

# **Automated Challan System**



**BS Final Year Project Report**

**Prepared by: Faheem Nawaz**

**Supervised by: Mam Ifrah Farrukh Khan**

**Department of Computer Science  
Quaid-i-Azam University Islamabad  
Session: (2014 -2018)**

**In the name of Allah, the Most Merciful, the Most Kind.**



# ACKNOWLEDGEMENT

First of all, I would like to extend my sincere and humble gratitude to **ALMIGHTY ALLAH** whose blessing and guidance has been a real source of all my achievements in my life, and who gave me the ability and knowledge to undertake this project and showed me the right path bestows me with his guidance.

I am especially grateful to my supervisor, **Mam Ifrah Farrukh Khan**, whose guidance made me complete this project. She helped me in improving my development skills with her helpful suggestions.

I would also like to thanks all my **friends** for their help, without which the completion of the project would not have been possible.

**Faheem Nawaz**

# Table of Contents

Automated Challan System.....	0
ACKNOWLEDGEMENT .....	2
Acronyms and abbreviations.....	9
Chapter 1 Software Project Management Plan .....	10
1.1 Introduction .....	10
1.1.1 Project Overview .....	10
1.1.2 Project Deliverables.....	10
1.2 Project Organization.....	10
1.2.1 Software Process Model .....	10
1.2.2 Roles and Responsibilities.....	10
1.2.3 Tools and Techniques .....	10
1.3 Project Management Plan.....	11
1.3.1 Analysis Phase Tasks Description .....	11
1.3.2 Design Phase Tasks Description.....	11
1.4 Gantt Chart .....	12
Chapter 2 Software Requirement Specification.....	14
2.1 Introduction .....	14
2.2 Existing Solution .....	14
2.3 Proposed Solution .....	14
2.4 Purpose .....	14
2.5 Scope .....	15
2.6 Overview .....	15
2.6.1 Overall Description.....	15
2.6.2 Product Perspective .....	15
2.6.3 User Characteristics .....	15
2.6.4 General Constraints .....	16
2.7 Assumptions and Dependencies .....	16
2.8 External Interface Requirements .....	16
2.8.1 User interfaces .....	16

2.8.2 Hardware Specifications .....	16
2.8.3 Software interfaces .....	16
2.8.4 Communication interfaces .....	17
2.9 Software Product Features .....	17
2.10 Software Quality Attributes .....	17
2.10.1 Reliability .....	17
2.10.2 Availability .....	17
2.10.3 Security .....	18
2.10.4 Maintainability.....	18
2.10.5 Portability .....	18
2.11 Database Requirements .....	18
2.12 List of Use Cases.....	18
2.13 Use Case Diagram.....	19
2.14 Usecase Descriptions.....	20
2.14.1 UC1 Login .....	20
2.14.2 UC2 Logout .....	21
2.14.4 UC4 Update Course/Semester/Batch.....	23
2.14.5 UC5 Generate Challan.....	24
2.14.6 UC6 Delete Course/Semester/Batch/Challan .....	25
2.15 Database design view .....	27
2.16 Entity Relation Diagram.....	28
Chapter 3 Software Design Description .....	29
3.1 Design overview.....	29
3.2 System Architecture Design.....	29
3.3 Architecture Diagram.....	30
3.4 Sequence Diagrams .....	31
3.4.1 Login.....	31
3.4.2 Add Course .....	31
3.4.3 Add User Profile.....	32
3.4.4 Select Courses.....	32

3.4.5 Delete Courses .....	33
3.4.6 Save Challan Form .....	33
3.4.7 View paid challans.....	34
3.4.8 Logout.....	34
Chapter 4 Implementation.....	35
4.1 Selected Platform .....	35
4.2 Selected Language.....	35
4.2.1 Java .....	35
4.2.2 Python.....	35
4.3 Libraries .....	35
4.3.1 Retrofit.....	35
4.3.2 Flask Restful.....	35
4.4 User Interfaces.....	36
4.4.1 Login.....	36
4.4.2 Register user interface .....	37
4.4.3 Generate challan .....	38
4.4.4 View challan details.....	39
4.4.5 Student Drawer .....	40
4.4.6 Admin Drawer .....	41
4.4.7 Manage courses/Add course .....	42
4.4.8 Course list.....	43
4.4.9 Manage/Add Semester.....	44
4.4.10 Semester list.....	45
4.4.11 Add Batch .....	46
4.4.12 Batch list.....	47
Chapter 5 Software testing.....	48
5.1 Introduction .....	48
5.2 Black box testing.....	48
5.3 White box testing .....	48
5.4 Test Cases.....	49

5.4.1 Login Test Case .....	49
5.4.2 Generate Challan Test Case.....	49
5.4.3 Register Student Test Case .....	50
5.4.4 Validate Student/Challan.....	50
5.4.5 Add Course/Batch/Semester.....	51
5.4.6 Delete Course/Batch/Semester/Challan.....	51
5.4.7 Update Course/Batch/Semester .....	52
Chapter 6 Future Conclusion .....	53
6.1 Future Work .....	53
References.....	54

# List of Figures

Figure 1: Gantt Chart .....	12
Figure 2 : Gantt Chart .....	12
Figure 3: Gantt Chart .....	13
Figure 4 : Gantt Chart .....	13
Figure 5 : Usecase Diagram.....	19
Figure 6:Database Design View .....	27
Figure 7 : ERD .....	28
Figure 8 : Architecture Diagram .....	30
Figure 9 : Sequence Diagram Login .....	31
Figure 10 : Sequence Diagram Add Courses .....	31
Figure 11 : Sequence Diagram Add User Profile .....	32
Figure 12 : Sequence Diagram Select Courses .....	32
Figure 13 : Sequence Diagram Delete Courses .....	33
Figure 14 : Sequence Diagram Save Challan Form.....	33
Figure 15 : Sequence Diagram View Challan Form .....	34
Figure 16 : Sequence Diagram Logout .....	34
Figure 17: Interface Login .....	36
Figure 18:Interface Registration .....	37
Figure 19:Interface Generate Challan .....	38
Figure 20:Interface View Challan.....	39
Figure 21 : Student drawer.....	40
Figure 22: Admin drawer.....	41
Figure 23 : Add course.....	42
Figure 24 : Course list.....	43
Figure 25 : Add semester .....	44
Figure 26 : Semester list .....	45
Figure 27 : Add batch .....	46
Figure 28 : Batch list.....	47



# List of Tables

Table 1 : Acronyms and abbreviations .....	9
Table 2:Tools and Techniques .....	11
Table 3:Languages and tools .....	16
Table 4:UC1.....	20
Table 5:UC2.....	21
Table 6:UC3.....	22
Table 7 : UC4.....	23
Table 8:UC5.....	24
Table 9:UC6.....	25
Table 10:UC7.....	26
Table 11:TC-001 .....	49
Table 12:TC-002 .....	49
Table 13:TC-003 .....	50
Table 14:TC-004 .....	50
Table 15:TC-005 .....	51
Table 16:TC-006.....	51
Table 17:TC-007.....	52

# Acronyms and abbreviations

Different acronyms and abbreviations are shown in table below.

**Table 1 : Acronyms and abbreviations**

<b>Acronyms</b>	<b>Abbreviation</b>
ACS	Automated Challan System.
QAU	Quaid-I-Azam university.
USER	Admin/Student currently using app.
UC	Use case.

# Chapter 1 Software Project Management Plan

## 1.1 Introduction

This document is a software Project Management Plan about my project i.e. challan automation system.

### 1.1.1 Project Overview

Our Project is an android based challan automation system specifically designed for the students of university. This system will allow students to choose their courses for registration and get fees challan form. Once challan form is generated admin will verify paid challans and student can view all his paid challans.

### 1.1.2 Project Deliverables

1. Software Project Management Plan (SPMP)
2. Software requirement Specifications (SRS)
3. Software Design Description (SDD)
4. Software Test Documentation (STD)

## 1.2 Project Organization

### 1.2.1 Software Process Model

Water fall Model is used for the development of this project. The main purpose for which I am choosing water fall model is that our requirements are clear and all the work is done in a sequence. Another reason to choose water fall model is that all universities are clear about their fee structure.

### 1.2.2 Roles and Responsibilities

I am the only one person who is working on this project. Therefore, all the roles and responsibilities are on my side. I have gathered the entire requirement to complete my Analysis phase. And after the refinement of these requirements I have developed the software requirement specification document.

### 1.2.3 Tools and Techniques

I have used the following tools for the documentation purposes.

Table 2: Tools and Techniques

<b>MS word</b>	For the Document Purpose
<b>Geny Model</b>	Online Tool for UML diagrams
<b>Paint</b>	For picture resizing and cropping
<b>Visio</b>	It is used for to develop the UML diagrams
<b>Project Libre</b>	For the Time Table

## 1.3 Project Management Plan

Tasks are divided into two parts Analysis Phase and Design Phase

### 1.3.1 Analysis Phase Tasks Description

#### Description

Initial requirements have been gathered. Then refinement has been performed. Then develop the use case diagrams. After which the functional and non-functional requirements are extracted.

#### Deliverables and Milestones

1. SPMP
2. SRS

#### Resources Needed

1. MS Office
2. Microsoft Visio
3. Geny Model
4. Paint
5. Internet

### 1.3.2 Design Phase Tasks Description

#### Description

System Sequence Diagram, Architecture Diagram and Class Diagram fall under the design phase.

#### Deliverables and Milestones

1. SDD
2. STD

#### Resources Needed

1. MS Office
2. Geny Model (Online Tool)
3. Paint
4. Internet

## Timetable

All the tasks that have assigned to me I have completed them successfully.

## 1.4 Gantt Chart

Name	Duration	Start	Finish
<input type="checkbox"/> <b>Analysis Phase</b>	57 days?	<b>27/11/17 08:00</b>	<b>13/02/18 17:00</b>
Understanding Problem	1 day?	27/11/17 08:00	27/11/17 17:00
Making SPMP Document	2 days	28/11/17 08:00	29/11/17 17:00
<input type="checkbox"/> <b>Analysis</b>	25 days?	<b>30/11/17 08:00</b>	<b>03/01/18 17:00</b>
Collect requirements	2 days	30/11/17 08:00	01/12/17 17:00
Refine requirements	1 day?	04/12/17 08:00	04/12/17 17:00
Make Document 1	2 days	05/12/17 08:00	06/12/17 17:00
<input type="checkbox"/> <b>Specific requirements</b>	15 days?	<b>07/12/17 08:00</b>	<b>27/12/17 17:00</b>
User interfaces	2 days	07/12/17 08:00	08/12/17 17:00
software interfaces	1 day?	11/12/17 08:00	11/12/17 17:00
Making Document 2	2 days	12/12/17 08:00	13/12/17 17:00
<input type="checkbox"/> <b>Finding Functional Requirments</b>	10 days	<b>14/12/17 08:00</b>	<b>27/12/17 17:00</b>
<input type="checkbox"/> <b>Identify usecases</b>	5 days	<b>14/12/17 08:00</b>	<b>20/12/17 17:00</b>
Refine usecases	2 days	14/12/17 08:00	15/12/17 17:00
Define usecase Text	2 days	18/12/17 08:00	19/12/17 17:00
Draw Usecase Diagram	1 day	20/12/17 08:00	20/12/17 17:00
Identify System Functions	2 days	21/12/17 08:00	22/12/17 17:00
Refine Functional Requirements	1 day	25/12/17 08:00	25/12/17 17:00
Make Document 3	2 days	26/12/17 08:00	27/12/17 17:00
<input type="checkbox"/> <b>Finding Non Functional Requirments</b>	1 day	<b>07/12/17 08:00</b>	<b>07/12/17 17:00</b>
Identify constraints	1 day	07/12/17 08:00	07/12/17 17:00
Making srs document	5 days	28/12/17 08:00	03/01/18 17:00

