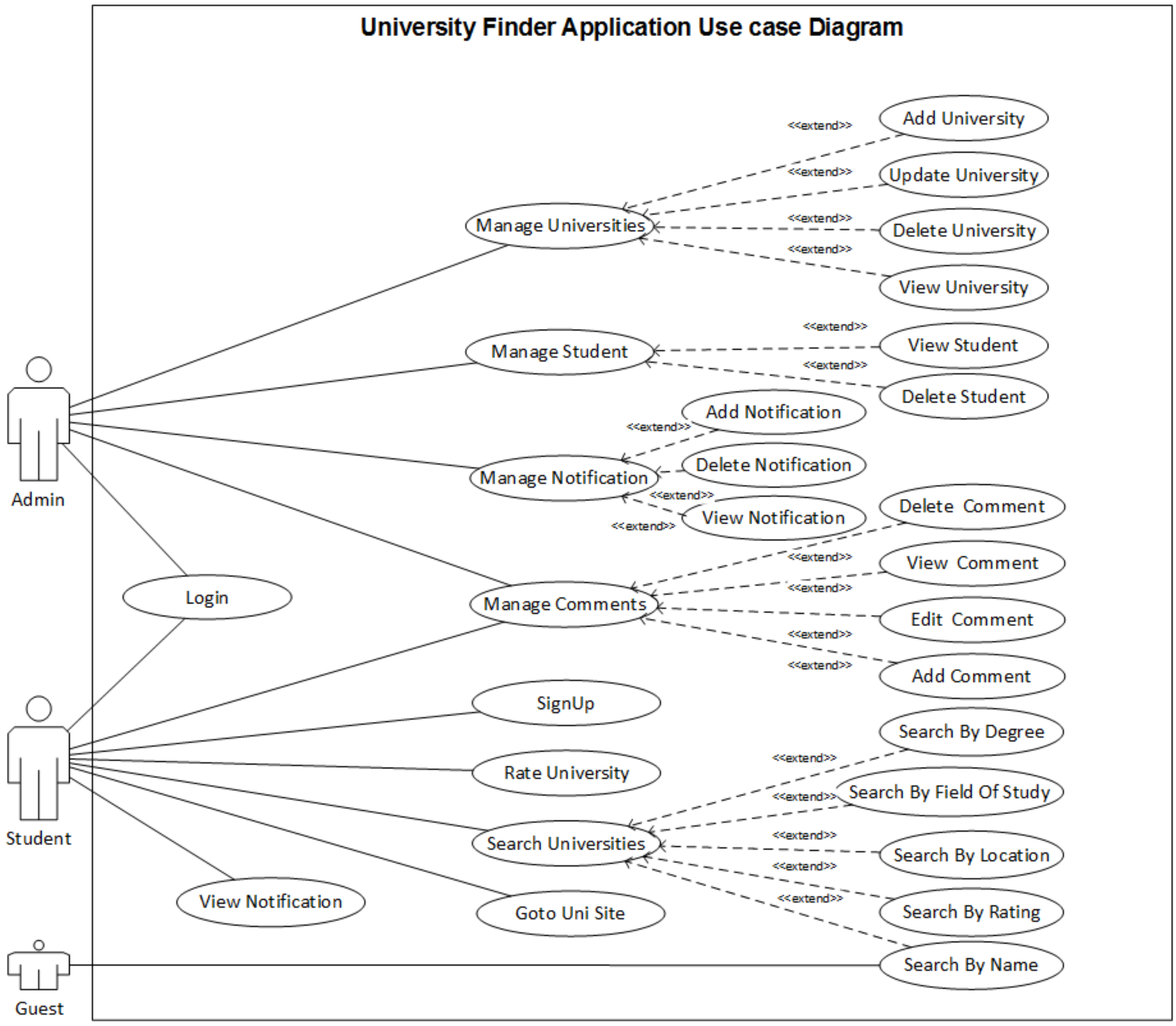


Figure 4-1 Use Case Diagram Overall Project

4.2 Functional Requirements

This subsection presents the identified functional requirements. The requirements are gathered at higher perspective level and drawn below in tabular form. [5]

Table 4-1 Functional Requirements

ID	Requirement	Justification
REQ-1	**Signup**	User create an account to use system.
REQ-2	**Login**	User login into system using email and password as valid user.
REQ-3	**Add University**	Admin add new university detail.
REQ-4	**Update University**	Admin update the university detail.
REQ-5	**View University**	Admin view selected university detail.
REQ-6	**Delete University**	Admin delete the selected university.
REQ-7	**Go to Site**	User access the site of university.
REQ-8	**View Students**	Admin view all register student.
REQ-9	**Delete Students**	Admin delete selected student.
REQ-10	**Add Notification**	Admin add notification.
REQ-11	**Delete Notification**	Admin delete selected notification.
REQ-12	**View Notification**	Student view notification about new events, admissions etc.
REQ-13	**Edit Notification**	Admin update notification.
REQ-14	**Edit Profile**	Student edit his/her profile.
REQ-15	**Delete Account**	Student delete his/her account.

REQ-16	**Guest Search**	Guest user search the universities only by entering name of university.
REQ-17	**Add Comments**	Registered user post comments about universities.
REQ-18	**Delete Comments**	Admin or comment's author delete posted comments.
REQ-19	**View Comments**	User view comments about university.
REQ-20	**Search by Name**	Student search universities by entering the name of university.
REQ-21	**Search by Field of Study**	Student search universities by selecting field of study.
REQ-22	**Search by Location**	Student search universities by entering specific location.
REQ-23	**Search by Ranking**	Student find universities by ranking.

4.3 Non-Functional Requirements

Non-functional requirements are the quality requirements that stipulate how well a software does what it has to do. These are Quality attributes of any system; these can be seen at the execution of the system and they can also be the part of the system architecture.

Accuracy:

The system will be accurate and reliable based on the design architecture. If there is any problem in the accuracy, then the system will provide alternative ways to solve the problem. If the user provides the wrong information, then system will show the pre-defined messages.

Usability:

The proposed system will be simple and easy to use by the users. The users will comfort in order to communicate with the system. The user will be provided with an easy interface of the system.

Accessibility:

The system will be accessible through internet and there should be no any known problem.

Performance:

The system performance will be at its best when performing the functionality of the system.

Reliability:

The proposed system will be reliable in all circumstances and if there is any problem that will be affectively handle in the design.

Security:

The proposed system will be highly secured; every user will be required registration and username/password to use the system. The system will do the proper authorization and authentication of the users based on their types and their requirements. The proposed system will be designed persistently to avoid any misuse of the application.

Chapter 5 Product/System Design

5.1 Introduction

We design our system by conducting meeting of our team, in which we discuss main features of our system. Then we design SRS based on collected requirements and we moved towards the use case to find out how our end users will interact with University Finder.

5.2 Product Features

- **Signup**

  User can sign up through his email address

- **Login**
  User can login through his/her email and password

- **Manage University**
  We can add, update, delete and view university information.

- **Manage Students**
  We can add student and delete student information.

- **Manage Notification**
  We can add, delete and view notification.

- **Manage Comments**

  We can add, edit and view comments.

- **Search Universities**
  Guest Users and Login Users can search any university.

5.3 User Interface

Main Activity

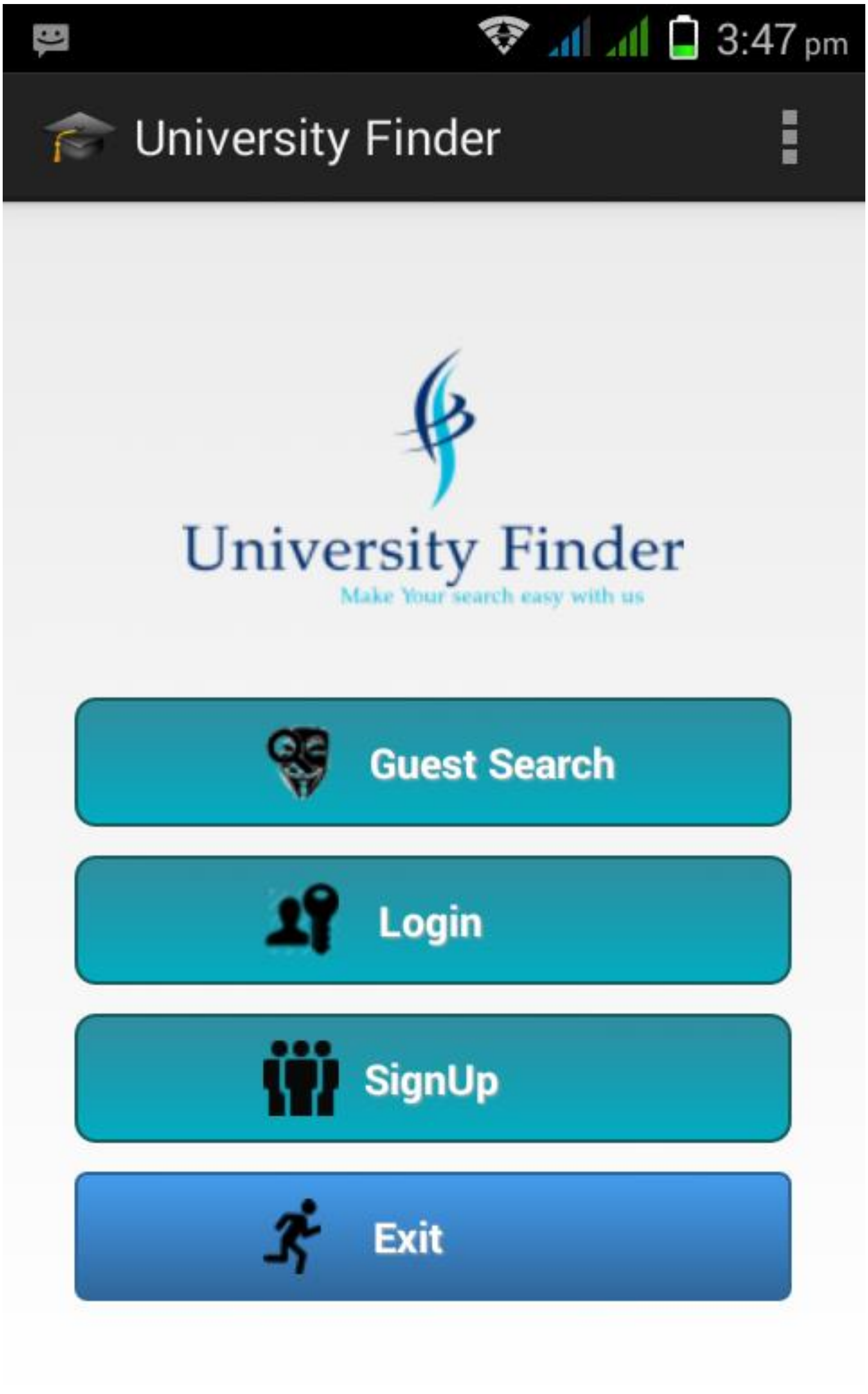
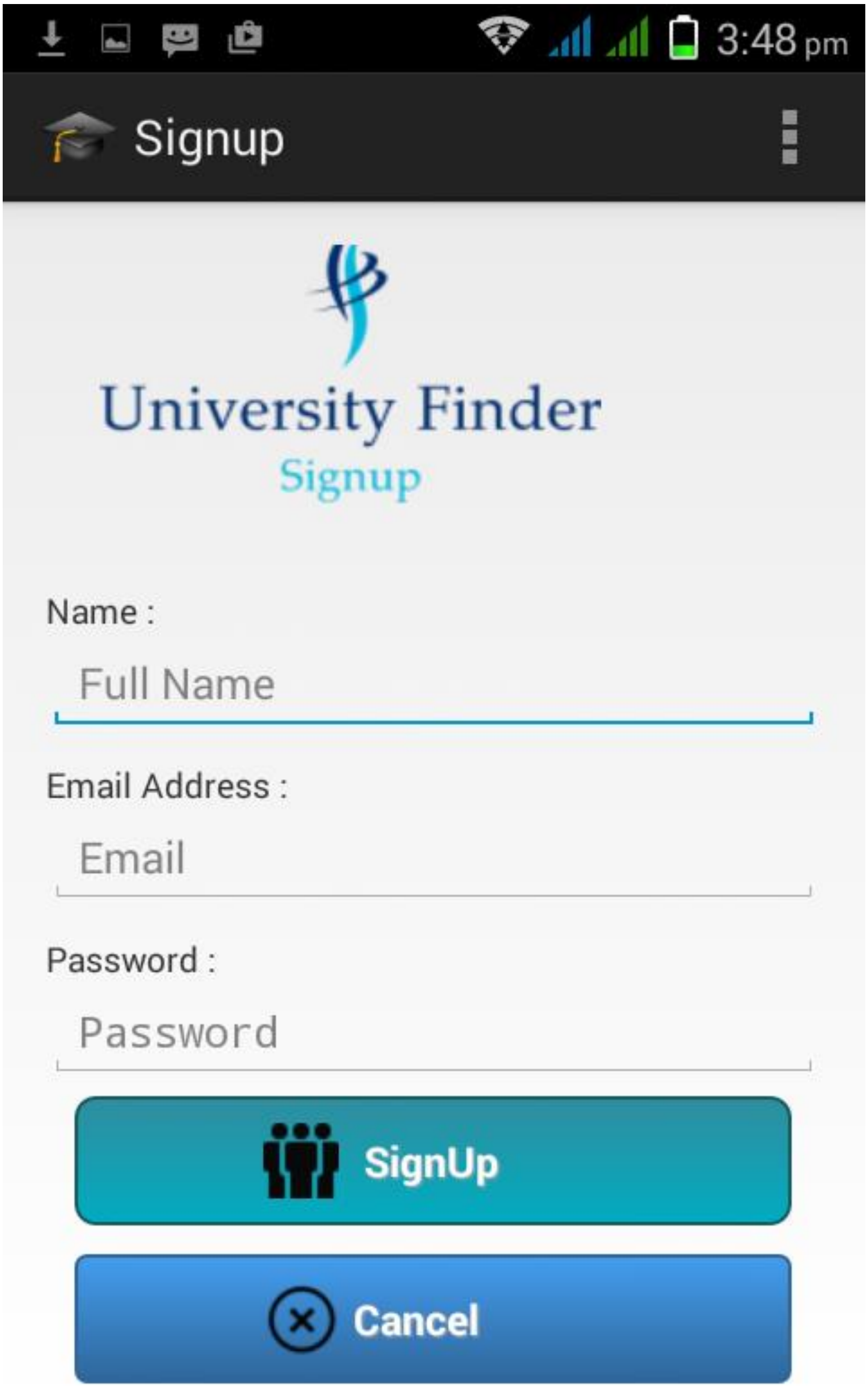


Figure 5-2 Main Activity

SignUp



3:48 pm

Signup

University Finder

Signup

Name :

Full Name

Email Address :

Email

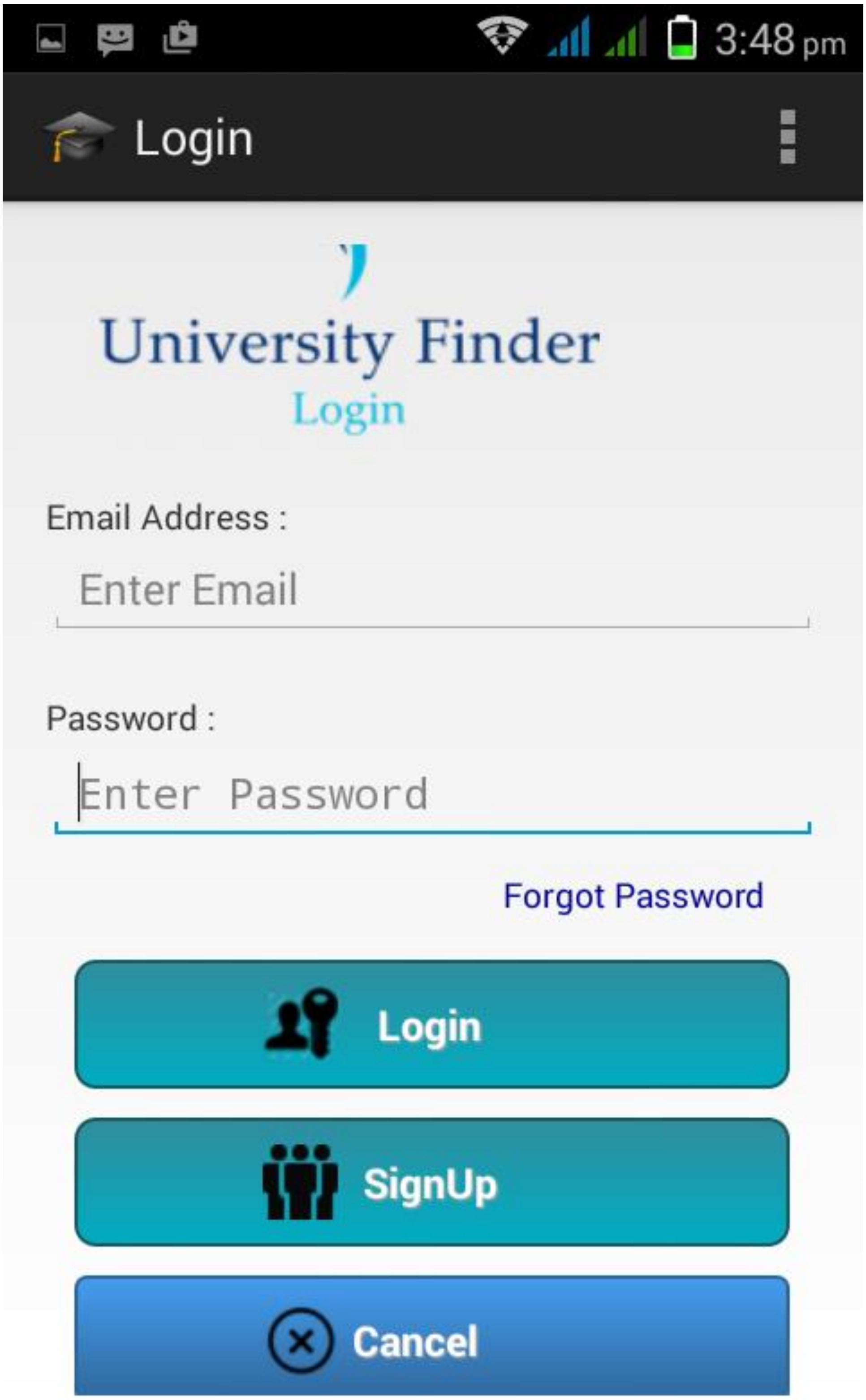
Password :

