

# **STUDENT SAVING ACCOUNT**

**Final Year Project**

**Session 2018-2022**

A project submitted in partial fulfillment of the degree of

**BS Computer Science**



Department of Computer Sciences

Faculty of Computer Science & Information Technology

Superior University, Lahore

Spring 2022

Type (Nature of project)	[ <input checked="" type="checkbox"/> ] <b>Development</b>	[ <input type="checkbox"/> ] <b>Research</b>	[ <input type="checkbox"/> ] <b>R&amp;D</b>
--------------------------	--	--	---

Area of specialization				
<b>FYP ID</b>	FYP-BCSM-F21-087			
<b>Project Group Members</b>				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-F18-332	Abdul Hanan	bcsm-f18-332@superior.edu.pk	
(ii)	BCSM-F18-138	Shahzal Aslam	bcsm-f18-138@superior.edu.pk	

### Plagiarism Free Certificate

This is to certify that, I \_\_\_\_\_ S/D of \_\_\_\_\_, group leader of FYP under registration no \_\_\_\_\_ at Computer Science Department, The Superior University, Lahore.

I declare that my FYP report is checked by my supervisor.

Date: \_\_\_\_\_ Name of Group Leader: Abdul Hanan Signature: \_\_\_\_\_

Name of Supervisor: Shahzad Namat

Designation: Lecturer

Signature: \_\_\_\_\_

HoD: Dr. Irfan Ud Din

Signature: \_\_\_\_\_

# Project Report

## Student Saving Account

### Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

\_\_\_\_\_

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

---

### PROJECT MANAGER

Comments: \_\_\_\_\_

\_\_\_\_\_

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

### HEAD OF THE DEPARTMENT

Comments: \_\_\_\_\_

\_\_\_\_\_

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

## **Dedication**

*This work is dedicated to my university for providing us support and a platform for implementing our idea.*

## Acknowledgment

We are very thankful to our supervisor and our mentor who gave us this golden opportunity to build our future is the most supportive person we have ever met; he guides us more We wish to express our immense gratitude to GOD Almighty for his mercy, guidance, and protection towards us for seeing us through the rigors of this work. We are greatly indebted to our supervisor **Sir Shahzad Namat** moreover all the staff our in department of Computer Science both academic and non-academic staff for their intellectual upbringing. Our special appreciation goes to our loving parents' brothers and sisters, whose moral and finical support cannot be overemphasized. Also, our sincere gratitude and special regards to our friends too many to mention whose encouragement led to the success of this work.

## **Executive Summary**

On our campus, there are students from different cities who are living in rental flats and rooms many students are not financially strong to spend money without facing any kind of problem, or this result in they cannot manage their budget. Our idea will help these students in different ways like in managing a budget as we are providing a credit system to students in university the cafeteria, in the bookshop or travel expenses.

Students do not have to pay while purchasing something from the cafeteria or bookshop they will pay at the end of the month at a single time it will help them in managing their budget or in this way they will not take the stress of getting out of budget in the mid of the month.

## Table of Contents

Dedication .....	v
Acknowledgments.....	v
Executive Summary.....	v
Table of Contents .....	vi
Chapter 1.....	1
Introduction .....	1
1.1. Background .....	2
1.2. Motivations and Challenges .....	2
1.3. Goals and Objectives .....	2
1.4. Literature Review/Existing Solutions.....	2
1.5. Gap Analysis.....	2
1.6. Proposed Solution .....	3
1.7. Project Plan.....	3
1.7.1. Work Breakdown Structure .....	3
1.7.2. Roles & Responsibility Matrix .....	4
1.7.3. Gantt Chart.....	4
1.8. Report Outline .....	5
Chapter 2.....	6
Software Requirement Specifications .....	6
2.1. Introduction .....	7
2.1.1. Purpose .....	7
2.1.2. Document Conventions .....	7
2.1.3. Intended Audience and Reading Suggestions .....	7
2.1.4. Product Scope .....	7
2.1.5. References .....	7
2.2. Overall Description .....	8
2.2.1. Product Perspective .....	8
2.2.2. Product Functions .....	8
2.2.3. User Classes and Characteristics.....	8
2.2.4. Operating Environment .....	8
2.2.5. Design and Implementation Constraints .....	8
2.2.6. User Documentation.....	9
2.2.7. Assumptions and Dependencies.....	9
2.3. External Interface Requirements.....	9
2.3.1. User Interfaces.....	9
2.3.2. Hardware Interfaces .....	9
2.3.3. Software Interfaces.....	10
2.3.4. Communications Interfaces .....	10
2.4. System Features .....	10
2.5. Other Nonfunctional Requirements.....	10
2.5.1. Performance Requirements.....	10
2.5.2. Safety Requirements.....	10
2.5.3. Security Requirements.....	11
2.5.4. Software Quality Attributes .....	11
2.5.5. Business Rules.....	11

2.6. Other Requirements .....	11
Chapter 3.....	12
Use Case Analysis.....	12
3.1. Use Case Model .....	13
3.2. Use Case Descriptions .....	14
Chapter 4.....	15
System Design .....	15
4.1. Architecture Diagram .....	16
4.2. Domain Model .....	17
4.3. Entity Relationship Diagram with data dictionary.....	18
4.4. Class Diagram.....	19
4.5. Activity Diagram .....	20
4.6. Deployment Diagram .....	21
Chapter 5.....	22
Implementation .....	22
5.1. Important Flow Control/Pseudo codes .....	23
5.2. Components, Libraries, Web Services and stubs .....	23
5.3. Deployment Environment .....	23
5.4. Tools and Techniques .....	24
5.5. Best Practices / Coding Standards .....	24
5.6. Version Control .....	24
Chapter 6.....	25
Testing and Evaluation.....	25
6.1. Use Case Testing.....	26
6.2. Equivalence partitioning.....	26
6.3. Boundary value analysis .....	27
6.4. Data flow testing.....	27
6.5. Unit testing .....	28
6.6. Integration testing .....	28
6.7. Performance testing .....	28
6.8. Stress Testing.....	28
Chapter 7.....	29
Summary, Conclusion and Future Enhancements.....	29
7.1. Project Summary .....	30
7.2. Achievements and Improvements .....	30
7.3. Critical Review .....	30
7.4. Lessons Learnt .....	30
7.5. Future Enhancements/Recommendations .....	31
Appendices .....	32
Appendix A: User Manual .....	32
Reference and Bibliography .....	33
Index .....	34

## List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

## List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

# Chapter 1

## **Introduction**

# Chapter 1: Introduction

Most of the students face financial issues they cannot maintain their budgets sometimes students also face issues related to cash/money like what if they don't have money for printing their assignments or if they do so then how can they buy food or what if no cash remains for fare? So in this scenario, this project will help them out of the above problem.