Figure 1: Gantt Chart

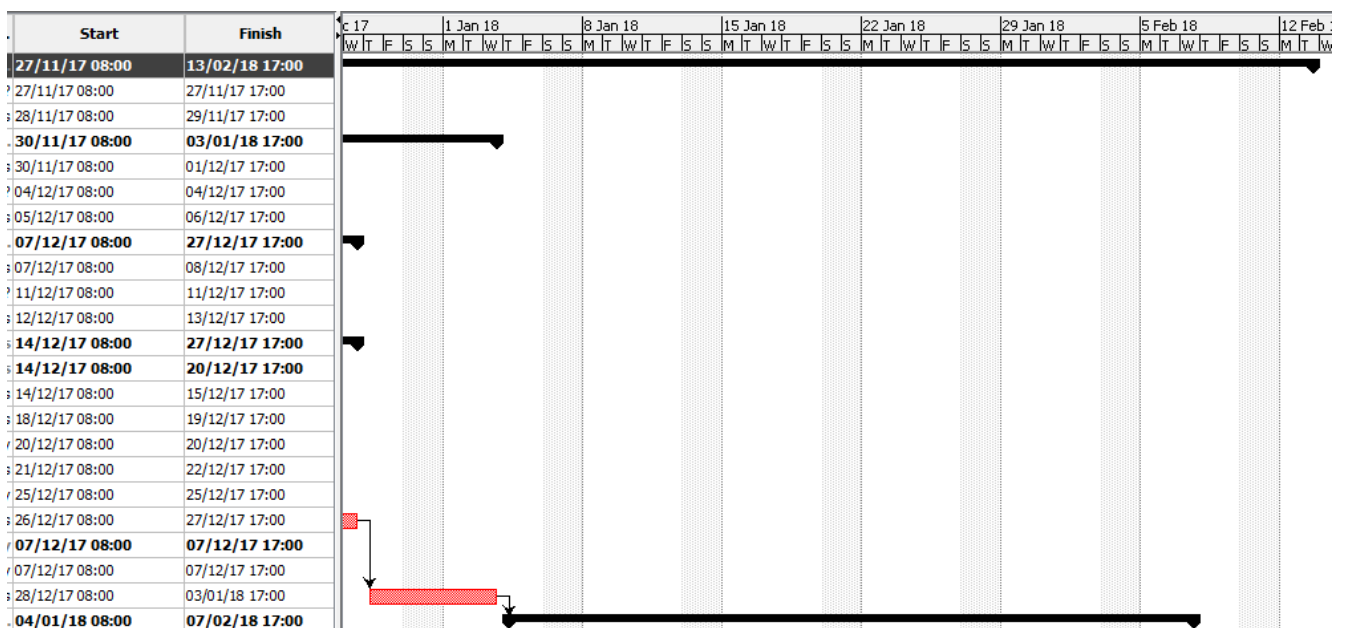


Figure 2 : Gantt Chart

<b>Design</b>	<b>25 days?</b>	<b>04/01/18 08:00</b>	<b>07/02/18 17:00</b>
Develop interface design	5 days	04/01/18 08:00	10/01/18 17:00
Refine interface design	2 days	11/01/18 08:00	12/01/18 17:00
Study retrofit	5 days	15/01/18 08:00	19/01/18 17:00
Study webservices	5 days	22/01/18 08:00	26/01/18 17:00
Study python	1 day?	04/01/18 08:00	04/01/18 17:00
Define pseudo code	5 days	29/01/18 08:00	02/02/18 17:00
Review pseudo code	3 days	05/02/18 08:00	07/02/18 17:00
<b>Validate design</b>	<b>2 days</b>	<b>08/02/18 08:00</b>	<b>09/02/18 17:00</b>
Evaluate design	2 days	08/02/18 08:00	09/02/18 17:00
Make final Document	2 days	12/02/18 08:00	13/02/18 17:00
<b>Coding</b>	<b>15 days?</b>	<b>14/02/18 08:00</b>	<b>06/03/18 17:00</b>
Code android interfaces	1 day?	14/02/18 08:00	14/02/18 17:00
Code webservice	1 day?	14/02/18 08:00	14/02/18 17:00
Integrate android with webservice	10 days	14/02/18 08:00	27/02/18 17:00
Implement all functions	15 days	14/02/18 08:00	06/03/18 17:00
Refine implementation	2 days	14/02/18 08:00	15/02/18 17:00
<b>Testing</b>	<b>4 days</b>	<b>16/02/18 08:00</b>	<b>21/02/18 17:00</b>
Verification	2 days	16/02/18 08:00	19/02/18 17:00
Validation	2 days	20/02/18 08:00	21/02/18 17:00

Figure 3: Gantt Chart

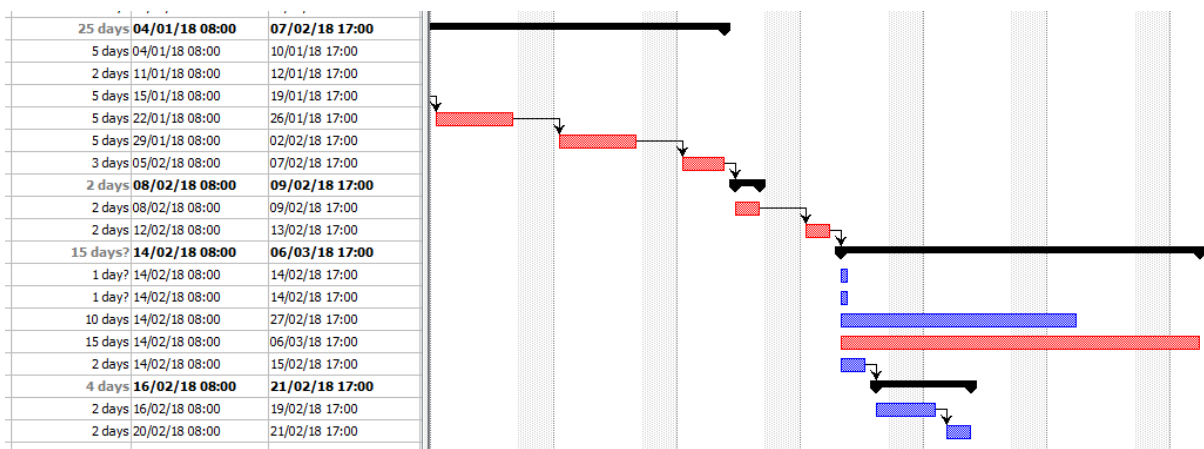


Figure 4 : Gantt Chart

# Chapter 2 Software Requirement Specification

## 2.1 Introduction

This chapter aims at defining the overall Software Requirements for “Challan Automation System “. Efforts have been made to define the requirements accurately. The final product will be having only features mentioned in this document.

## 2.2 Existing Solution

There are different challan automation system running in different universities and organizations as well. But as I am making system for university students so mostly there are two different types of challan systems working in universities. For example, in Quaid-e-Azam university there is semester-based system so the challan form will come according to that. In simple words you can say that in this university a student has to pay according to the fees which have been selected in the prospectus for that specific degree and specific semester, it doesn't matter how many credit hours you are studying.

Another type of challan system is mostly used in different universities. For example, in Allama Iqbal open university there is credit hour-based challan form. They have a fixed amount for 3 credit hours which is Rs. 1930 and their challan form is calculated according to the credit hours the student had taken.

## 2.3 Proposed Solution

The solution which I had proposed is that my system provides both functionalities i.e. semester-based challan form and credit hour-based challan form. So, in this way my system becomes generic and it can be used in every university whether it is following semester-based system or credit hour-based system.

## 2.4 Purpose

The purpose of this SRS document is to provide a detailed overview of software product, its parameters and goals. It will illustrate the purpose and complete declaration for the development of system. This document describes the project's intended audience, all the functional as well as non-functional requirements. It also defines how the client, developer and audience see the product and its functionality.

## 2.5 Scope

The Project is android based challan automation system specifically designed for the students of university. This system provides both semesters based and credit hour-based challan form which will help universities to automate their challan according to the course registration process which they were following.

This system will perform the following operations

- Students can register themselves and then login to get their Challan form.
- Students can view their paid challans.
- Students can select specific courses while generating challan.
- The system will calculate fee and provide automated challan form.
- Admin can add/remove courses, batches and semesters.
- Admin can validate students and paid challans.

## 2.6 Overview

The remaining sections of this chapter provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section2 of this document. Section 3 gives the functional requirements, data requirements and constraints and assumptions made while designing the software. Section 3 also gives the specific requirements of the product. Section 3 also discusses the external interface requirements and gives detailed description of functional requirements.

### 2.6.1 Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

### 2.6.2 Product Perspective

The Challan Automation system is basically a generic challan automation system which helps students to get their challan form according to system running in their universities i.e. Semester based or Credit hour Based. The student has the android platform for the selection of its courses to get its challan form and the data is stored in MySQL database.

### 2.6.3 User Characteristics

There are two types of users involved: Student and University Admin.



1. Users must have technical expertise to use smartphone.
2. Users may have to be trained for using the application.

### 2.6.4 General Constraints

The constraints are mentioned below:

1. Application must be installed in the phone.
2. Only registered users can use the application.
3. The system must be user-friendly.

## 2.7 Assumptions and Dependencies

1. System is dependent on the challan form type which they were using (semester based or credit hour based).
2. The users should know the English language, as the interface will be provided in English.

## 2.8 External Interface Requirements

### 2.8.1 User interfaces

Challan automation system should be designed for ease of use, providing help instructions, and appropriate error messages for user inputs. The Challan automation system makes sure at every point, that the user spends most of the time using the application rather than figuring out how to use it. The background will be light in color with dark colored font to enhance the contrast and visibility. Controls which allow the user to interact with the application will be clear and imply their functionality within the application.

The home screen offers a menu with a list of functions that the application performs. The user can select one of the options on the menu, and is taken to the respective screen. Every screen displays the menu on the top. The user can click on any one of the options and is taken to the screen of their choice.

### 2.8.2 Hardware Specifications

This application will work on android phones and tablets. Since the application must run connected to internet, Android devices must have internet connection to run this application.

### 2.8.3 Software interfaces

Table 3: Languages and tools

Languages	Tools
Java	Android Studio
XML	Android Studio

MySQL	Xampp Server
Python3.6 (backend development)	Spyder

## Web Service for Connectivity

Restful Webservice (API in python) [1].

### 2.8.4 Communication interfaces

All the interaction between users and the system will be performed over the internet, so Challan automation system will use the HTTP protocol.

## 2.9 Software Product Features

Summary of major functions that the system should perform are listed below.

**Register account:** This function will allow students to get registered.

**Login account:** This function will allow university admin and student to login account.

**Select Courses:** This function will allow student to select their specific courses.

**Generate Challan:** This function will generate the Challan form for the students.

**Validate:** System provide interface to validate paid challans and valid students.

**Manage:** Admin can add/remove semesters, courses, batches.

## 2.10 Software Quality Attributes

Software system attributes define overall factors that affect run-time behavior, application design, and user experience. To develop high quality application, software system attributes are the benchmarks that describe system's intended behavior within the environment for which it was built. Here is detail of some software system attributes.

### 2.10.1 Reliability

1. System must produce correct and consistent results.
2. System must provide Challan form after the selection of courses.
3. System must calculate fees correctly.

### 2.10.2 Availability

The system will be available all the time but it requires internet connection.

### **2.10.3 Security**

The security section describes the need to control access to the data. This includes controlling who may view and alter application data.

1. Only registered students and university admins can use this application.
2. A student who uses this application should have a login id and password.
3. Any modification (add, delete, update) for the database will be synchronized and done only by the administrator.
4. Information about challan form will be (add, update and delete) by students and administrator.

### **2.10.4 Maintainability**

1. The program will use modular approach so that it would be easier to update or change code when needed.
2. Updates in the software will be done in some separate files so that actual product will not be disturbed.

### **2.10.5 Portability**

The system can be easily modified for a new environment. This software can be installed in any university, if modified properly. There is no portability for different OS.

## **2.11 Database Requirements**

MySQL database will be used for this system to store all information. MySQL is a free, open-source database management system (DBMS for short). A DBMS is a system that manages databases and connects them to software.

Some of the reasons to use MySQL are

1. Ubiquitous
2. It's easy to find help
3. It's easy to learn
4. Scalable

## **2.12 List of Use Cases**

1. Register
2. Login
3. Logout
4. Add Course
5. Update Course
6. Delete Course

7. Add Semester
8. Update Semester
9. Delete Semester
10. Add Batch
11. Update Batch
12. Delete Batch
13. Delete Challan
14. Verify Student
15. Verify Challan
16. Generate Challan
17. View Challan form

## 2.13 Use Case Diagram

Use case diagram is graphical representation of user interaction with system. Use case diagram represents that how users interact with the components of system. So, all use cases are mentioned in diagram and how the users interact with use cases is shown by diagram.

