Student’s Facilitator

Final Year Project – Mid Report

Session 2016-2020

A project submitted in partial fulfilment of the

COMSATS University Degree

of

BS in Computer Science / Software Engineering (CUI)



Department of Computer Science

COMSATS University Islamabad, Lahore Campus

09 December 2020

# Evaluation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Group Members** (To be filled by students) | | | | (To be filled by supervisor) |
| Sr.# | Reg. # | Student Name | \*Signature | Obtained Marks  (Total Marks: 10) |
| (i) | FA16-BCS-232 | Ahmad Hassan Minhas |  |  |
| (ii) | FA16-BCS-054 | Ayishm Azeem |  |  |
| (iii) |  |  |  |  |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others.

Supervisor Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Supervisor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Remarks (if any): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

The advance era of technology has facilitated us in many ways and is helping us by making each task easier. By using this advancement of technology we will be working on an android application that will be a facilitation system for University students.

Student’s facilitation system will be providing its users a handy access to most useful resources that may make the users comfortable and less burdened. This application holds some interesting features that are added after considering the needs of students. Students will get an access to Semester notes that will be arranged subject wise, this feature is added to provide complete helping material to students for the good knowledge of subject, which will also help them to nourish their concepts. The feature of carpooling will be a great advantage to students. This will help the drivers to earn and passengers to get cheap rides. Hostelites suffers with finding their residence area near university. So for them a feature is added that helps them find accommodation for students. And a tutor will also be provided to students through this application, who will be seeking one to one attention of a person. Other queries of students will be answered through a discussion forum that will be designed to cater students in need of help within their environment.

This application will be acting as a student’s assistant. We will deploy this application in our university as well as places where it will be demanded.

Table of Contents

[Evaluation 2](#_Toc28121707)

[Chapter 1: Introduction 7](#_Toc28121708)

[1.1 Introduction 7](#_Toc28121709)

[1.2 Goals and Objectives 7](#_Toc28121710)

[1.3 Problem Statement 8](#_Toc28121711)

[1.4 Assumptions & Constraints 8](#_Toc28121712)

[1.5 Project Scope 8](#_Toc28121713)

[Chapter 2: Requirement Analysis 9](#_Toc28121714)

[2.1 Actors 9](#_Toc28121715)

[2.2 Requirement Elicitation 9](#_Toc28121716)

[2.2.1 Functional Requirements 9](#_Toc28121717)

[2.2.2 Non-Functional Requirements 10](#_Toc28121718)

[2.3 Use Case Description 11](#_Toc28121719)

[2.4 Use case design 20](#_Toc28121720)

[2.5 Software development life cycle 21](#_Toc28121721)

[Chapter 3: System Design 22](#_Toc28121722)

[3.1 Work breakdown structure (WBS) 22](#_Toc28121723)

[3.2 Activity Diagrams 23](#_Toc28121725)

[3.3 Network Diagram 31](#_Toc28121726)

[Chapter 4: System Testing 32](#_Toc28121727)

[4.1 Test Cases 32](#_Toc28121728)

[Chapter 5: Conclusion 51](#_Toc28121729)

[5.1 Problems faced and lessons learned 51](#_Toc28121730)

[5.2 Project Summary 52](#_Toc28121731)

[5.3 Future Work 52](#_Toc28121732)

[Chapter 6: References 54](#_Toc28121733)

**List of Figures**

[Figure 1. Admin Use case 21](#_Toc28243120)

[Figure 2. Students Use case 21](#_Toc28243121)

[Figure 3. V- Model 22](#_Toc28243122)

[Figure 4. WBS 23](#_Toc28243123)

[Figure 5. Registration 24](#_Toc28243124)

[Figure 6. Notes Sharing 25](#_Toc28243125)

[Figure 7. Carpooling 26](#_Toc28243126)

[Figure 8. Discussion Forum 27](#_Toc28243127)

[Figure 9. Hostels 28](#_Toc28243128)

[Figure 10. Tutor 29](#_Toc28243129)

[Figure 11. Profile 30](#_Toc28243130)

[Figure 12. Logout 31](#_Toc28243131)

[Figure 13.Sequence diagram of Login](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243132) 32

[Figure 14.Sequene diagram of registration process. 32](#_Toc28243133)

[Figure 15.Sequence diagram of Notes View/Download](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243134) 33

[Figure 16.Sequence diagram of uploading notes](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243135) 33

[Figure 17.Sequence diagram of posting hostel ads 34](#_Toc28243136)

[Figure 18. Sequence diagram of hostel view/contact](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243137) 34

[Figure 19.Sequence diagram of driver](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243138) 35

[Figure 20.Sequence diagram of taking ride by passenger](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243139) 35

[Figure 21.Sequence Diagram of tutor](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243140) 36

[Figure 22. Network Diagram 36](#_Toc28243141)

[Figure 23. Collaboration diagram of login page 37](#_Toc28243142)

[Figure 24. Collaboration diagram of Notes sharing 38](#_Toc28243143)

[Figure 25.Collaboration diagram of Hostel 38](#_Toc28243144)

[Figure 26. Carpooling collaboration diagram](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243145) 39

[Figure 27.Discussion forum](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243146) 39

[Figure 28. Collaboration diagram of Edit profile 40](#_Toc28243147)

[Figure 29. Collaboration diagram of tutors](../../../../D:/FYP%20material/FYP%20REPORT/students%20facilitator%20FYP1%20REPORT.docx" \l "_Toc28243148) 40

# Introduction

## Introduction

Through our application we want to help Students. We want to create an application that will facilitate students to get there university life easy. We are trying to create an application that is handy for students. Through this app students can earn some money, share their thoughts, notes and they also can place ads in order to solve residential and other issues. This app will help student to communicate with university members and will help them to find a good source for learning. Our main purpose of this app is to save students time and some effort forgetting information about any course and other resource problems.

## Goals and Objectives

In order to use this app Student will have to create an account. After that student will be allowed to use this app. We are giving following facilities in our app.

* There will be an activity for notes uploading and downloading. Student will upload and download notes and will be able to search notes through this activity.
* There will be an activity where student will able to book his ride to destination and other student will post a request for give ride to other student in order to get cash.
* There will be an activity where students will post their ads related to their hostels and other students will see those ads and interact with them as well.
* There will be an activity for discussion purpose. Student will ask question and give answers to them on this activity.
* There will be also an activity for tutors available for related course. And they will be Students that will register as tutors.

## Problem Statement

Freshies in university face many problems during their first semester in University. Being unaware of the rules and regulations, curriculum and future academic decisions, Also they are helpless to get any academic assistance, except from teachers. It is difficult for some students to stay organised and cope up with educational burden, so they lack behind and starts to get demotivated. Students from out of city and country find difficulties in getting nearby residencies. They keep wandering to places to look for accommodation being unnoticed by any availability of accommodation near to university. With that students, also considering hostelites having conveyance issues, pays high fare and few of those who have some trust issues with the public service providers keep those tasks pending that requires travelling. There are many problems that each and every student face related to some features added into Students’ facilitator application. The aim is to provide a helping hand to students, by joining them on a same platform. It will provide organised notes of all departments, sorted according to semester, subjects, and chapters with other helping material. A discussion forums with a friendly disciplined and helping environment to get notified with solution of common problems and suggestions regarding any issue that rises. Or it can be said that this application will be a bridge that is created among students.

## Assumptions & Constraints

* This application will provide assistance to University members only.
* Student will only be able to register with their registration number.
* This is an online application, which requires the access of Internet connection.
* Driving licence will be necessary for registration as a driver to carpool.
* The project will release in May, 2020.

## Project Scope

No such application provides all the above features in a single platform. Universities do provide students with the facility of their own portal that helps them to keep a track of their academic performance as well as a service portal that helps students with problems related to administration level. This can be accessed anywhere at any time, helping students to get advantage of their mobile devices. This will help students to deal with issues in an automated manner. Carpooling will have a great impact on students ease of transportation, as students travel in the same time schedule, off timings or holidays in case, thus makes it feasible by limiting it to campus. Application will be implemented according to android application standards.

# Requirement Analysis

This chapter outlines the functional and non-functional requirements of the system. It will provide the list of the users of this applications.

## Actors

* Admin
* Students

## Requirement Elicitation

### Functional Requirements

**FR01: Register**

* Students will be able to register with their University student ID
* Admin will register with their university email address and password.

**FR02: Login**

* Students and admin will be able to login into the application with an ID and password with which they registered.

**FR03: Logout**

* Application will allow any type of user to log out.

**FR04: Authentication**

* The system will verify user’s registration and login to confirm that user belongs to University.
* An error will be generated with wrong entry of login **credentials**.

**FR05: Notes Sharing**

* Students will be shown the notes related to the semester and department they belong to.
* The system will allow students to download the documents.
* Using the drop down menus, students will be able to filter notes with departments, semester and subject wise.
* Reporting option in this module will report admin about the irrelevant, unusable or any issue in the uploaded document.
* A button will allow user to get to the interface where students will be able to upload documents with few details that will be department, semester, subject and the topic to which notes belong to.