Password

Figure 5-3 Sign Up

Figure 5-4 Login

Login



Email Address : Enter Email

Password : Enter Password

Figure 5-4 Login

Admin Interface

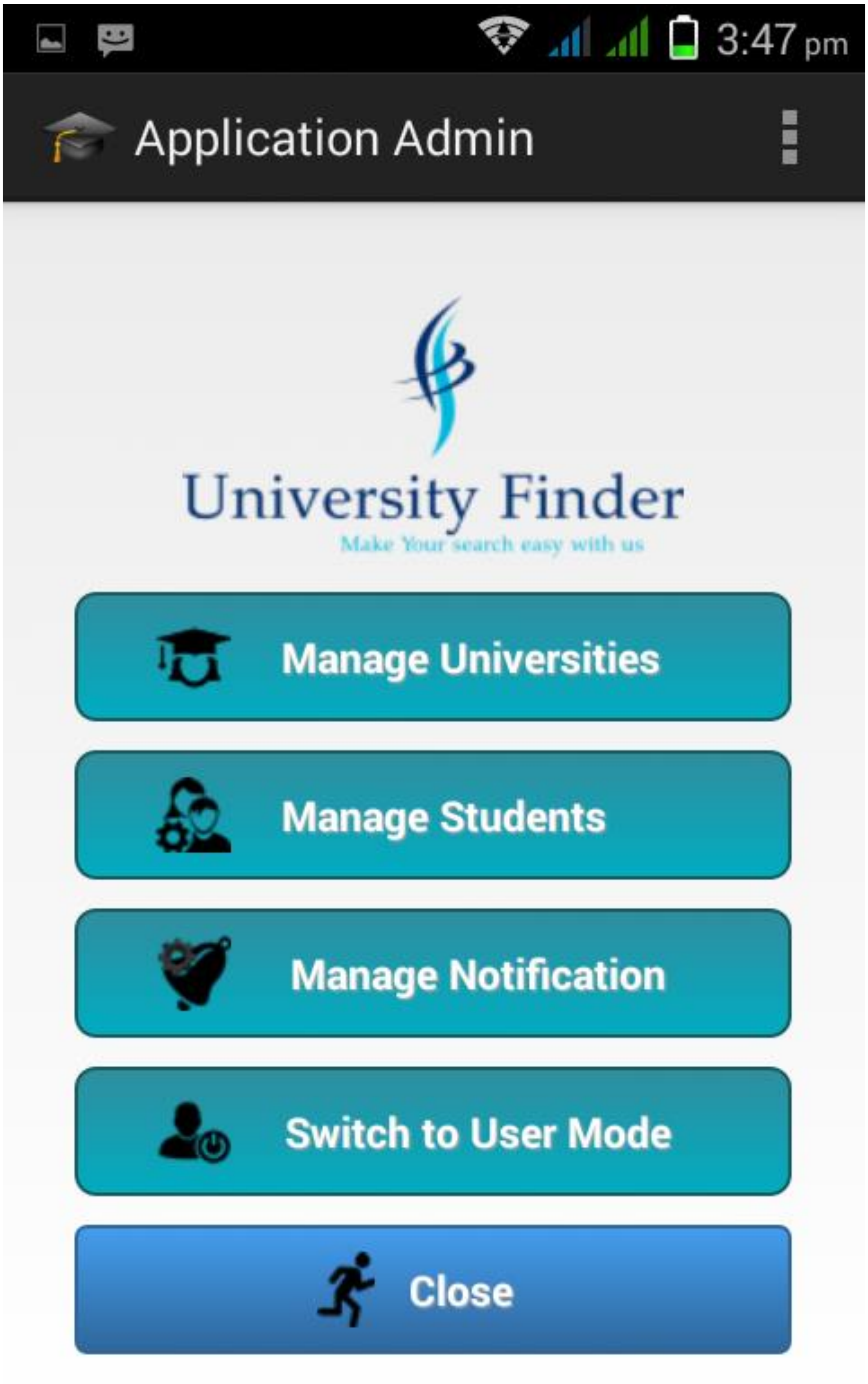


Figure 5-5 Admin Interface

Student Activity Interface

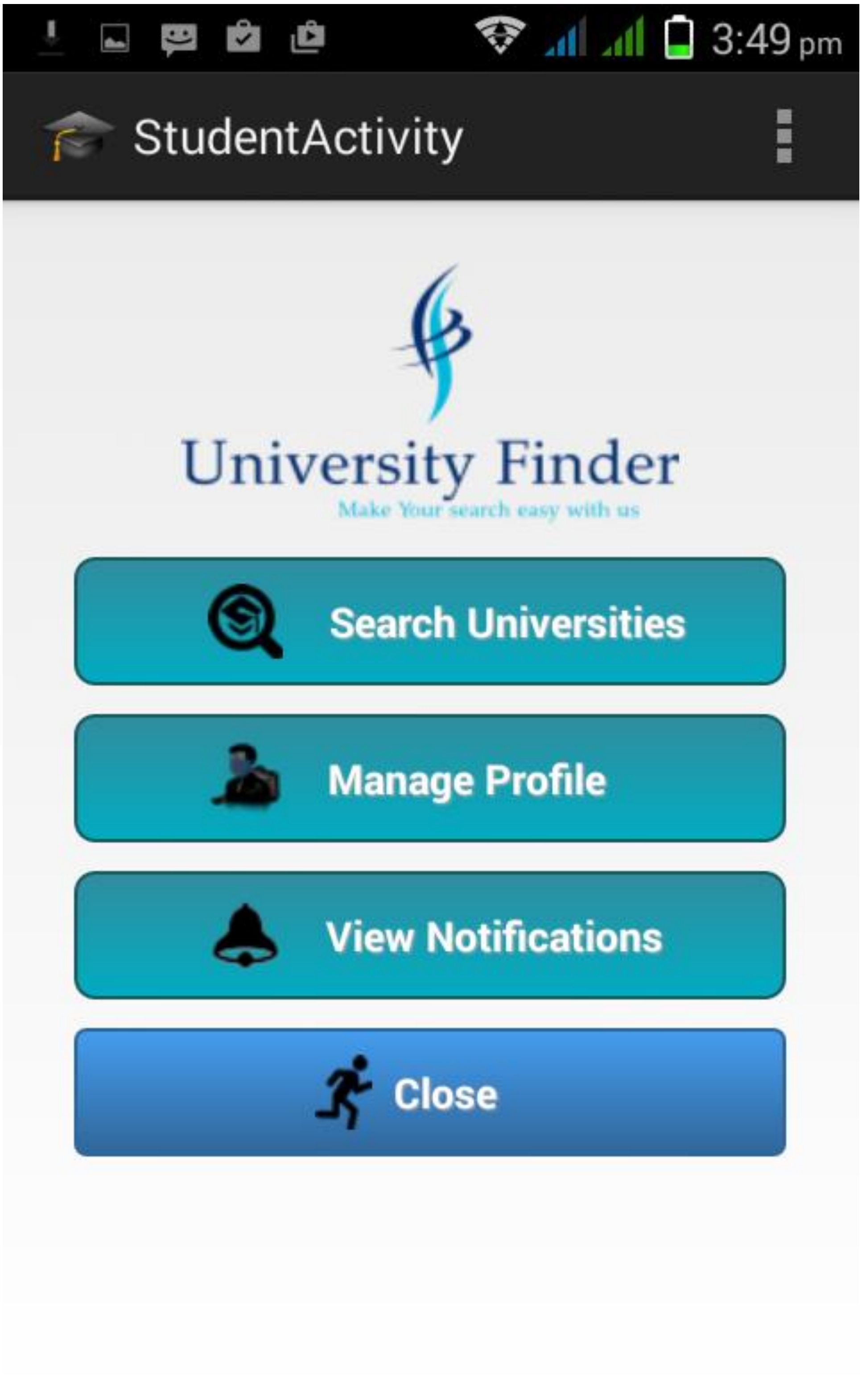


Figure 5-6 Student Activity Interface

Search University Interface

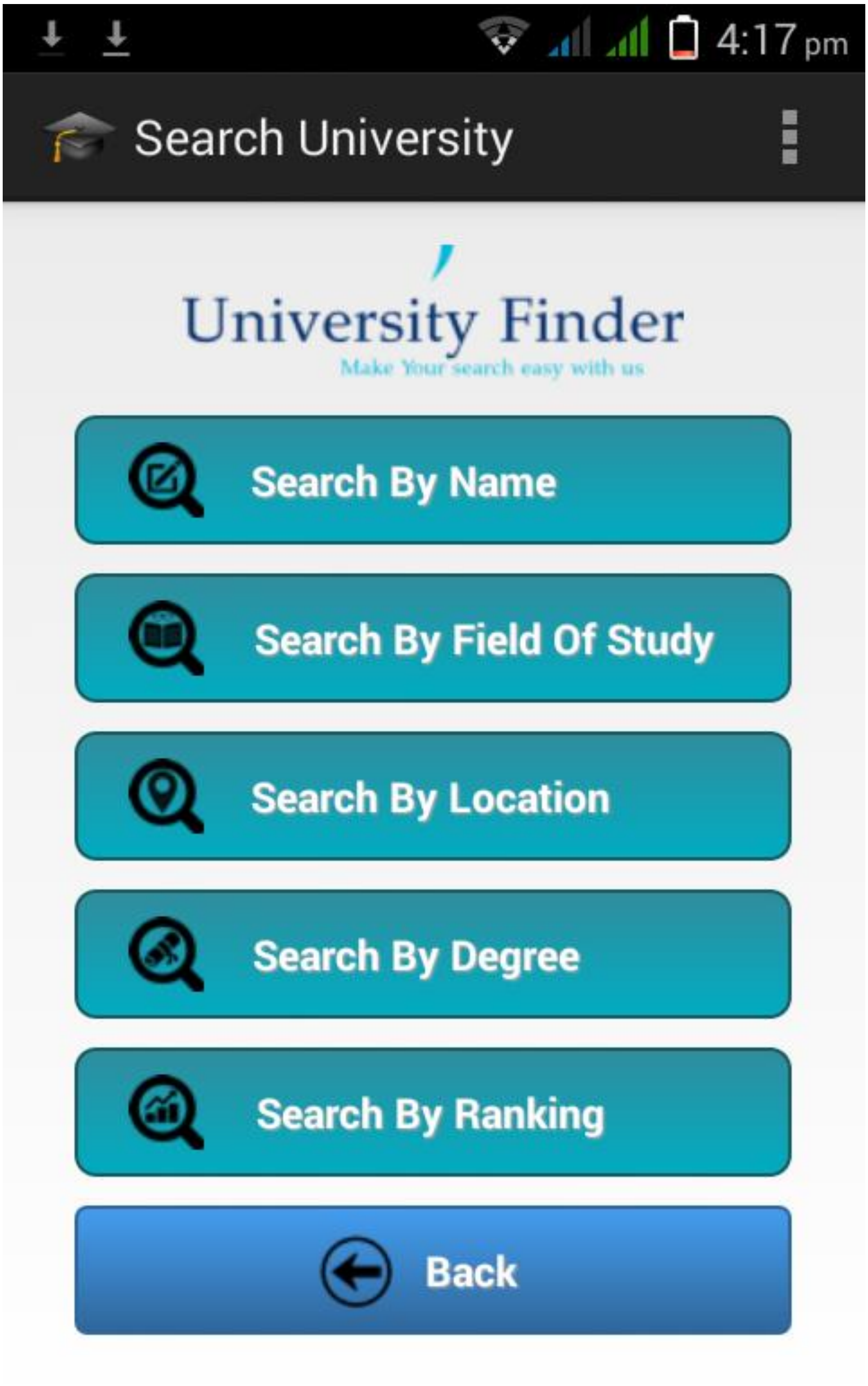


Figure 5-7 Search University Interface

Manage Profile Interface

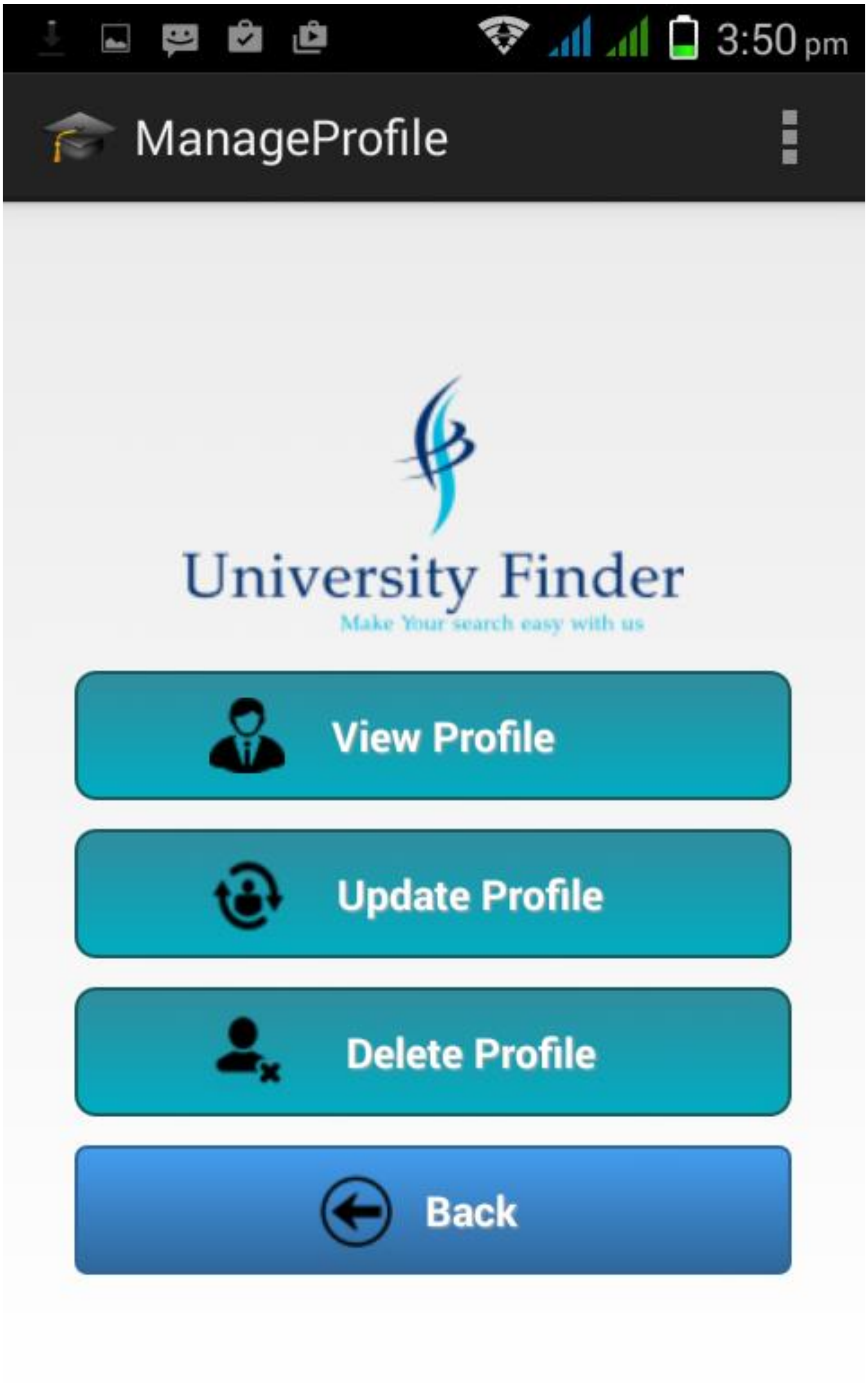
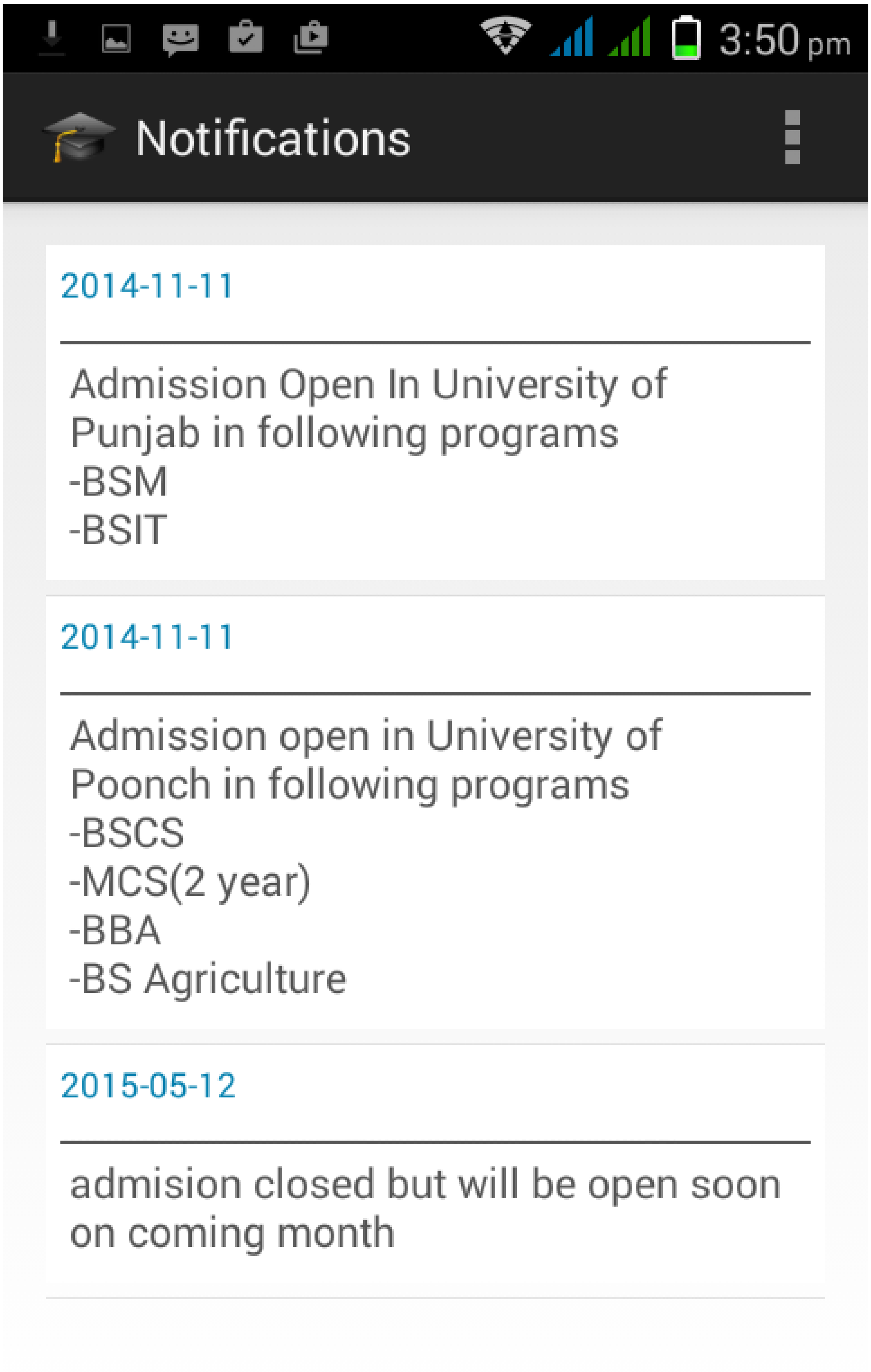


Figure 5-8 Manage Profile Interface

View Notifications



Notifications

2014-11-11

Admission Open In University of Punjab in following programs
-BSM
-BSIT

2014-11-11

Admission open in University of Poonch in following programs
-BSCS
-MCS(2 year)
-BBA
-BS Agriculture

2015-05-12

admision closed but will be open soon on coming month

Figure 5-9 View Notifications

5.4 Non-Functional Requirements

Non-functional requirements are the quality requirements that stipulate how well a software does what it has to do. These are Quality attributes of any system; these can be seen at the execution of the system and they can also be the part of the system architecture.

- **Accuracy:**

The system will be accurate and reliable based on the design architecture. If there is any problem in the accuracy then the system will provide alternative ways to solve the problem. If the user provides the wrong information then system will show the pre-defined messages.

- **Usability:**

The proposed system will be simple and easy to use by the users. The users will comfort in order to communicate with the system. The user will be provided with an easy interface of the system.

- **Accessibility:**

The system will be accessible through internet and there should be no any known problem.

- **Performance:**

The system performance will be at its best when performing the functionality of the system.

- **Reliability:**

The proposed system will be reliable in all circumstances and if there is any problem that will be affectively handle in the design.

- **Security:**

The proposed system will be highly secured; every user will be required registration and username/password to use the system. The system will do the proper authorization and authentication of the users based on their types and their requirements. The proposed system will be designed persistently to avoid any misuse of the application. [6]

Chapter 6 Software Design

6.1 Introduction

Two tier architecture model will be used. The main reason behind using this model is that there will be direct communication between the client and server. There is no intermediate between client and server.

