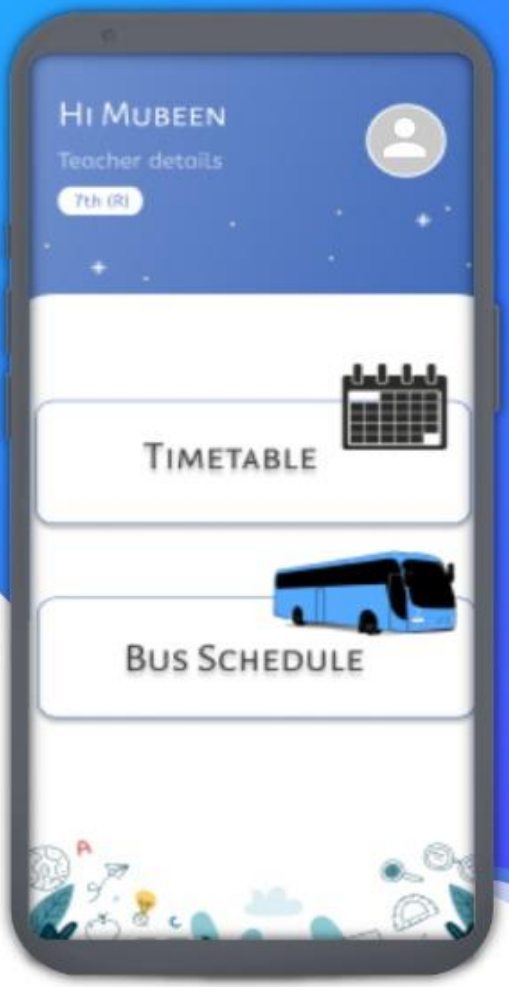


UOSify

Your Schedules in your hand

Find Schedules

- Class Timetable
- Bus Schedules
- Show timetable of particular class

Instant Notifications

- Notification before every class, informing room No. and subject
- Notification for change in schedule
- Notification for change in buses routes and time

Bus Tracking

- Track nearby buses
- Provide information of bus stops and routes

Our Team
Hajira Raheem, Rabia Ashraf,
Fizza Nasir, Mubeen Fatima

CS & IT department, University of Sargodha

Document Information

Category	Information
Customer	Institute
Project	UOSify app
Document	Documentation of UOSify
Document Version	
Identifier	
Status	Final
Author(s)	Fizza Nasir, Mubeen Fatima, Rabia Ashraf, Hajra Rahim
Approver(s)	PM
Issue Date	21 jun, 2022
Document Location	
Distribution	1. Advisor 2. PM 3. Project Office

Table of Contents

Chapter 1

Proposal

1.1 Abstract

In this era of technology, almost everything is automated. Manual Work has been digitized. The workload is now shared by many shoulders. If we particularly talk about the University's traditional systems, important notices regarding exams, timetables, bus schedules, and different events were displayed on boards which usually result in a rushed environment. Especially newcomers are the ones who face many difficulties regarding this. The "UOSify '' provides an improved student service, enhanced information sharing as well as teacher's facilities. Applications can handle the basic details that students or teachers should have. The details include Timetable of all classes, tracking and scheduling of Bus system. This application is revolutionary to document processing. The main objective of the proposed university management system is to computerize the existing system and reduce manpower and time consumption.

1.2 Background and Justification

After going through many websites and projects, it has been seen that innovation can be done in it. Some of the websites that we are especially going to discuss are Bus Tracking Application by Kashipara [1], and 1000 Projects [2] Timetable Management System by Free Projectz [3], MY CLASS CAMPUS [4], CampCodes [5].

If we talk about the features shared by **Bus tracking System** from the two references [1] [2], the users can track bus status by using Google Map and GPS. Besides this feature, no more information is provided to users like if any other bus goes through the same route or in case of any complaint whom he/she heads to.

Lastly, the **Timetable Management System** from the mentioned references [3] [4] [5] has characteristics in common; the users, especially teachers and students can view only their schedules and an adequate amount of administrative time and effort is saved. In this system there is no feature of notifications. It can become more user-friendly by adding features like alert messages before every class and class cancellation. Rooms with working pieces of equipment, unoccupied rooms, etc. options can also be added.

1.3 Project Methodology

- **SDLC for UMS**

System Development Life Cycle (SDLC) is going to be one of the best methods for the creation of this application. Since all the necessary requirements and vision for the creation of this app are crystal clear. SDLC phases are:

1. Requirement gathering
2. Analysis
3. Designing
4. Coding

5. Testing
6. Implementation

All the SDLC phases will be implemented **iteratively**. After gathering requirements for it, these requirements will be analyzed and further designed. After coding and testing this app, we will be able to continue further. As our android based application has lots of modules, it requires testing techniques to ensure working of the model so we will use the iterative model. Based on requirements and testing, we will get a model of the application. If the trained model works properly then we will launch the app otherwise redo the training.

- ## Database

For UOSify we will use the Firebase database.