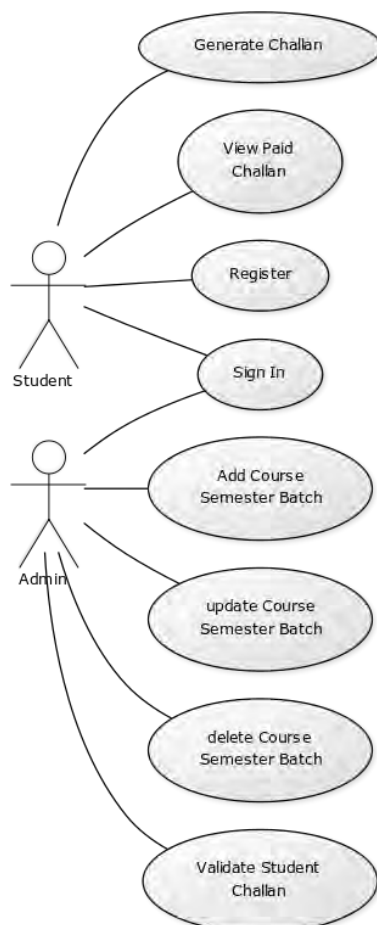


Figure 5 : Usecase Diagram

## 2.14 Usecase Descriptions

### 2.14.1 UC1 Login

Table 4:UC1

<b>ID</b>	<b>UC1</b>
<b>Name</b>	Login
<b>Primary Actor</b>	Student and Admin.
<b>Stockholder's and interests:</b>	<b>Admin:</b> wants to manage students, semesters, batches and courses <b>Student:</b> To generate new fee challan and to view his paid challan history.
<b>Input</b>	Email and Password.
<b>Pre-conditions</b>	User should already register.
<b>Post-conditions</b>	<ol style="list-style-type: none"><li>1. The user is logged in to the system.</li><li>2. The user has access to use the functionality of the system.</li></ol>
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. User opens the app.</li><li>2. System shows the login screen and ask for the credentials.</li><li>3. User enter his/her username and password.</li><li>4. The system authenticates the username and password with the Database.</li><li>5. The user gains access to the app functionality.</li></ol>
<b>Alternatives:</b>	<ol style="list-style-type: none"><li>1. App crash user.</li><li>2. Login screen doesn't ask for credentials.</li><li>5.a Credentials are not correct.</li><li>5.b Student is not verified by the admin.</li></ol>
<b>Frequency of occurrence:</b>	Every time user wants to Login.

## 2.14.2 UC2 Logout

Table 5:UC2

<b>ID</b>	<b>UC2</b>
<b>Name</b>	Logout
<b>Primary Actor</b>	Student and Admin.
<b>Stockholder's and interests:</b>	<b>Admin:</b> wants to manage students, semesters, batches and courses <b>Student:</b> To generate new fee challan and to view his paid challan history.
<b>Input</b>	N/A
<b>Pre-conditions</b>	User should already log in.
<b>Post-conditions</b>	The user is logged out of the system.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. The user is done using the system and he clicks the logout button.</li><li>2. The system logs the user out and clear shared preferences file.</li><li>3. The system redirects to the default screen.</li></ol>
<b>Alternatives:</b>	2 a. File is not clear properly.
<b>Frequency of occurrence:</b>	Every time user wants to Logout.

### 2.14.3 UC3 Add Course/Semester/Batch

Table 6:UC3

<b>ID</b>	<b>UC3</b>
<b>Name</b>	<b>Add Course/Semester/Batch</b>
<b>Primary Actor</b>	Admin.
<b>Stockholder's and interests:</b>	<b>Admin:</b> wants to manage students, semesters, batches and courses <b>Student:</b> To generate new fee challan and to view his paid challan history.
<b>Input</b>	All details of Course/Semesters/Batch.
<b>Pre-conditions</b>	User should already log in.
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. System stores a new record to the database.</li> <li>2. Course/Semester/Batch is available for display.</li> </ol>
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user navigate to add course/semester/batch.</li> <li>2. The system shows a form asking for the necessary information.</li> <li>3. The user fill form.</li> <li>4. The User clicks on add button to add the information to the database.</li> <li>5. The system shows a message that the record is saved successfully.</li> </ol>
<b>Alternatives:</b>	<ol style="list-style-type: none"> <li>3. The User enters doesn't fill form properly. The System shows a message that fill complete form.</li> </ol>
<b>Frequency of occurrence:</b>	Every time user wants to Logout.

## 2.14.4 UC4 Update Course/Semester/Batch

Table 7 : UC4

<b>ID</b>	UC4
<b>Name</b>	<b>Update Course/Semester/Batch</b>
<b>Primary Actor</b>	Admin.
<b>Stockholder's and interests:</b>	<b>Admin:</b> wants to manage students, semesters, batches and courses <b>Student:</b> To generate new fee challan and to view his paid challan history.
<b>Input</b>	All the updated details of Course/Semesters/Batch.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User should already log in.</li> <li>2. Course/Semester/Batch is already in database.</li> </ol>
<b>Post-conditions</b>	<ol style="list-style-type: none"> <li>1. Course/Semester/Batch id updated.</li> </ol>
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user navigate to Courses/Semesters/Batches list.</li> <li>2. The system will show available Courses/Semesters/Batches.</li> <li>3. The user click on update button.</li> <li>4. The form with the previous inform will be populated user change information according to his need and press done button.</li> <li>5. The system shows a message that the record is updated successfully.</li> </ol>
<b>Alternatives:</b>	<ol style="list-style-type: none"> <li>3 a) System doesn't show Courses/Semesters/Batches list.</li> <li>4 a) User doesn't fill complete form.</li> </ol>
<b>Frequency of occurrence:</b>	Every time user wants to Update Course/Semester/Batch.



## 2.14.5 UC5 Generate Challan

Table 8:UC5

<b>ID</b>	<b>UC5</b>
<b>Name</b>	Generate Challan
<b>Primary Actor</b>	Student.
<b>Stockholder's and interests:</b>	<b>Admin:</b> wants to manage students, semesters, batches and courses <b>Student:</b> To generate new fee challan and to view his paid challan history.
<b>Input</b>	List of courses (student want to register).
<b>Pre-conditions</b>	Student is login.
<b>Post-conditions</b>	Challan is generated and details of challan are displayed.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user navigate to generate challan interface.</li> <li>2. The system shows a list of all available courses of that batch.</li> <li>3. The user select all his desired courses.</li> <li>4. User click generate challan button.</li> <li>5. The System Shows a detail of generated challan.</li> </ol>
<b>Alternatives:</b>	<ol style="list-style-type: none"> <li>2 a) Desired course does not exist.</li> <li>3 a) User select less courses than limit.</li> <li>3 b) User select more courses than upper limit.</li> </ol>
<b>Frequency of occurrence:</b>	Every time user wants to generate challan.

## 2.14.6 UC6 Delete Course/Semester/Batch/Challan

Table 9:UC6

<b>ID</b>	<b>UC6</b>
<b>Name</b>	<b>Delete Course/Semester/Batch/Challan</b>
<b>Primary Actor</b>	Admin.
<b>Stockholder's and interests:</b>	<b>Admin:</b> wants to manage students, semesters, batches and courses <b>Student:</b> To generate new fee challan and to view his paid challan history.
<b>Input</b>	Id of the specific Course/Semesters/Batch/Challan.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User should already log in.</li> <li>2. Course/Semester/Batch/Challan is already in database and displayed to user.</li> </ol>
<b>Post-conditions</b>	Course/Semester/Batch/Challan is deleted and disappeared from screen.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user navigate to Course/Semester/Batch/Challan list.</li> <li>2. The System shows the list of Courses/Semesters/Batches with update and delete button.</li> <li>3. The User clicks delete button.</li> <li>4. The system shows a message that if he really wants to delete the course.</li> <li>5. User clicks on confirm button.</li> <li>6. System shows message that the record is successfully deleted.</li> </ol>
<b>Alternatives:</b>	3 a) The User search for a course that does not exist. System shows a message that nothing is found.
<b>Frequency of occurrence:</b>	Every time user wants to delete Course/Semester/Batch/Challan.

## 2.14.7 UC7 Verify Challan/Student

Table 10:UC7

<b>ID</b>	<b>UC7</b>
<b>Name</b>	<b>Verify Challan/Student</b>
<b>Primary Actor</b>	Admin.
<b>Stockholder's and interests:</b>	<b>Admin:</b> wants to manage students, semesters, batches and courses <b>Student:</b> To generate new fee challan and to view his paid challan history.
<b>Input</b>	Id of the specific Student/Challan.
<b>Pre-conditions</b>	<ol style="list-style-type: none"> <li>1. User should already log in.</li> <li>2. Student/Challan record is already in database and displayed to user.</li> </ol>
<b>Post-conditions</b>	Student/Challan is verified and disappeared from screen.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user navigates to validate Challan/Student screen.</li> <li>2. The System shows all the new/unverified Challans/Students/</li> <li>3. The User clicks the verify button.</li> <li>4. System verify that Challan/Student and remove it from list.</li> </ol>
<b>Alternatives:</b>	2. a) The system shows no Challan/Student.
<b>Frequency of occurrence:</b>	Every time user wants to verify Student/Challan.