6.2 High Level Design

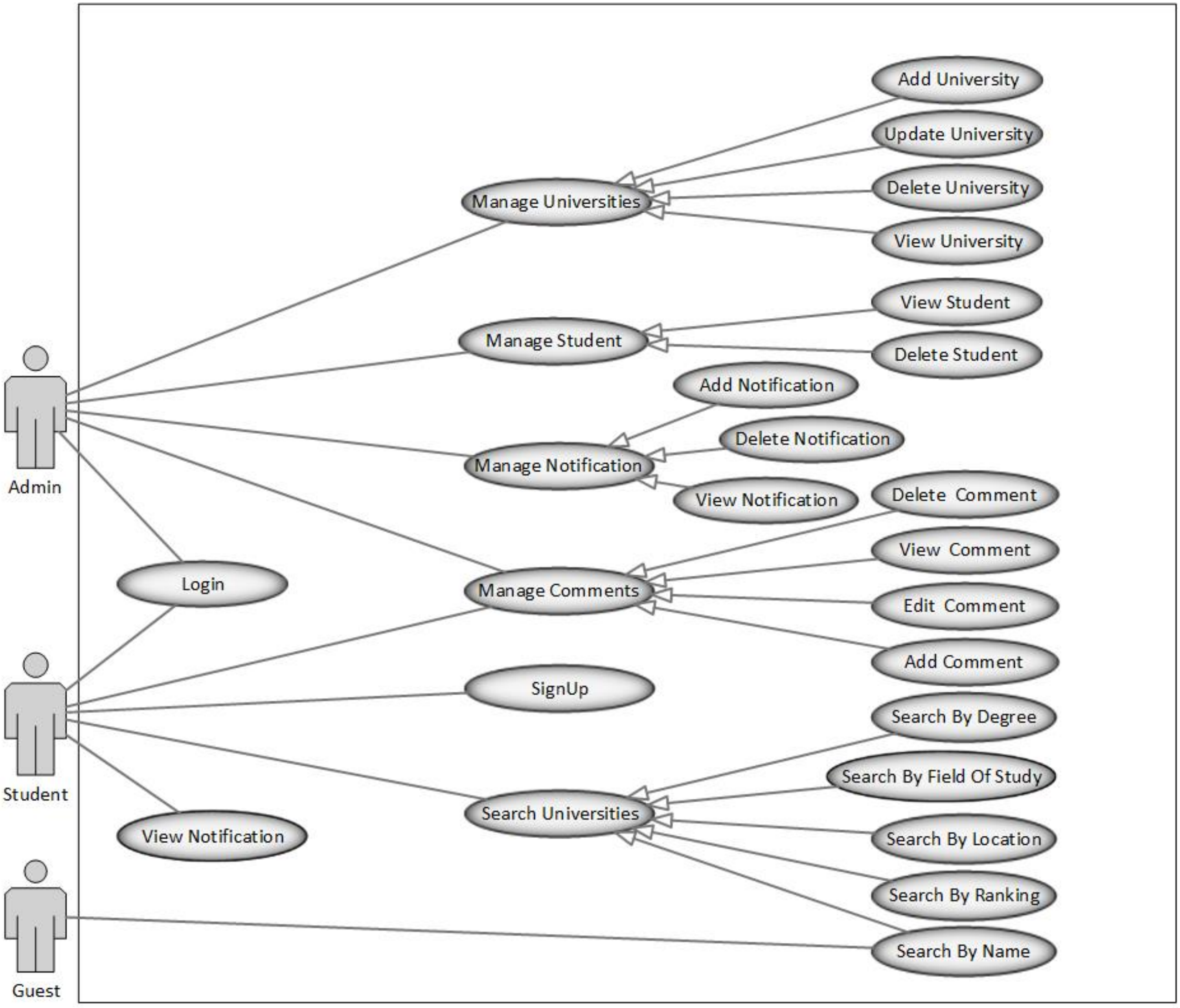


Figure 6-1 Use Case Diagram of UF

6.3 Detailed Design

Activity Diagram

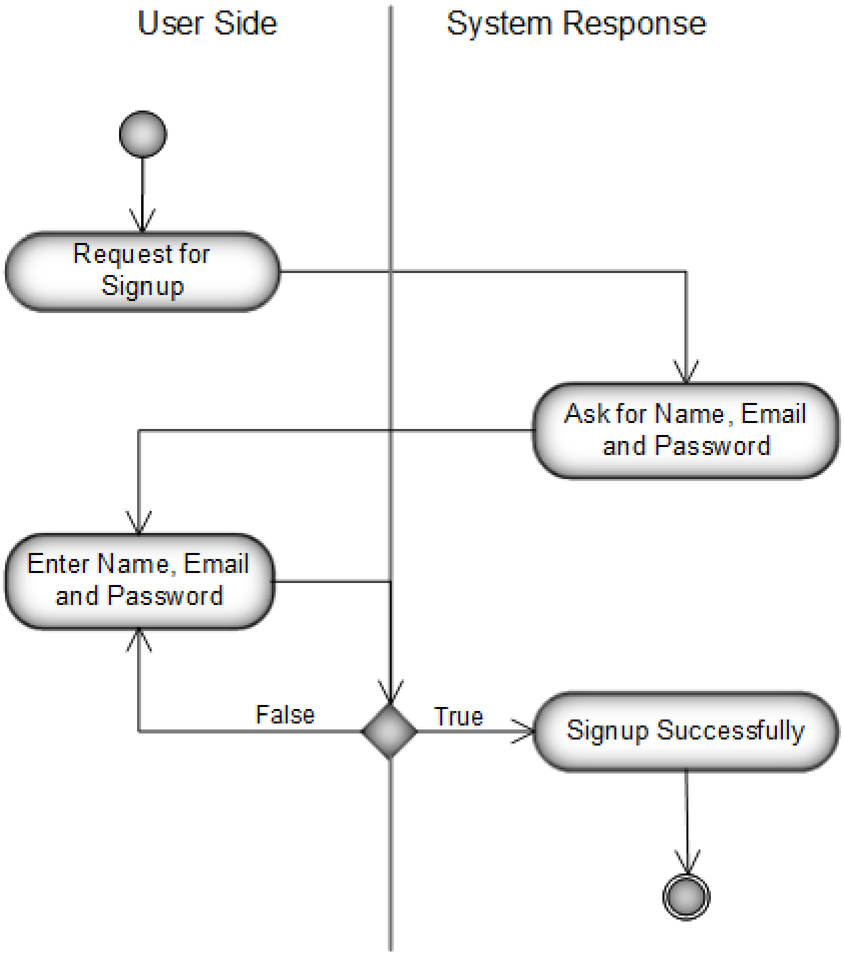


Figure 6-2 Overall Activity Diagram of UF

Login

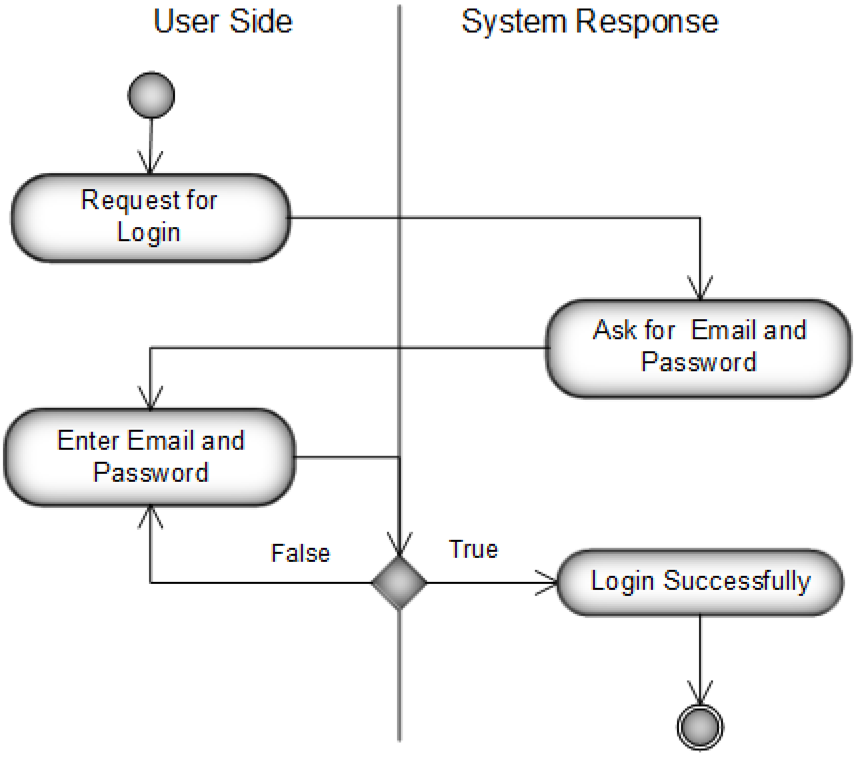


Figure 6-3 Activity Diagram to Login

Search by Location

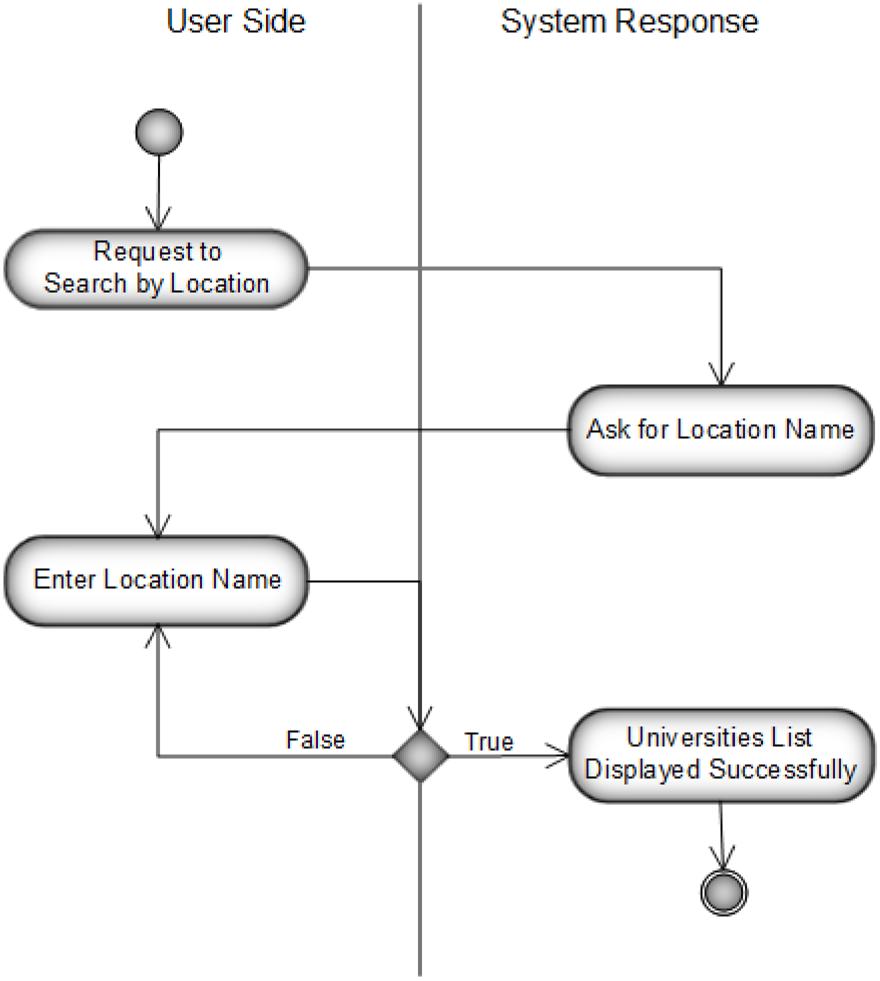


Figure 6-4 Activity Diagram for search location

Search by Field of Study

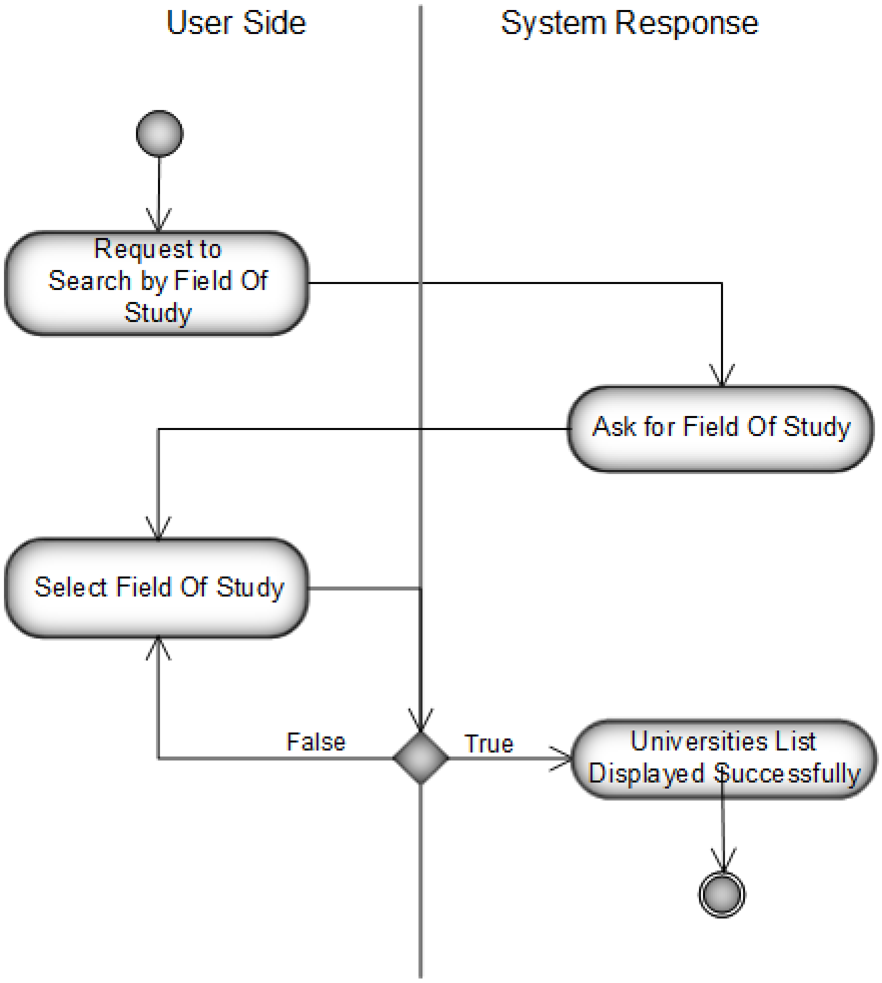


Figure 6-5 Activity Diagram to search by field of study

Search By Degree

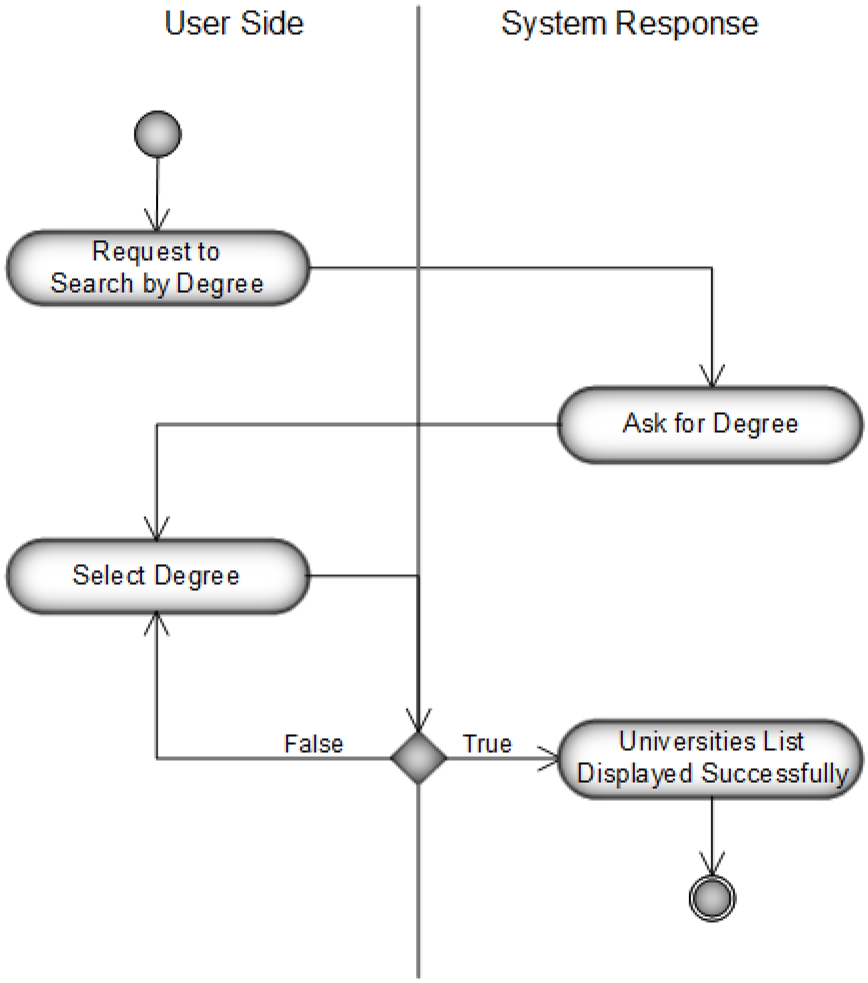

User Side
System Response
Request to Search by Degree
Ask for Degree
Select Degree
False
True
Universities List Displayed Successfully