**FR06: Carpooling**

* The system will allow students to post and request a trip.
* Posted trip will include car details, destination, vacant seats, time and cost.
* The system will notify the passengers about the driver that posted the same destinations as they will be willing to travel to.
* It will allow them to accept trip and communicate with drivers through a chat box.

**FR07: Discussion Forum**

* System will allow user to post their queries and problems
* An answer button will allow students to reply to the particular post.
* System will prioritize the post with the greater number of likes by showing it at the top of news feed to all the users.

**FR08: Hostels**

* A wall with list of hostel will be viewed by the user.
* The system allows the user to post an advertisement of an accommodation with details including the room picture, its size, location and rent.
* Users will be allowed to report fake content shared on the wall.

**FR09: Tutor**

* A list of tutors with their specific details will be displayed and students will be given an opportunity to contact a tutor through chat box
* A student will also be able to post for a position of tutor.
* The system will only allow students of 5th or above semester to be eligible for the position of a tutor.

**FR10: Profile**

* The system will allow admin and students to update and save their profile.

### Non-Functional Requirements

**NFR01: Performance**

* Minimized response time of an application.

**NFR02: Availability**

* The application will always be available to its user, as its services will be needed 24/7.

**NFR03: Usability**

* The system will be easy to use as it will be interactive, and will not require training of users.

**NFR04: Security**

* System will only allow University students to access an application, or those holding the university email address to access admin site.
* System will not allow users to view each other’s profile.
* System will ban users that will be found violating rules of application

**NFR05: Extensibility**

* The system shall allow adding more features to provide much better services in future.

**NFR06: Extensibility**

* It will be easy to maintain, to upgrade it in future

### Traceability Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Technical Requirements | Functional Requirement | Status | Actors | Type | Priority | Complexity | Use Case ID | Activity Diag | Sequence Diagram | Collaboration Diagram |
| 01 | User will be able to register in the system. | Register |  | Student, admin | functional |  |  | 01 | Fig5 | Fig 14 | - |
| 02 | User will be able to log into the system. | Login |  | Student, admin | functional |  |  | 02 | Fig 5 | Fig 13 | Fig 24 |
| 03 | User will be able to log out from the system | Logout |  | Student, admin | functional |  |  | 09 | fig 12 | - | - |
| 04 | User will be verified | Authentication |  | Student, admin | functional |  |  | 03 | - | - | - |
| 05 | Students will be able to view, upload, and download notes. | Notes Sharing |  | Student | functional |  |  | 04 | Fig 6 | Fig 15,16 | Fig 25 |
| 06 | Students will be able to take and give ride | Carpooling |  | Student | functional |  |  | 05 | Fig 7 | Fig 19,20 | Fig 27 |
| 07 | Students will be able to get their problems/queries sorted. | Discussion Forum |  | Student | functional |  |  | 06 | Fig 8 | Fig 22 | Fig 28 |
| 08 | Students will be view and post new hostel details | Hostels |  | Student | functional |  |  | 07 | Fig 9 | Fig 17,18 | Fig 26 |
| 09 | Students will be able to hire and register as tutors | Tutor |  | Student | functional |  |  | 08 | Fig 10 | Fig 21 | Fig 30 |
| 10 | Students will be able to edit their profile | Profile |  | Student, admin | functional |  |  | 10 | Fig 11 | - | Fig 29 |

## Use Case Description

|  |  |
| --- | --- |
| **Use Case ID:** | 01 |
| **Use Case Name:** | Registration |
| **Actors:** | Student, Admin |
| **Description:** | This use case describes the process by which the related actors can register. |
| **Pre - Condition:** | The actor must not be registered and have no record before. |
| **Post - Condition:** | The actor should be successfully registered and have a record in database. |
| **Normal Flow Of Events:** | * The actor will open the application. * The actor will tap the “Sign up” button and the system will lead to Sign up activity. * The student will enter his/her credentials i.e. Student ID, username, Password etc. * Admin will register through their university email address and new password. * The actor will tap “Submit” button and will be headed to Login activity. * The actor should be registered successfully and redirect actor to login activity. |
| **Alternative flows:** | If an actor is already registered with the given credentials then redirect him/her to Sign up instead.  Inform actor if any of the credentials are in violation of the rules. |
| **Exceptions:** | <<includes>> |

|  |  |
| --- | --- |
| **Use Case ID:** | 02 |
| **Use Case Name:** | Login |
| **Actors:** | Student, Admin |
| **Description:** | This use case describes the process by which the related actors can Login. |
| **Pre - Condition:** | The actor must be registered and have record in system. |
| **Post - Condition:** | The actor should be successfully logged in and have an online record in database. |
| **Normal Flow Of Events:** | * The actor will head to log in activity in application. * The actor will tap the “Log in” button and the system will lead to log in activity. * The actor will enter his/her Student ID/ Student email or username, Password etc. * The actor will tap "Log in” button and will be headed to home activity. * The actor should be logged in successfully and lead to actor to home activity. |
| **Alternative flows:** | If an actor is not registered with the Same Student Id then it will redirect to sign up activity to sign up before log in.  Inform actor if Student Id and Password are incorrect. |
| **Exceptions:** | <<includes>> |

|  |  |
| --- | --- |
| **Use Case ID:** | 03 |
| **Use Case Name:** | Authentication |
| **Actors:** | Student, Admin |
| **Description:** | This use case authenticate that actor belongs to the university. |
| **Pre - Condition:** | The actor must be a university student. |
| **Post - Condition:** | The actor should be successfully registered and have a record in database. |
| **Normal Flow Of Events:** | * The actor will be authenticated by its Student Id if he/she is from university. |
| **Alternative flows:** | If actor is not from the university then he cannot register. |
| **Exceptions:** | <<includes>> |

|  |  |
| --- | --- |
| **Use Case ID:** | 04 |
| **Use Case Name:** | Notes Sharing |
| **Actors:** | Student |
| **Description:** | This use case describes the process by which students can access, upload and report notes. |
| **Pre - Condition:** | The actor must be logged in to application as a Student. |
| **Post - Condition:** | The actor will able to see content on notes sharing activity. |
| **Normal Flow Of Events:** | * The actor will tap on Notes Sharing activity. * The actor will filter the notes by specifying the details i.e. department, semester and subject. * They will download notes and will report and delete them by taping on download, report and delete button respectively. * The actor will upload notes by tapping on upload button. * The actor will be shown a pop up form where he/she will fill the details about notes. * The actor will tap on "confirm and upload" button. * The actor will be shown a success message and notes will be updated in system. |
| **Alternative flows:** | If the notes could not be downloaded, system will show the actor an error message about the system error or internet error.  System will show a try again and back button. |
| **Exceptions:** | none |

|  |  |
| --- | --- |
| **Use Case ID:** | 05 |
| **Use Case Name:** | Carpooling |
| **Actors:** | Student |
| **Description:** | This use case describes the process by which the actor can book a ride and post req to give a ride. |
| **Pre - Condition:** | The actor must be logged in to application as student. |
| **Post - Condition:** | The actor will be able to manage carpooling activity. |
| **Normal Flow Of Events:** | * The actor will open the carpooling activity. * The actor will select whether he will take a ride or give a ride. * The actor will enter the details i.e. ride time, car detail, destination etc. * The actor will tap "post" button. * The request will be posted in system. * Every Actor will see a notification according to their request criteria. * The actor will accept or reject the request   1. Actor will contact to the related actor. * After the ride actor will rate the other actor. |
| **Alternative flows:** | If request is not be posted, system will show an error message. |
| **Exceptions:** | none |

|  |  |
| --- | --- |
| **Use Case ID:** | 06 |
| **Use Case Name:** | Discussion Forum |
| **Actors:** | Student |
| **Description:** | This use case describes the process by which the actor can ask question and give answer on the discussion forum. |
| **Pre - Condition:** | The actor must be logged in to application as student. |
| **Post - Condition:** | The actor will be able to ask and answer the questions. |
| **Normal Flow Of Events:** | * The actor will open the Discussion forum. * The actor will see the recent questions posted by the students on a timeline view. * The actor will type his/her question in text box in the top. * The actor will tap the post button. * The actor will give likes will answer the questions and will give response to the answers given. * The actor will report any abusive comment. * Every Actor will see a notification in response of their question and answers. |
| **Alternative flows:** | If question cannot be post system will show an error message. |
| **Exceptions:** | none |

|  |  |
| --- | --- |
| **Use Case ID:** | 07 |
| **Use Case Name:** | Hostels |
| **Actors:** | Student |
| **Description:** | This use case describes the process by which the actor can post an ad about hostels and contact the actor. |
| **Pre - Condition:** | The actor must be logged in to application as student. |
| **Post - Condition:** | The actor will be able see the ads and contact the actor activity. |
| **Normal Flow Of Events:** | * The actor will open the hostels activity. * They will see all ads. * The actor will choose a hostel and tap on contact button. * The actor will tap on place add button. * The actor will fill details about hostel ads. * The actor will confirm and post add. * The ad will be placed in the system. |
| **Alternative flows:** | If request cannot be post system will show an error message. |
| **Exceptions:** | none |