## **.1. Background**

In universities there are students of different financial backgrounds some of them can afford the universities expenses but most of them cannot do so, many students are from other cities who also have to manage their rents or other expenses.

## **.2. Motivations and Challenges**

At the start, we can face problems related to that how to convince students to use this idea or to explain how this project will help them in managing their budget.

## **.3. Goals and Objectives**

Our main objective is to facilitate the student as much as we can to help with their financial problems or in managing their budget.

## **.4. Literature Review/Existing Solutions**

Run time there is no solution implemented in our university but in other universities of Lahore like UET implemented this system manually, but that system is not much help for the students.

## **.5. Gap Analysis**

At that time our students are not facilitated with any kind of idea that will help them in managing their budget and will facilitate them in different ways and this idea will be very useful or helpful especially for those students who are from other cities and in Lahore they are living on rent or trying to minimize their expenses or controlling their budget.

## **.6. Proposed Solution**

In our project, student do not have to pay on book shops or in cafes on time, one will pay the total cash in the end of the month at a single time means to say that we are providing a credit system to students every month and at the end of each month student must clear it.

## **.7. Project Plan**

Firstly, we will cover all the problems by providing solutions to it. We will launch the beta version, or you can say the testing version on our own university campus to check how our project goes or what its impact on our students.

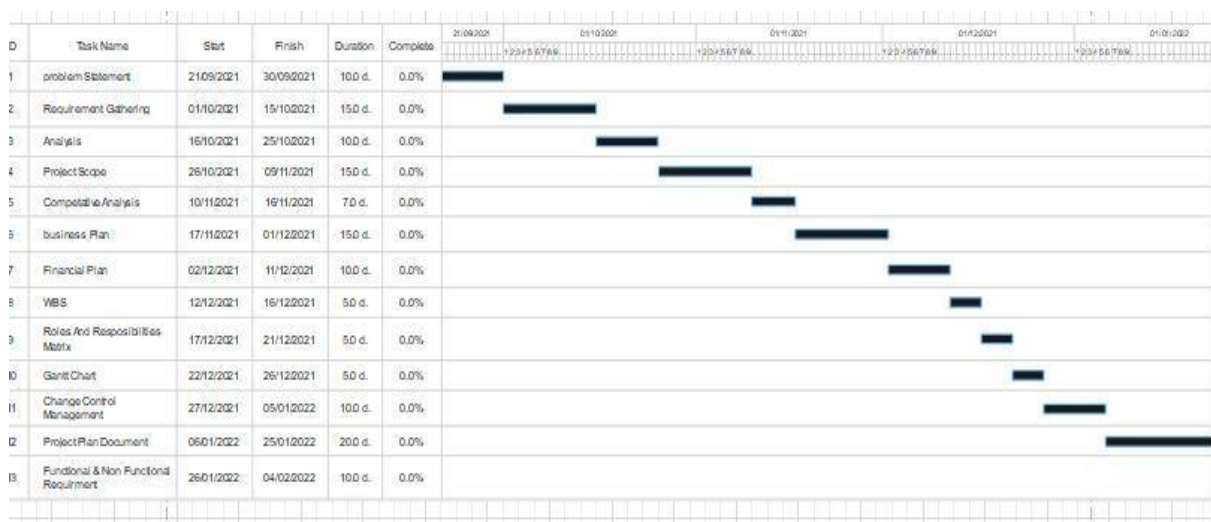
### **7.1. Work Breakdown Structure**

- We will create documentation
- for Software analysis
- Testing (beta version
- analysis) Implementation
- Future upgradation

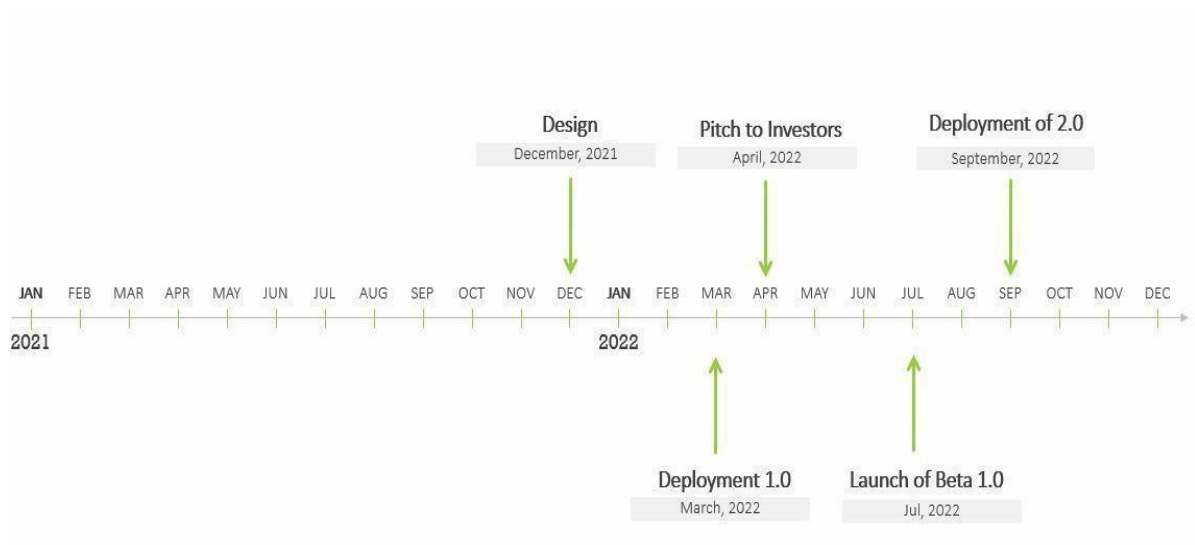
## 7.2. Roles & Responsibility Matrix

<i>Positions</i>	<i>Rules and Responsibilities</i>
Team head	Manage team Control workflow Check or manage the implementation of the project
Software analyzer	Provide the latest software architecture support Provide different software management or development ideas
Designer	Design the UI/UX of the App Design the theme and color scheme Design user-friendly UI
Developer	Develop the app Manage the upgradation by using the latest tools and technologies
Admin	Manage Students side Provide user support Report to team leader

## 7.3. Gantt Chart



## .8. Report Timeline



# Chapter 2

## Software Requirement Specifications

## Chapter 2: Software Requirement Specifications

### **.1. Introduction**

#### **1.1. Purpose**

Basically, this project maintains the credit record of the student and it relates to the university portal it also generates notification alerts on student phone. One can use a credit system with their student card because is also synced with the chip on the student card.