Figure 6-6 Activity Diagram to search by degree

Search By Name

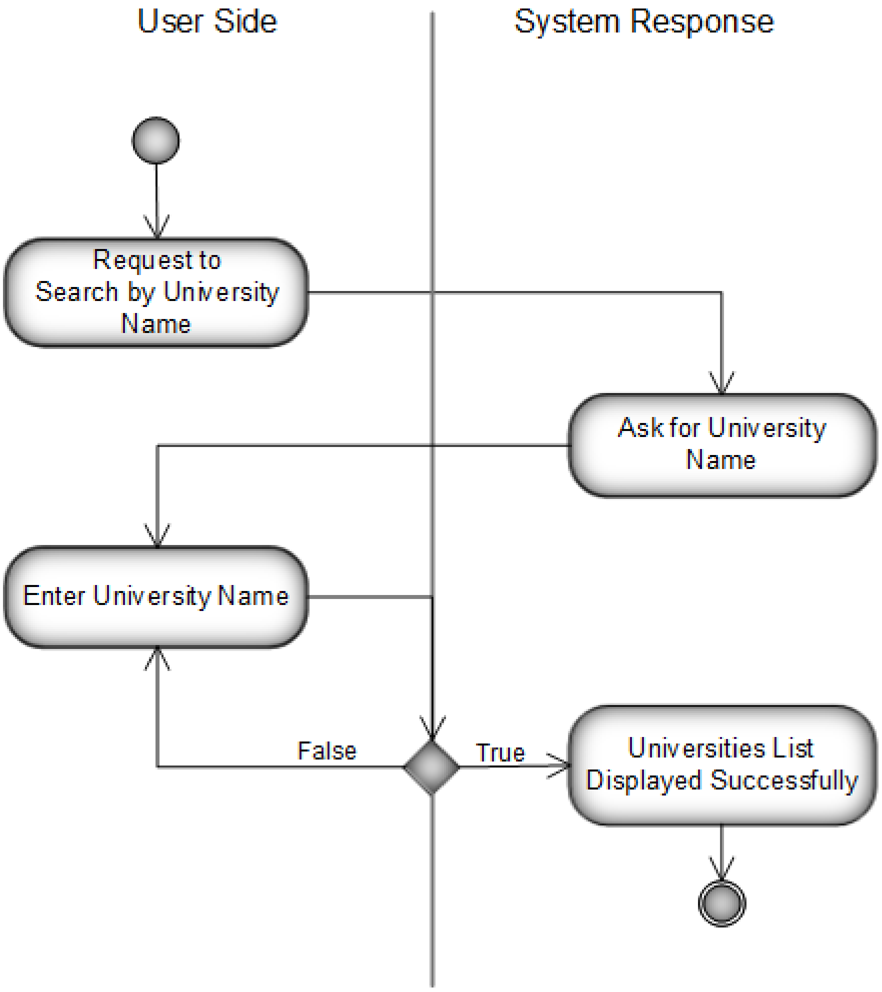


Figure 6-7 Activity Diagram to search by name

Search By Ranking

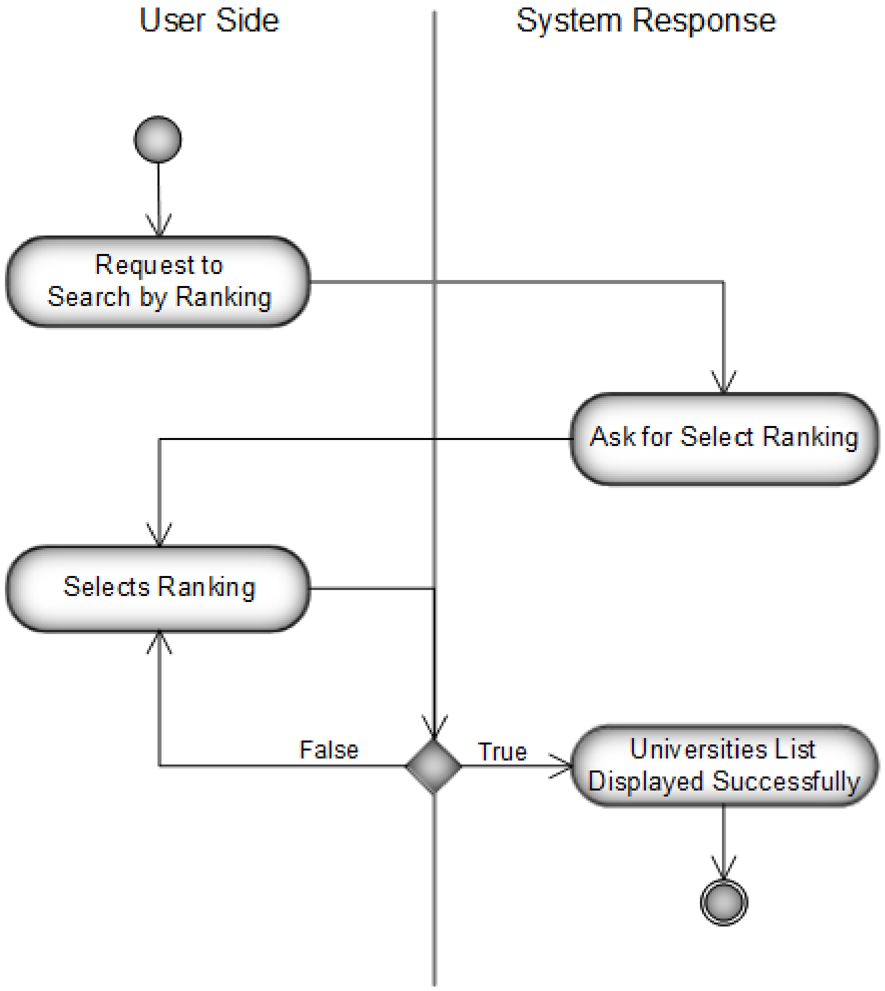


Figure 6-8 Activity Diagram to search by ranking

Edit Profile

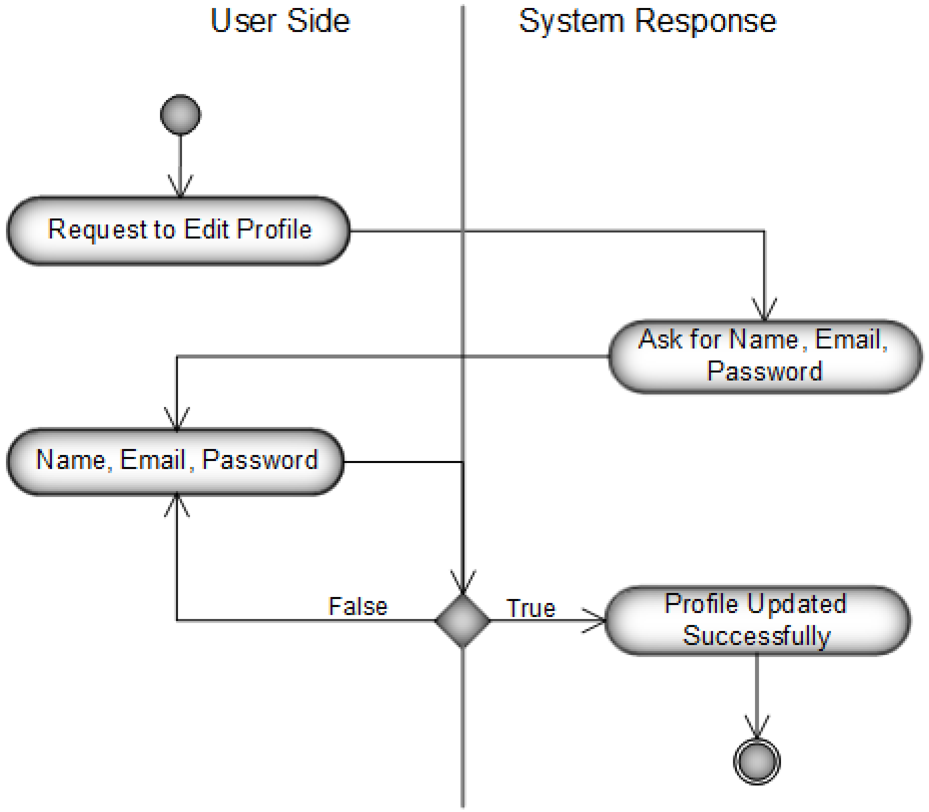


Figure 6-9 Activity Diagram to edit profile

Delete Account

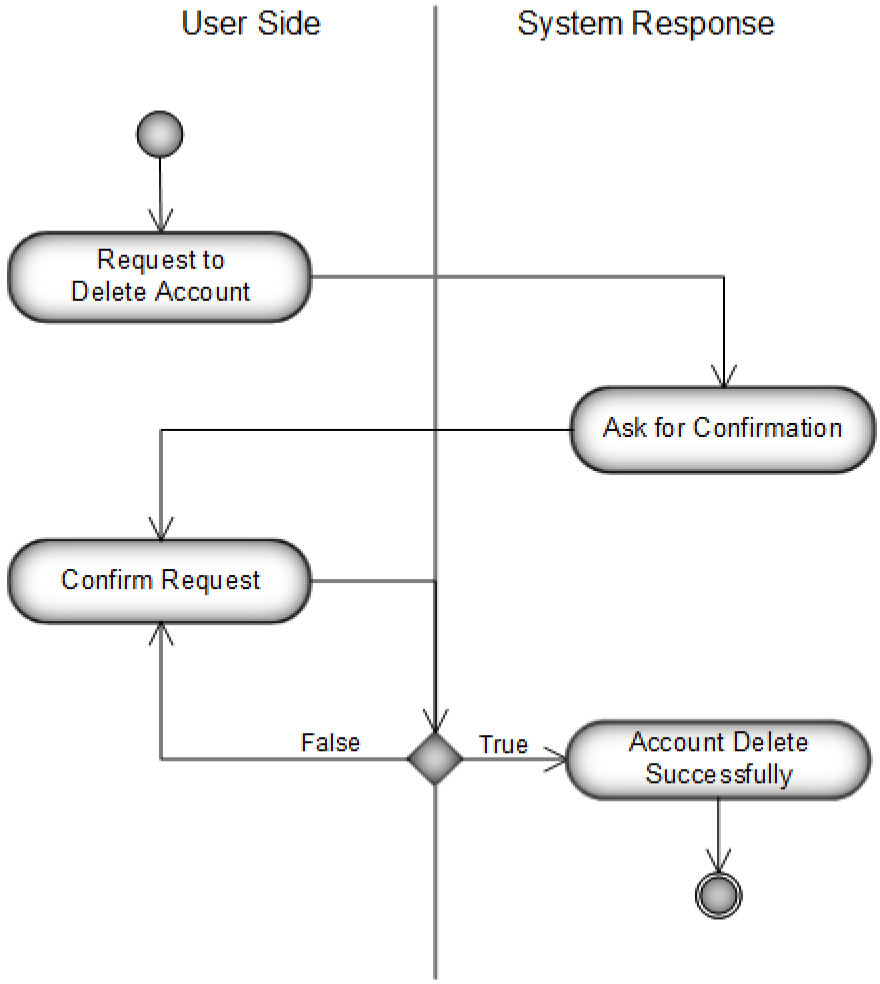


Figure 6-10 Activity Diagram to delete account

Add Comment

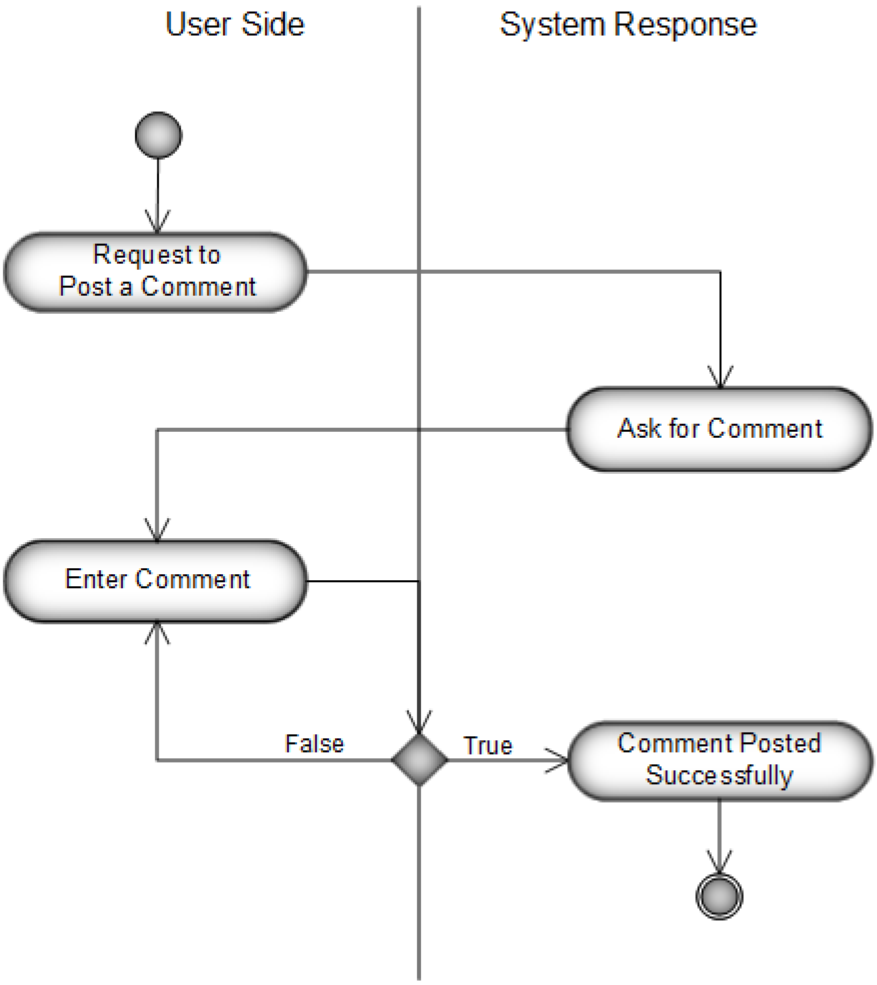


Figure 6-11 Activity Diagram to Add Comment

Edit Comment

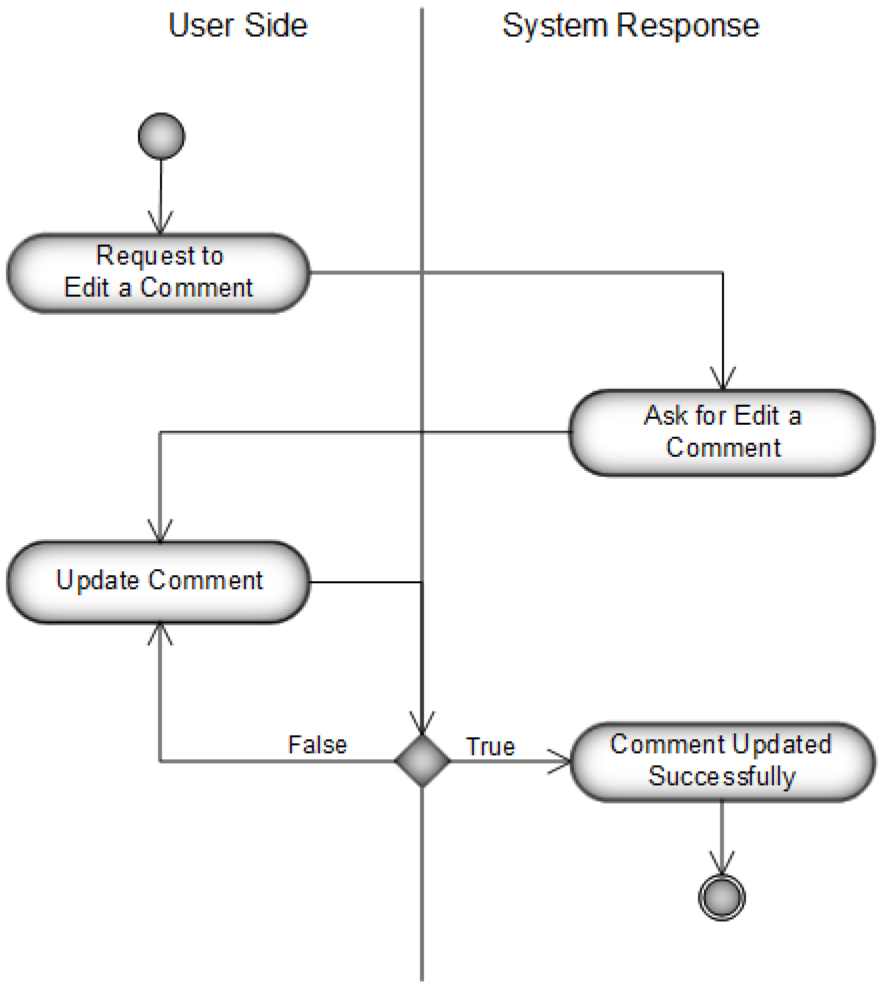


Figure 6-12 Activity Diagram to edit comment

Delete Comment

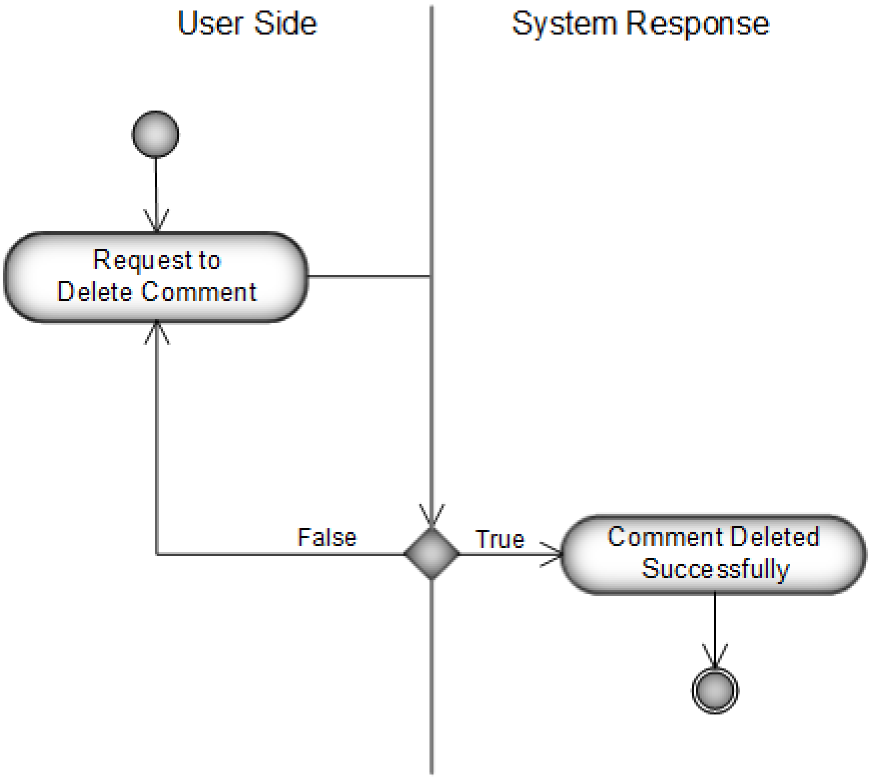


Figure 6-13 Activity Diagram to delete comment

View Students

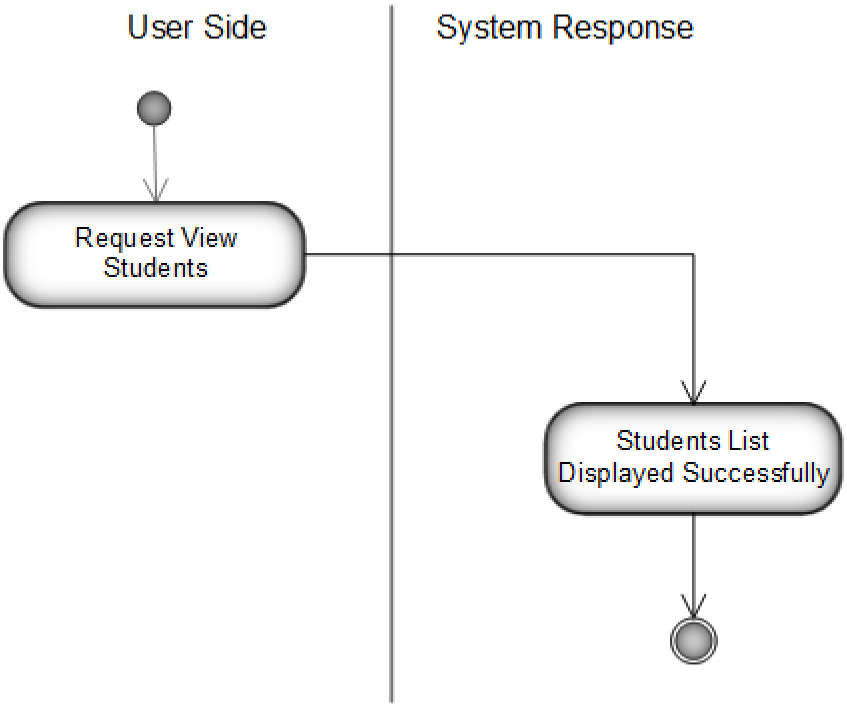


Figure 6-14 Activity Diagram to view student

Add Notification

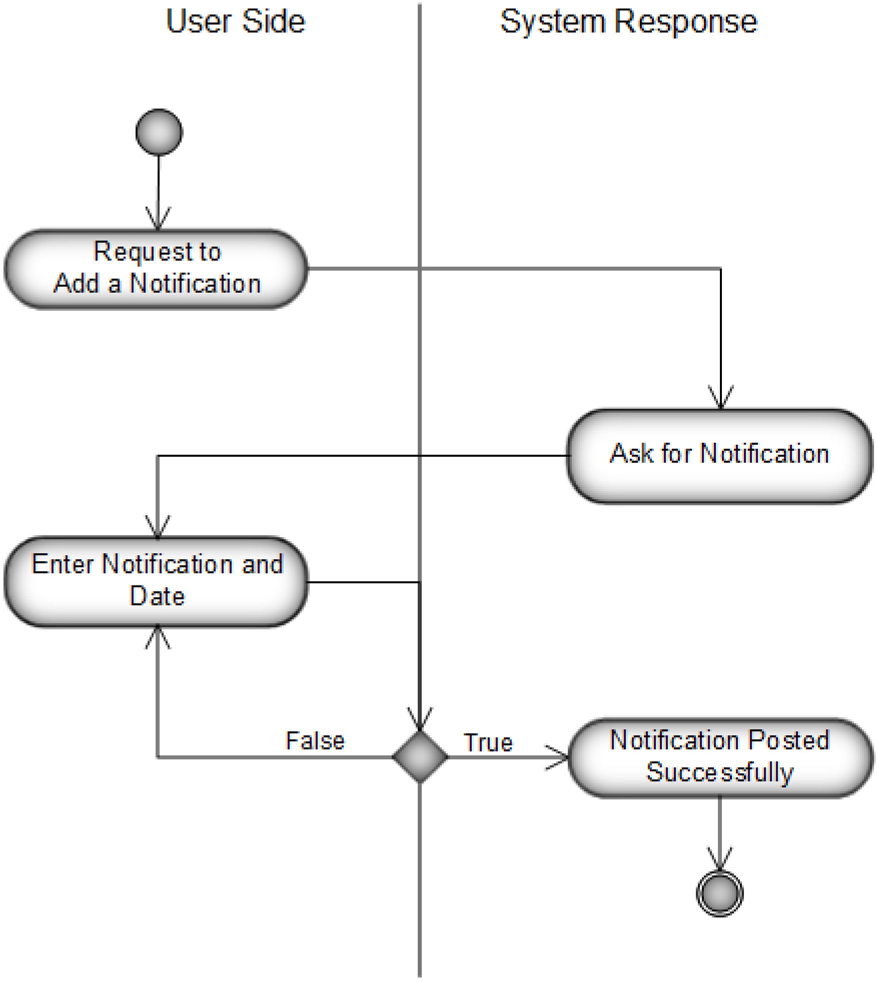

User Side
System Response
Request to Add a Notification
Ask for Notification
Enter Notification and Date
False
True
Notification Posted Successfully

Figure 6-15 Activity Diagram to add notifications

Edit Notification

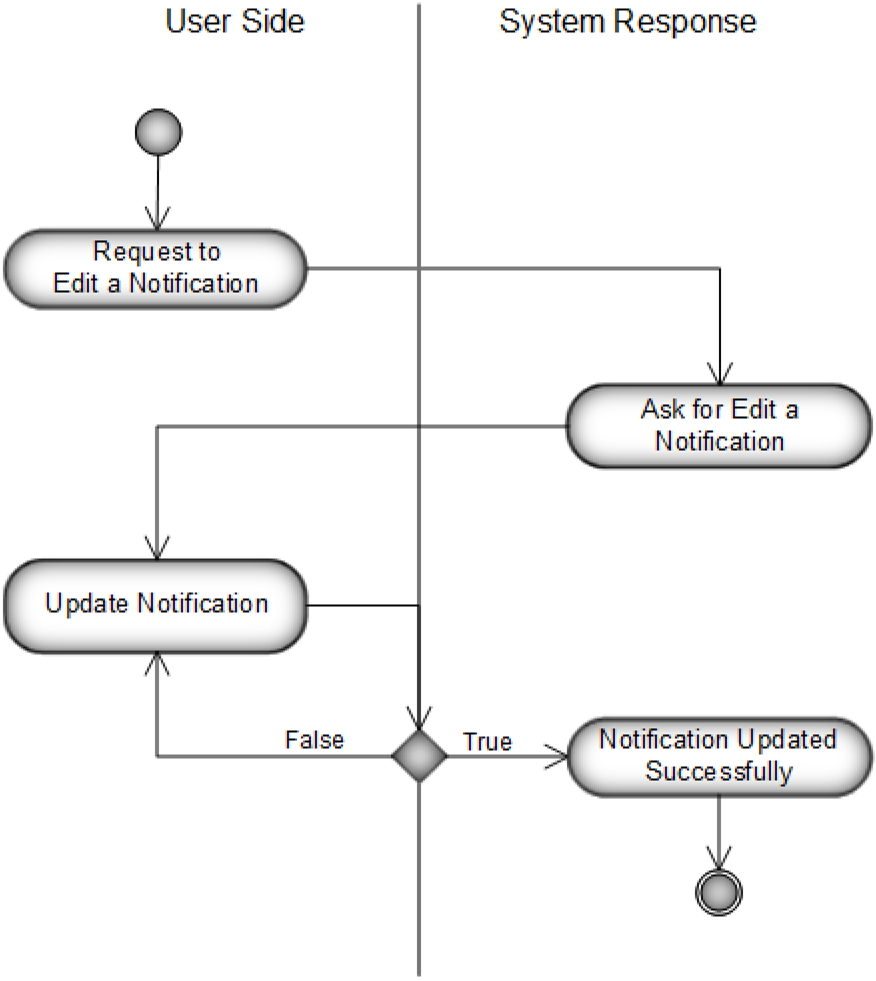

User Side
System Response
Request to Edit a Notification
Ask for Edit a Notification
Update Notification
False
True
Notification Updated Successfully

Figure 6-16 Activity Diagram to edit notifications

Delete Notification

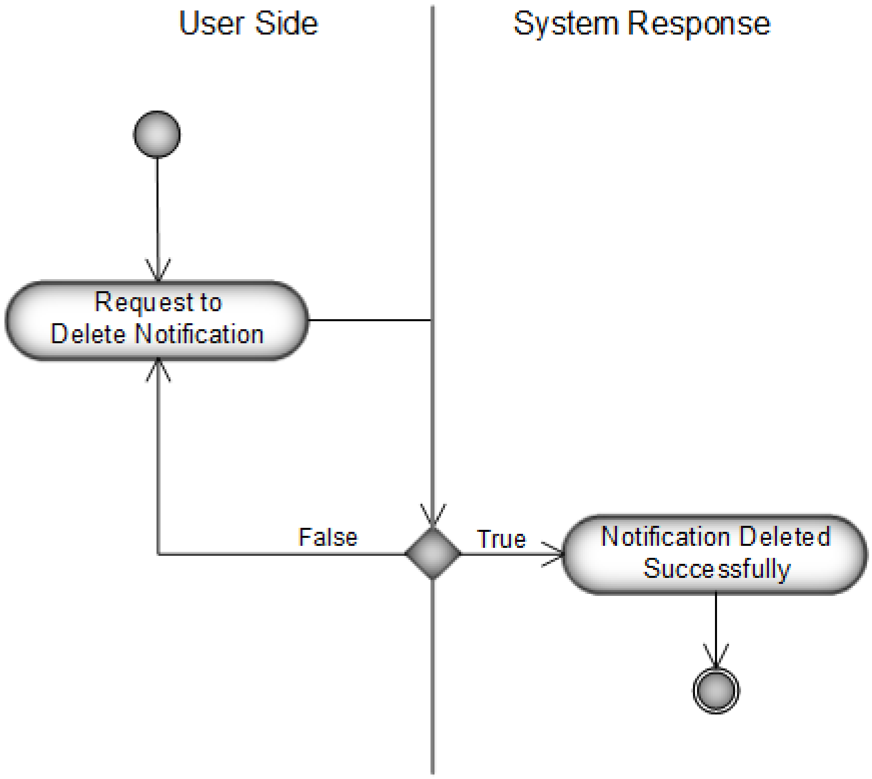

User Side
System Response
Request to Delete Notification
False
True
Notification Deleted Successfully