|  |  |
| --- | --- |
| **Use Case ID:** | 08 |
| **Use Case Name:** | Tutor |
| **Actors:** | Student |
| **Description:** | This use case describes the process by which the actor can contact tutors and register ass a tutor. |
| **Pre - Condition:** | The actor must be logged in to application as student. |
| **Post - Condition:** | The actor will be able to manage tutors activity. |
| **Normal Flow Of Events:** | * The actor will open the tutor activity. * The actor will see a list of all available tutors. * The actor will contact a tutor. * The actor will register as a tutor by tapping on register as a tutor button. * The actor will confirm and post his/her position as a tutor. * The actor will be verified if he/she is a student of 5th semester or more. * The actor will be registered as a tutor. |
| **Alternative flows:** | If request cannot be post system will show an error message. |
| **Exceptions:** | none |

|  |  |
| --- | --- |
| **Use Case ID:** | 09 |
| **Use Case Name:** | Logout |
| **Actors:** | Student, Admin |
| **Description:** | This use case describes the process by which the actor can log out of the system |
| **Pre - Condition:** | The actor must be logged in to application. |
| **Post - Condition:** | The actor will be logged out, and will be shown login page again. |
| **Normal Flow Of Events:** | * The actor will tap logout * The actor will get a confirm message * The user will confirm and will be logged out |
| **Alternative flows:** | If confirmation cancelled, user isn’t logged out then. |

|  |  |
| --- | --- |
| **Use Case ID:** | 10 |
| **Use Case Name:** | Edit Profile |
| **Actors:** | Student, Admin |
| **Description:** | This use case describes the process by which the actor can edit his/her profile. |
| **Pre - Condition:** | The actor must be logged in to application. |
| **Post - Condition:** | The actor will be able to manage profile activity. |
| **Normal Flow Of Events:** | * The actor will open his/her profile. * The actor will apply changes to his/her profile i.e. change name, change display picture. |
| **Alternative flows:** | If changes cannot be made system will show an error message. |
| **Exceptions:** | <<includes>> |

## Use case design

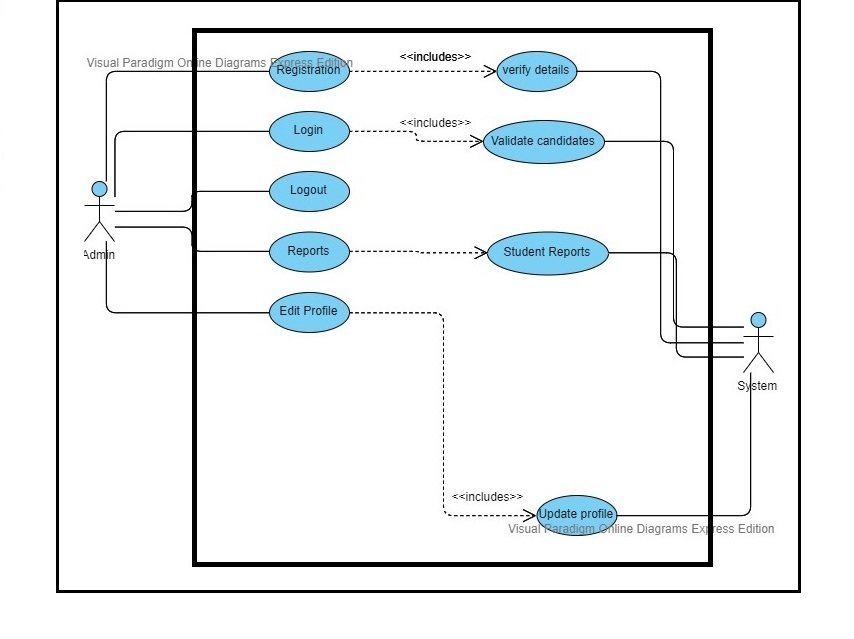


Figure 11. Admin Use case

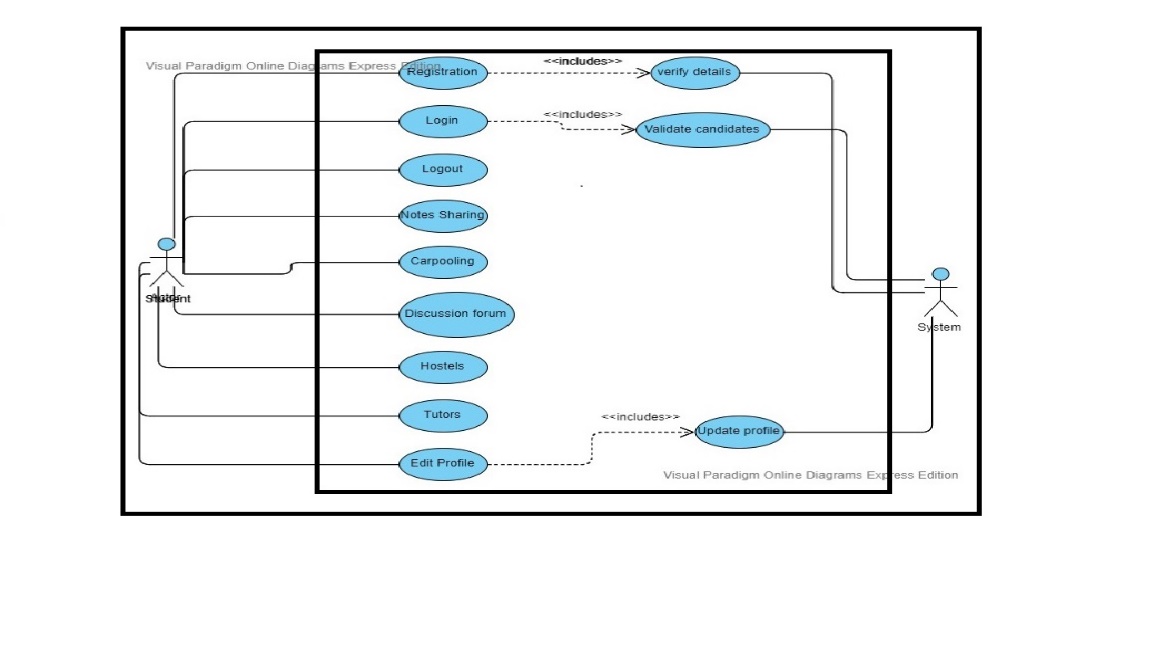


Figure 2. Students Use case

## Software development life cycle

Figure 3. V- Model

We will use the V-Model.

* The advantages of the V-Model are following:
* Easy in use.
* Error findings will be easy.
* Flaws will be lesser.
* Works well for small projects where requirements are easily understood.

# System Design

## Work breakdown structure (WBS)

## 

Figure 4. WBS

## Activity Diagrams

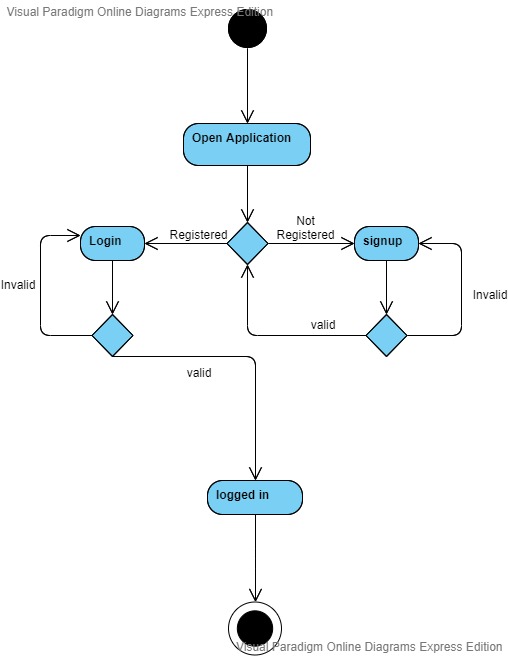
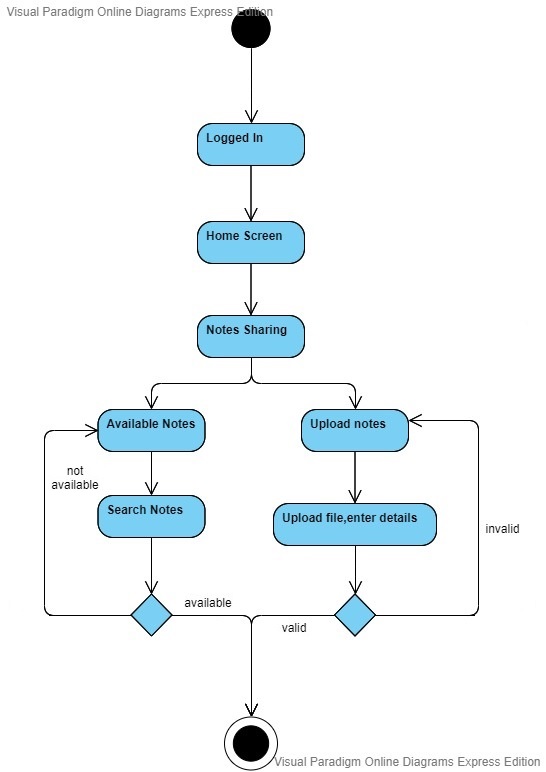
Figure 5. Registration