- ## Modules
  - **Login:** Firstly, the user will have to log in to access any module.
  - **Timetable:** In this module, Admin can update, manage, and edit timetables according to requirements. Teachers and students can see their full schedule of classes. Pop-up messages will appear before starting class that will show the timing and room number of that class. Teachers can cancel and reschedule their classes.
  - **Bus Tracking and Scheduling:** Timing, Bus routes, Bus drivers and Bus numbers will display in this module. Google maps will be used to determine how much time a bus will take to reach a certain point. There will be a complaint box page for complaints.

- ## Tools

  Tools that will use by UOSify are following:

  - **Android Studio** (teachers, Students and Bus driver portals will implement in android Studio)
  - **Firebase**
  - **Sublime** (admin portal will implemented in sublime)

- ## Languages

The front-end design of UOSify will be in **XML, HTML, CSS, and JAVA Script**.

The back-end design of UOSify will be in **JAVA and PHP.**

1.4 Project Scope

This proposed UOSify is an Android and Web based application. This app has features of:

- **Login.**
  Timetable management (which includes students, teachers, and classes information), free room availability (class switching and sending pop - up notification to inform). Everyone will be able to see just his own timetable.
- **Bus tracking and scheduling** (which cover all buses and driver information including bus timing, routes, driver contact number and a complaint box in terms of any query). It cannot provide more accuracy than the Google GPS system.
- It requires a **strong internet connection** to operate.

1.5 High level Project Plan

ID	Task Name	Duration	Start	Finish
1	Requirment gathering	2 weeks	8/11/2021	19/11/2021
2	Analysis and Proposal	3 weeks	19-11-2021	9/12/2021
3	Documentation	5 weeks	9/12/2021	12/1/2022
4	Desgin	3 weeks	12/1/2022	1/2/2022
5	Coding	5 weeks	1/2/2022	7/3/2022
6	Devlopment	5 weeks	7/3/2022	8/4/2022
7	Implementation	6 weeks	8/4/2022	19/5/2022
8	Testing	4 weeks	19/5/2022	16/6/2022

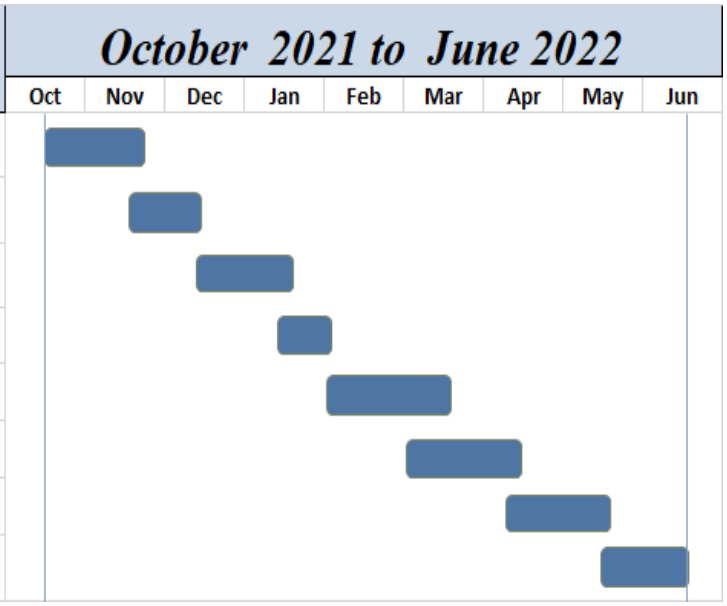


FIGURE 1 GANTT CHART

Chapter 2

Software Requirement Specifications

2.1 Introduction

2.1.1 Purpose of Document

This document describes different accepts of our proposed system. The main objective of our proposed work is to facilitate students, teachers, and admin, regarding maintaining timetable. Currently, most of the work is done manually, but this is the era of digitalization, so these tasks should be done digitally. This proposed system will provide timely and easy access to information.

2.2 Overall System Description

2.2.1 Product Perspective

The proposed application "UOSify" is an online android-based application. This application will provide different modules like the timetable, bus tracking availability. The students and teachers can only view the info provided in the modules. But teachers have limited access to make some changes in the timetable. The System will also have an ADMIN who has rights to manage resources, especially in the timetable management module. There are basic two types of users one is the student, and the other is a faculty member. Each user facilitates with a different account number having a profile along with a password for private use. The two types of users differ from each other according to access limits.

2.2.2 Product Features

There are three different users who will be using this product:

- **Admin**

The features that are available for the Administrator are:

- The administrator has all the rights over the app.
- Can create/delete an account.
- Can view the accounts.
- Can change the password.
- Can hide any kind of features from both users.
- Insert/delete/edit the information available on the app.
- Can access all the accounts of the faculty members/students.

- **Teacher**

The features available for the faculty members are:

- Can view timetable
- Can reserve or cancel class
- Can view buses info.
- Can view Tracking

- **Student**

The features available for the students are:

- Can view their timetable.
- Can view buses info.
- Can view Tracking
- Can view notifications
- Only CR/GR can make reserve rooms

2.2.3 Design and Implementation Constraints

The Product is developed using XML for front end and Java for backend. The Web portal should be designed using HTML, CSS, Java Script, and PHP for backend the backend database for this is Firebase. The product is accomplished with a login facility so that specific function is available to specific users.