Figure 6-17 Activity Diagram to delete notifications

View Notification

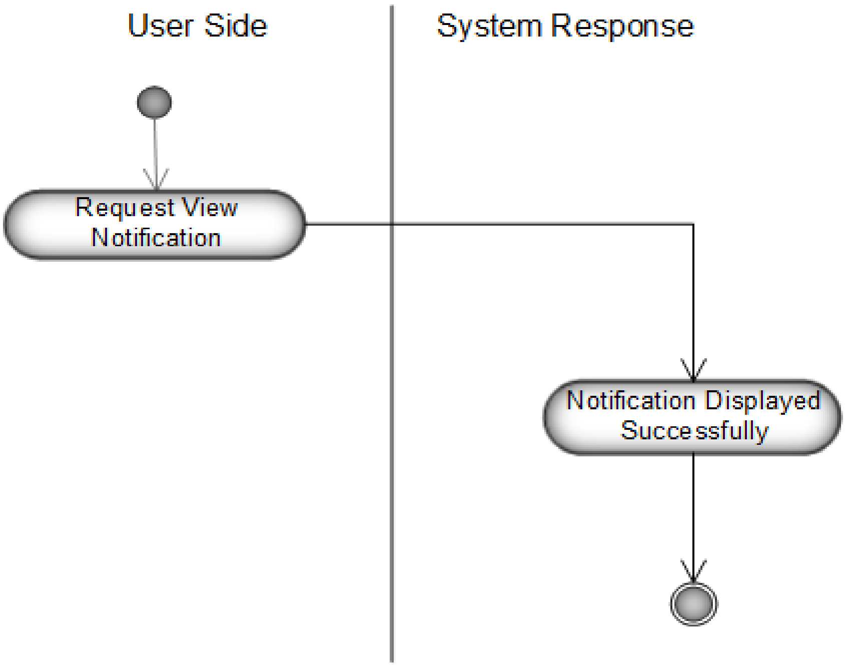

User Side
System Response
Request View Notification
Notification Displayed Successfully

Figure 6-18 Activity Diagram to view notification

Add University

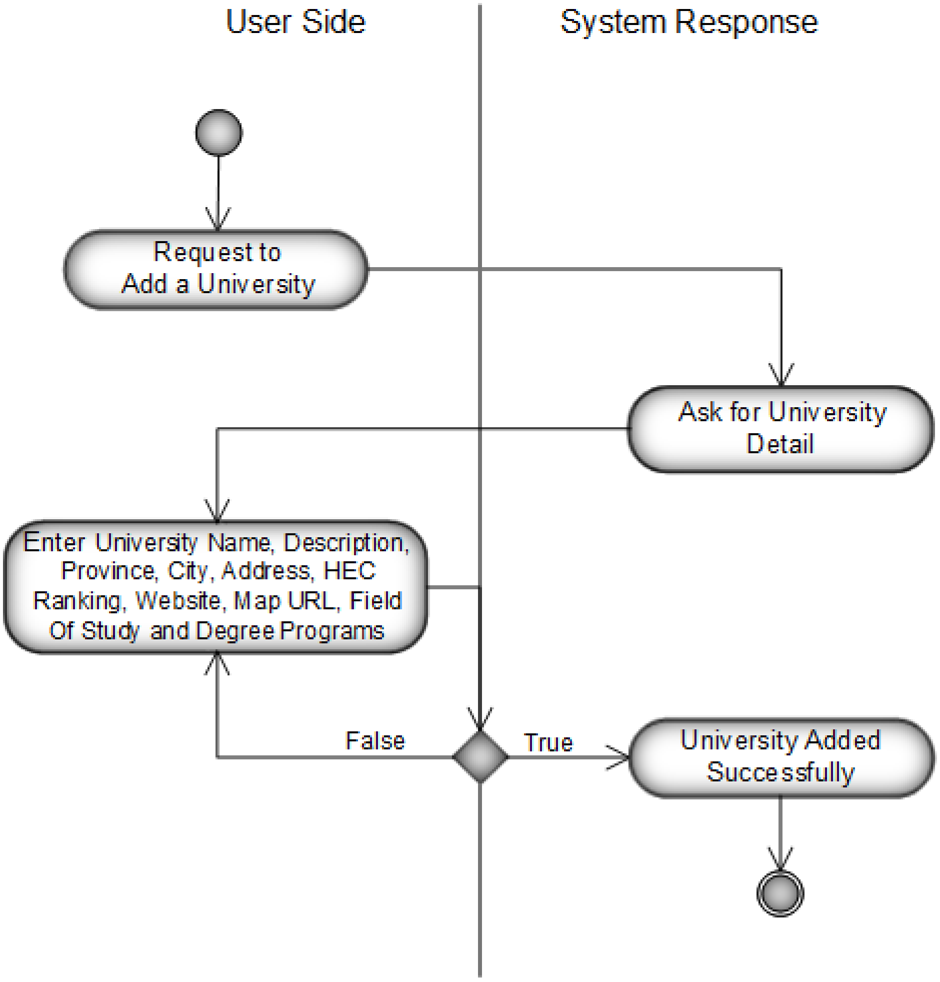

User Side
System Response
Request to Add a University
Ask for University Detail
Enter University Name, Description, Province, City, Address, HEC Ranking, Website, Map URL, Field Of Study and Degree Programs
False
True
University Added Successfully

Figure 6-19 Activity Diagram to add university

Edit University

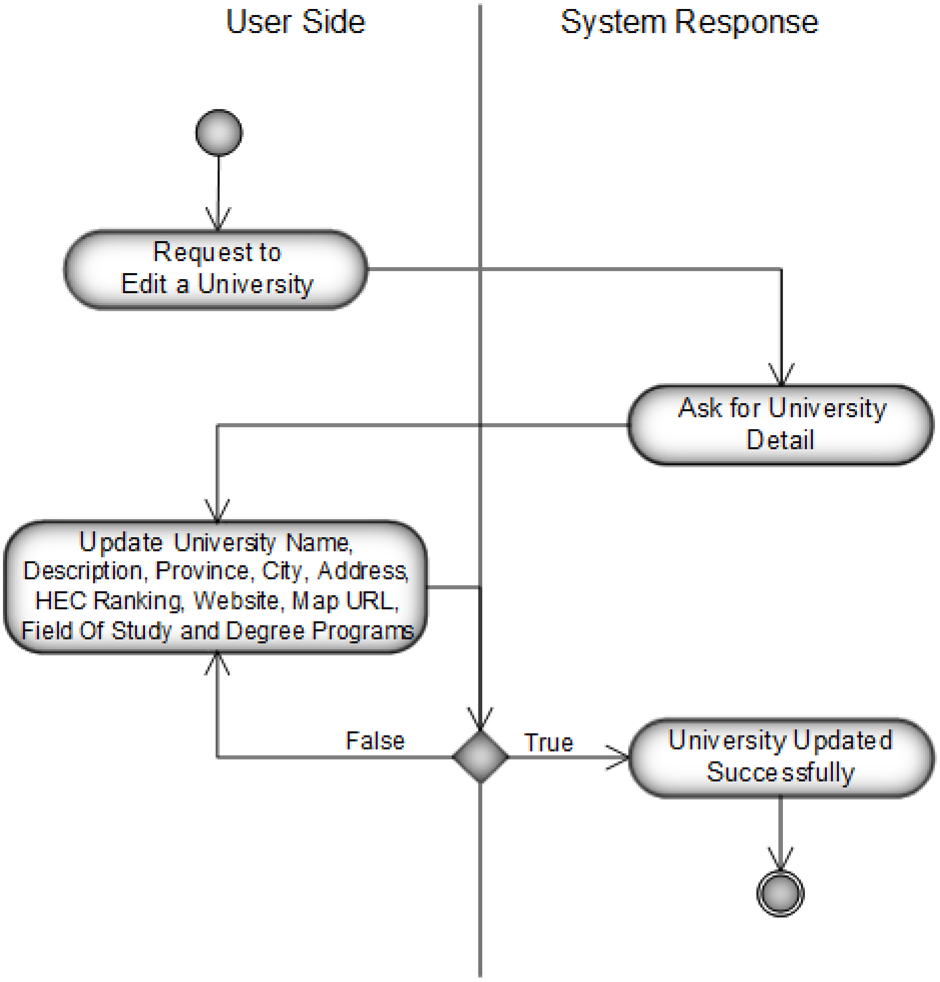

User Side
System Response
Request to Edit a University
Ask for University Detail
Update University Name, Description, Province, City, Address, HEC Ranking, Website, Map URL, Field Of Study and Degree Programs
False
True
University Updated Successfully

Figure 6-20 Activity Diagram to edit university

View Universities

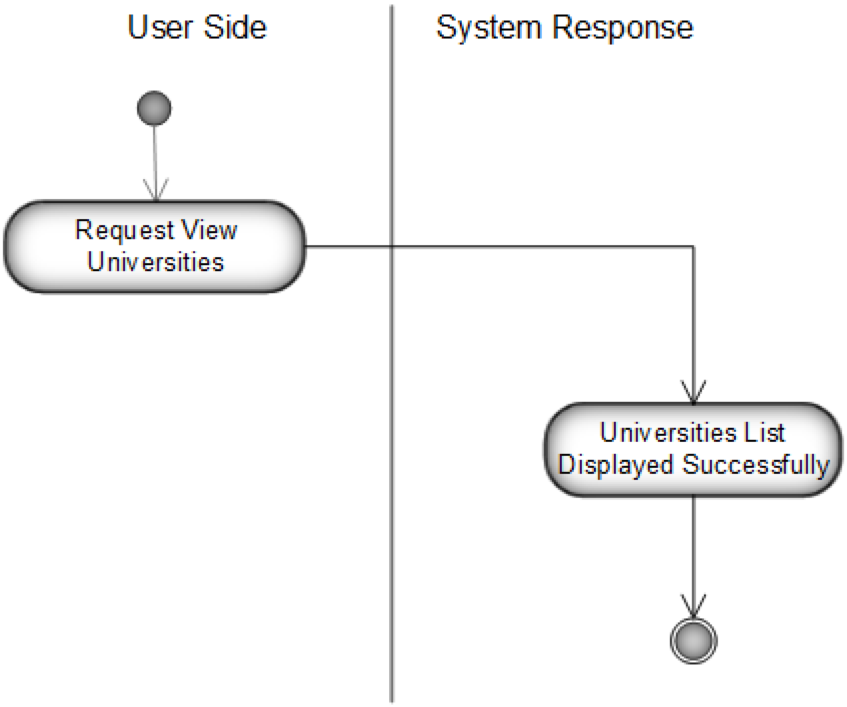

User Side
System Response
Request View Universities
Universities List Displayed Successfully

Figure 6-21 Activity Diagram to view university

Delete University

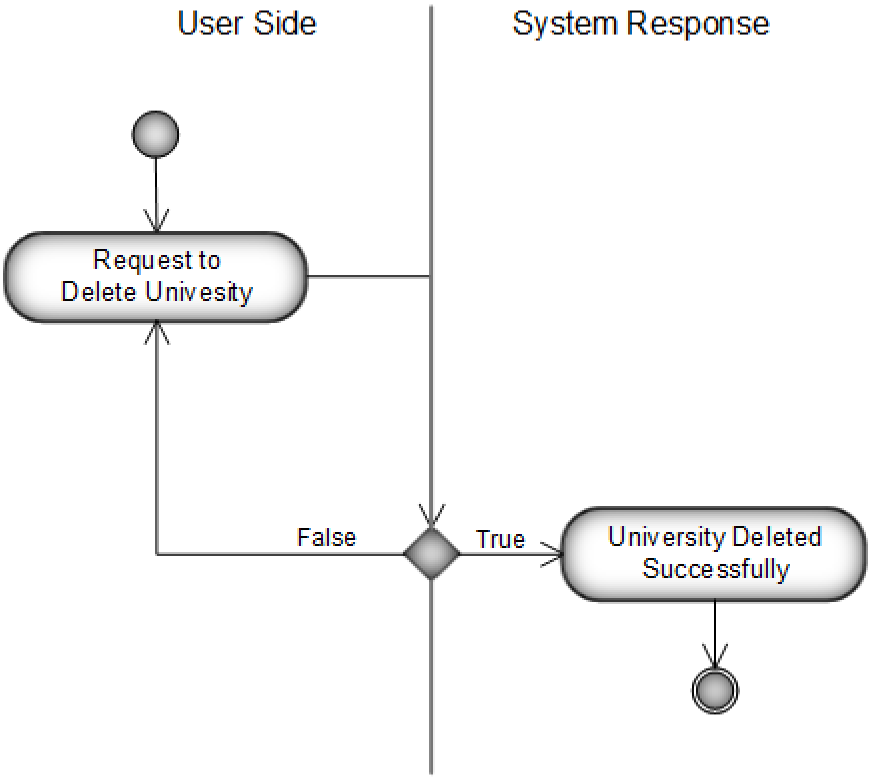


Figure 6-22 Activity Diagram to delete university

Logout

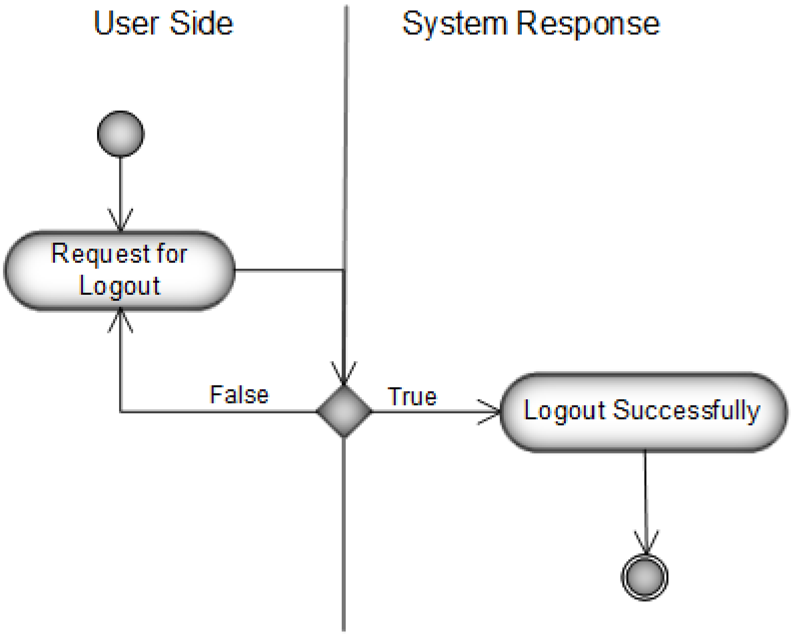


Figure 6-23 Activity Diagram to logout

Sequence Diagram:

Login

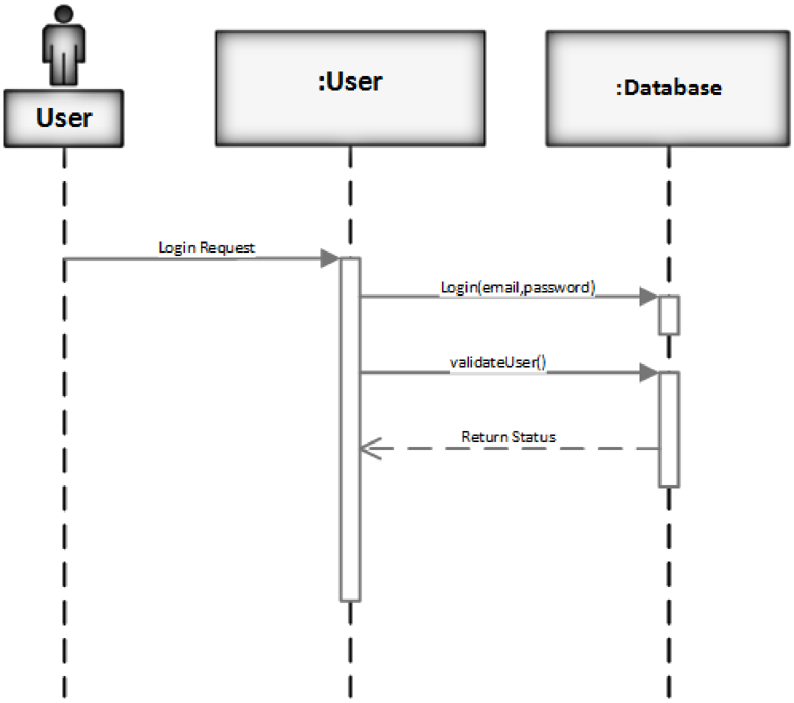


Figure 6-24 Sequence Diagram for login

Signup

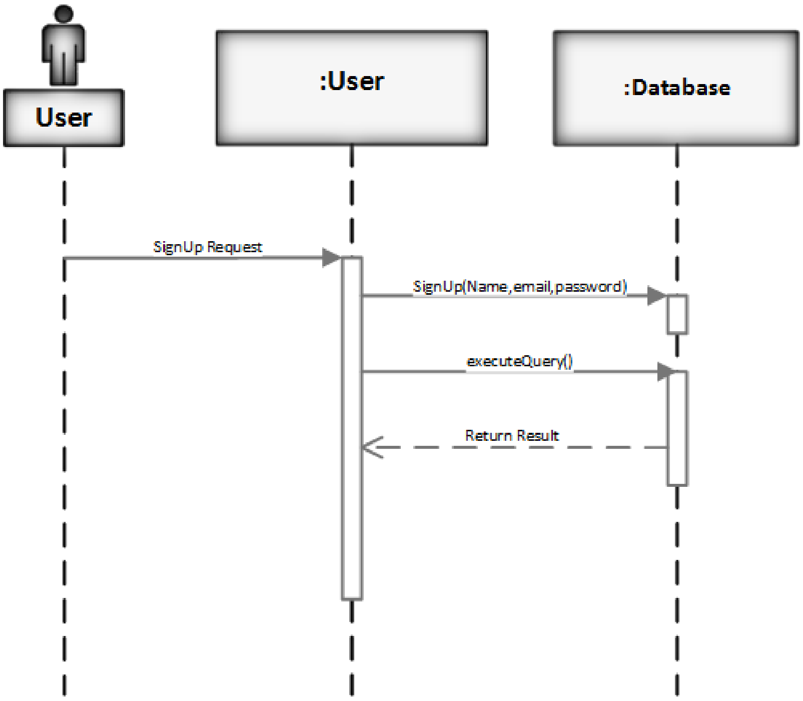


Figure 6-25 Sequence Diagram for signup

Search by Name

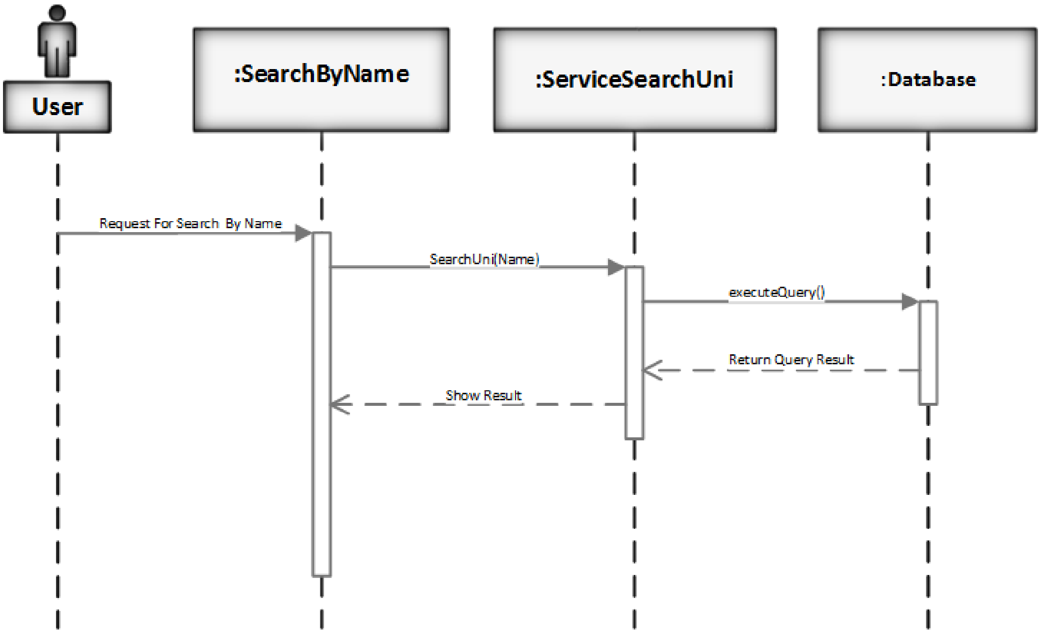


Figure 6-26 Sequence Diagram to search by name

Search By Ranking

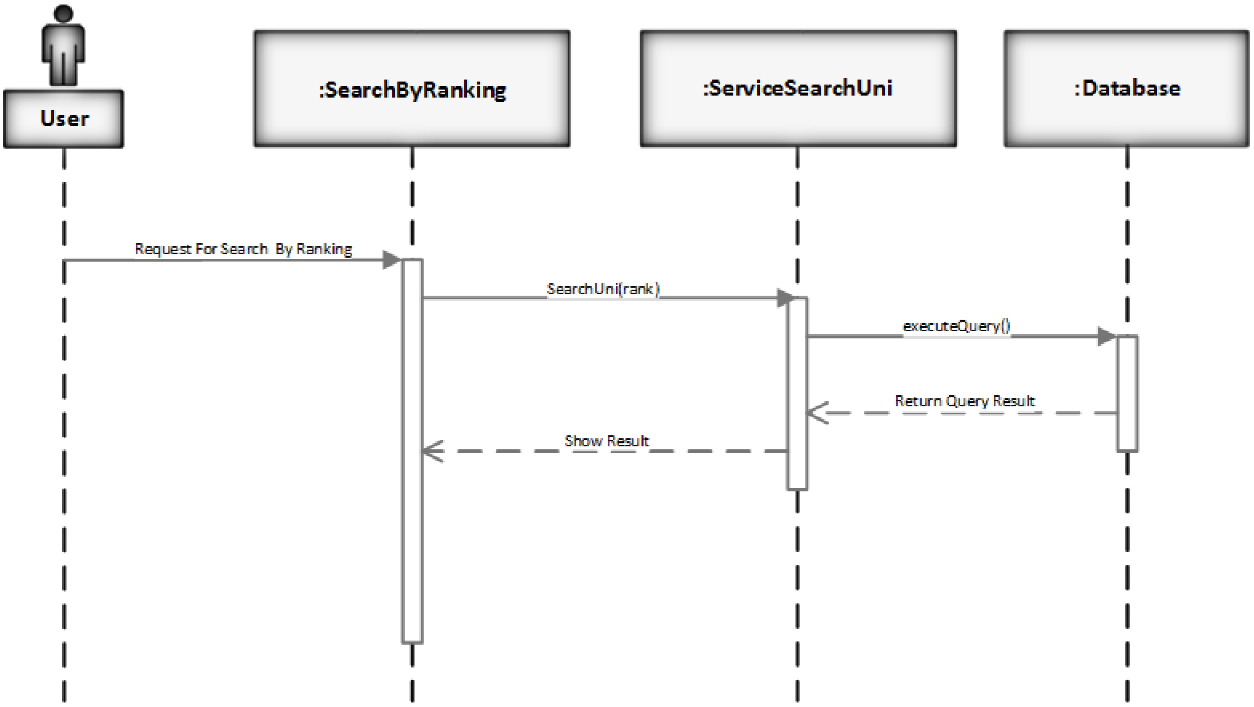


Figure 6-27 Sequence Diagram to search by ranking

Search by Degree

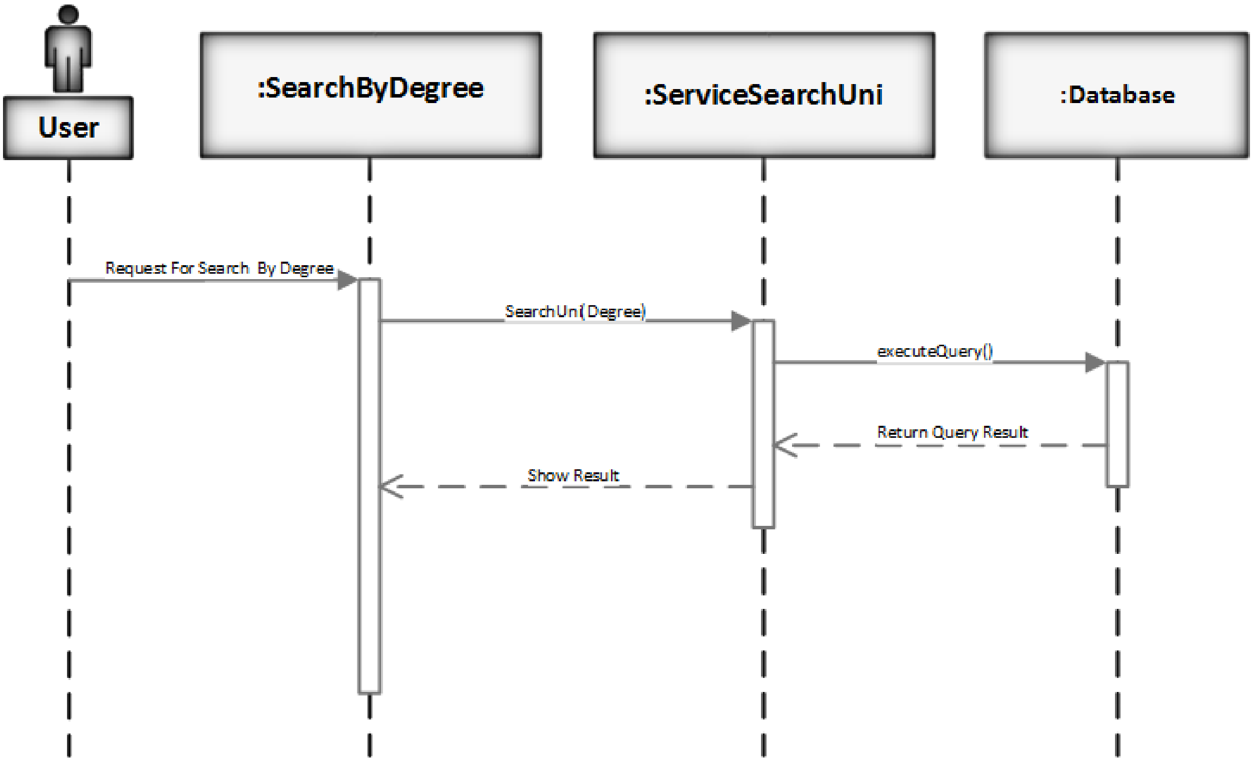


Figure 6-28 Sequence Diagram to search by degree

Search by Location

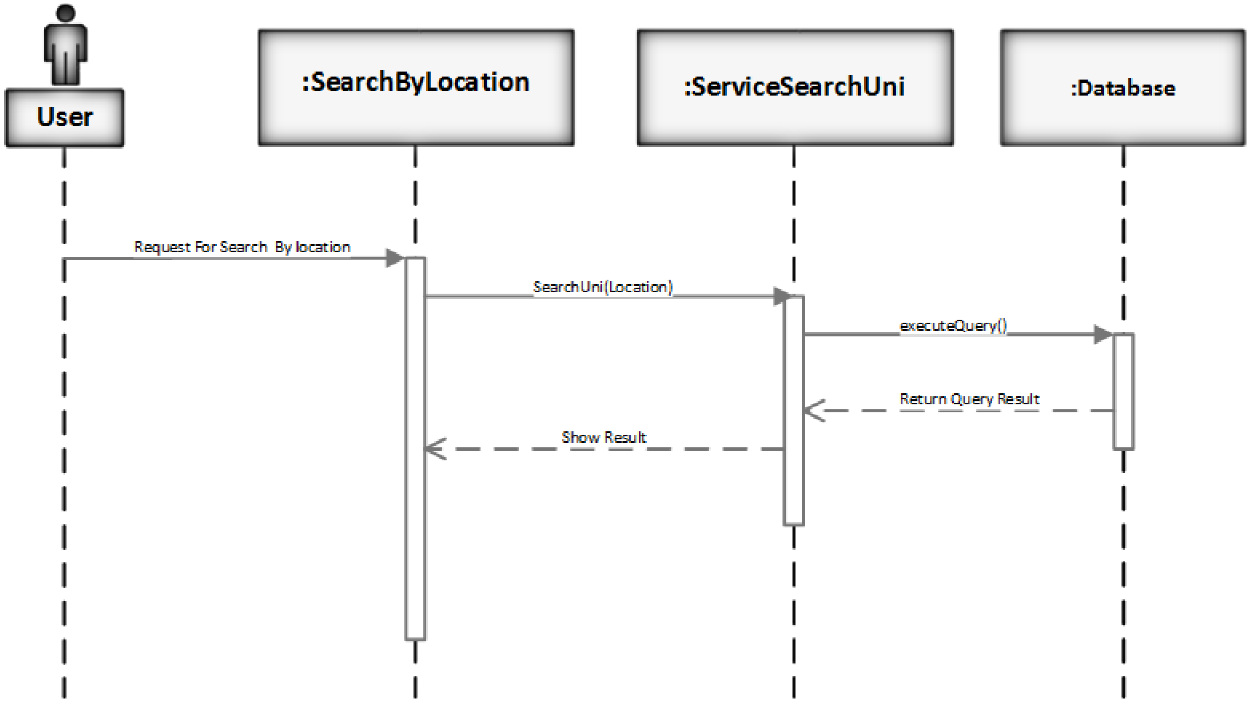


Figure 6-29 Sequence Diagram to search by locations

Search By Field Of Study

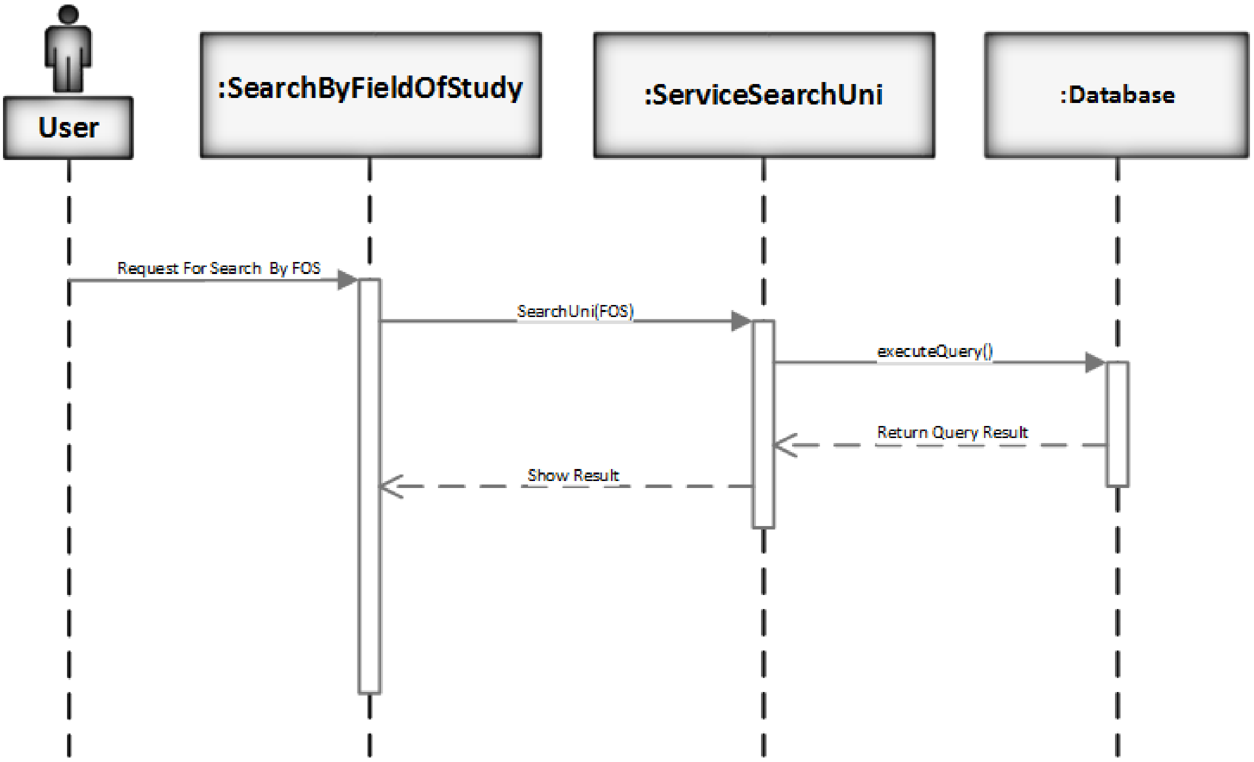


Figure 6-30 Sequence Diagram to search by field of study

Edit Profile

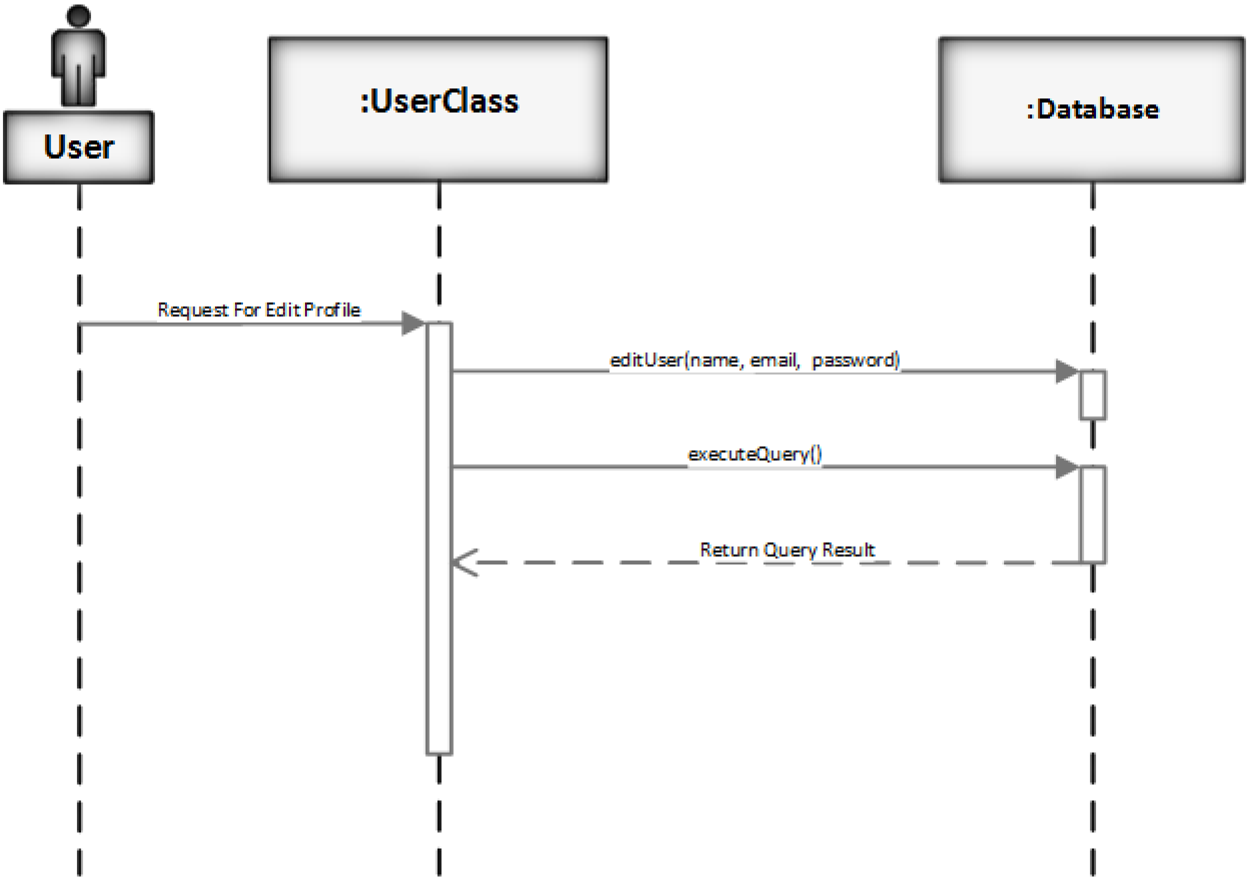