#### **1.2. Document Conventions**

The UI of our application is user-friendly and easy to understand. We will use that type of font style or size that will be easily readable or understandable. We will use modern themes for the UI. Besides this stuff our notification alert is also of that type that one will not have to open the app our notification describes the detail and concisely explain the alert.

#### **1.3. Intended Audience and Reading Suggestions**

Firstly, we will do some seminars on our campus for explaining the idea or its works and in the end of the seminar we will collect feedback from different students and ask them about what kind of changes we can do further in this idea and after launching beta version we will again ask and collect comments on this project of different students to know the flaws and problems to cover up in the next version.

#### **1.4. Product Scope**

This is reliable for students at universities and colleges in maintain their monthly budget or they can also save some cash in this way this is a credit system for students. Just like we get credit cards from bank and use that money for different purposes, student can buy everything from bookshop and cafeteria on credit and at the end of the month one should have to clear its monthly pending balance. So whole month student will not worry about cash.

#### **1.5. References**

## **.2. Overall Description**

### **2.1. Product Perspective**

This product is only applicable for students at universities and colleges to help them or to relief them from their money spending stress. It is a new idea which is right now not available in any other university or colleges, but some universities are providing discount when a student buy something from university book shop.

### **2.2. Product Functions**

Maintain the credit record of every student

Student can also deposit or save some cash for future use

Application is sync with the portal so in case if any student does not clear their pending dues its portal will get blocked.

Generate notification alerts

Synchronized with the chip on student card.

### **2.3. User Classes and Characteristics**

Users of this application is bookshop or cafeteria to store the detail of student that when one bought something on credit or students are also users of this application to check details and to deposit cash or to check their balance.

### **2.4. Operating Environment**

This application is applicable for android 4.4 to latest and IOS users. It will work on chip card technology for back tap. Chip Card reader technique is used in it or also the latest payment gateway is implemented in it.

### **2.5. Design and Implementation Constraints**

First issue is that we have to use university database for this project so, in case maybe we will face some issue, but it can be overcome by convince university that they can give access to us with some restrictions. Or the next problem that we can face is that we also have to collaborate with the admin staff of the university so we also have to convince them about the security levels or many other things. We can also face some issues or will know some gaps

present in this idea when we launch beta version which will be covered after getting user response and feedback.

## **2.6. User Documentation**

We will arrange seminars in the university to explain the working of our project but we will not give any kind of manual with our product because its an application-based project so no manual is required for it or the application will also be user friendly or easy to understandable.

## **2.7. Assumptions and Dependencies**

First thing first, if university will not give access to their database so maybe it will become difficult for us to achieve the goal of this project because our application relates to the university portal. So, in this case we will try to use another method for database. Next assumption is that if this project work according to our expectation, then we will do this for teachers too.

## **.3. External Interface Requirements**

### **3.1. User Interfaces**

We will check internet connectivity and many other stuffs on the background of splash screen.

Home screen will be user-friendly, and buttons are of check balance on student side and a button to get student detail or to enter detail of what student buy.

There is a drawer in which you can see your login details.

On student side one can also see their balance details.

Login screen on the start and there is a logout button in drawer.

On home screen a push notification box generates like error message box.

### **3.2. Hardware Interfaces**

Application interacts with smartphone through Background processes Communication protocol with database is through APIs

One can read card on phone by a tap.

### **3.3. Software Interfaces**

Software will be intellectually designed and home screen will be user friendly and buttons are of check balance on student side and a button to get student detail or to enter detail of

what student buy. There is a drawer in which you can see your login details. On student side one can also see their balance details. Login screen on the start and there is a logout button in drawer. On home screen a push notification box generates like error message box.

### **3.4. Communications Interfaces**

The communication interface of our project is to read and write student data from their university email. This thing can be approach by the access of student information from admin of university. Our APIs are also gives us JASON response after generating HTTPs request to server.

## **.4. System Features**

Maintain the credit record of every student

Student can also deposit or save some cash for future use

Application is sync with the portal so in case if any student does not clear their pending dues its portal will get blocked.

Generate notification alerts

Synchronized with the chip on student card.

## **.1. Other Nonfunctional Requirements**

### **1.1. Performance Requirements**

In our project, student do not have to pay on book shop or in café on time, one will pay the total cash in the end of the month at a single time means to say that we are providing credit system to students every month and in the end of each month student must clear it.

### **1.2. Safety Requirements**

This project is fully secured we will use secure database of university. Secure login feature is used in application which is full secure and safe. Our source code is also encrypted, or login is also encrypted.

### **1.3. Security Requirements**

Our application is secure from attackers as we will use secure database of university. Secure login feature is used in application which is full secure and safe. Our source code is also encrypted, or login is also encrypted.

### **1.4. Software Quality Attributes**

Our application is easy to understandable, as cafeteria management and book shopkeeper also understand by its GUI. It is more flexible for student and use high-speed server response, which need few seconds for response. This app is also easy to maintain because UI is modern and development technology is modern.

### **1.5. Business Rules**

In this app the criteria is that when student deposit his/her money it will get some returns like some paper prints or photocopies are gifts for him/her. In addition, if student does not pay dues, then he/she will get fine challan and get their portal block.