2.2.4 Assumption and Dependencies

The product needs the following third-party product.

- Firebase database for storing data.
- Android Studio to develop the Product.
- Google Maps

Assumptions and Dependencies

- Coding should be error free.
- Application should be user friendly.
- All the information should be stored in a database that is accessible by app.
- It should provide fast access to the database.
- Users may access info if he/she has internet connection.
- Users must have their correct usernames and passwords to access info.

2.2.5 User characteristics

There are various kinds of users for the product. The users include:

- Transportation head and Department Timetable manager who will be acting as the controller and he will have all the privileges of the administrator.
- Faculty members will be using the above features by accessing the UOSify.
- Students who will be using the above features by accessing the UOSify.

2.2.6 Operating environment

The application will run on android devices (smartphones and tablets). For proper functioning, it will require a database and Google Maps.

2.2.7 System constraints

- **Software constraints:** Programming will be done in JAVA, XML, HTML, CSS, Java Script, PHP, and the development environment will be an android studio and in sublime. Database will be firebase.
- **The cultural constraints:** Interfaces are only in English; no other language option is available.
- **User constraints**: Users should have basic knowledge of using Google maps.

2.3 External Interface Requirements

- **Programming language:** XML, HTML, CSS, JAVA Script (front end) and JAVA and PHP (back end)
  - We will use XML for layout. The elements (Text view, Buttons, List view, etc.) show on the screen. To set the attributes of these elements (e.g. The text color, background color, visibility, font, width, height, etc.).
  - We will use JAVA for coding. This page controls all the elements of XML with time. We can give default attribute values for different elements in XML, which will be used (for

that element) in Activity (app) until we change that attribute in the corresponding JAVA file for that element.

- The admin portal that will be the web portal, its front end should be designed using HML, CSS, Java Script and Backend would be in PHP.

- **Application:** Android studio 2021 (we will use android studio to write the programs to develop the app) and Sublime for admin portal designing.
- **Database:** Firebase (we will use a database to manage large amounts of data, Data security, and easy data search).
- **Operating system:** Android **(**smartphone and tablets).

In this project, we will use Google maps service for the bus tracking system.

2.4 Functional Requirements

Modules

- **Login**
  Firstly, the user will have to login to access any module.
- **Display Detail**
  Timetable, bus Tracking and Bus Info will be displayed to users.
- **Select Module**
  Users can select modules according to the type of information required by them.
- **Timetable View**
  Teachers, Students, and Admin can view the timetable at any time.
- **Timetable Updating**
  Admin can update and edit the timetable.
- **Timetable Notification**
  Messages will be conveyed to students before class that will show timing and room no of class
- **Cancel Class**
  Teacher can cancel class.
- **Reschedule Class**
  Teacher/ CR/GR can reschedule class.
- **Bus information view**
  Users can view information on bus schedules, routes, and drivers.
- **Tracking**
  Drivers can start tracking and users can view it.
- **Bus information update notification**
  A notification will be generated if any information is updated.

2.5 Non-functional Requirements

2.5.1 Performance Requirements

- UOSfiy performance will not be affected by large number of users at a time.
- UOSify provides accurate and updated information to Students.
- Due to its user-friendly interface, it is reliable for students and teachers as well.

2.5.2 Safety Requirements

In UOSify for the prevention of data loss or any other type of modification in data, Data-Backup is maintained that will provide backup in the case of any damage or data loss.

2.5.3 Security Requirements

- In UOSify no student can access the data of any other student. All data of students will be saved in the databases and only the admin can access that data and can make modifications to it.
- In UOSify ID and Password, Verification is provided to users.
- Only authorized users can access and use the system.

2.5.4 User Documentation

The proposed system will include a user manual, it will include an app overview, complete configuration of used software (such as Firebase and android studio), technical details, backup procedure, and contact information. The database will be created in Firebase.

Chapter 3

Design Documentation

3.1 Introduction

3.1.1 Purpose of Document

This document describes different aspects of our proposed system. It mainly describes the designing aspects of the system which include diagrams like system architect diagram, component diagram, class diagram etc.

3.1.2 Risks and Volatile Areas

The design structure of our system is very flexible. We can easily add a new module in the existing system. We don't need to change the full structure of a system; new module will be added easily in the existing design in future. The risks which can occur in the system are following

- System may crash.
- Any software or hardware failure.
- Any intruders in terms of hacker can affects our system.
- No connectivity with the internet.
- Database may crash.
- Server can be down.