Figure 6. Notes Sharing

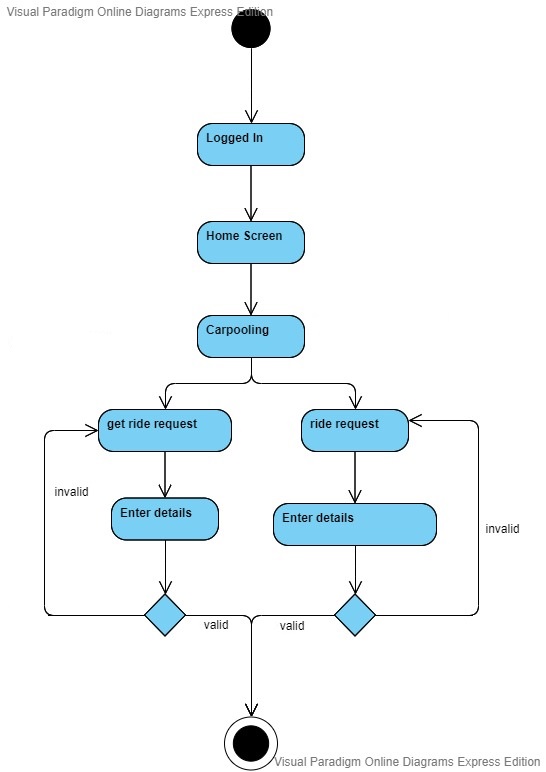


Figure 7. Carpooling

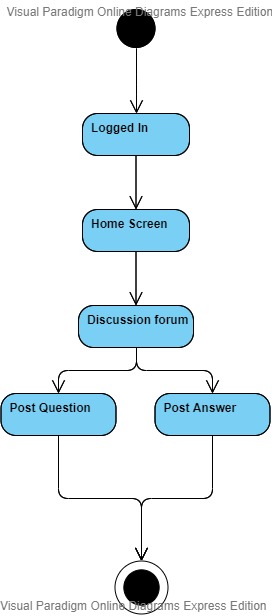
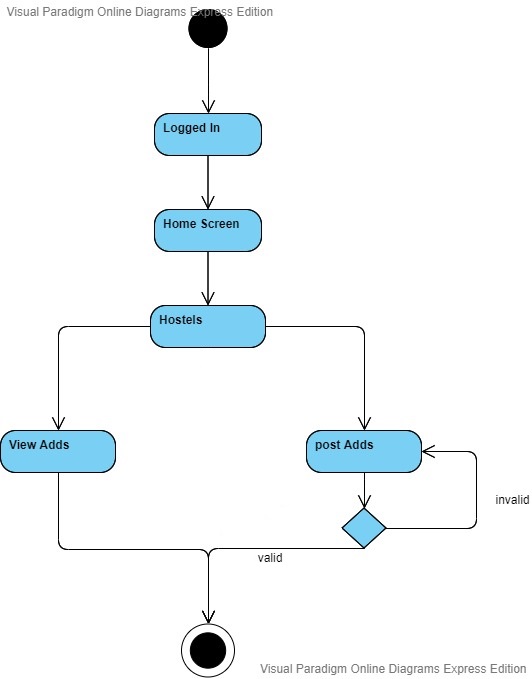


Figure 8. Discussion Forum

Figure 9. Hostels

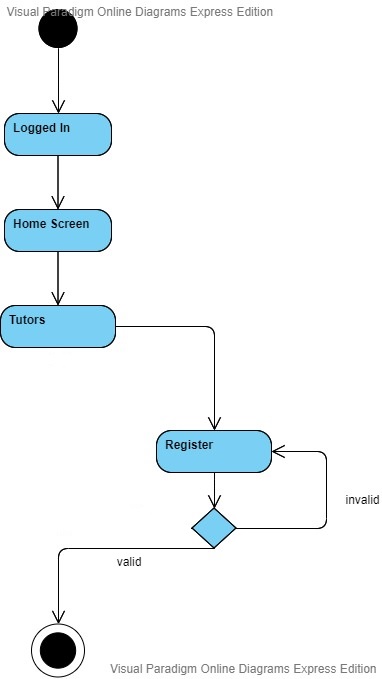
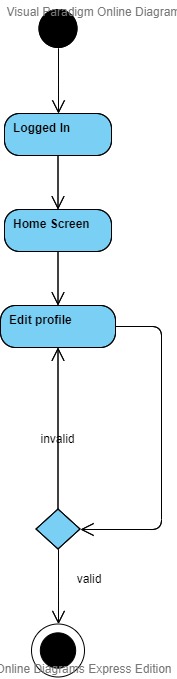


Figure 10. Tutor

Figure 11. Profile

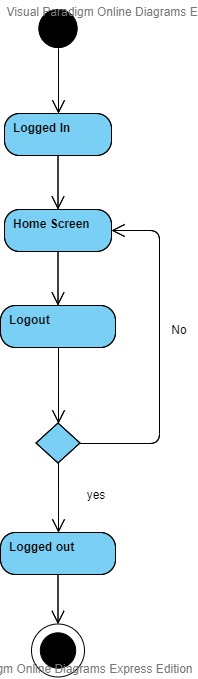


Figure 12. Logout

## Sequence Diagram

# 

Figure 13.Sequence diagram of Login

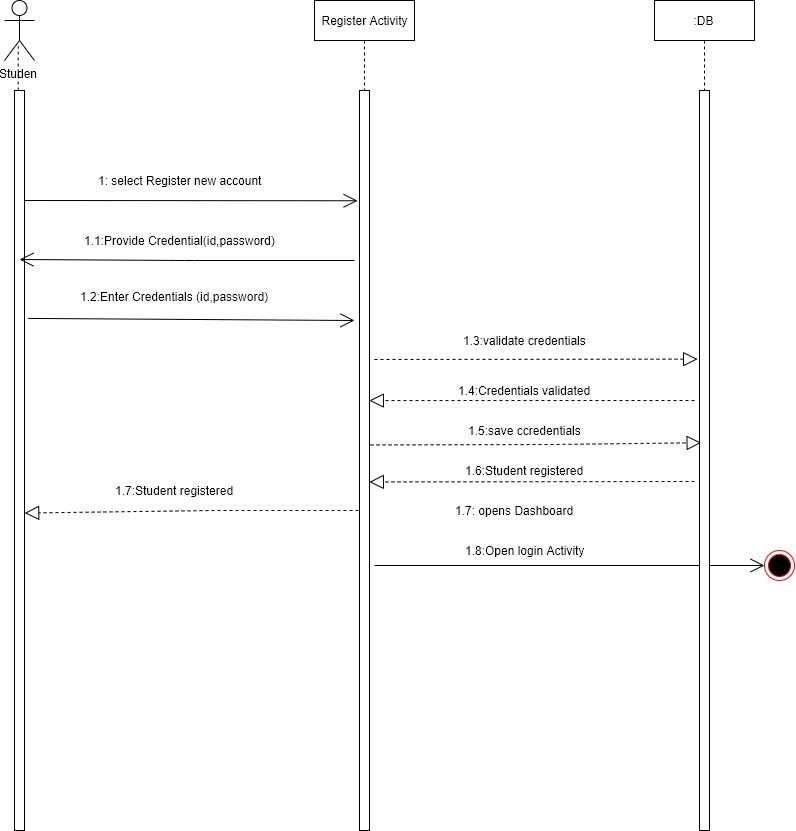


Figure 14.Sequene diagram of registration process.