## **.2. Other Requirements**

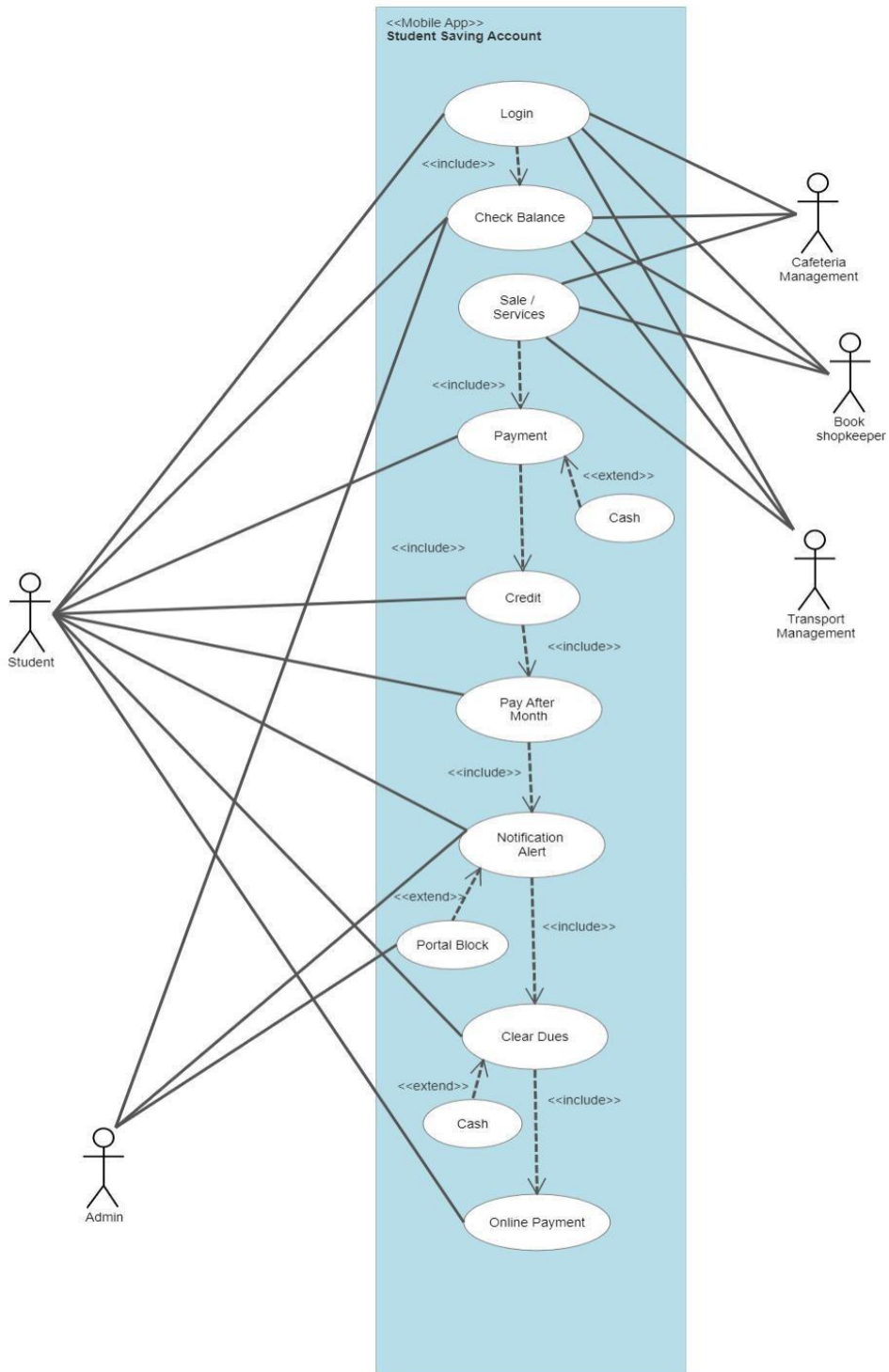
Oure first legal requirement is to get student information from admin and act on those students who do not pay dues and get fine. Database of students need admin verification, and this is the main thing by which our idea is based on. University admin faculty can only give us access on some privacy policies agreement.

# Chapter 3

## Use Case Analysis

# Chapter 3: System Analysis

## 3.1. Use Case Model



## 3.2. Use Case Descriptions

### Admin:

Admin can access all the data available in the database.

Admins have rights to block and unblock user (student).

Admin can add/remove user (bookshop user/cafeteria user). Admin can see credit details of user (student).

### User:

There is multiple type of users that can use app on their own platform:

- Student
  
- Bookshop User
  
- Cafeteria User

Student can check balance, their transection history their status and profile.

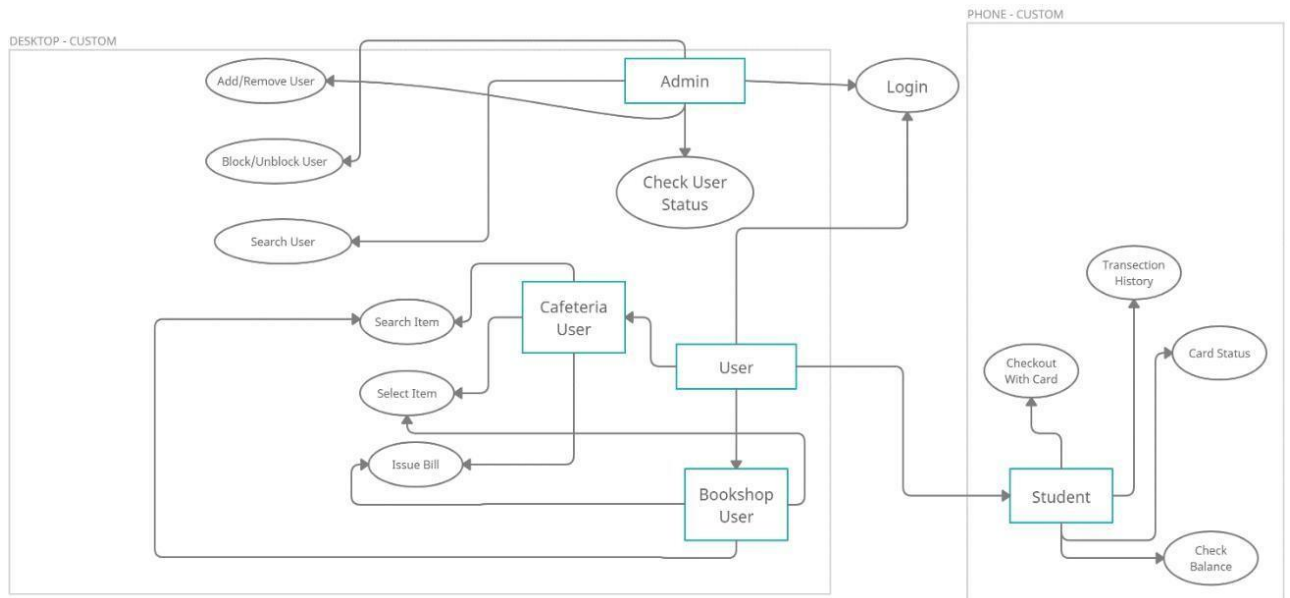
Bookshop User and Cafeteria user can add/remove items, provide tap to pay on student card facility and manage stock of items.