3.2 System Architecture

3.2.1 Sub-System / Component / Module Level Architecture

- **Component Diagram:**

Components of UML System Component Diagram of UOSify are:

- Timetable Component
- Bus Schedule Component
- Login Component
- Student Component
- Teacher Component
- Admin Component
- Driver Component

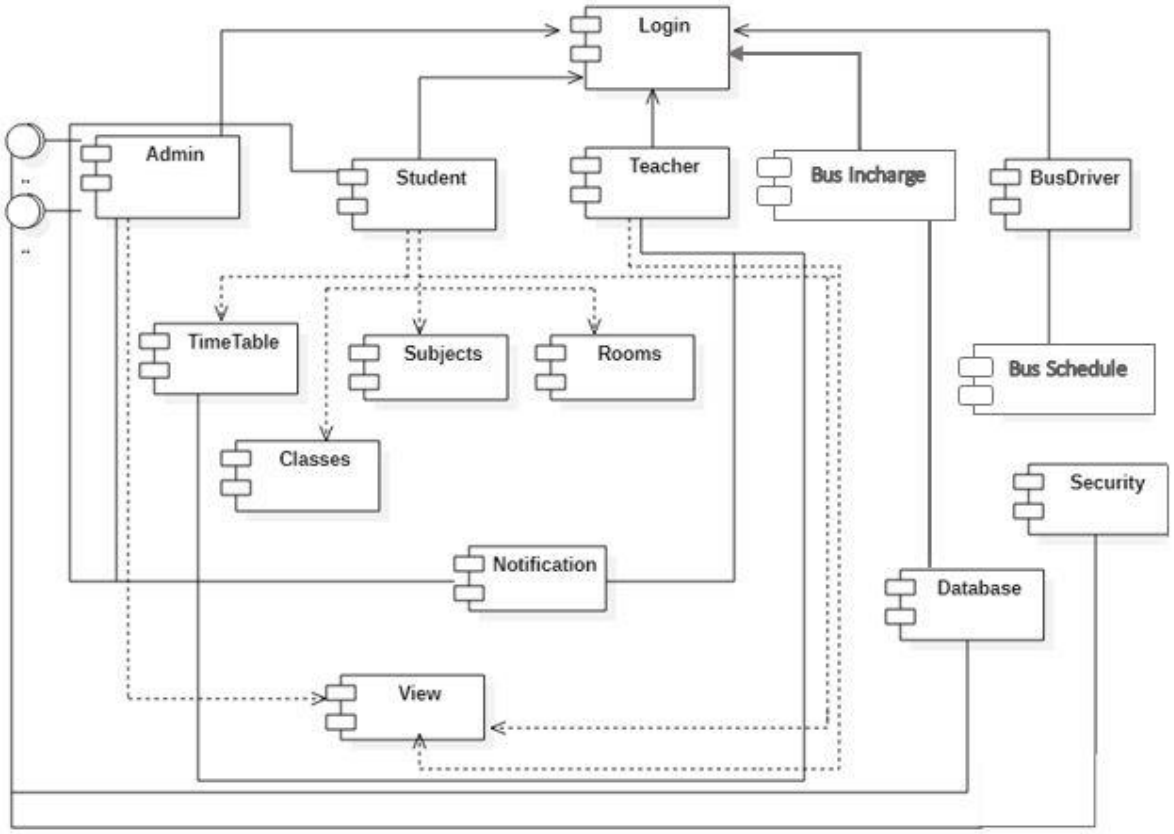


Figure 2: Component Diagram of System

3.2.2 Sub-Component / Sub-Module Level Architecture

Timetable Subcomponent:

This is the Timetable component diagram for UOSify in which main components are login, teacher, student, and admin. The mentioned users are provided with credentials to login into the system. Those credentials are already stored in the database. The users can perform or see only those functionalities defined in particular module