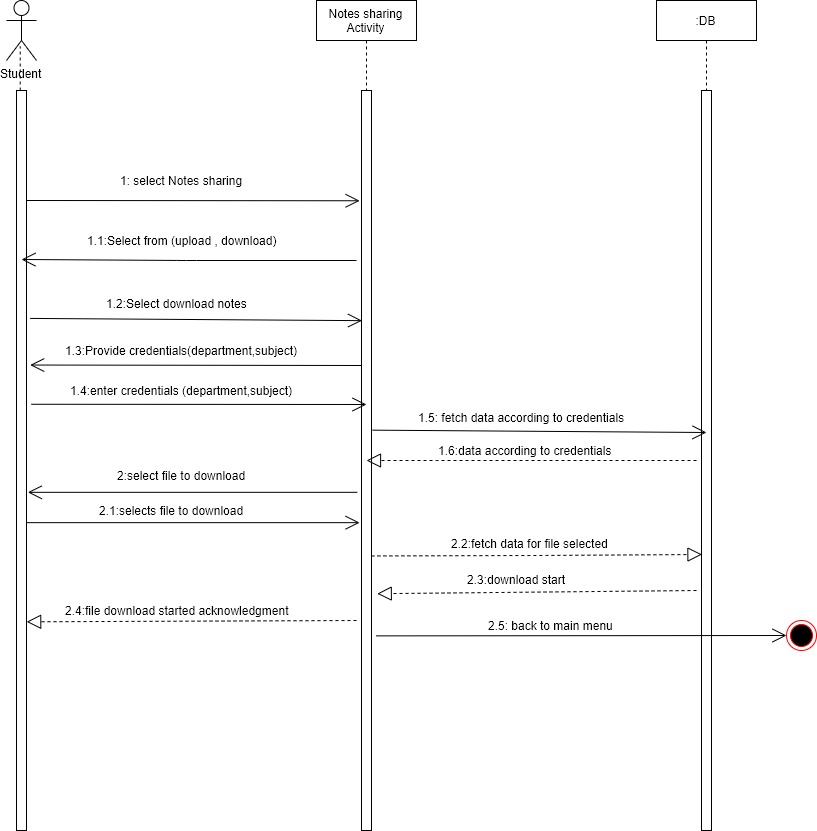


Figure 15.Sequence diagram of Notes View/Download

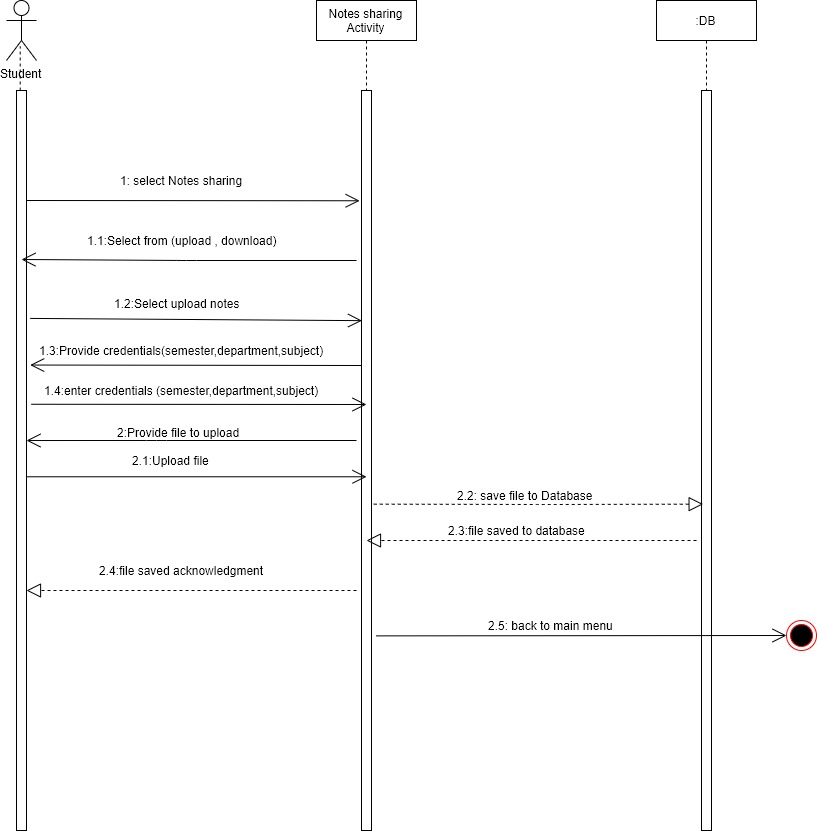


Figure 16.Sequence diagram of uploading notes

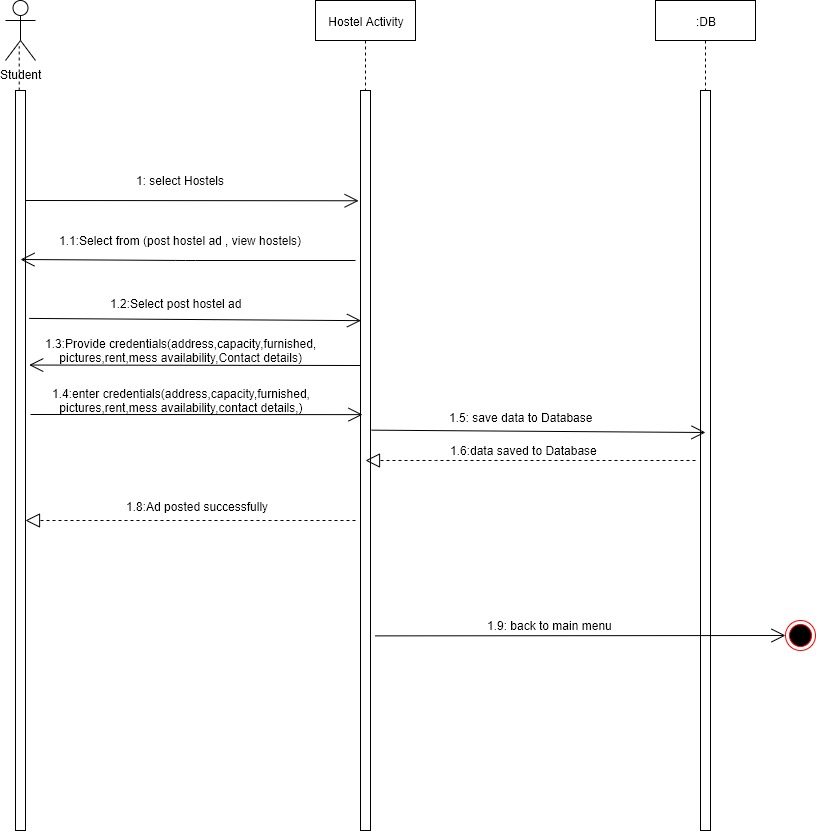


Figure 17.Sequence diagram of posting hostel ads

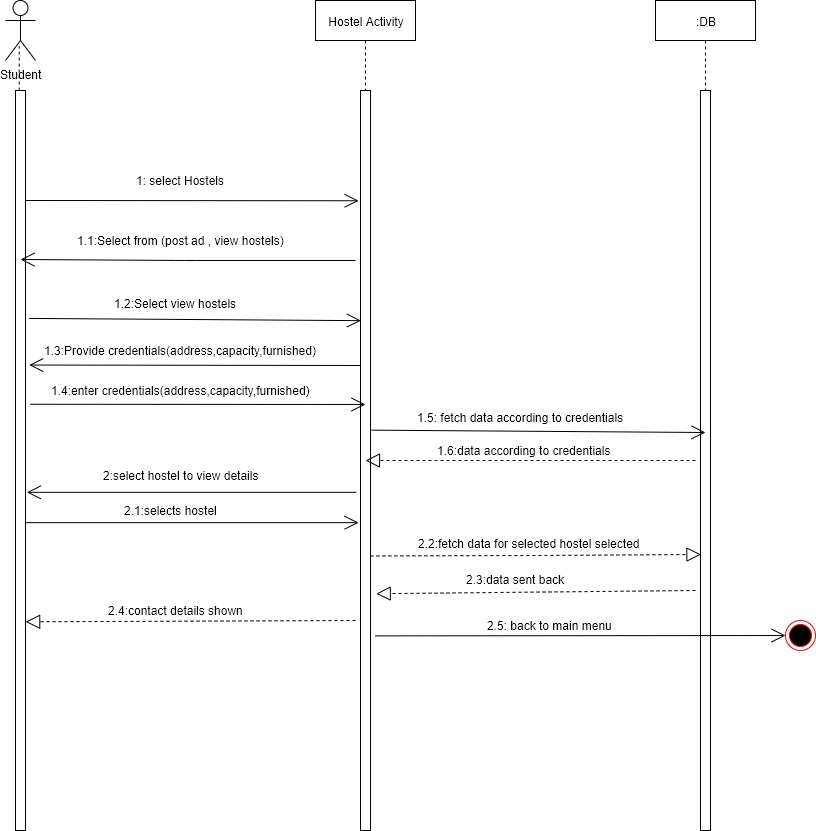


Figure 18. Sequence diagram of hostel view/contact

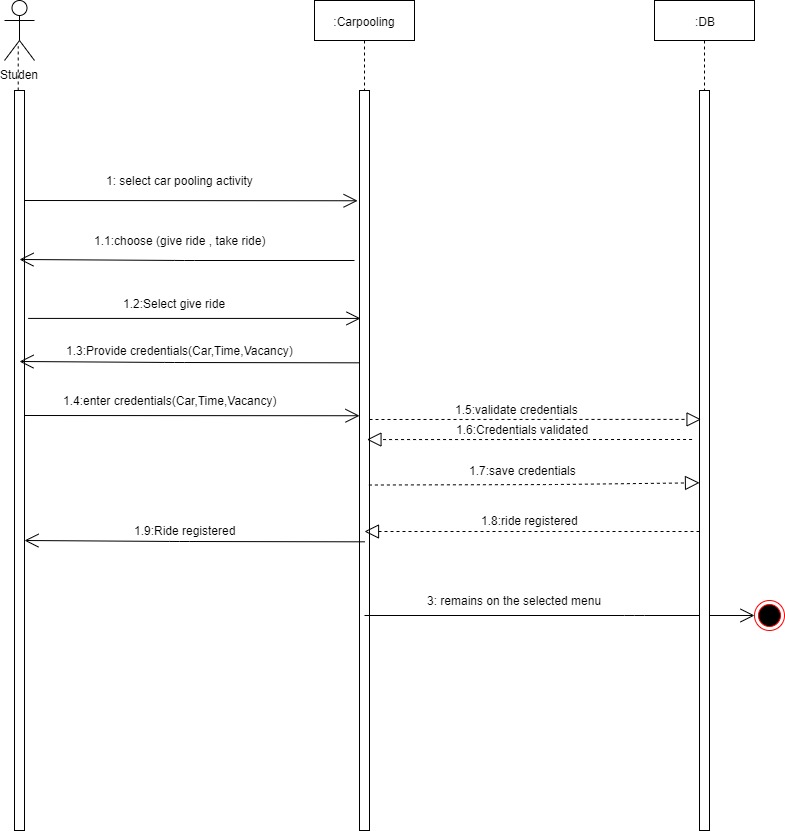


Figure 19.Sequence diagram of driver

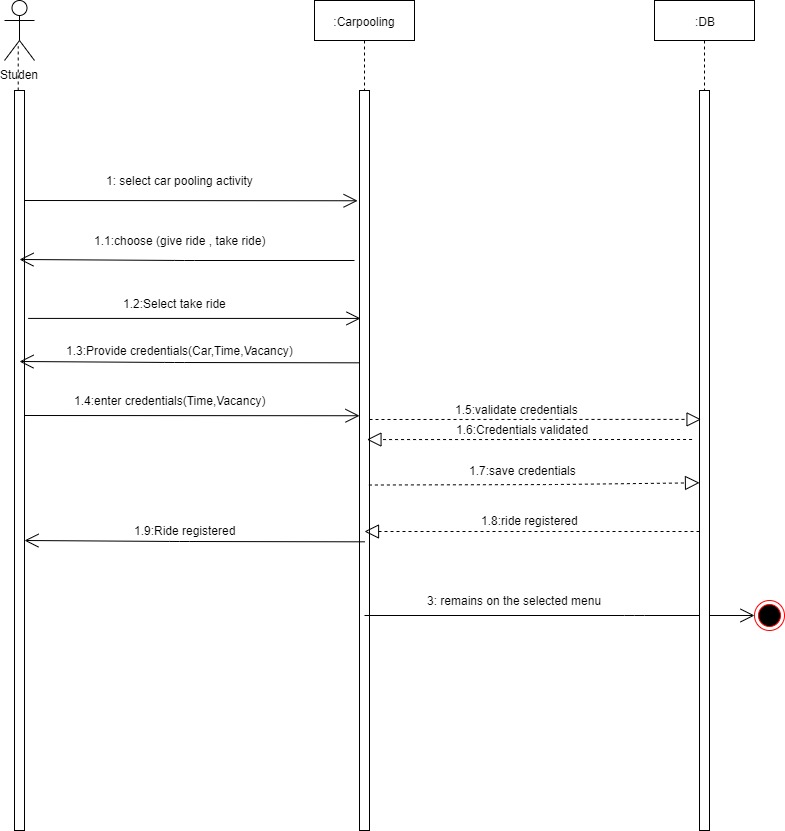