# Chapter 4

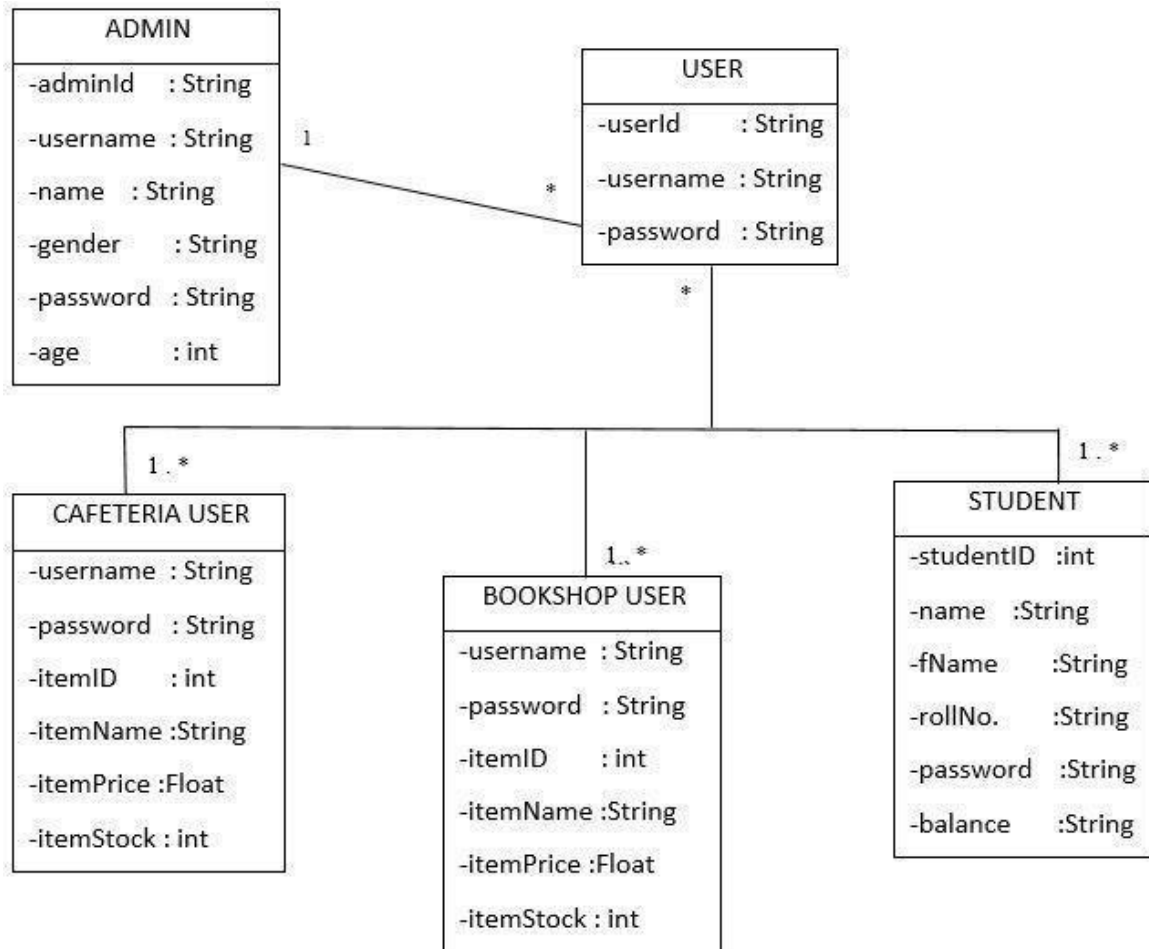
## System Design

## Chapter 4: System Design

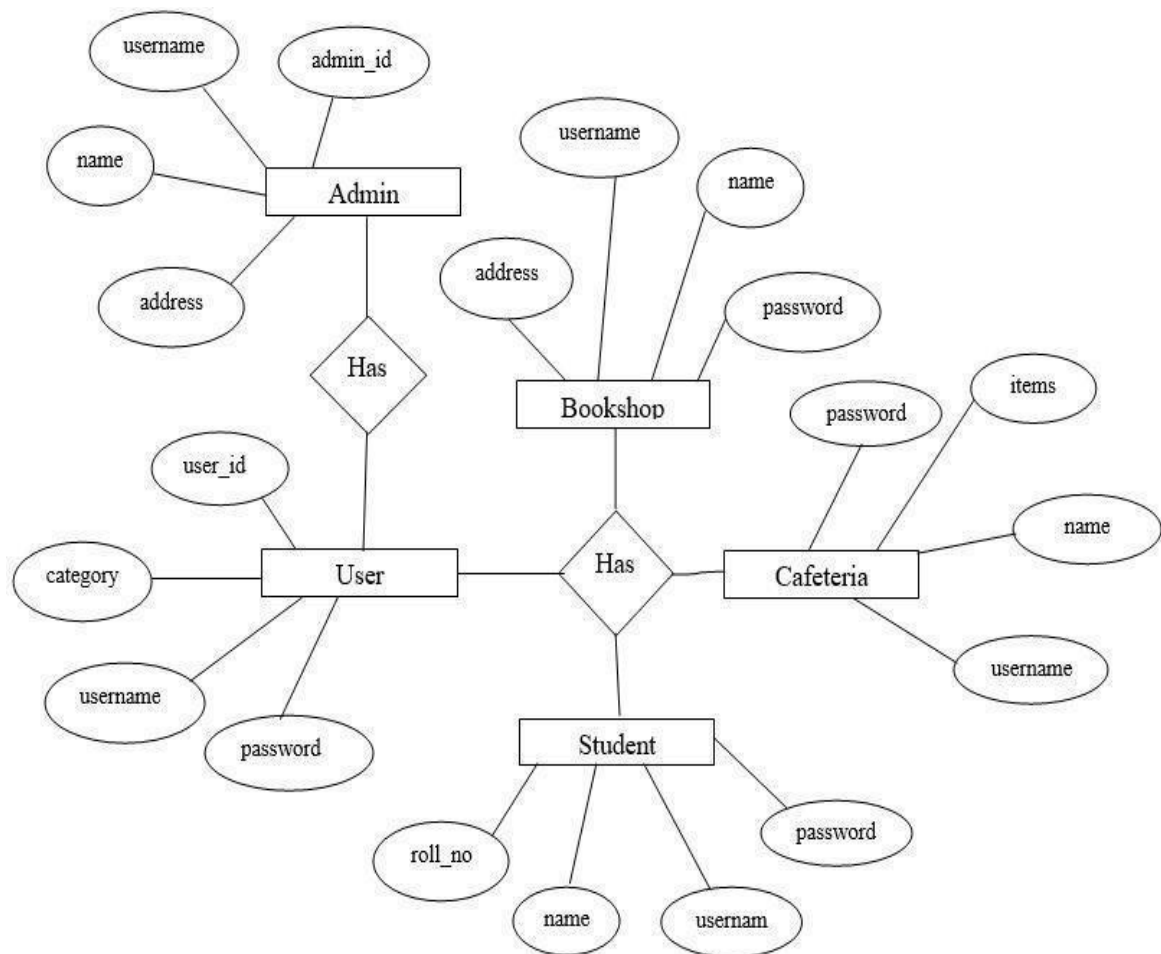
### 4.1. Architecture Diagram



## 4.2. Domain Model



### 4.3. Entity Relationship Diagram with data