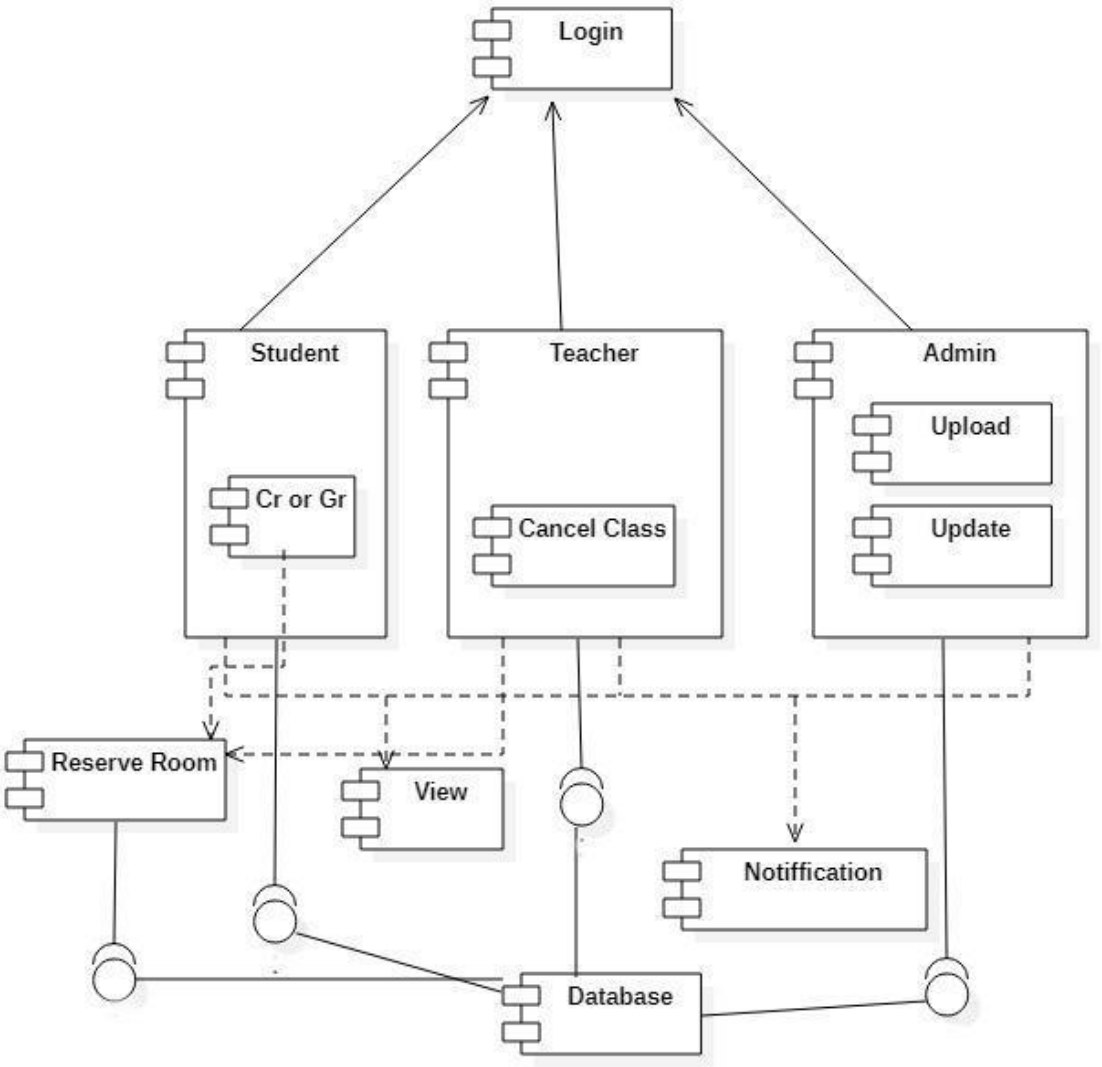


Figure 3: Component Diagram of Timetable

Bus Schedule Subcomponent:

In Bus schedule component diagrams Main components are driver, admin etc. The mentioned users are provided with credentials to login into the system. Those credentials are already stored in the database. The users can perform or see only those functionalities defined in particular module. Only admin can send notification if any of the data is updated.

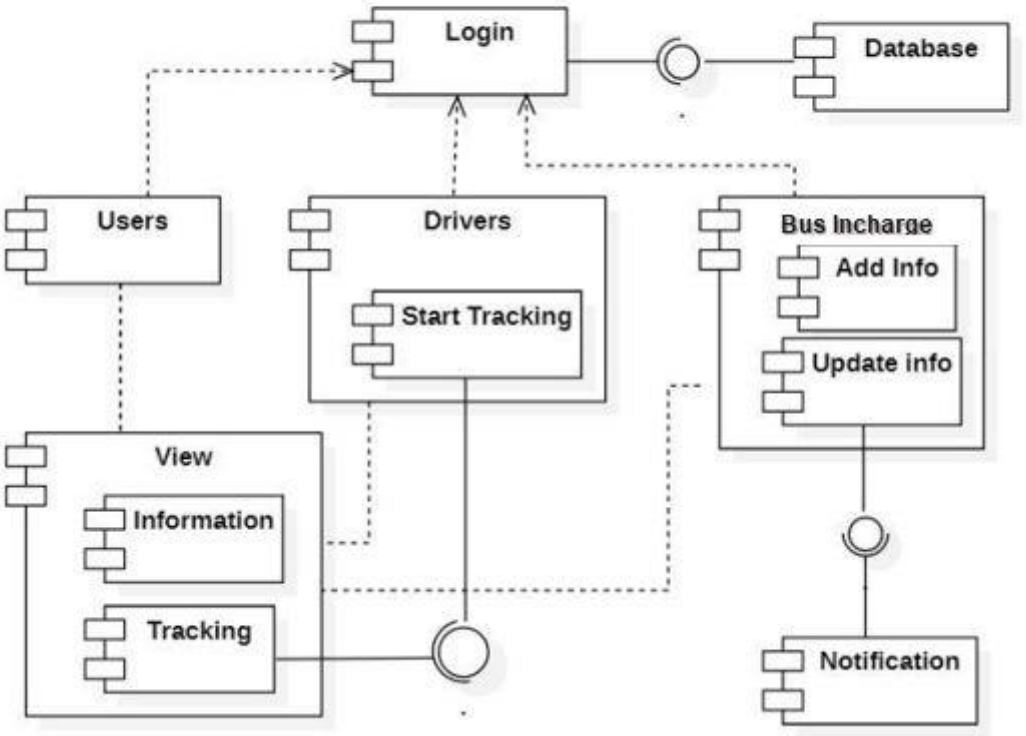


Figure 4: Component Diagram of Bus schedule module

3.3 Detailed System Design

3.3.1 Sequence Diagram

- **Admin**

This is the admin sequence diagram where:

- Admins need to login.
- After Login Admin can View Timetable, View Notification and can also Update Timetable.
- Admin can also logout from the system.