Figure 20.Sequence diagram of taking ride by passenger

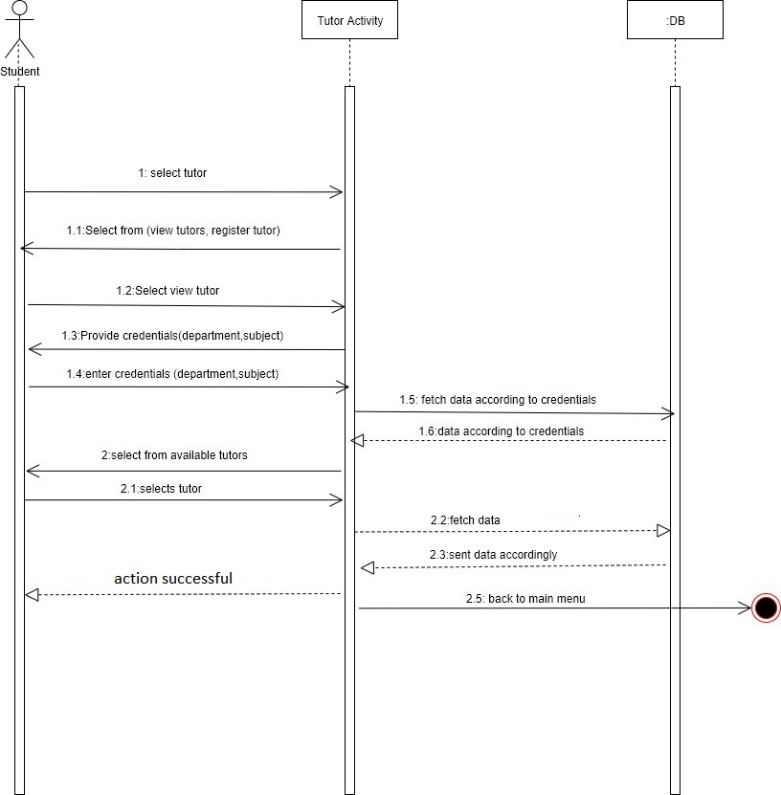


Figure 21.Sequence Diagram of tutor

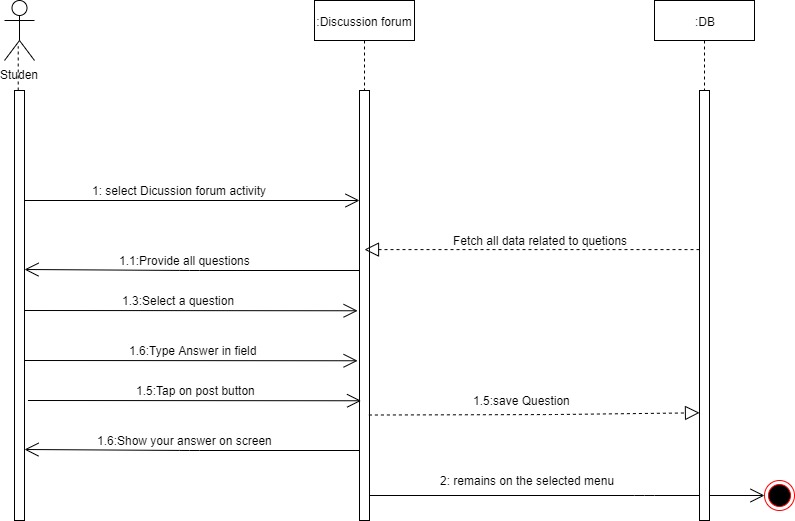


Figure 22. Sequence Diagram of Discussion Forum

## Class Diagram

## Database Diagram

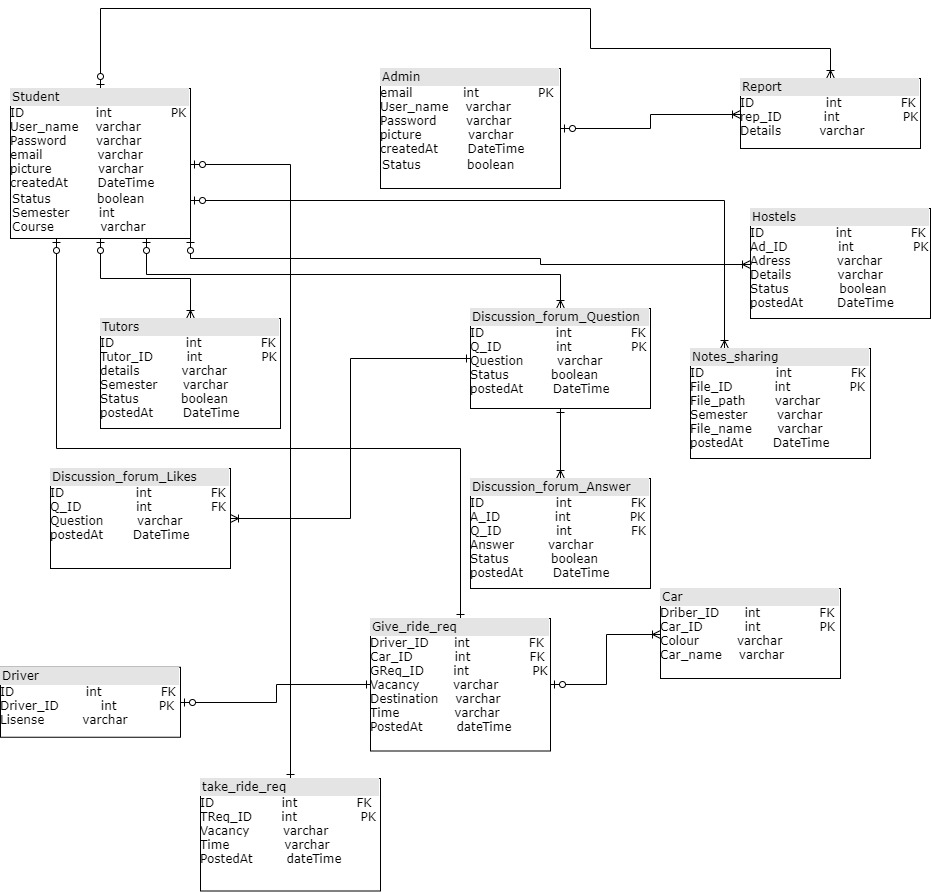


Figure 23. Database Diagram

## Network Diagram

Figure 24. Network Diagram

## Collaboration Diagram

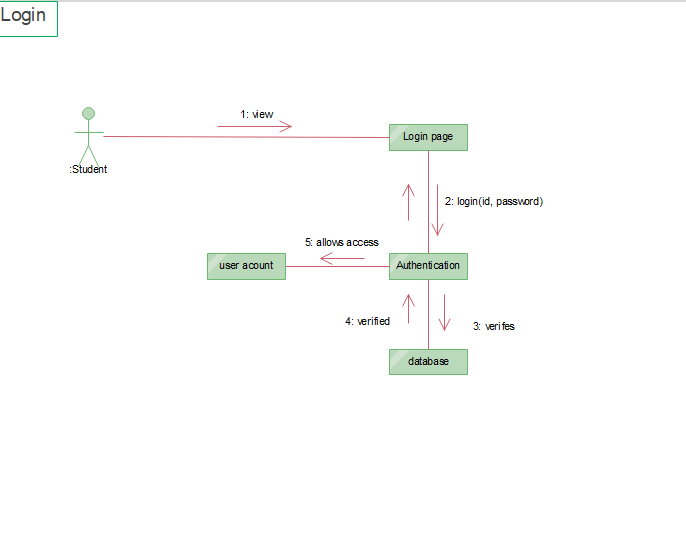


Figure 25. Collaboration diagram of login page