dictionary



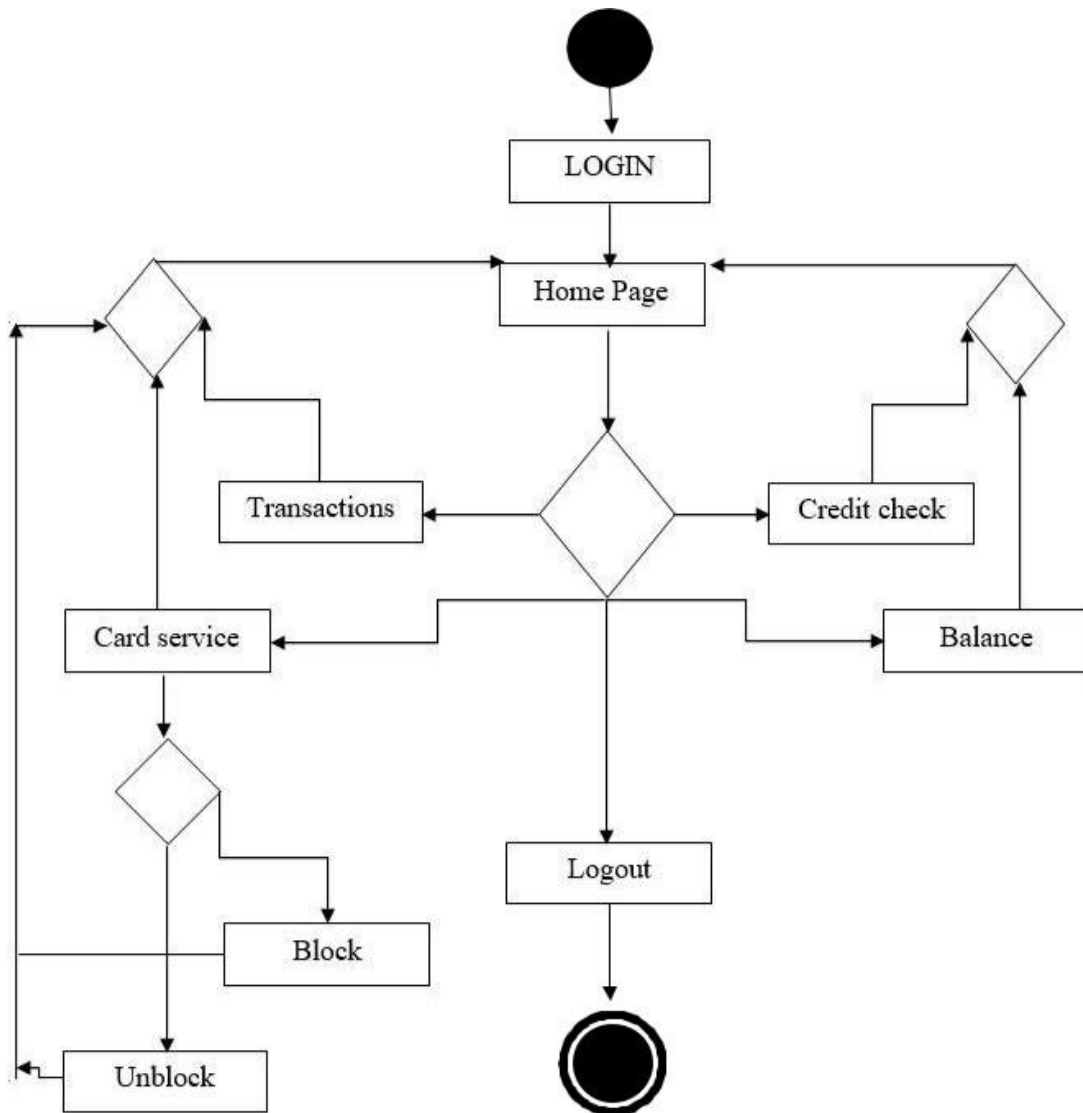
## 4.4. Class Diagram

Admin	
-adminId	: int
-username	: String
-name	: String
-gender	: String
-password	: String
-age	: int
+login()	
+search()	
+status()	
+removeUser()	
+addUser()	
+blockUser()	