## 2.15 Database design view

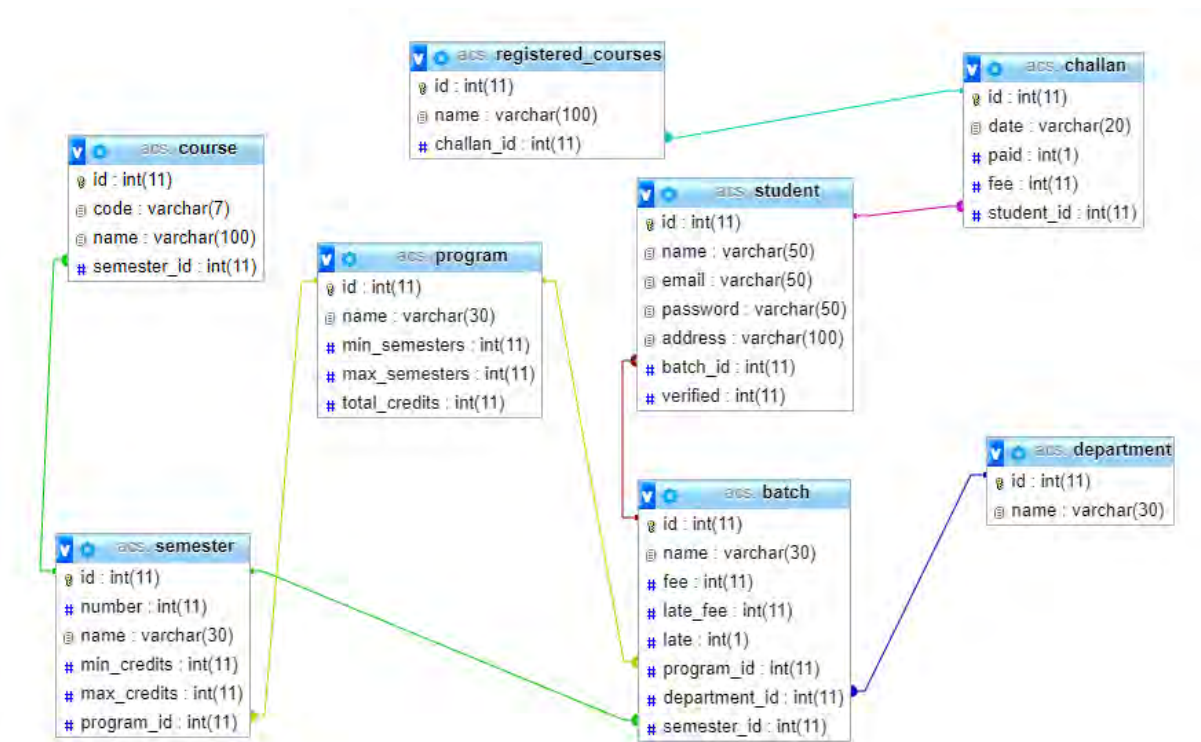


Figure 6: Database Design View

## 2.16 Entity Relation Diagram

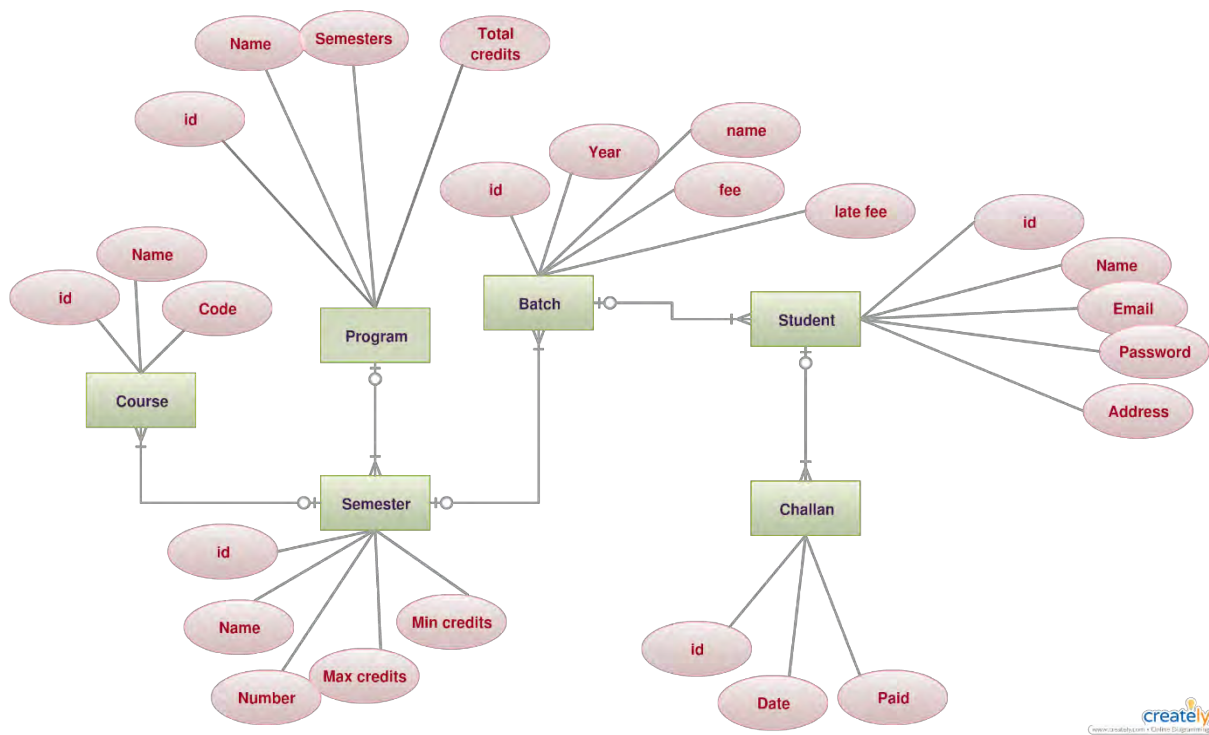


Figure 7 : ERD

# Chapter 3 Software Design

## Description

This chapter reviews the design description for the system of ACS. In this chapter introduction of system design description, system architecture design, detailed description of components and user interfaces are described in detail. Sequence diagrams are also designed in this chapter.

### 3.1 Design overview

System design describes the system at architecture level, services of system, data management of system and complete structure of system. In design phase of system, user interfaces or screen images of system are designed. In addition, system sequence diagram that is an interaction diagram, which shows sequence of interaction with system and user, State chart diagram of system is designed that describes multiple states of system.

### 3.2 System Architecture Design

Architectural design is a creative process where we design a system organization that will satisfy the functional and non-functional requirements of a system. System architecture design is the set of significant decisions about how multiple components of a software system interact with each other, the selection of the structural elements and their interfaces by which the system is composed. System architecture describes organization, styles, patterns, responsibilities, collaborations, connections, and motivations of a system and major subsystems.