Figure 6-31 Sequence Diagram to edit profile

Delete Account

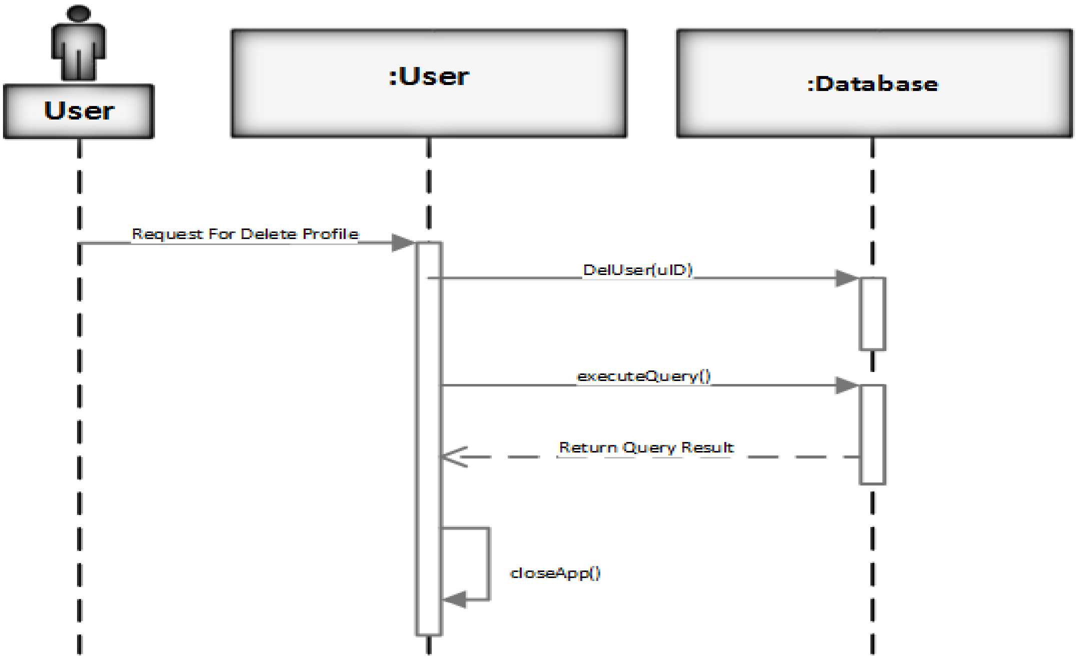


Figure 6-32 Sequence Diagram to delete account

Add Comment

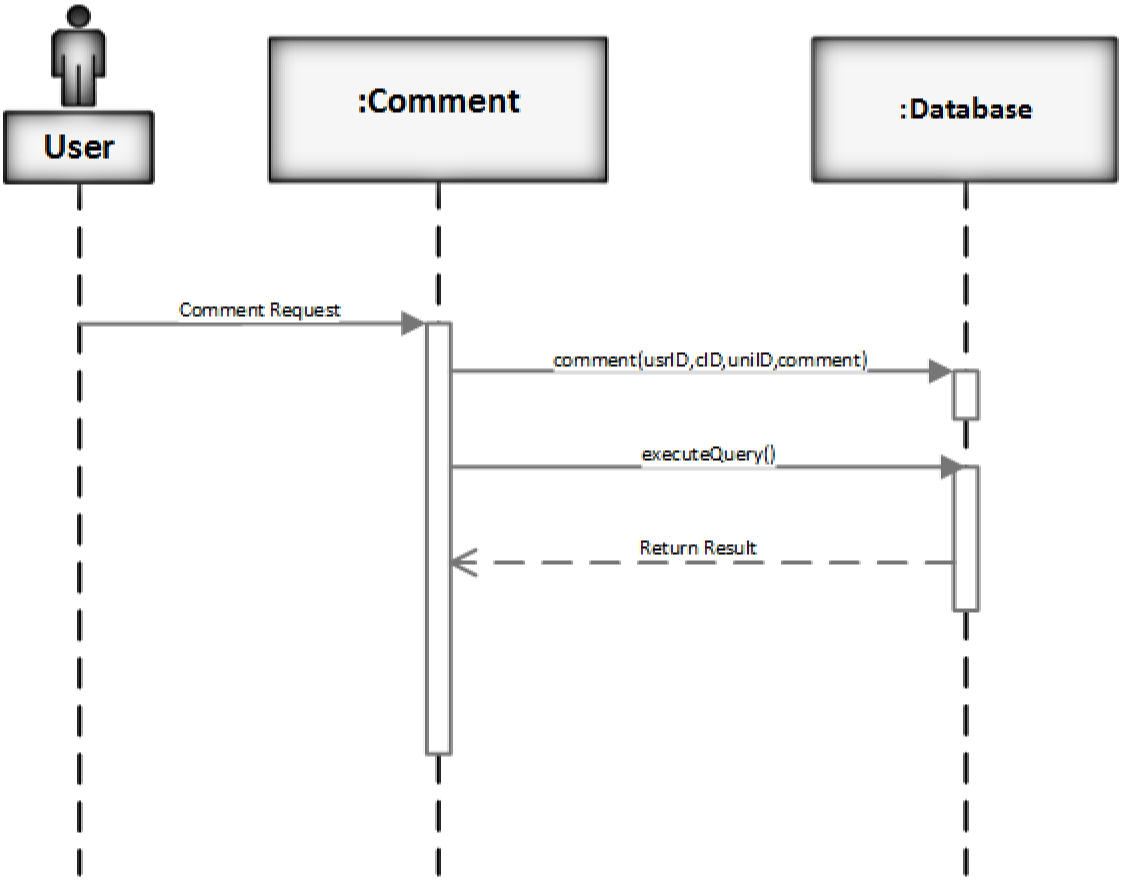


Figure 6-33 Sequence Diagram to add comment

Edit Comment

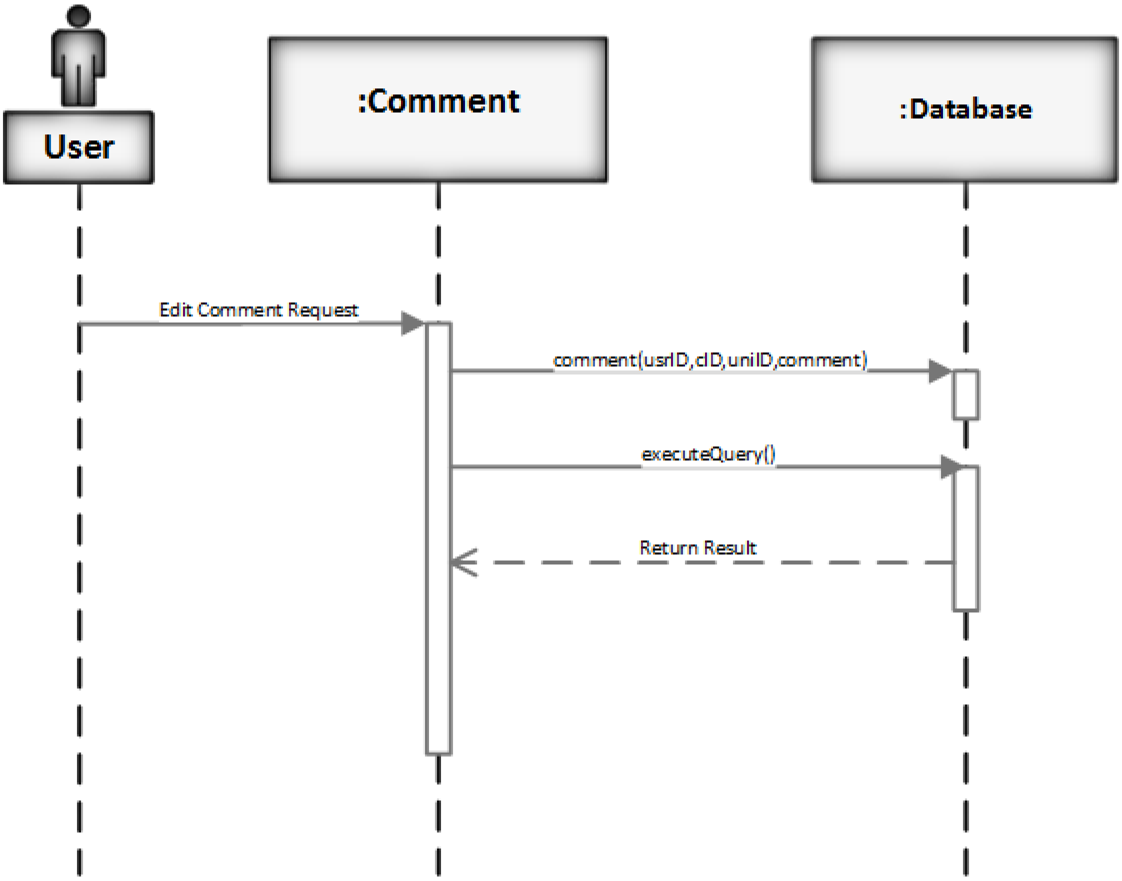


Figure 6-34 Sequence Diagram to edit comment

Delete Comment

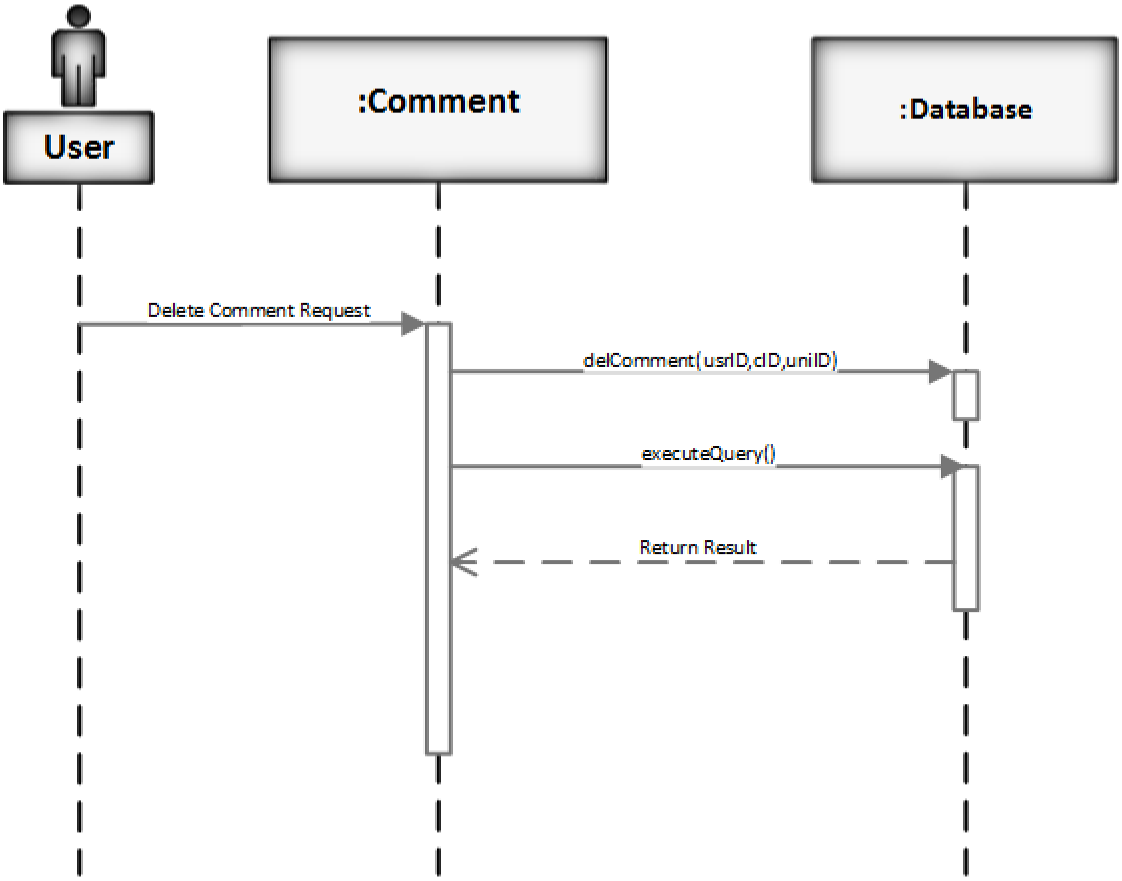


Figure 6-35 Sequence Diagram to delete comment

Add University

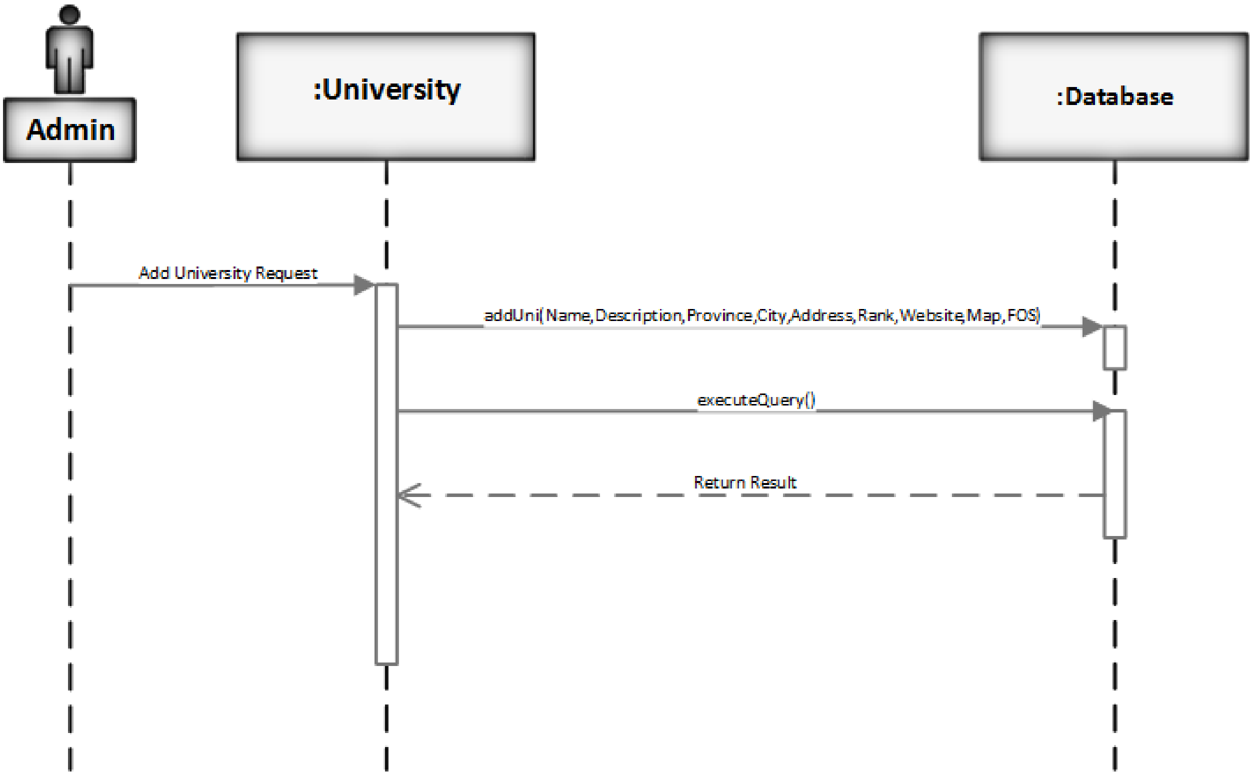


Figure 6-36 Sequence Diagram to add university

Edit University

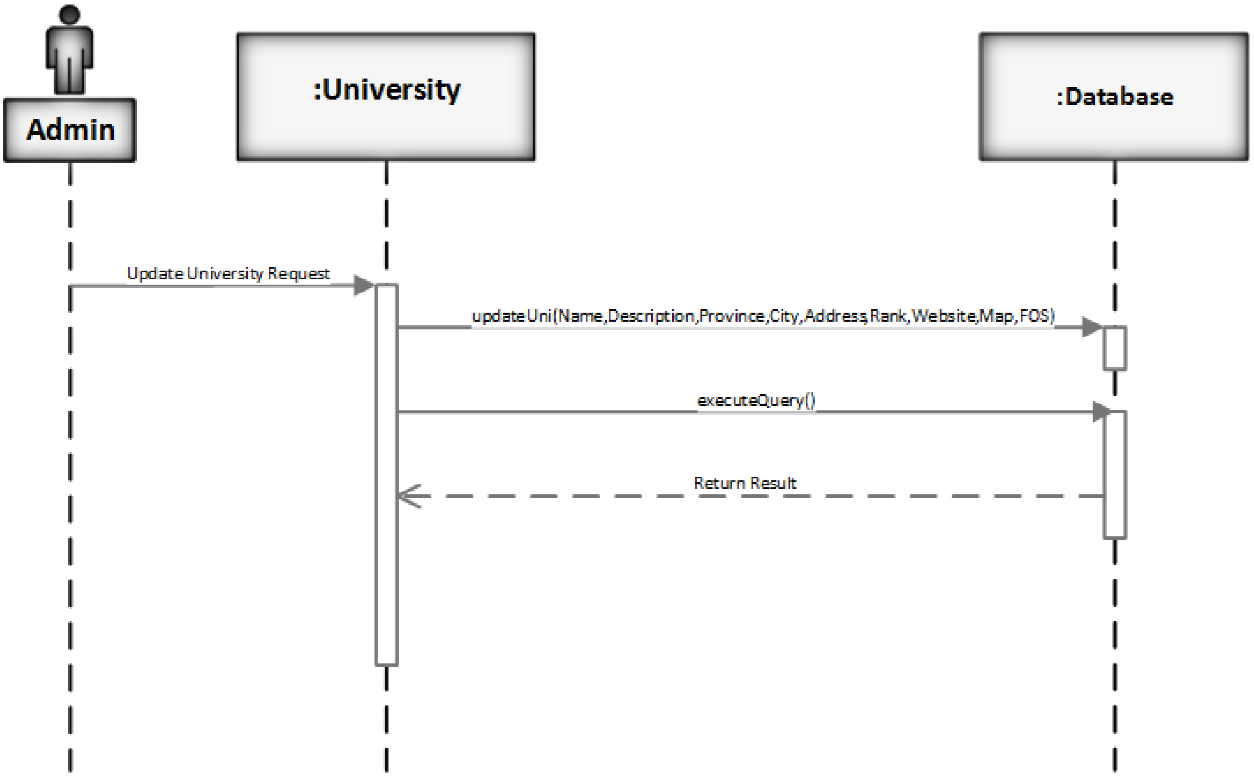

Admin
:University
:Database
Update University Request
updateUni(Name,Description,Province,City,Address,Rank,Website,Map,FOS)
executeQuery()
Return Result