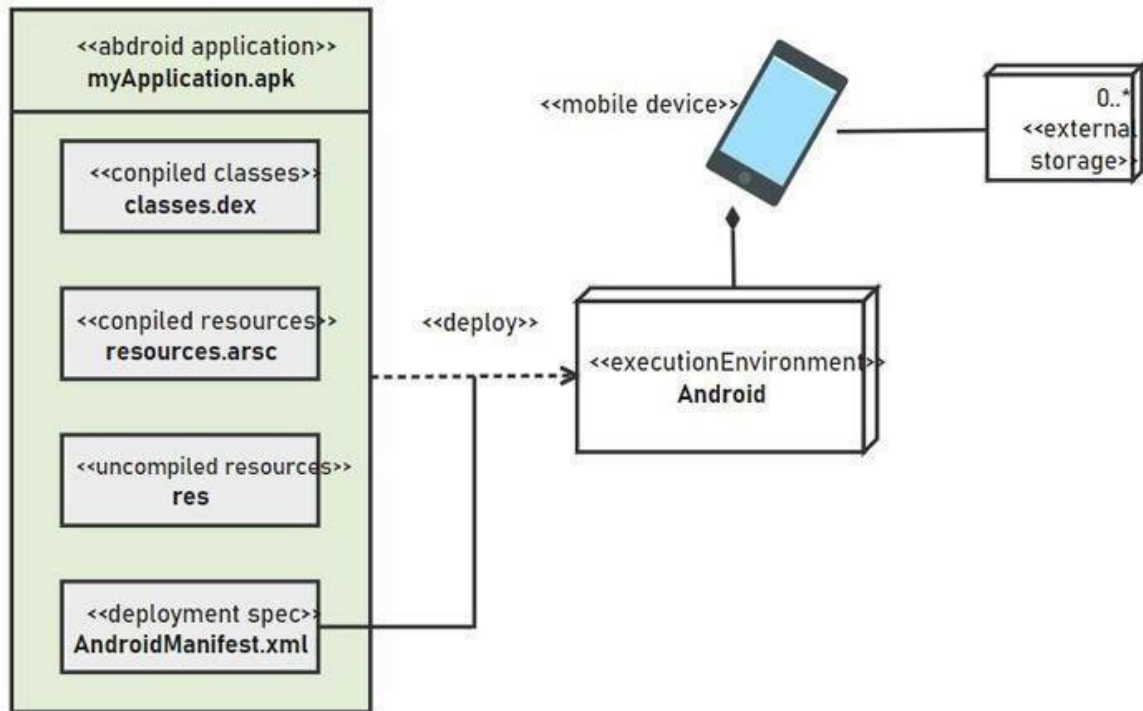
USER	
-userId	: String
-username	: String
-password	: String
-itemID	: int
-itemName	:String
-itemPrice	: Float
-itemStock	: int
+login()	
+searchItem()	
+removeItem()	
+addItem()	
+Checkout()	

Student	
-studentID	: String
-name	: String
-rollNo	: String
-password	: String
+login()	
+credit()	
+transactionhistory()	
+checkBalance()	
+cardStatus()	

### 4.5. Activity Diagram



## 4.6. Deployment Diagram



# Chapter 5

## Implementation

## Chapter 5: Implementation

In this chapter, we talk about the tools and technologies that are used to develop this project. Our basic concern is the latest technology like Flutter, React Native or Jetpack Compose etc.... We create our project in a cross browsing platform so we choose Flutter to develop this app and desktop app.

### 5.1. Important Flow Control/Pseudo codes

### 5.2. Components, Libraries, Web Services, and stubs

#### Components:

Dart Main modules.

Use GetX (MVC architecture).

Use Bindings, Controllers, Views, Widgets and Page Routs.

#### Libraries:

lib getX ^ [version]

Library of third-party dependency like make charts.

#### Web Services:

Email & OTP Authentication.

Linking With UMS.

#### Stubs:

For Now, we are working with the local server and rest API. In future, we'll buy an AWS server for hosting and database management.

### 5.3. Deployment Environment

Our main deployment environment is:

Play Store

App Store

On the university end, we can maintain that by the web application or desktop application.

## 5.4. Tools and Techniques

### Tools:

VS Code  
XAMPP  
Android Studio

### Technologies:

Dart  
PHP  
JSON

## 5.5. Best Practices / Coding Standards

### Our best practice of Coding Standards is:

Use architectures like MVC or MVVM to make UX and Coding Flow better. Use camel case technique to declare or initialize anything in code.

Use private, public, and protected modifiers to precise things in code.

## 5.6. Version Control

Version control depends upon the up-gradation of Mobile software or in-app bugs fixes. Firstly, we launch the beta version of this app for testing. After beta testing, we launch this with version 1.0.0 in the market after this we can control our application version control easily.

# Chapter 6

## Testing and Evaluation

## Chapter 6: Testing and Evaluation

### 6.1. Use Case Testing

#### 4. Use case for Login

Main Success Scenario	Step	
A: Actor	1	A: Insert username & password
S: System	2	B: Validity check automatically from DB.
	3	C: Allow access to Account.
Extensions	2A	Username not valid
	2B	Password not valid

5.

6.

#### 7. Use case for add user

Main Success Scenario	Step	
A: Actor	1	A: Insert Login Details
S: System	2	B: Validate username, phone number, ID details
	3	C: Allow access to submit to Database
Extensions	2A	Entity must be filled
	2B	Duplication not allowed

### 6.1. Equivalence partitioning

## 6.2. Boundary value analysis

Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.

So, these extreme ends like Start- End, Lower- Upper, Maximum-Minimum, Just Inside-Just Outside values are called boundary values and the testing is called boundary testing.

The basic idea in boundary value testing is to select input variable values at their:

1. Minimum
2. Just above the minimum
3. A nominal value
4. Just below the maximum
5. Maximum

## 6.3. Data flow testing

Data Flow testing is one of the testing strategies, which focuses on the data variables and their values, used in the programming logic of the software product, by making use of the control flow graph. The basic idea behind this form of testing, is to reveal the coding errors and mistakes, which may result into improper implementation and usage of the data variables or data values in the programming code. In this Project We have done static data flow testing in which We have study and analysis the code without performing the actual execution of the code such as wrong header files or library files use or syntax error. While testing this We have remove anomalies from the code like variables are defined but they are not used. Some variables are defined twice.