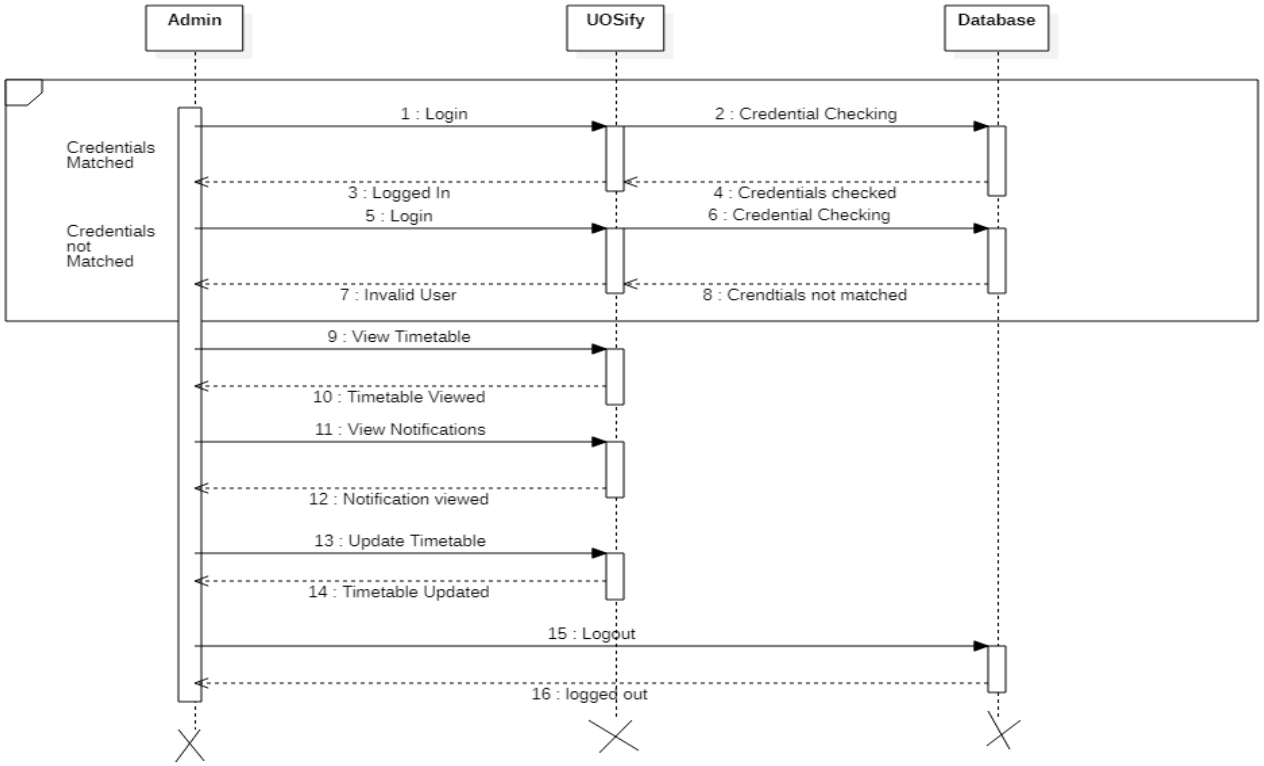


Figure 5: Sequence Diagram of Admin

- **Teacher**

This is the Teacher sequence diagram where: -

- Teachers need to login.
- Can access any module after credentials verification
- Teacher can view Timetable
- Teachers can also cancel class.
- Teachers can reschedule classes
- Teachers can View Bus info and tracking also.
- Teachers can also logout from the system