### 3.3 Architecture Diagram

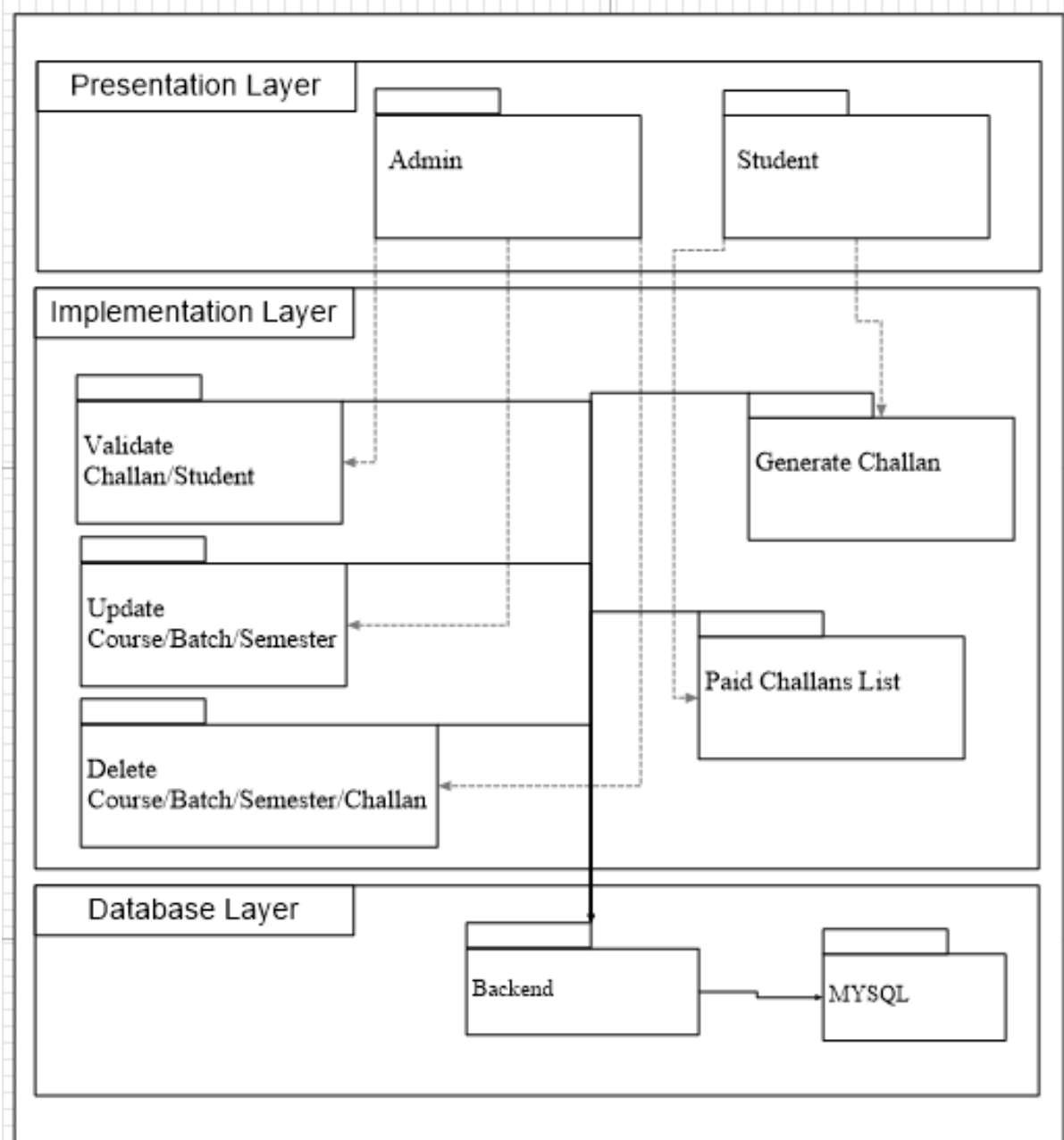


Figure 8 : Architecture Diagram

## 3.4 Sequence Diagrams

### 3.4.1 Login

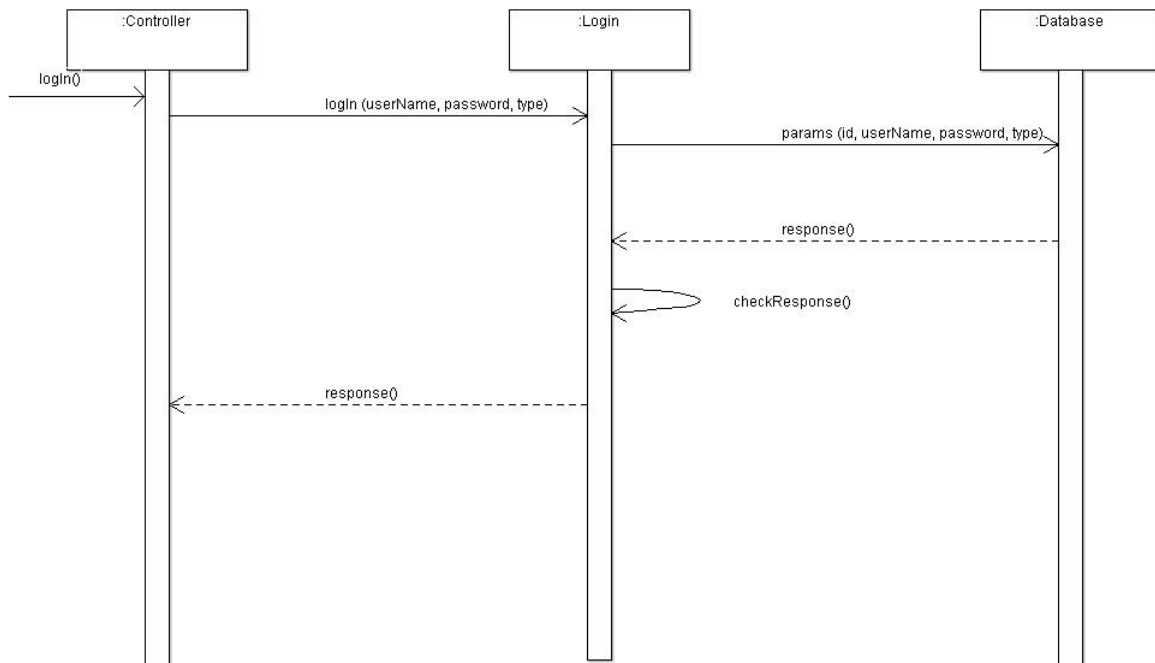


Figure 9 : Sequence Diagram Login

### 3.4.2 Add Course

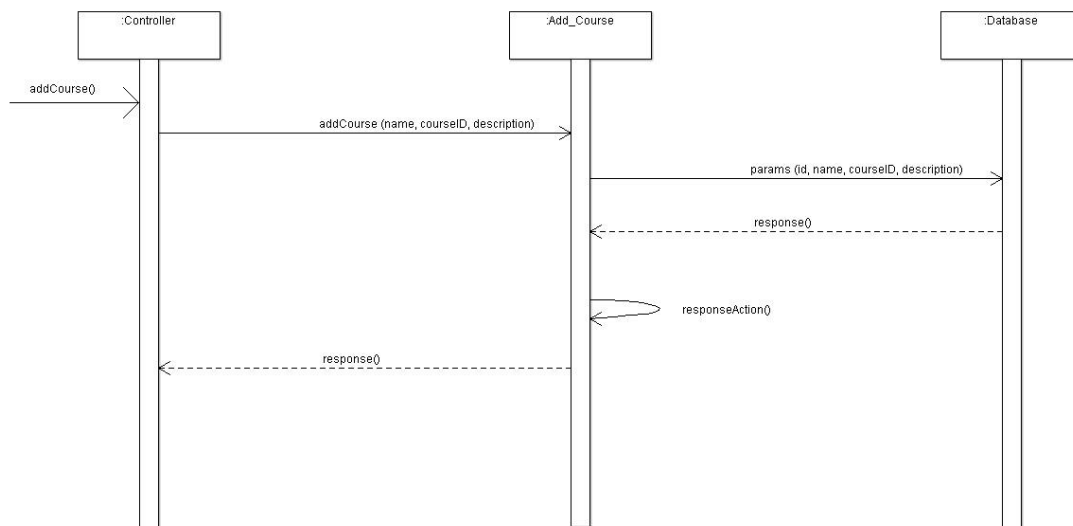


Figure 10 : Sequence Diagram Add Courses



### 3.4.3 Add User Profile

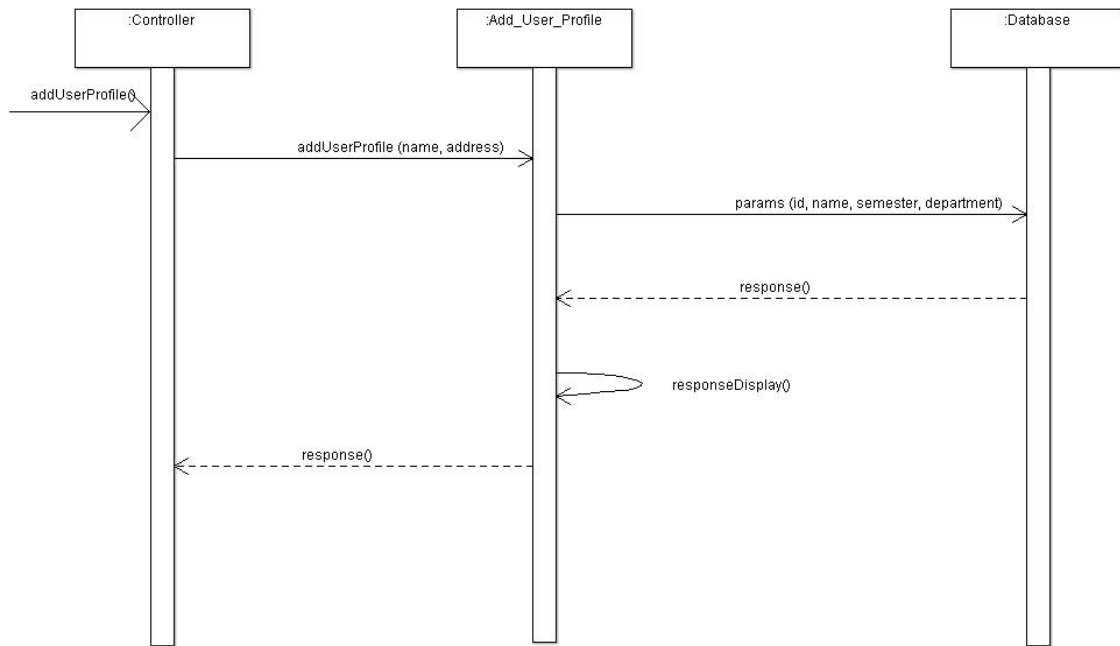


Figure 11 : Sequence Diagram Add User Profile

### 3.4.4 Select Courses

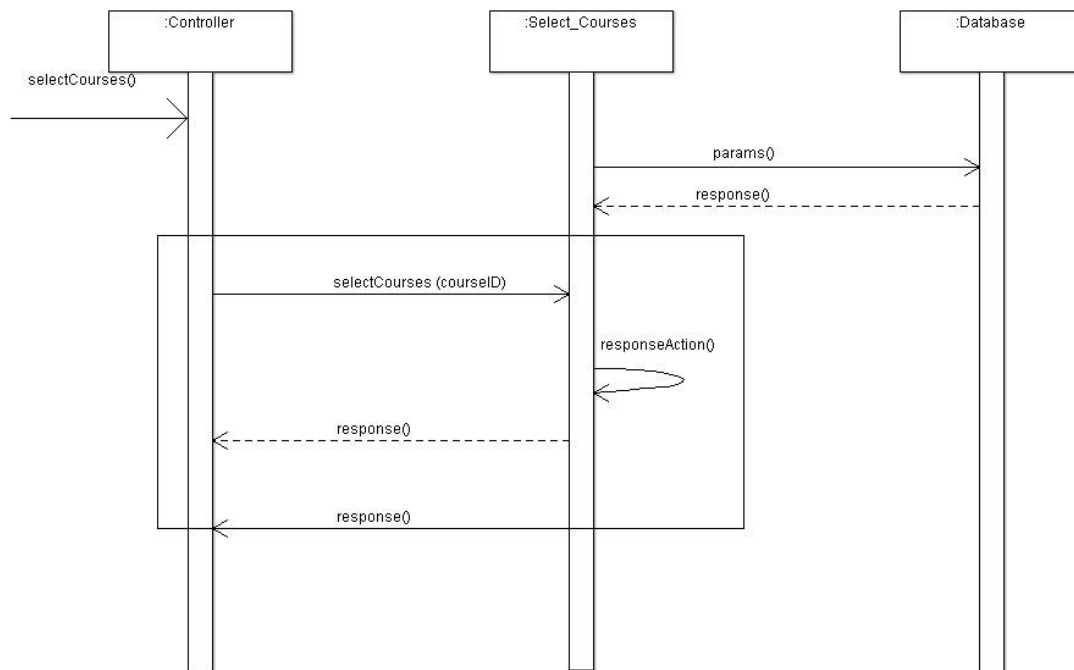


Figure 12 : Sequence Diagram Select Courses

### 3.4.5 Delete Courses

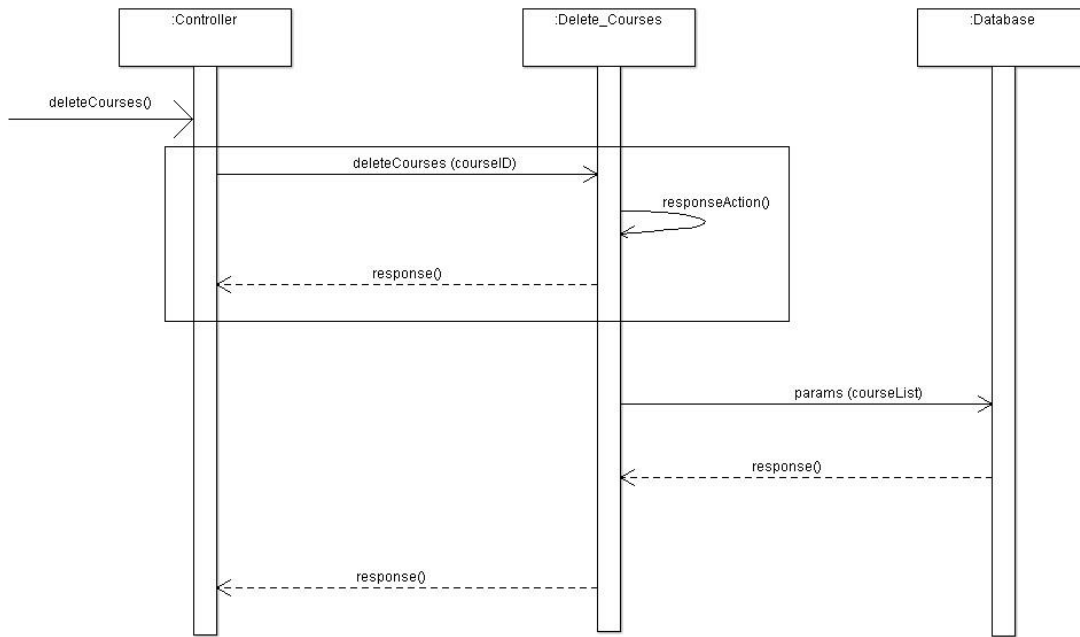


Figure 13 : Sequence Diagram Delete Courses

### 3.4.6 Save Challan Form

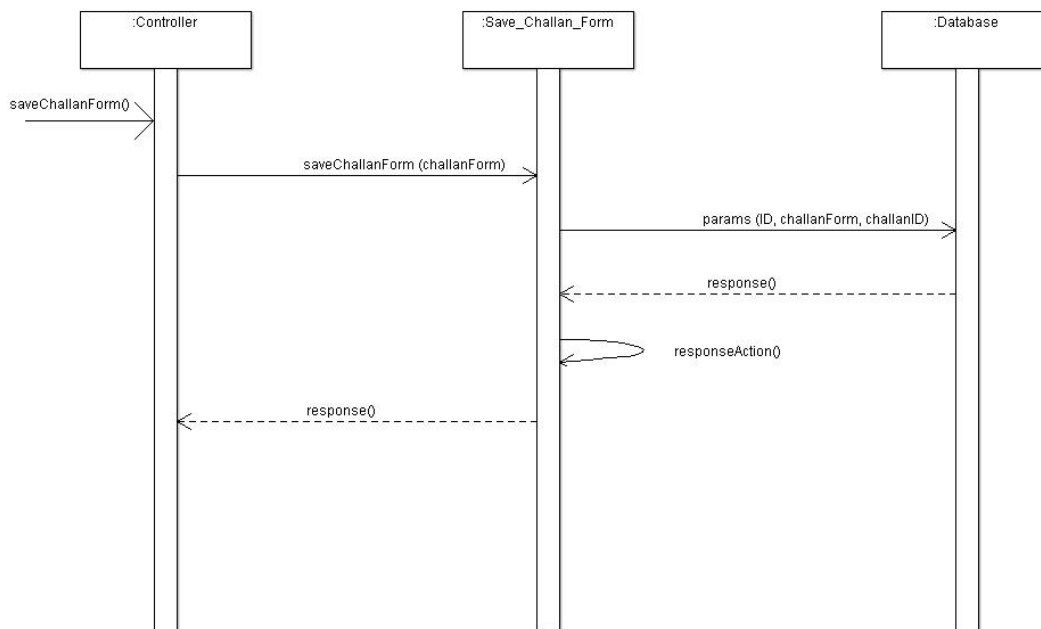


Figure 14 : Sequence Diagram Save Challan Form

### 3.4.7 View paid challans

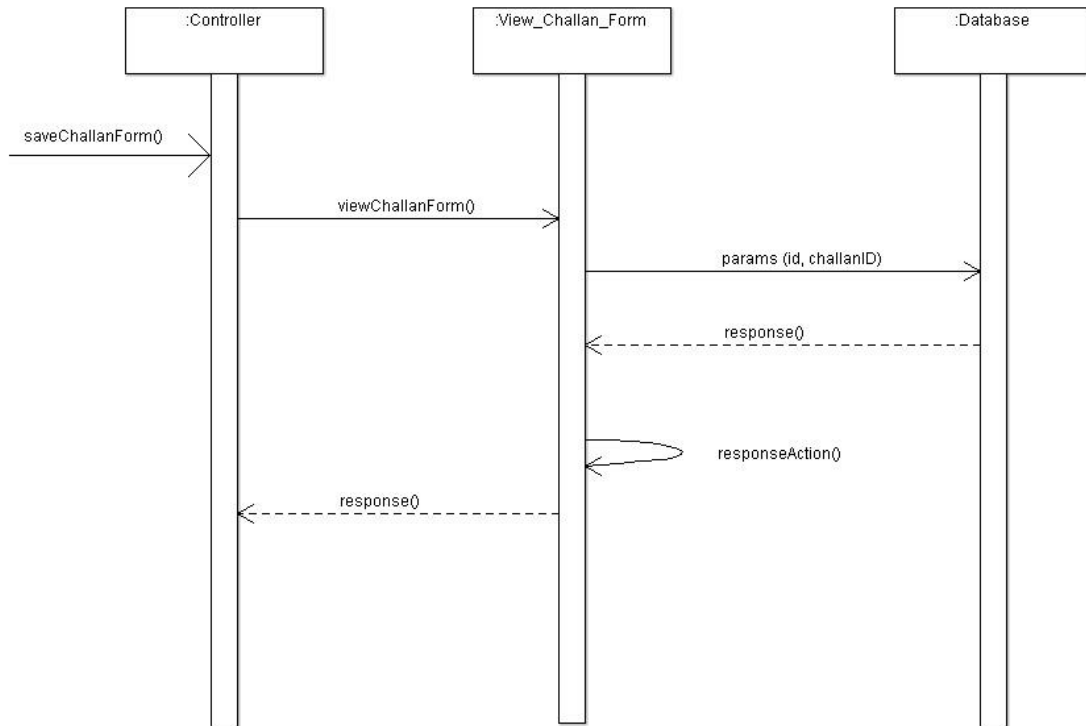


Figure 15 : Sequence Diagram View Challan Form

### 3.4.8 Logout

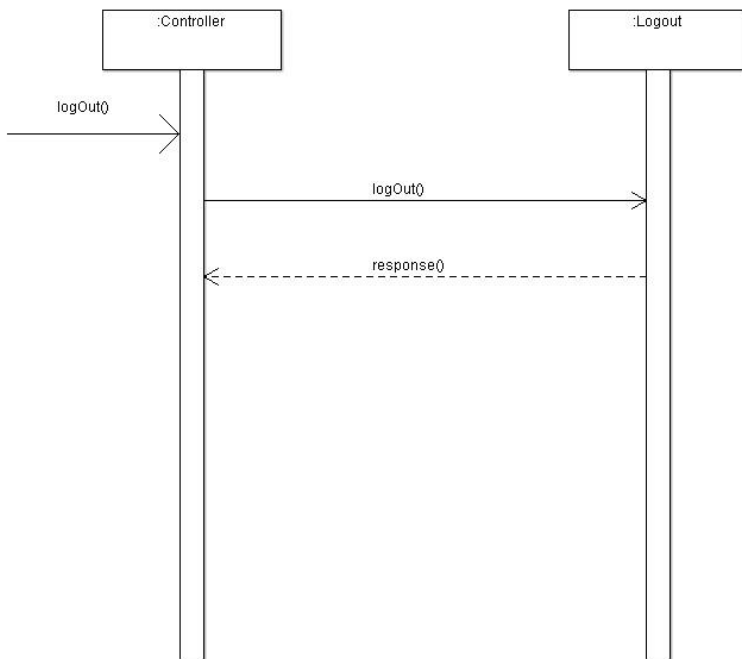


Figure 16 : Sequence Diagram Logout

# Chapter 4 Implementation

This chapter is about the implementation of system. In this chapter selected platform, language and api's for the implementation of this project are described in detail.

## 4.1 Selected Platform

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

## 4.2 Selected Language

### 4.2.1 Java

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible [2].

### 4.2.2 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales [3].

## 4.3 Libraries

### 4.3.1 Retrofit

Retrofit is a type-safe HTTP client for Android and Java – developed by Square (Dagger, OKHTTP) [4].

### 4.3.2 Flask Restful

Flask-RESTful is an extension for Flask that adds support for quickly building REST APIs. It is a lightweight abstraction that works with your existing ORM/libraries. Flask-RESTful encourages best practices with minimal setup. If you are familiar with Flask, Flask-RESTful should be easy to pick up [5].