Figure 6-37 Sequence Diagram to edit university

Delete University

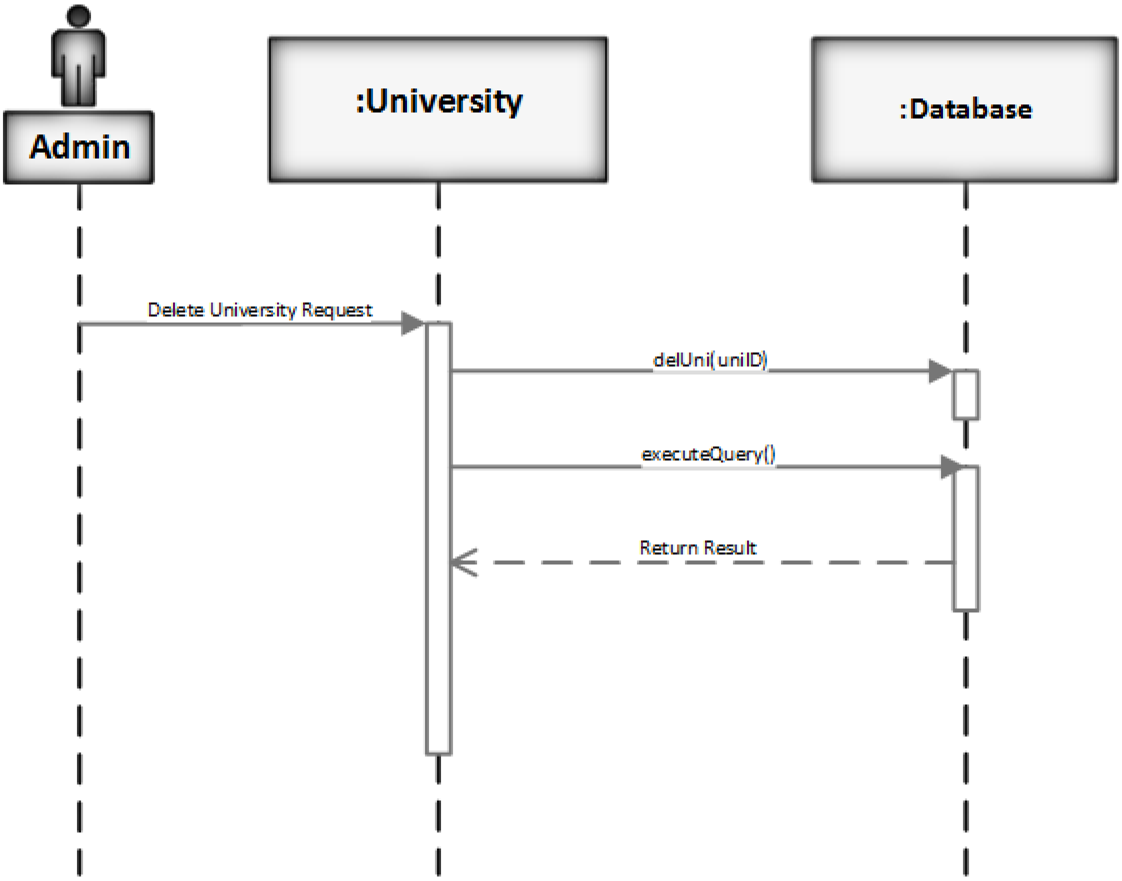


Figure 6-38 Sequence Diagram to delete university

View Universities

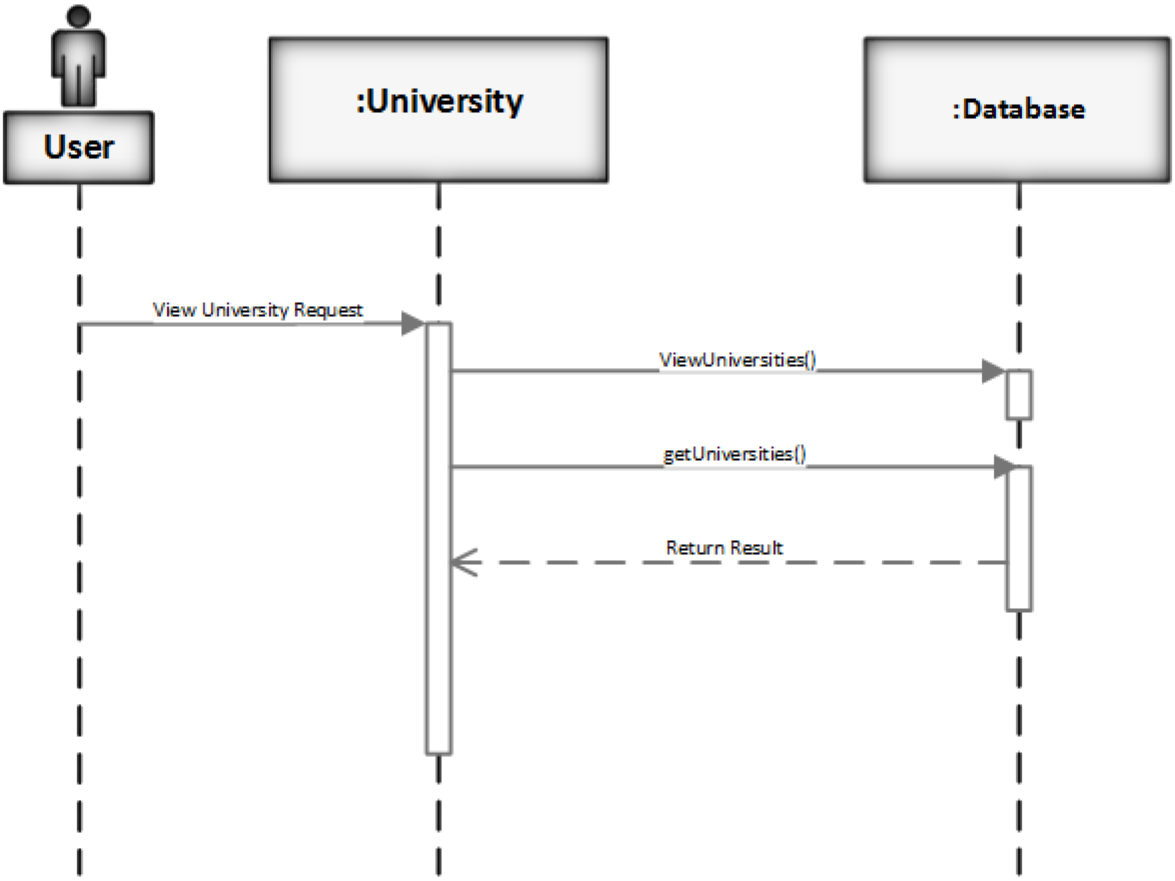


Figure 6-39 Sequence Diagram to view University

View Students

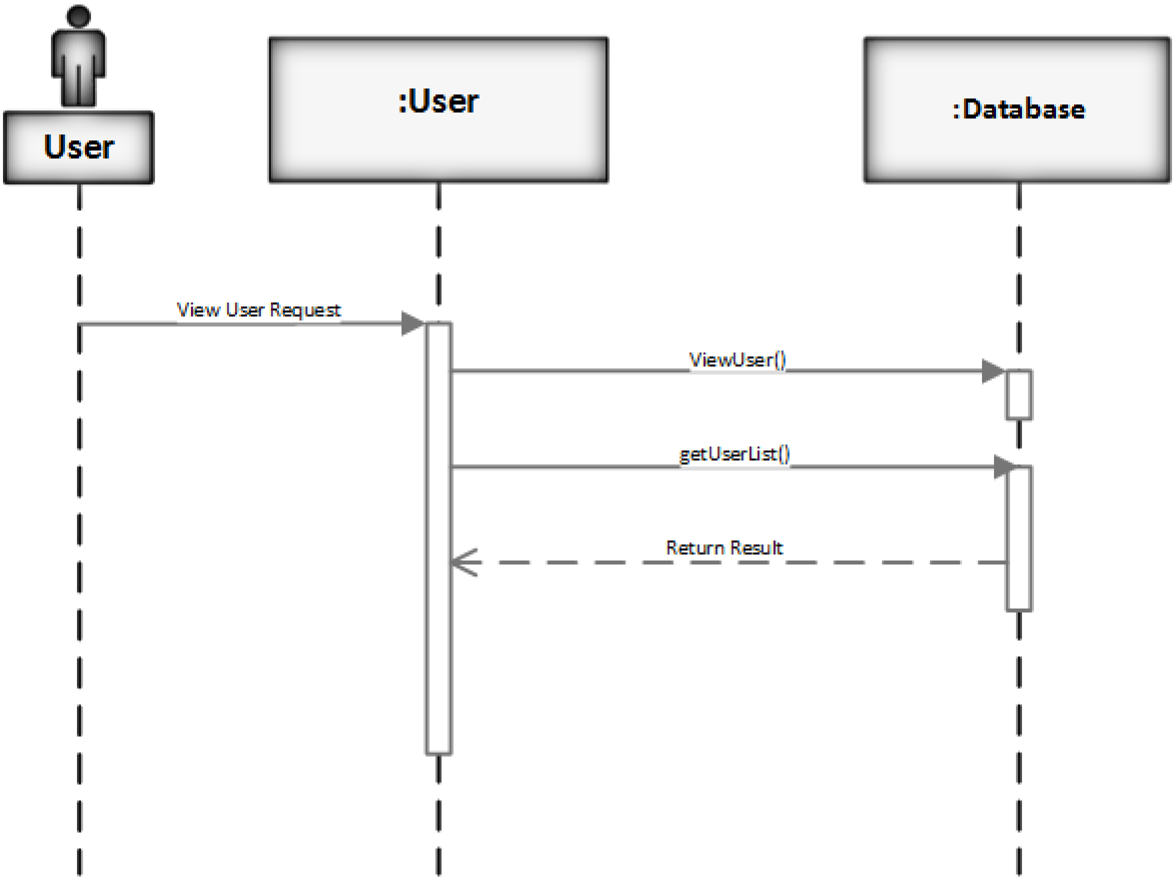


Figure 6-40 Sequence Diagram to View Student

Add Notification

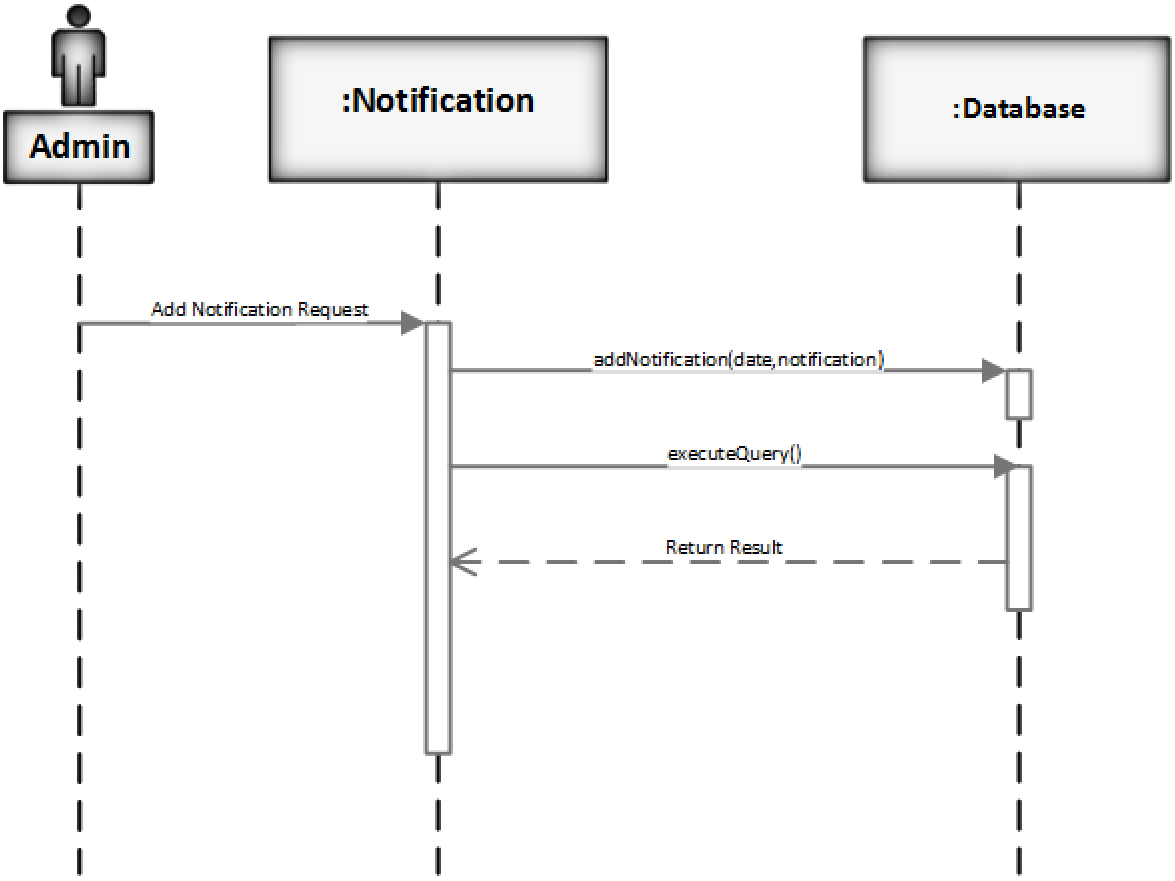


Figure 6-41 Sequence Diagram to Add notifications

View Notification

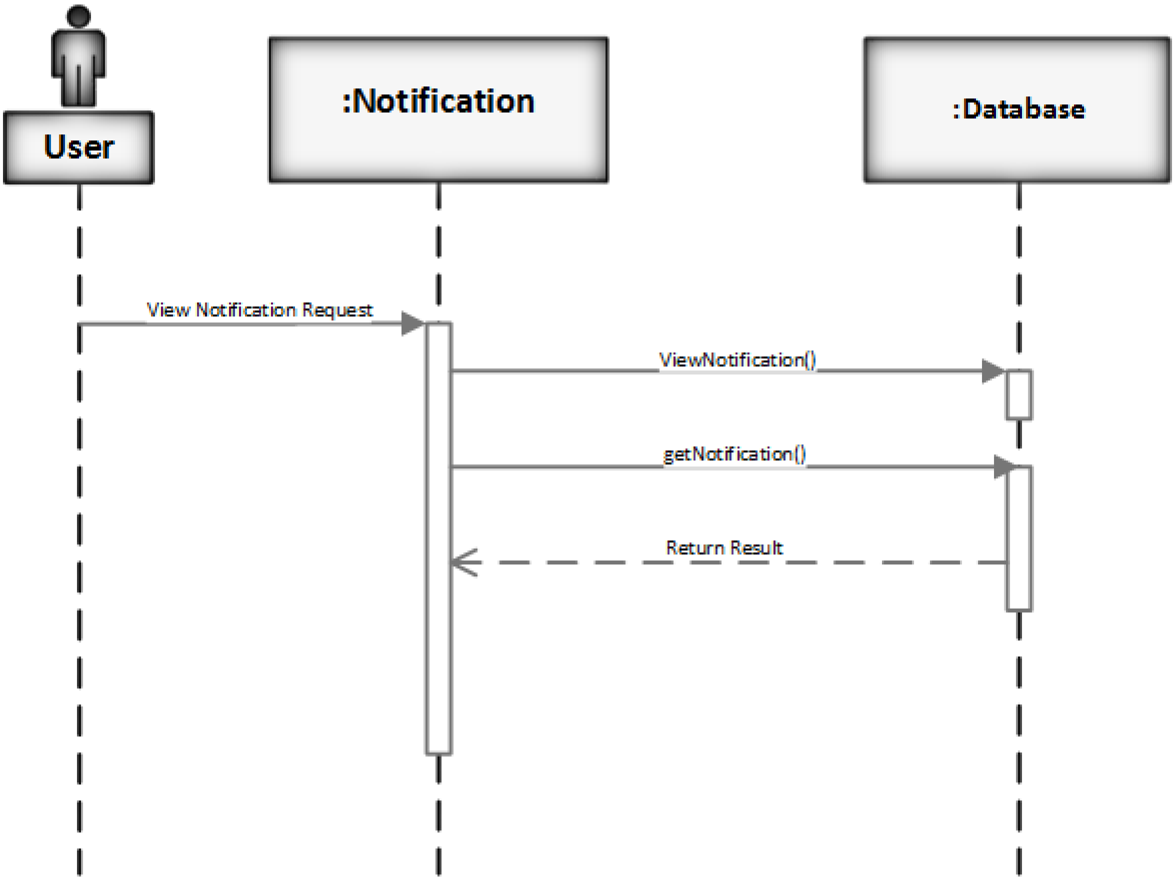


Figure 6-42 Sequence Diagram to view notifications

Delete Notification

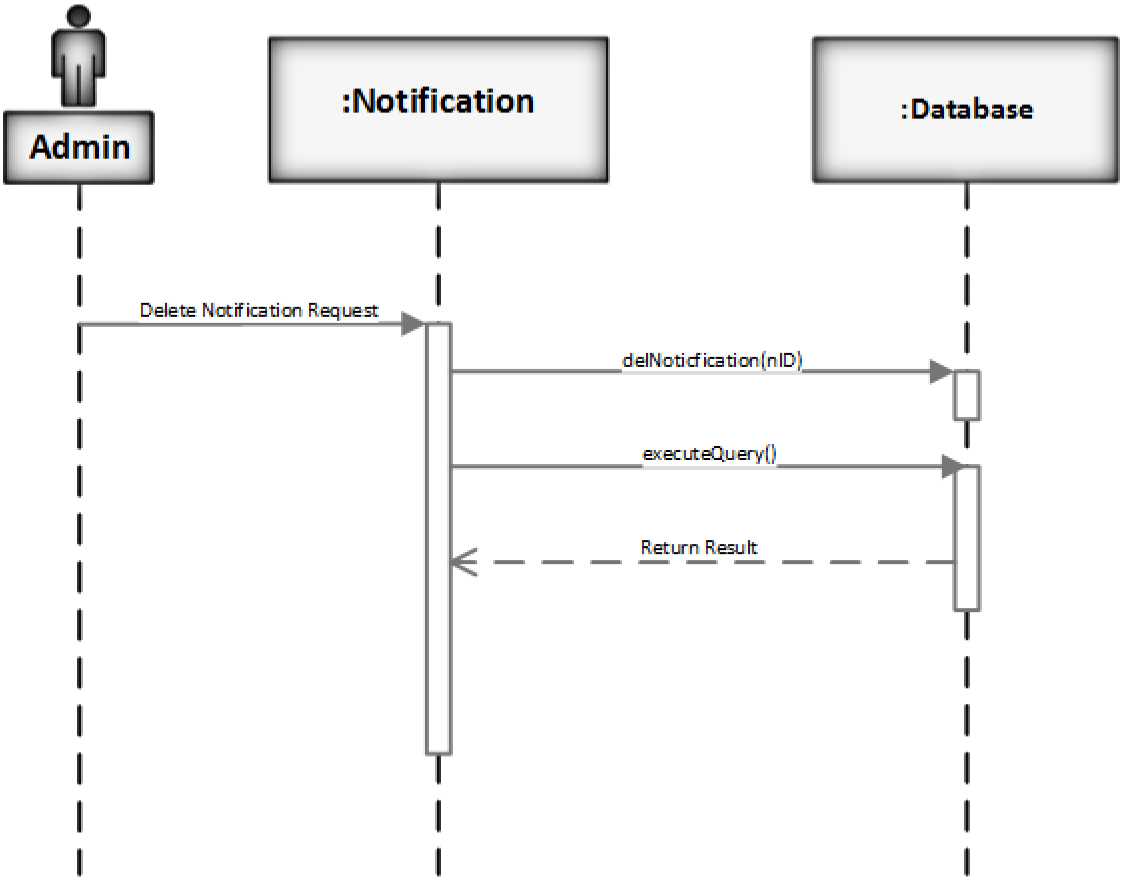


Figure 6-43 Sequence Diagram to delete notifications

Edit Notification

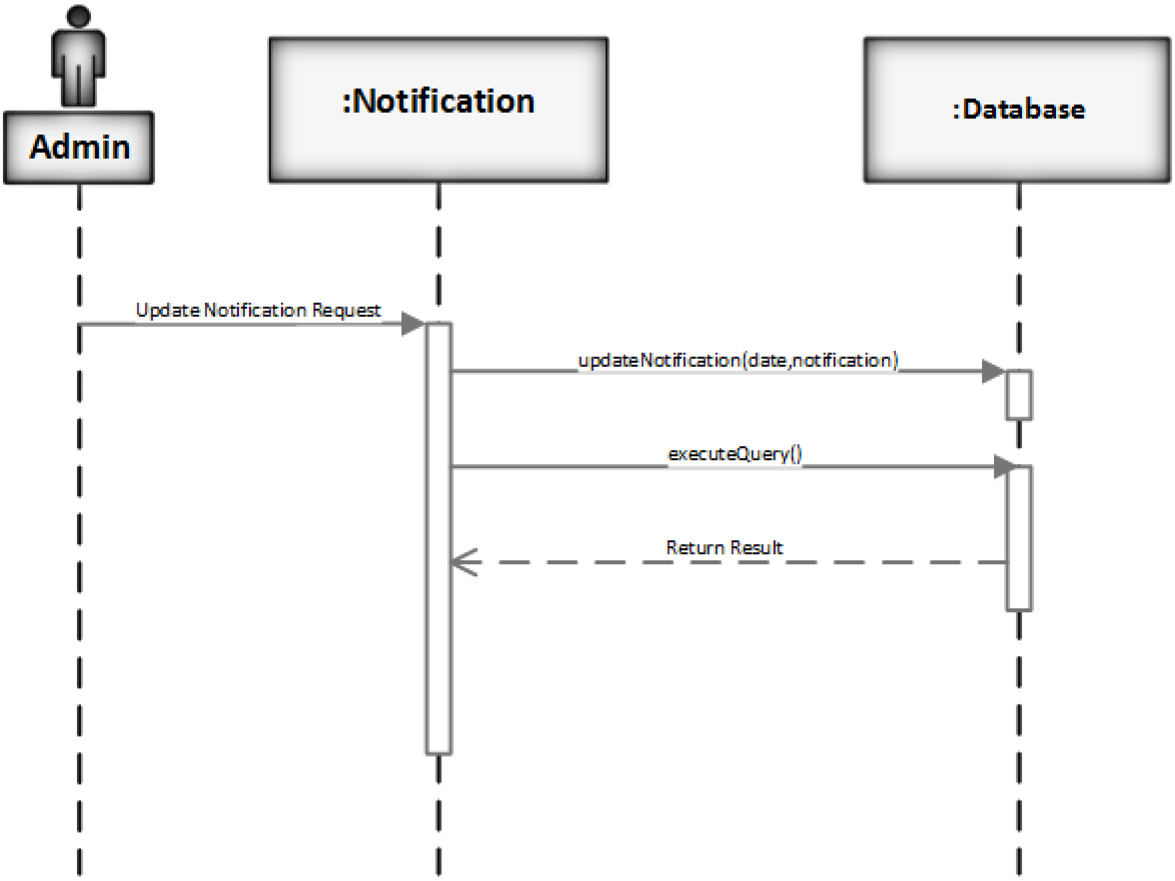

Admin
:Notification
:Database
Update Notification Request
updateNotification(date,notification)
executeQuery()
Return Result

Figure 6-44 Sequence Diagram to edit notifications

Chapter 7 Implementation

7.1 Introduction

Following software are used in development of university finder application.

- Windows OS
- Eclipse Luna
- JDK 7
- MySQL
- Android Development Tools Plugins
- Android virtual Device
- Adobe Photoshop
- Adobe Illustrator
- Adobe Dreamweaver
- GenY motion Emulator
- Apache Webserver

7.2 Verification

System testing is an essential step for the development of the reliable and error free system. Testing is the process executing a problem with the explicit intention of finding errors i.e. making the problem fail and test cases are devised with the purpose in mind. A successful test is one which does not find an error.

Login

Table 7-1 Test Case for Login

Test Case ID: TC-UFA-001 **Test Type:** Manual-black box. **Component:** User authentication facility	**Test Item:** User authentication **Test Case Name:** Login **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of user authentication	
Pre-Condition	**Post Condition**
User should have their own email and password.	System successfully authorized the user and registered users to use system.
Input Data/Event	**Expected output Data**
In case of UFA login. Enter email, password.	Successfully login.
Test Case Result	
Test is performed successfully.	
Actual Input Data/Event	**Actual Output Data/Event**
1. Enter “admin@uf.com” as username and “admin@123” as password. **2.** Click on “Login” button.	Successfully login.
1. Enter “Tayyab.muzamal@gmail.com” as username and”tayyab123” as password. **2.** Click on “Login” button.	Successfully login.
1. Enter “admin@uf.com” as username and “dfsd” as password. **2.** Click on “Login” button.	Login Failed.
1. Enter “admin@uf.com” as username and “abcde” as password. **2.** Click on “Login” button.	Login Failed.
Test Case Result	
Test is performed successfully.	

Add Notification

Table 7-2 Test Case for Add notifications

Test Case ID: TC-UFA-003 **Test Type:** Manual-black box. **Component:** Manage Notifications	**Test Item:** Manage Notification **Test Case Name:** Add Notification **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of add notification.	
Pre-Condition	**Post Condition**
Admin and should have authorized to Add Notification.	Notification posted successfully.
Input Data/Event	**Expected output Data**
In case of UFA Add Notification, Admin enters notification, and select date from calendar and then click on Add Notification button.	Notification posted successfully.
Test Case Result	
Test is performed successfully.	
Actual Input Data/Event	**Actual Output Data/Event**
1. Enter “Admission Open in BSSE in Mirpur University of Science and Technology” as notification and select “2017-08-10”, as date and click on Add Notification button.	Notification posted successfully.
2. Enter “1234” as notification and click on Add Notification button.	“Invalid notification”.
3. Enter “” as notification and click on Add Notification button.	“Invalid notification”.
Test Case Result	
Test is performed successfully.	

Delete Notification

Table 7-3 Test Case for Delete Notifications

Test Case ID: TC-UFA-004 **Test Type:** Manual-black box. **Component:** Manage Notification facility	**Test Item:** Manage Notification **Test Case Name:** Delete Notification **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of delete Notification.	
Pre-Condition	**Post Condition**
Admin should have authorized to delete Notification.	Notification Deleted Successfully from the system.
Input Data/Event	**Expected output Data**
In case of UFA delete Notification Tap and hold on Notification and select Delete button.	Notification Deleted Successfully.
Test Case Result	
Test is performed successfully.	

View Universities

Table 7-4 Test Case for View Universities

Test Case ID: TC-UFA-005 **Test Type:** Manual-black box. **Component:** Manage Universities	**Test Item:** Manage Universities **Test Case Name:** View Universities **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of view Universities.	
Pre-Condition	**Post Condition**
Admin should have authorized to view Universities.	Universities listed in the system.
Input Data/Event	**Expected output Data**
In case of UFA View Universities admin clicks on view Universities button.	Universities list displayed.
Test Case Result	
Test is performed successfully.	
Actual Input Data/Event	**Actual Output Data/Event**
Click on view Universities button.	Universities list displayed.
Test Case Result	
Test is performed successfully.	

Delete University

Table 7-5 Test Case for Delete University

Test Case ID: TC-UFA-006 **Test Type:** Manual-black box. **Component:** Manage Universities	**Test Item:** Manage Universities **Test Case Name:** Delete University **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of delete University.	
Pre-Condition	**Post Condition**
Admin should have authorized to delete University.	University Deleted Successfully from the system.
Input Data/Event	**Expected output Data**
In case of UFA delete University Tap and hold on University and select Delete button.	University Deleted Successfully.
Test Case Result	
Test is performed successfully.	

Search by Name

Table 7-6 Test Case for Search By Name

Test Case ID: TC-UFA-007 **Test Type:** Manual-black box. **Component:** Search Universities	**Test Item:** Search Universities **Test Case Name:** Search By Name **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of search University by name.	
Pre-Condition	**Post Condition**
User should have authorized to Search university.	University found successfully in the system.
Input Data/Event	**Expected output Data**
In case of UFA Search by name, Enter University name and then clicks on Search University button.	University found successfully.
Test Case Result	
Test is performed successfully.	
Actual Input Data/Event	**Actual Output Data/Event**
1. Enter “Mirpur University of Science and Technology” as University name. **2.** Click on Search University button.	University found successfully.
1. Enter “” as University name, **2.** Click on Search University button.	Invalid input.
Test Case Result	
Test is performed successfully.	

Search by Location

Table 7-7 Test Case for Search By Location

Test Case ID: TC-UFA-008 **Test Type:** Manual-black box. **Component:** Search Universities	**Test Item:** Search Universities **Test Case Name:** Search by Location **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of search University by Location.	
Pre-Condition	**Post Condition**
User should have authorized to Search university.	University found successfully in the system.
Input Data/Event	**Expected output Data**
In case of UFA Search by location, enter city name and then clicks on Search University button.	University found successfully.
Test Case Result	
Test is performed successfully.	
Actual Input Data/Event	**Actual Output Data/Event**
1. Enter “Mirpur” as City name. **2.** Click on Search University button.	University found successfully.
1. Enter “:D” as University name, **2.** Click on Search University button.	University not found.
Test Case Result	
Test is performed successfully.	

View Notification

Table 7-8 Test Case for View Notifications

Test Case ID: TC-UFA-010 **Test Type:** Manual-black box. **Component:** View Notification	**Test Item:** View Notification **Test Case Name:** View Notification **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of view Notification.	
Pre-Condition	**Post Condition**
User should have authorized to view Notification.	Notification displayed in the system.
Input Data/Event	**Expected output Data**
In case of UFA View Notification user clicks on view Notification button.	Notification list displayed.
Test Case Result	
Test is performed successfully.	
Actual Input Data/Event	**Actual Output Data/Event**
Click on view Notification button.	Notification list displayed.
Test Case Result	
Test is performed successfully.	

Search by Field of Study

Table 7-9 Test Case for Search by Field of Study

Test Case ID: TC-UFA-009 **Test Type:** Manual-black box. **Component:** Search Universities	**Test Item:** Search Universities **Test Case Name:** Search By Field of Study **Release Version:** 1.0
Test Case Description: The test case will determine the current functionality of search University by Field of Study.	
Pre-Condition	**Post Condition**
User should have authorized to Search university.	University found successfully in the system.
Input Data/Event	**Expected output Data**
In case of UFA Search by Field of Study, Select Field of Study and then clicks on Search University button.	University found successfully.
Test Case Result	
Test is performed successfully.	
Actual Input Data/Event	**Actual Output Data/Event**
1. Select “Engineering” as Field of Study. **2.** Click on Search University button.	University found successfully.
Test Case Result	
Test is performed successfully.	

Chapter 8 Discussion and Conclusion

8.1 Solution Review

We will design an application to facilitate students which will help users to find the suitable university as well as that will match to their academic abilities. Our target audience for this kind of tool is primarily students. Best time when students aren't explicitly connected to an institution where they are able to obtain guidance and can easily compare different universities from each aspect. We are looking forward for ways to fill up the gap between students and institutes with the help of latest technology which will be improve the patterns of polishing someone's skills by choosing the right platform without wasting the precious time of their life, instead investing it in a right direction.

8.2 Project Review

Discuss how well you addressed the project, and what you might do differently if you were to do it again. In particular, make sure to identify how you handled any problems that arose during the project.

8.3 Key Skills

Skills needed for this project are given as under:

- Android Development
- Adobe Photoshop
- Adobe Illustrator
- Database

8.4 Future Work

In future, we have many things in our mind to add into the UniversityFinder. For example, we are making this application on international level. Most common international Universities information are also searched and viewed form UniverstyFinder.

8.5 Conclusion

This application will facilitate students through which student users to find the suitable university as well as that will match to their academic abilities. Our target audience for this kind of tool is primarily students. Best time when students aren't explicitly connected to an institution where they are able to obtain guidance and can easily compare different universities from each aspect. We are looking forward for ways to fill up the gap between students and institutes with the help of latest technology which will be improve the patterns of polishing someone's skills by choosing the right platform without wasting the precious time of their life, instead investing it in a right direction.

Our application allows users to find the university by:

- Field of study.
- Degree programs.
- Location.
- Ranking.

- Name.

System will show the list of top universities that having information about their offered courses, duration, admission schedule, pre-requisites, fee information and address of every universities to the end users. Result will be filtered to the users for their best match.

References

[1] http://www.enterpriseappstoday.com/management-software/3-approaches-to-building-mobile-apps-which-is-best.html/ accessed on [6th May 2017]

[2] Gong, Lei, and Cong Zhou. "Development and Research of Mobile Termination Application Based on Android [J]." *Computer and modernization* 8.26 (2008): 85-88.

[3] http://developer.android.com/reference/android/os/AsyncTask.html/ accessed on [22nd April 2017]

[4] Goadrich, M.H. and Rogers, M.P., 2011, March. Smart smartphone development: iOS versus Android. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 607-612). ACM.

[5] https://www.quora.com/What-is-waterfall-development-model-in-android-app-development/ accessed on [11th May 2017]

[6] Kurkovsky, Stan. "Engaging students through mobile game development." *ACM SIGCSE Bulletin* 41, no. 1 (2009): 44-48.