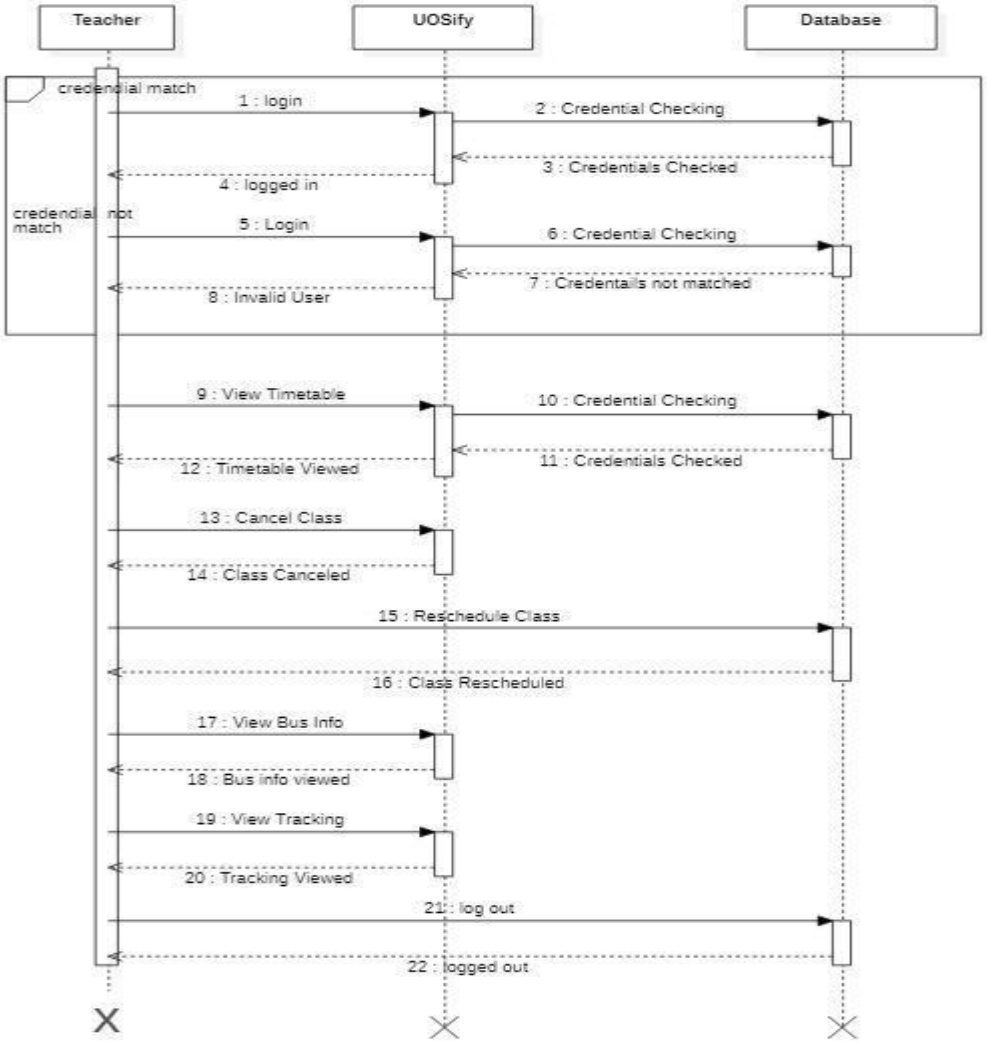


Figure 6: Sequence Diagram of Teacher

- **Student**

This is the student sequence diagram where:

- Students need to login.
- Can access any module after credentials verification
- Students can view Timetable
- Student can also view Bus tracking
- After credentials verification of CR/GR, they can reschedule a class
- Student can also logout.