## 4.4 User Interfaces

### 4.4.1 Login

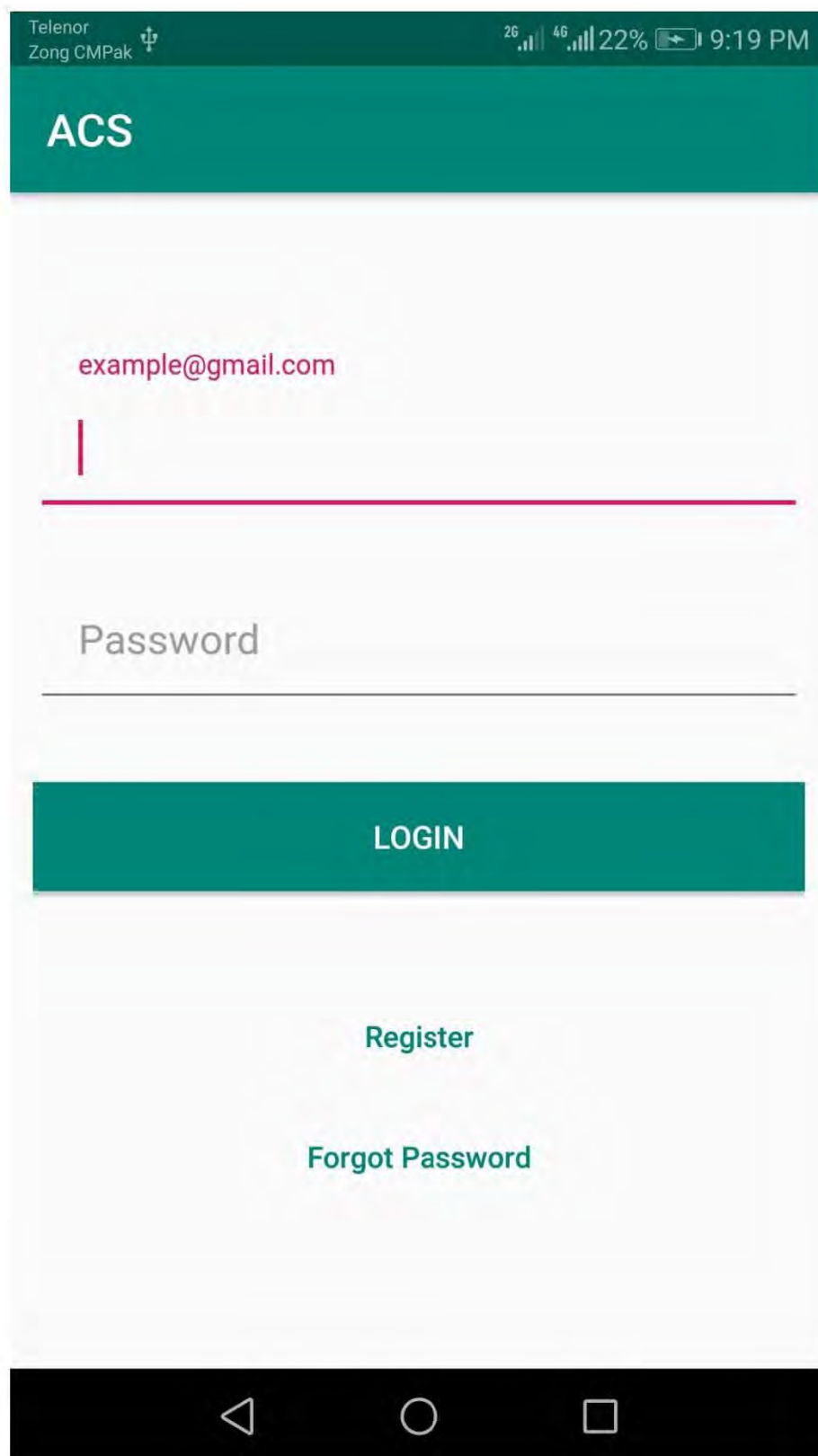



Figure 17: Interface Login

## 4.4.2 Register user interface

Telenor  
Zong CMPak

900 B/s 2G 4G 26% 9:37 PM

# ACS

 BS2015

Name

---

example@gmail.com

---

password

---

Address

---

**REGISTER**

[Login](#)