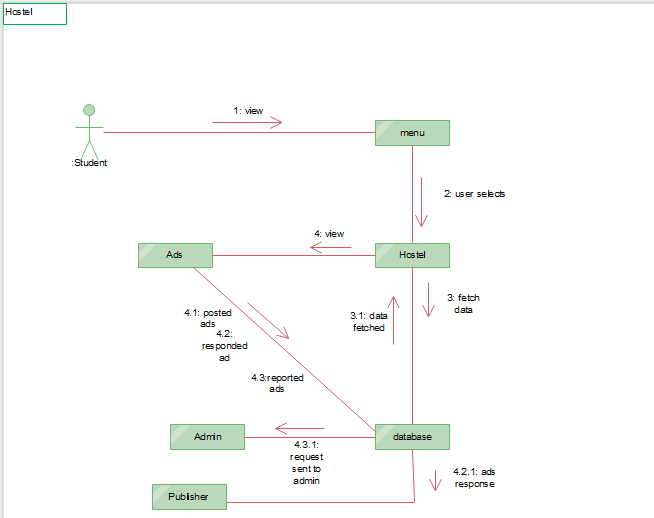
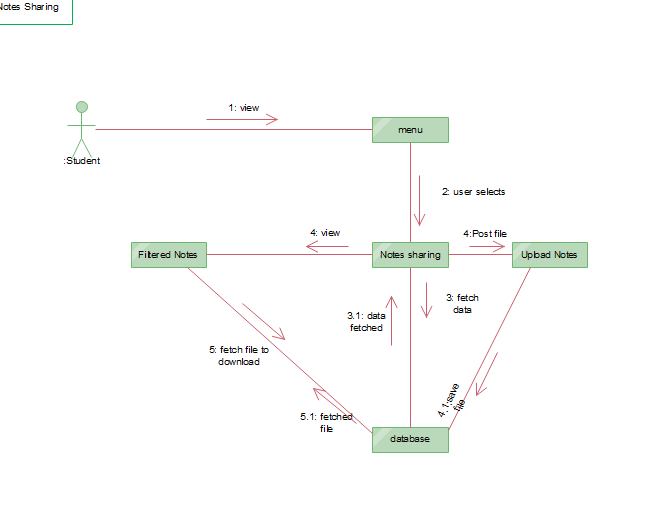
Figure 26. Collaboration diagram of Notes sharing

Figure 27.Collaboration diagram of Hostel

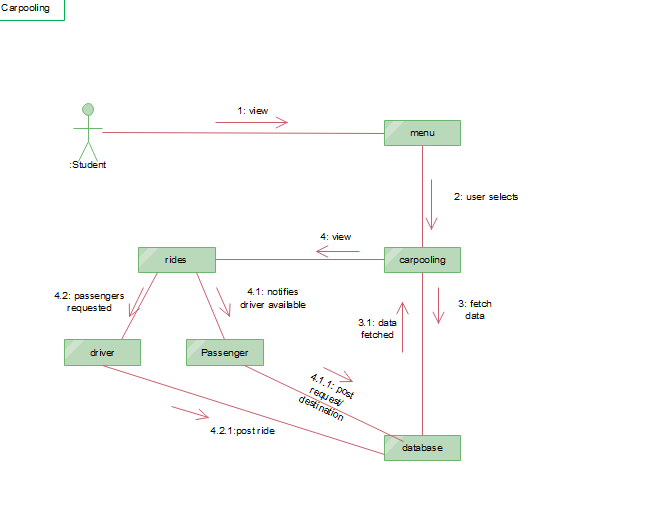
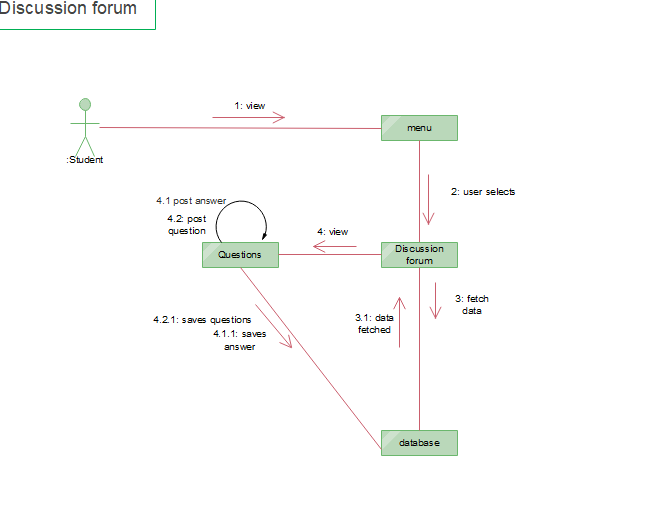


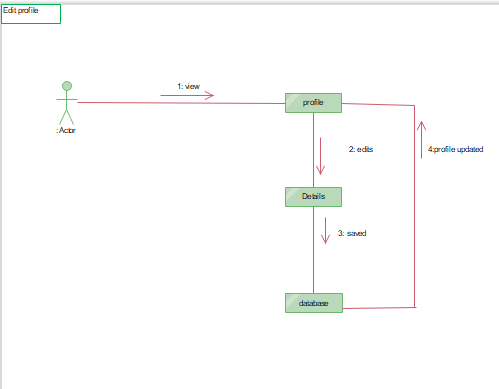
Figure 28. Carpooling collaboration diagram



# 

Figure 29.Discussion forum

# 



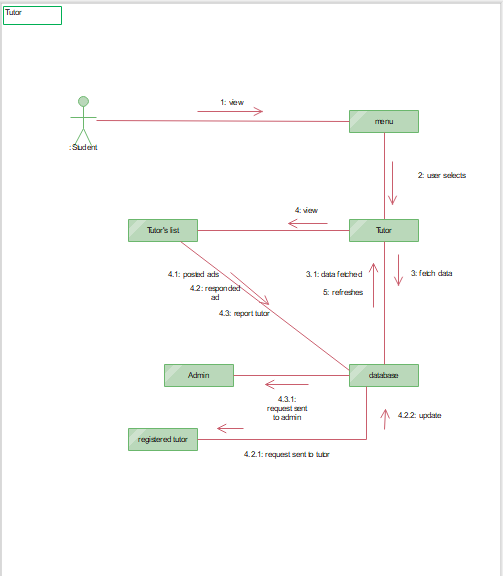
Figure 30. Collaboration diagram of Edit profile