## 6.4. Unit testing

8. Testing is done to check whether the individual modules of the source code are working properly. i.e., testing each unit of the application separately by the developer in developer's environment.
9. • We have done this testing my keeping in mind the goal of the system. In this testing We have carefully checked the naming conventions.
10. • we have clearly checked the flow is correct between inside and outside the module.
11. • Verify the check statements

12. • Check the extreme boundaries values of output
13. • Loops are correct or not

## 6.1. Integration testing

## 6.2. Performance testing

Performance testing plays a vital role. It is the only way to determine an application's behavior at different levels of load and users. The goal of Performance Testing is not to find bugs but to eliminate performance bottlenecks. I have performed performance testing manually. many active sessions of the same application should be open for testing.

Manually, we have determined the number of active sessions, number of database connections open, and number of threads running. The amount of CPU time, the memory being used, and few more parameters can also be obtained. A quick preliminary database performance test is done manually to get the necessary information.

## 6.1. Stress Testing

Testing is defined as a type of Software Testing that verified the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. The goal of stress testing is to analyze the behavior of the system after a failure. For stress testing to be successful, a system should display an appropriate error message while it is under extreme conditions. A quick check on the database for some preliminary database performance tests can also be done manually. A performance tester can also simulate his performance tests from several different client machines reaching out to the server, but that is not really a very effective way of running a performance test

# Chapter 7

## Summary, Conclusion and Future Enhancements

## Chapter 7: Summary, Conclusion & Future Enhancements

### 7.1. Project Summary

We are providing credit system for students, studying in the universities. In our project, student do not have to pay on book shop or in café on time, one will pay the total cash in the end of the month at a single time means to say that we are providing credit system to students every month and in the end of each month student must clear it. If in case student did not pay or clear his/ her dues, admin has the right to block his/her ums portal.

### 7.2. Achievements and Improvements

We are providing facility to students like they can easily manage their budget and their most o the financial issues get resolved with the help of this project. and with the time we will add more functions after the feedbacks of students or if we feel like that we should have to add or improve anything, we will do so.

### 7.3. Critical Review

Basically, this project maintains the credit record of the student and it relates to the university portal and it also generate notification alerts on student phone. One can use credit system with their student card because it is also sync with the chip on the student card.

### 7.4. Lessons Learnt

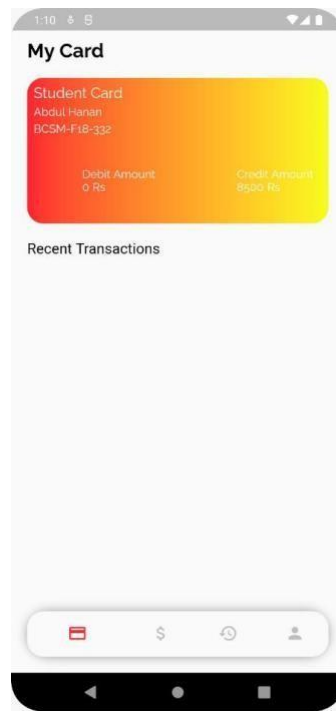
Without teamwork, it was not possible to build this application, and besides that through this App. But We fully try to make the Application and complete the project, we learn so many things like: Our team has learned that Time Management decide the success and failure of the project so it is necessary to do the tasks on the schedule time otherwise it will create burden and may led the project towards failure. And smart way to handle bugs and error.

## **7.5. Future Enhancements/Recommendations**

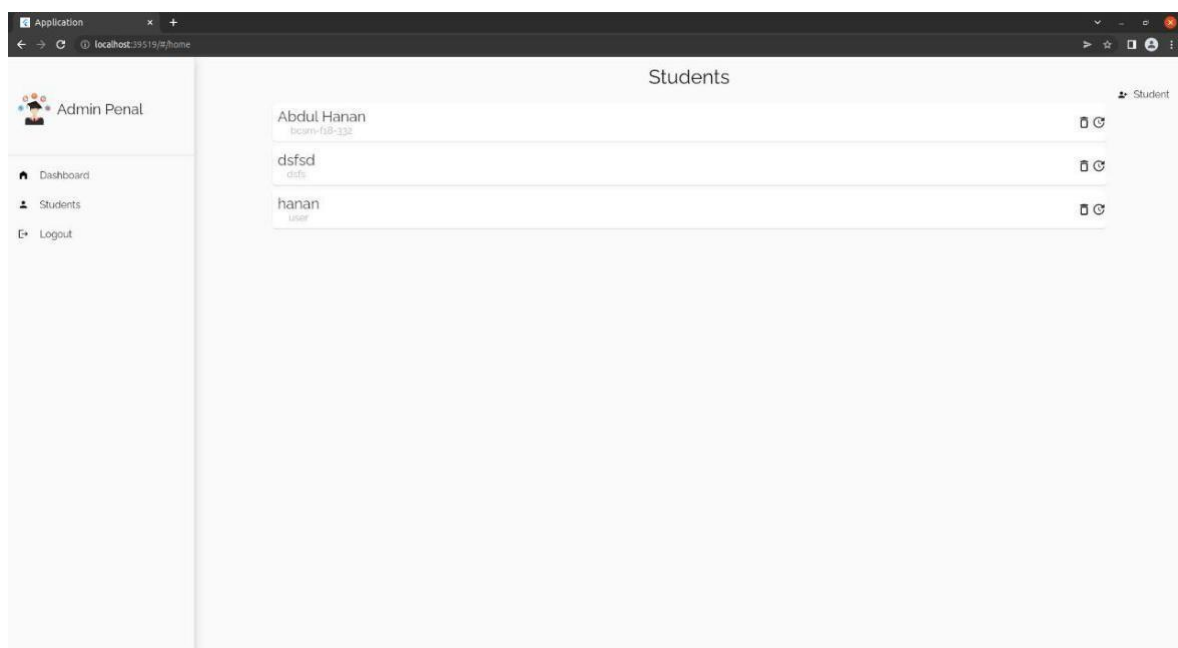
Many Enhancements are required in this project. This project must use High security layers so that data should be flow in between the concerned people. As well as need to add new smart feature on it that make this project more mechanized and efficient.

# Appendices

## Appendix A: User Manual



Mobile-App



Admin-Penal

# Reference and Bibliography

## Reference and Bibliography

- [1] M. Sher, M. Rehman, "*Title of the Paper*" Conference name/Journal Name, Edition, Volume, Issue, ISBN/ISSN, PP, Publisher/City-Country, Year.

# Index

## Index

[A]

[B]

[C]