Figure 18:Interface Registration

### 4.4.3 Generate challan

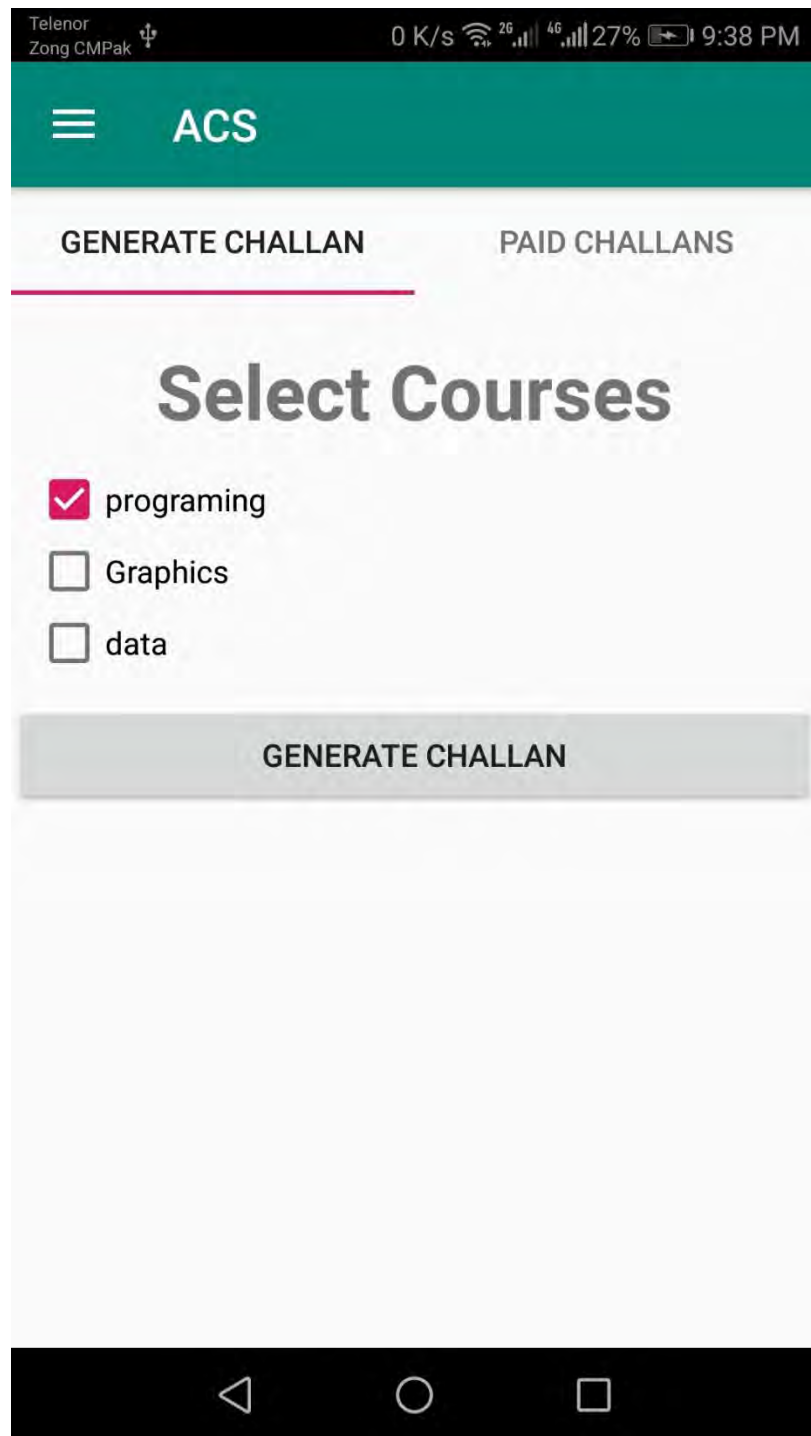


Figure 19: Interface Generate Challan

#### 4.4.4 View challan details

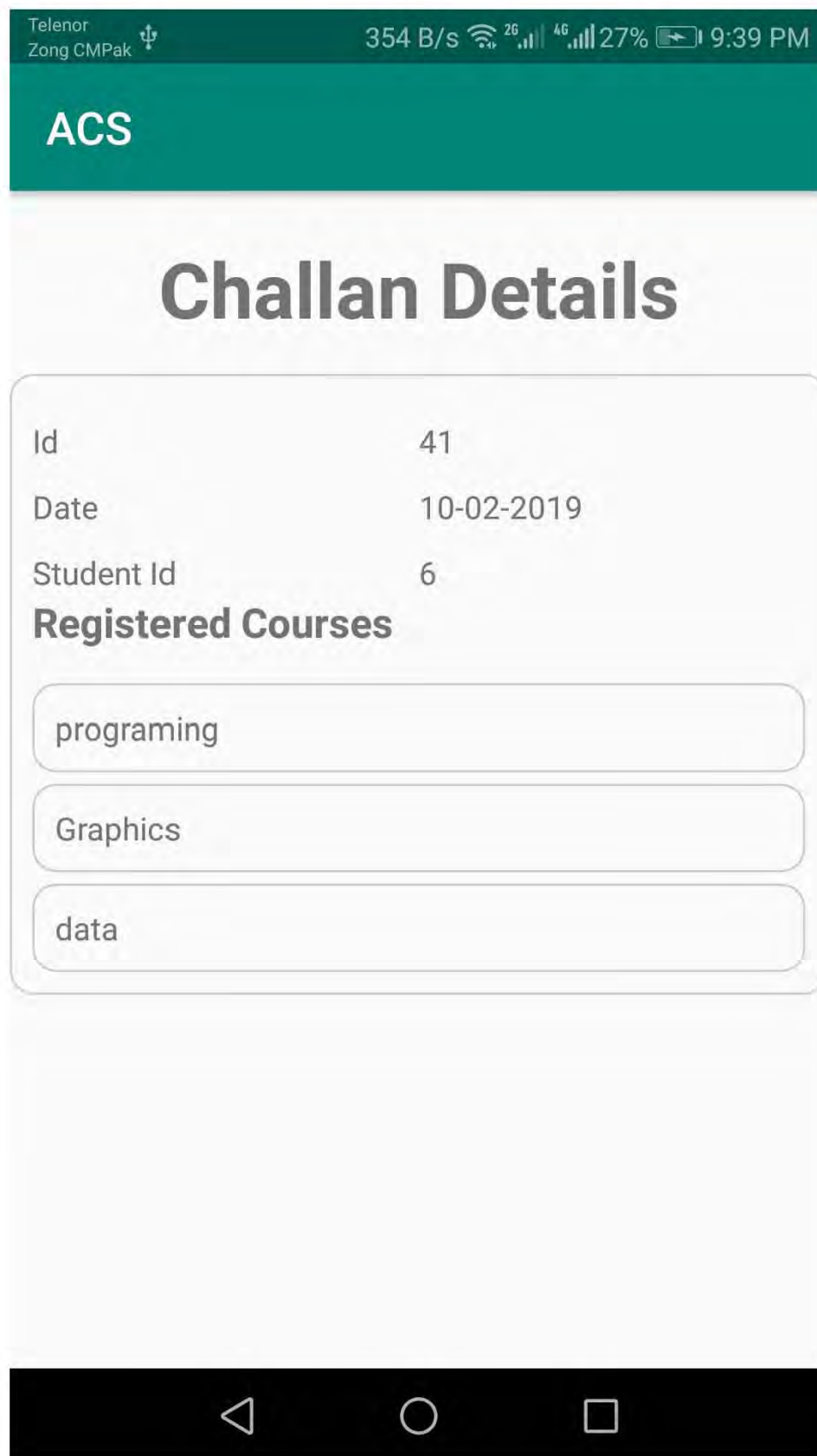


Figure 20:Interface View Challan



#### 4.4.5 Student Drawer

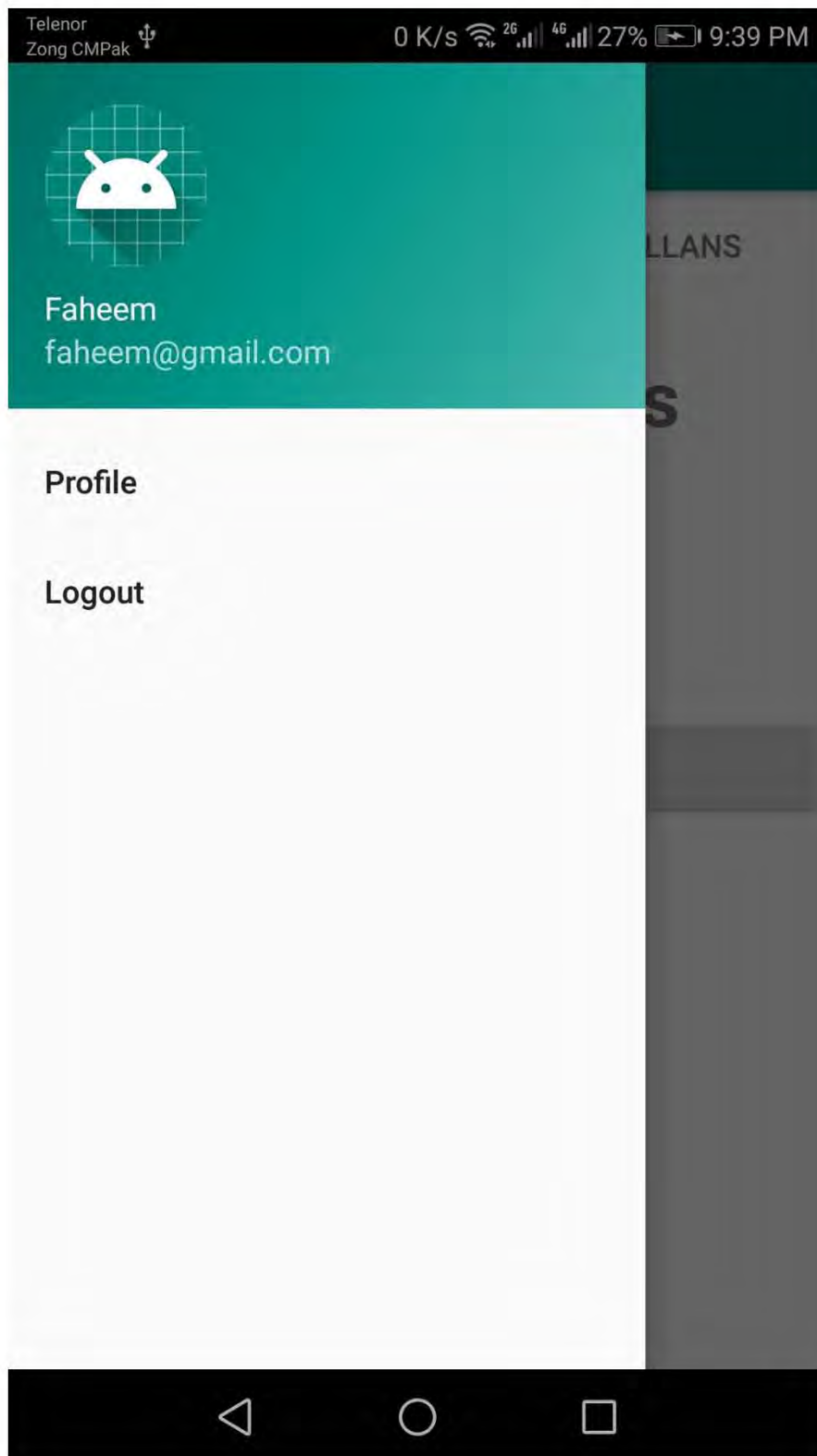


Figure 21 : Student drawer

## 4.4.6 Admin Drawer

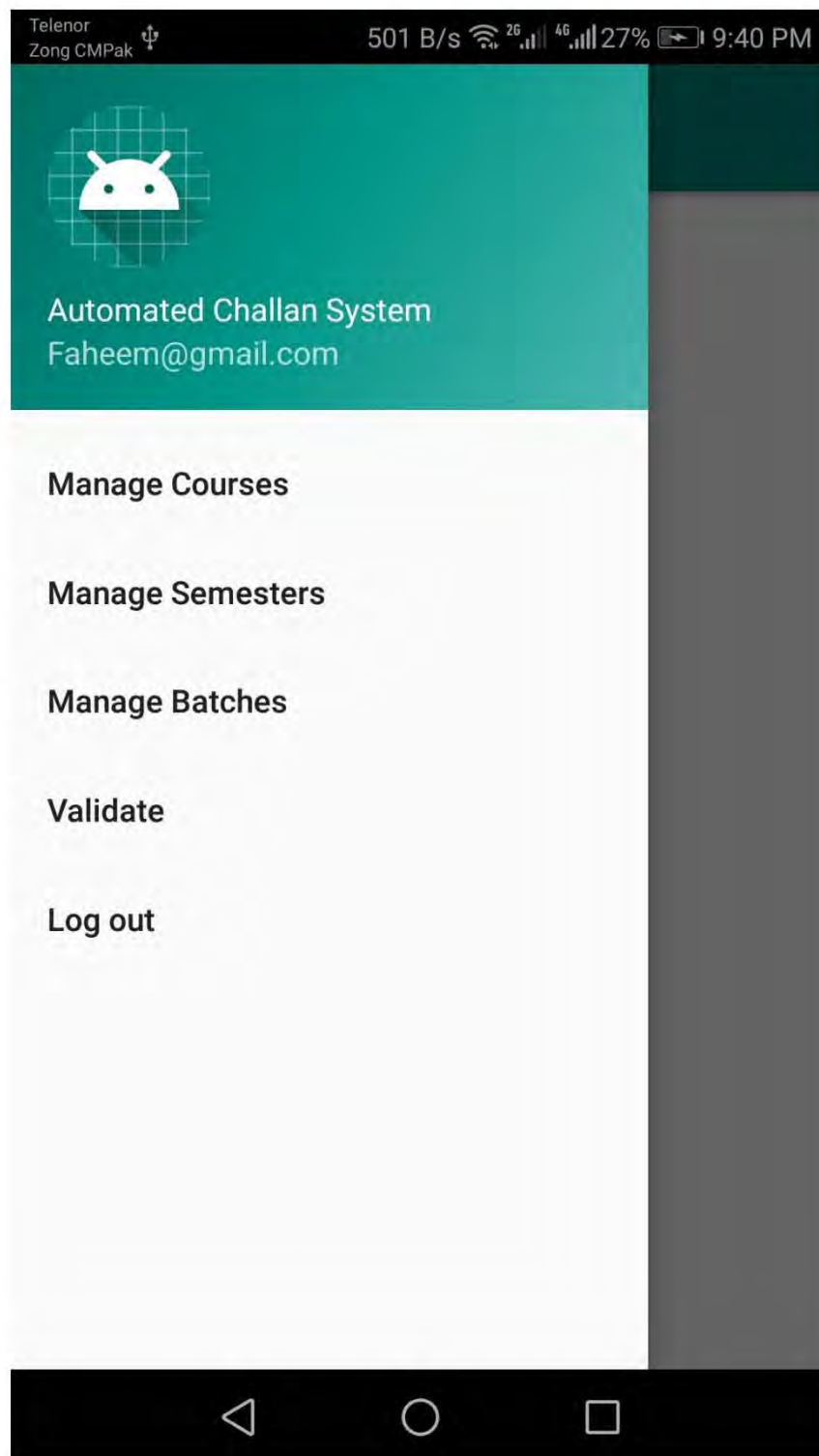


Figure 22: Admin drawer

#### 4.4.7 Manage courses/Add course

Telenor  
Zong CMPak

1.3 K/s 2G 4G 27% 9:40 PM

← ACS

ADD COURSE COURSE LIST

Semester  
BS-1st(fall-2014)

Course code  
cs-101

Name  
Introduction to Computing

ADD

Figure 23 : Add course

#### 4.4.8 Course list

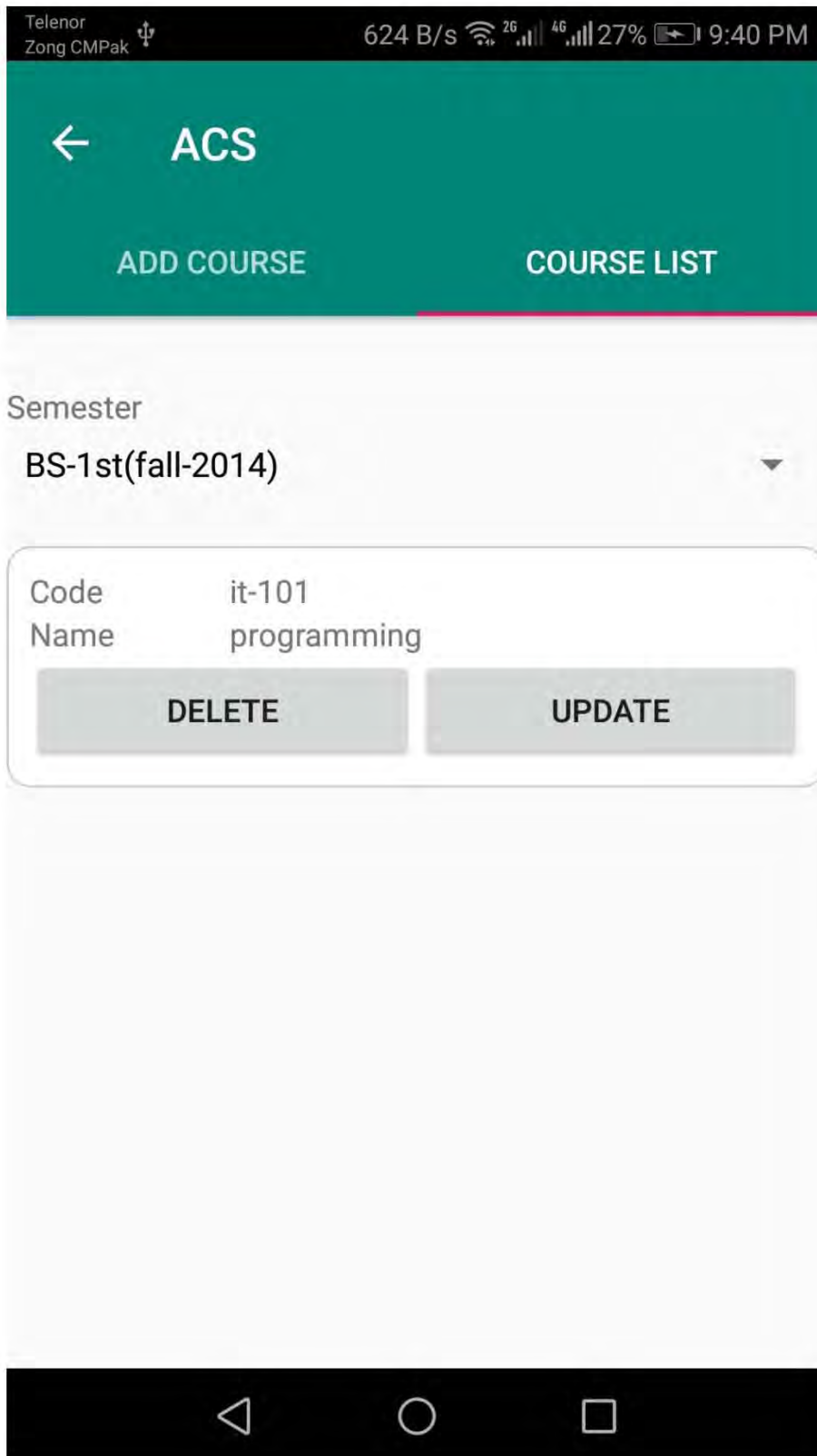


Figure 24 : Course list

#### 4.4.9 Manage/Add Semester

The screenshot shows a mobile application interface for managing semesters. At the top, the status bar displays 'Telenor Zong CMPak', data speed '334 B/s', signal strength, 4G connectivity, and 27% battery at 9:40 PM. Below the status bar is a teal header with a back arrow and the text 'ACS'. Underneath the header are two tabs: 'ADD SEMESTER' (which is selected) and 'SEMESTER LIST'. The main content area contains a form with the following fields:

- Program:** A dropdown menu with 'BS' selected.
- Name:** A text input field containing 'BS-fall2019'.
- Number:** A text input field containing '6'.
- Min credits:** A text input field containing '12'.
- Max credits:** A text input field containing '21'.