Figure 31. Collaboration diagram of tutors

# System Testing

## Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 1 | **Test Case Name:** | Registration |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Registration process for students shall be successful. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student shall be the member of university, and is signing up in the system for the first time. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps on “Sign up” | A registration form is displayed in result. |  |  |
| 2. | The student enters the registration number username and password. | System validates the user’s information. |  |  |
| 3. | Students submits their details by tapping “Submit”. | Login page is displayed for the user then. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | Student successfully registers into the system.  Student’s record will be saved in the database. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 2 | **Test Case Name:** | Registration |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Admin |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Registration process for Admin shall be successful. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Admin shall be the member of university, and is signing up in the system for the first time. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Admin taps on “Sign up” | A registration form is displayed in result. |  |  |
| 2. | The admin enters their username, university email address and a new password in particular in specific fields. | System validates the user’s information. |  |  |
| 3. | Admin submits their details by tapping “Submit”. | Login page is displayed for the user then. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | Admin successfully registers into the system.  Admin’s record will be saved in the database. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 3 | **Test Case Name:** | Login |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | The process through which user can log in. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Students should have registered before, and shall have the record in database. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps on “Log in” | A Login form is displayed. |  |  |
| 2. | The student enters the registration number password. | System checks the record. Validates the information entered |  |  |
| 3. | Students taps “Login”. | * System verifies the details from database. * Home page of the system is displayed to the user. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | Student successfully enters into the system. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 4 | **Test Case Name:** | Login |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Admin |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | The process through which admin can log in. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Admin shall already have an account to login, and shall have the record in database. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Admin taps on “Log in” | A Login form is displayed. |  |  |
| 2. | Admin enters an email address and password. | * System checks the record. Validates the information entered |  |  |
| 3. | Admin taps “Login”. | * Home page for admin is displayed to the user. * System verifies the details from database. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | Admin successfully enters into the system. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 5 | **Test Case Name:** | View and download notes |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Access notes. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Students must be logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Students tap on “Notes” in menu on homepage. | System loads a screen where notes are uploaded. |  |  |
| 2. | Student filters notes  Taps “department” dropdown menu | * Lists displays. * Notes filtered according to selected department. |  |  |
| 3. | Student taps on “Semester” dropdown menu. | * . Lists displays. * Notes filtered according to selected Semester. |  |  |
| 4. | Student taps on “Subject” dropdown menu | * Lists displays. * Notes filtered according to selected subject. |  |  |
| 5. | Tap “download” against particular document | * Downloads notes |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | After tapping filtering options, filtered notes are displayed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 6 | **Test Case Name:** | Report notes |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ahmad Hassan | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able to report notes. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be logged in.  Student must be in the notes section. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Student taps on “Report” button. | System displays a form to report document. |  |  |
| 2 | Student enters details | - |  |  |
| 3. | Student taps on “Submit” | System displays a message “Reported”. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Report is sent to the Admin.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 7 | **Test Case Name:** | Upload notes |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able to upload notes. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Student taps on “Upload Notes” button. | System displays a screen to upload documents. |  |  |
| 2 | Student chooses file to upload.  Students enter details related to document. | - |  |  |
| 3. | Student taps on “Submit” | System displays a message “successfully uploaded”. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Notes are uploaded with specific details.**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 8 | **Test Case Name:** | Carpooling, take ride |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ahmad Hassan | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able to upload notes. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Students tap on “Carpool” in menu on homepage. | System loads a screen for carpooling process. |  |  |
| 2. | Students taps “destination” | Dropdown list of destination points will be displayed. |  |  |
| 3. | Students enters extra details i.e. timings, their name. | - |  |  |
| 4. | Student taps on “post” | Request is sent.  A message is displayed “We’ll notify you about the matching ride” |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student is notified about the ride available.**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 9 | **Test Case Name:** | Carpooling, as driver |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able to upload notes. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be in carpool section. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps “Post as Driver” button | System loads a screen that shows the ads of hostels. |  |  |
| 2. | Students taps on “destination” | Dropdown list of destination points will be displayed. |  |  |
| 3. | Students enters extra details i.e. timings, their name, vacant seats, car details, cost. | - |  |  |
| 4. | Student taps on “post” | Request is sent.  A message is displayed “We’ll notify you about passengers available for the ride” |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student is notified about the ride someone wants to avail.**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 10 | **Test Case Name:** | Hostels |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 30, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able to view and request for accommodation available on the page. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be in logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps “hostels” in menu on homepage. | System loads a screen that shows accommodations available. |  |  |
| 2. | Students taps “contact icon” | A request form will be displayed for user to enter their details. |  |  |
| 3. | User enters request details for hostel in the form. | - |  |  |
| 4. | Student taps on “Confirm” | Request is sent.  A message is displayed “Your request sent, We will contact you in a while” |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student successfully views hostels list.**  **Students can successfully request for particular ad posted.**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 11 | **Test Case Name:** | Hostels, post accommodation. |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 30, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able post ads for hostels. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be logged in.  Student must be in hostels section. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps “Post ad” | System loads a form. |  |  |
| 2. | Students enters the details to post an ad. | - |  |  |
| 4. | Student taps on “Submit” | A post is uploaded in view list. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student successfully posts ad.**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 12 | **Test Case Name:** | View/Request tutors |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ahmad Hassan | **Design Date:** | November 30, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able to view and contact tutors displayed on the screen. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps “Tutors” in menu on homepage. | System loads a screen where notes are uploaded. |  |  |
| 2. | Student taps on “contact” button against a tutor details | System displays a form. |  |  |
| 3. | Student enters the required details | - |  |  |
| 4. | Student taps on “Submit” | A message is displayed “Request sent, tutor will contact you back in a while”. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student successfully views the list of tutors.**  **Student requests to hire a tutor.**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 13 | **Test Case Name:** | Register as a tutor |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students will be able to register as a tutor. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be in Tutor’s section.  Student must be in or above 5th semester. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps “Register as tutor” button | System loads a form. |  |  |
| 2. | Students enters details. | - |  |  |
| 4. | Student taps on “post” | System posts ad in view list of tutors. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student successfully registers as a tutor and will receive requests by users willing to hire them**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 14 | **Test Case Name:** | View/Reject request received by a tutor. |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students accepts or rejects whatever request they have received by students. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be in Tutor’s section.  Student must be in or above 5th semester. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps “view requests” button | System displays list of requests, if received. |  |  |
| 4. | Student taps on “reject request” | System send notification to the student requested for the desired tutor. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student successfully delete requests by users willing to hire them**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 15 | **Test Case Name:** | View/Accept request received by a tutor. |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 29, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students accepts or rejects whatever request they have received by students. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student must be in Tutor’s section.  Student should have registered as a tutor.  Student must be in or above 5th semester. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1. | Student taps “view requests” button | System displays list of requests, if received. |  |  |
| 4. | Student taps on “Accept request” | System sends notification to the student requested for the desired tutor. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | **Student successfully accepts requests received by students.**  **Database is updated.** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 16 | **Test Case Name:** | Edit Profile |
| **System:** | **Student’s Facilitation System** | **Subsystem:** | Student and Admin |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 30, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Student and Admin can edit their own profile. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Student and Admin must be logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Student and Admin clicks on “Profile” button. | System displays profile. |  |  |
| 2 | Student and Admin edit details | - |  |  |
| 3 | Student and Admin clicks on “Submit” button. | System validates details and displays “Successfully Edited” message |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | Profile is edited. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case #:** | 17 | **Test Case Name:** | Logout |
| **System:** | Student’s facilitator System | **Subsystem:** | Students and Admin |
| **Designed By:** | Ayishm Azeem | **Design Date:** | November 30, 2019 |
| **Executed By:** |  | **Execution Date:** |  |
| **Description:** | Students and Admin can logout from system. | | |

|  |  |
| --- | --- |
| **Pre-Conditions:** | Students and Admin must be logged in. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comment** |
| 1 | Students and Admin clicks on “Logout” button. | System displays a login page. |  |  |

|  |  |
| --- | --- |
| **Post-Conditions:** | Students and Admin will be logged out from system. |

## Unit/Integration/Acceptance Testing

### Unit Testing

The purpose of unit testing is to evaluate units of the system. This type of testing helps to make sure that all units are working properly. Units that are supposed to be tested are mentioned as following.

* Admin Unit
* Student Unit
* Carpooling Unit
* Notes Unit
* Question/Answers Unit

### Integration Testing

Integration testing is performed by combining modules, which then gives the results according to the usage. Each module will be tested with every possibility of interaction with it by the user.

### System Testing

In this type of testing the system is tested with all of its units and the modules that will be interacting with users. This ensures that the whole system is working properly.

### Acceptance Testing

Acceptance testing is done to confirm the requirements fulfilled by the system, and evaluation is done on a purpose to check if it is meeting user’s demand or not.

# Conclusion

## Problems faced and lessons learned

Student facilitation system has been documented after some challenging tasks. We did have concepts that we grabbed throughout the educational time period in University, that helped us to face challenges with strength.

The filtering and selection of features to be added in our application was a challenging task, as we had to provide a full fledge working prototype to facilitate university students. It was a bit hard for us to analyse, filter and prioritize the problems faced by University students in everyday life. That were majorly required to be added in the application.

The end users will not be technical, so considering that we had problems designing an interface that would help our intended users to perform the tasks we wanted them to perform. Moving, forward after formalizing a design to actually go ahead for our application, we had troubles to test and arrive at a design that was suitable for both our needs and the students to use.

Diagram creation was another task that was cumbersome. For that we reviewed the concepts that were given to us in the subject of Introduction to Software Engineering. Also, learnt through tutorials and grabbed information from technical communities in such a way to help us design the efficient flow of application.

All the problems raised during this time period had solutions, and also helped us to revise the important concepts that will be needed in future use. We explored about the techniques that will be applied in developing Students’ facilitation system.

## Project Summary

Student facilitation system is an android application that will be implemented to fulfil high quality modern mobile applications standards. It will be free to access, except the charges that only include if one avails some services like carpooling. The architecture of the application is designed in such a way that it will be reliable, scalable, extensible, and performing, it will be meeting high security and availability standards. It will not be allowing any third party services, as will be limiting the scope to university environment. The idea of this application is feasible in all ways. It will be the most important need of university students waiting for their assistance and finding few services that make their tasks to be performed with efficiency on the same platform. The services provided by this application on a single platform are already working well in market, yet with no such famous application exists that holds all of it.

The result of this project will be an android application, which will be connected to a high functioning server ready to be deployed at a place in its need or demand.

To develop this project with an ease and without facing any major hurdles. We will be following the development model mentioned above and go through every phase by which we will be ensuring an orderly approach to developing this project.

## Future Work

The main aim to develop this application was to help students and provide them an assistant with the use of technology. In future we will add a course add drop feature. Instead of visiting to offices students will be able to request for an add or drop of courses through Students’ facilitation system. It will lessen the burden on students and will save their time

# References

[1] <https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm>

[2]<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml-collaboration-diagram/>

[3] <https://www.draw.io/> for diagrams