At the bottom of the form is a grey button labeled 'ADD'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Figure 25 : Add semester

#### 4.4.10 Semester list

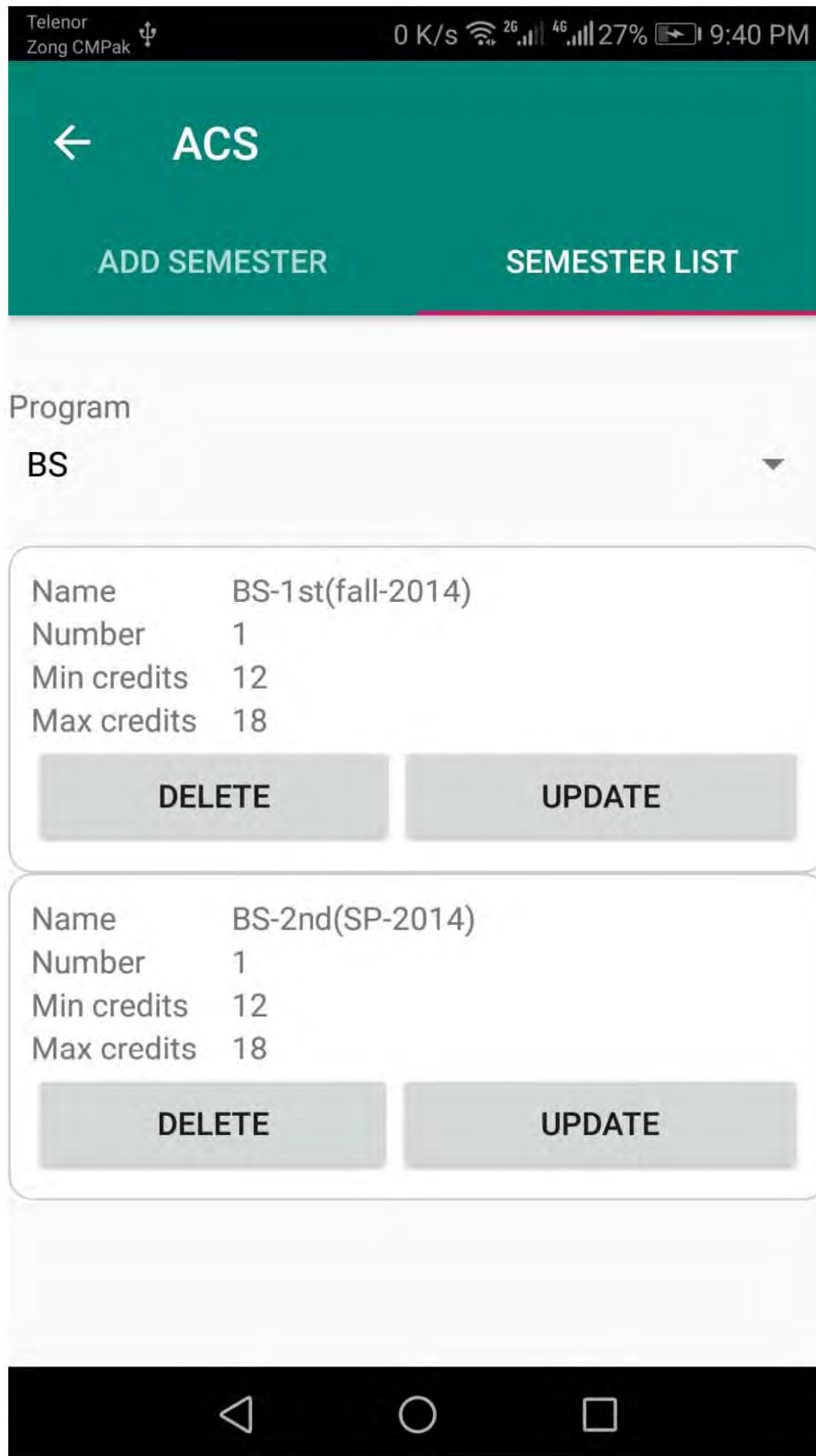


Figure 26 : Semester list

#### 4.4.11 Add Batch

The screenshot shows a mobile application interface for adding a batch. At the top, the status bar displays 'Telenor Zong CMPak', signal strength, data speed (2.5 K/s), and battery level (27%). The app header is green with a back arrow and the text 'ACS'. Below the header are two tabs: 'ADD BATCH' (active) and 'BATCH DETAILS'. The form contains the following fields:

- Department: Computer Science
- Program: BS
- Semester: BS-1st(fall-2014)
- Name: BS2019
- Normal Fee: 32000
- Late Fee: 35000
- Late
- ADD button

Figure 27 : Add batch

#### 4.4.12 Batch list

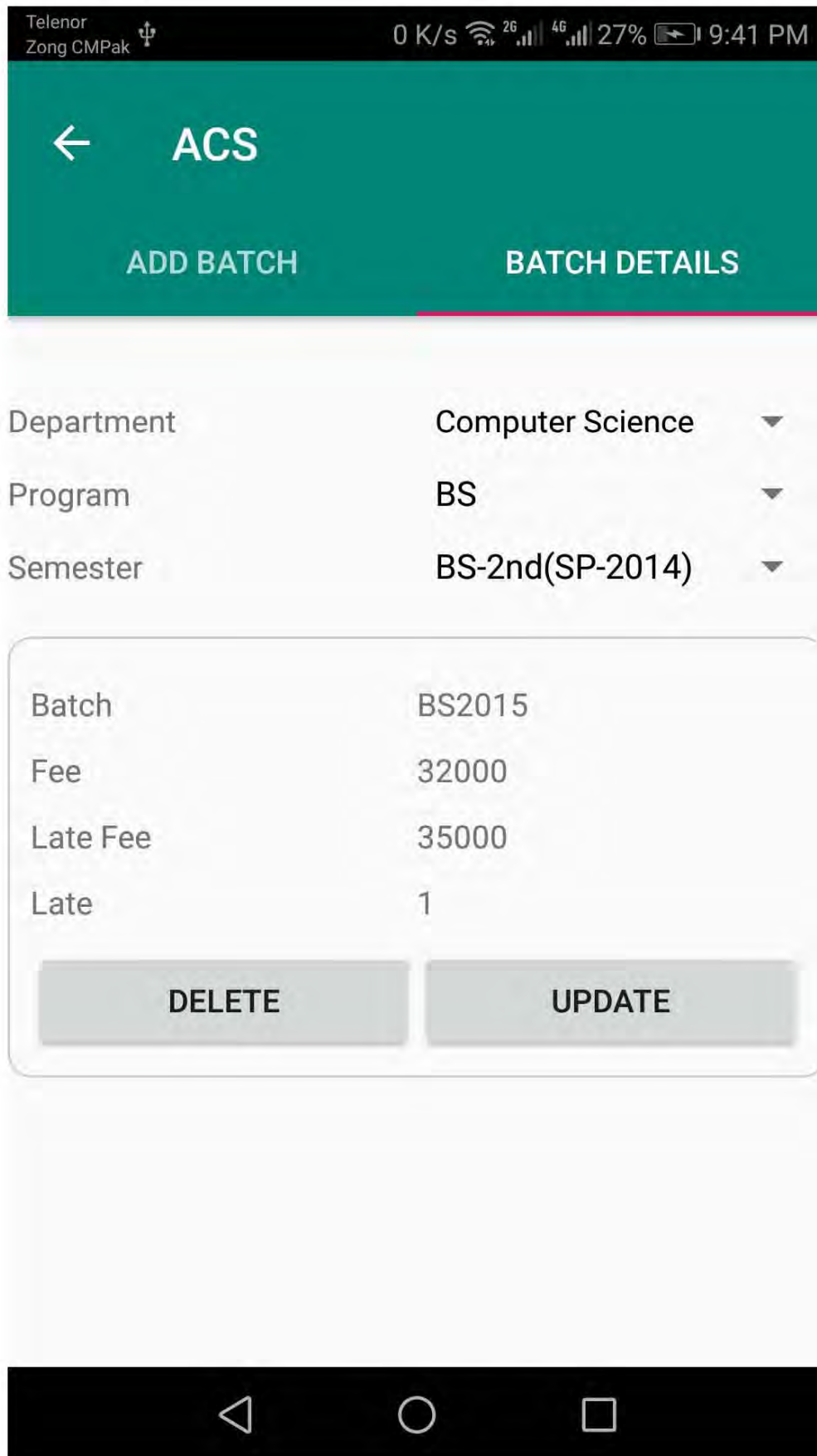


Figure 28 : Batch list



# Chapter 5 Software testing

## 5.1 Introduction

Software testing is a process of verifying and validating a software application or program. Software testing also identifies important defects, flaws, or errors in the application code that must be fixed.

The system testing is to check the verification and validation of software. System Testing tests all components and modules that are new, changed, affected by a change, or needed to form the complete application. The system test may require involvement of other systems but this should be minimized as much as possible to reduce the risk of externally-induced problems. Testing the interaction with other parts of the complete system comes in Integration Testing. The emphasis in system testing is validating and verifying the functional design specification and seeing how all the modules work together.

There are two basics of software testing:

1. Black box testing
2. White box testing

## 5.2 Black box testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. When performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

## 5.3 White box testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called internal testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

## 5.4 Test Cases

### 5.4.1 Login Test Case

Table 11:TC-001

<b>Test Case ID:</b> TC-001	
<b>Purpose</b>	Login to Account.
<b>Setup</b>	1. Install ACS app and create account.
<b>Instructions</b>	1. Open app. 2. Put email and password and press login button.
<b>Expected Result</b>	User Redirect to Student Activity.

### 5.4.2 Generate Challan Test Case

Table 12:TC-002

<b>Test Case ID:</b> TC-002	
<b>Purpose</b>	Generate new challan.
<b>Setup</b>	Login in to student account.
<b>Instructions</b>	1. Select your desired courses. 2. Press generate challan button.
<b>Expected Result</b>	Challan is generated and displayed with all registered courses.

### 5.4.3 Register Student Test Case

Table 13:TC-003

<b>Test Case ID:</b> TC-003	
<b>Purpose</b>	Create new account.
<b>Setup</b>	Install ACS app.
<b>Instructions</b>	<ol style="list-style-type: none"><li>1. Open app and select your desired batch from list.</li><li>2. Put name, email password and cnic.</li><li>3. Press Register button.</li></ol>
<b>Expected Result</b>	Registration is successful message is displayed and user is redirected to login activity.

### 5.4.4 Validate Student/Challan

Table 14:TC-004

<b>Test Case ID:</b> TC-004	
<b>Purpose</b>	To validate paid challan or actual student.
<b>Setup</b>	Login to admin account.
<b>Instructions</b>	<ol style="list-style-type: none"><li>1. Go to validate fragment and search for desired student/challan.</li><li>2. Press validate challan button.</li></ol>
<b>Expected Result</b>	Challan/Student is validated or removed from screen.

### 5.4.5 Add Course/Batch/Semester

Table 15:TC-005

<b>Test Case ID:</b> TC-005	
<b>Purpose</b>	To add new record of Course/Batch/Semester.
<b>Setup</b>	Login to admin account.
<b>Instructions</b>	<ol style="list-style-type: none"><li>1. Go to Manage Courses/Batches/Semesters Activity.</li><li>2. Add all the required Details of Course/Batch/Semester.</li><li>3. Press add button.</li></ol>
<b>Expected Result</b>	A toast is displayed that record is added successfully.

### 5.4.6 Delete Course/Batch/Semester/Challan

Table 16:TC-006

<b>Test Case ID:</b> TC-006	
<b>Purpose</b>	Delete record of Course/Batch/Semester/Challan.
<b>Setup</b>	Login to admin account.
<b>Instructions</b>	<ol style="list-style-type: none"><li>1. Go to list of Courses/Batches/Semesters/Challans.</li><li>2. Find your desired Course/Batch/Semester/Challan in list.</li><li>3. Press Delete button.</li></ol>
<b>Expected Result</b>	Record is deleted and disappeared from screen.

## 5.4.7 Update Course/Batch/Semester

Table 17:TC-007

<b>Test Case ID:</b> TC-007	
<b>Purpose</b>	Update record of Course/Batch/Semester.
<b>Setup</b>	Login to admin account.
<b>Instructions</b>	<ol style="list-style-type: none"><li>1. Go to list of Courses/Batches/Semesters.</li><li>2. Find your desired Course/Batch/Semester/Challan in list.</li><li>3. Press update button.</li><li>4. Give updated details of Course/Batch/Semester.</li><li>5. Press done button.</li></ol>
<b>Expected Result</b>	Message is displayed on screen that record is updated.

# Chapter 6 Future Conclusion

This system is implemented for Student and admin and it will provide a great ease for generating new challans from anywhere so it provides time and space efficiency. ACS also provide facility of viewing all paid challans so now student not need to keep all fee slips with him at clearance.

## 6.1 Future Work

1. In future we will add a module in system so that student can login with finger print.
2. Student can pay fee online from app.
3. Student can view his academic records.

# References

- [1] <https://www.tutorialspoint.com/restful/>
- [2] <https://docs.oracle.com/javase/7/docs/api/>
- [3] <https://docs.python.org/3/>
- [4] <https://square.github.io/retrofit/2.x/retrofit/>
- [5] <https://flask-restful.readthedocs.io/en/latest/>