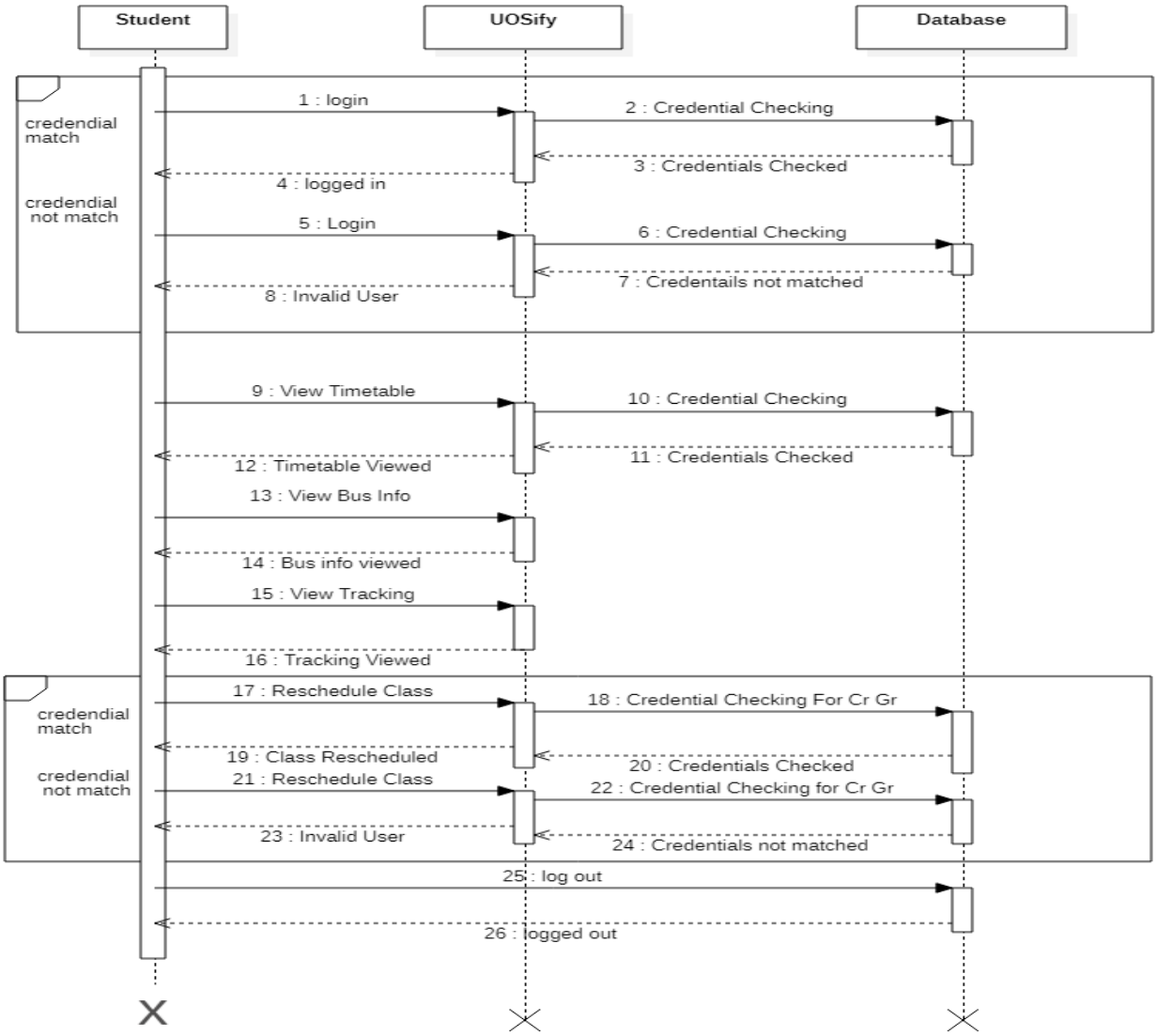


Figure 7: Sequence Diagram of Students

- **Bus In charge**

In Bus incharge sequence diagram:

- Bus incharge need to login.
- Can access any module after credentials verification
- Can update bus info, View bus info.
- Can also logout from the system.

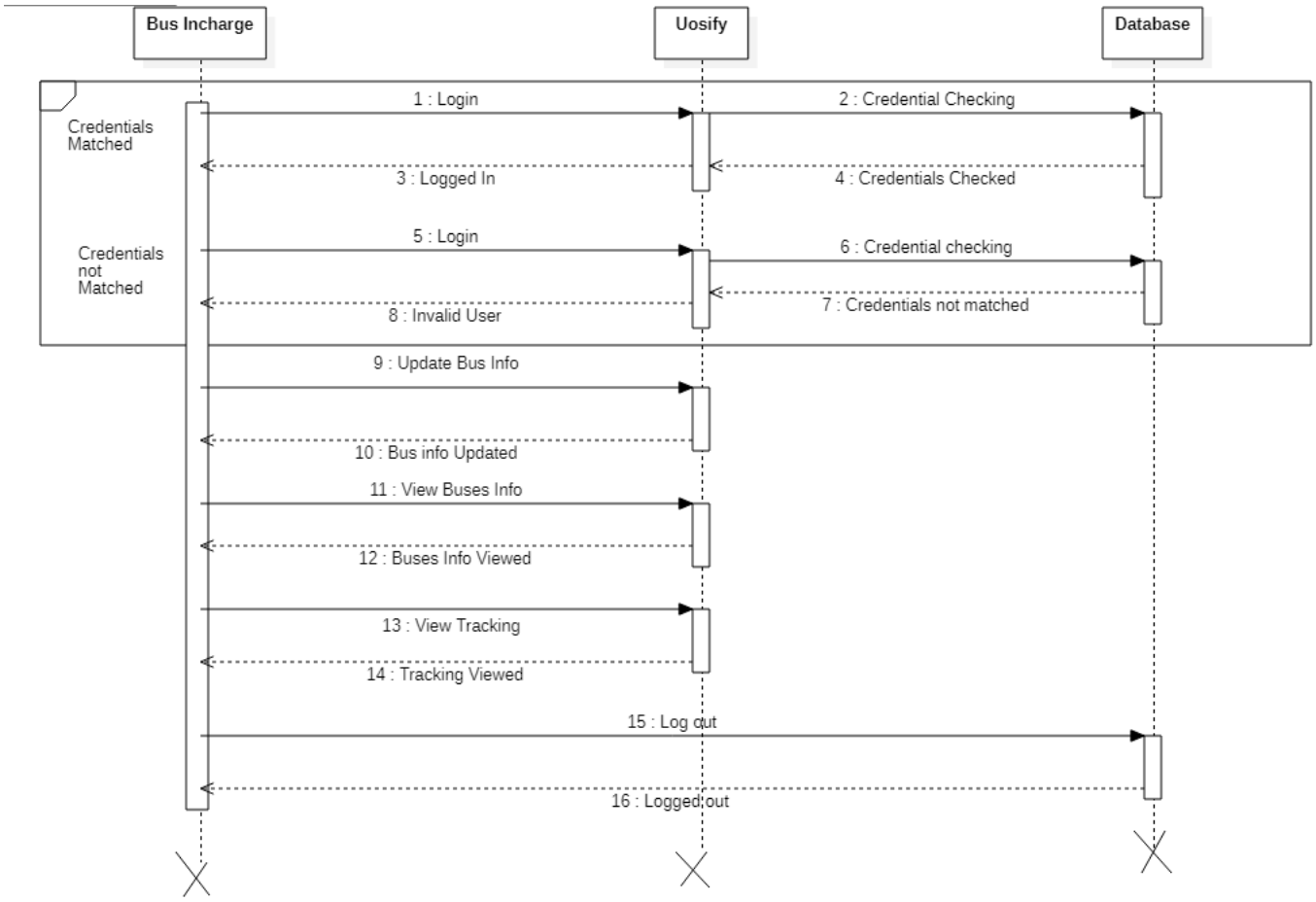


Figure 8: Sequence Diagram of Bus In charge

- **Driver**

In this diagram: -

- Drivers need to login.
- Can access any module after credentials verification
- Can start and view tracking.
- Can also logout from the system.

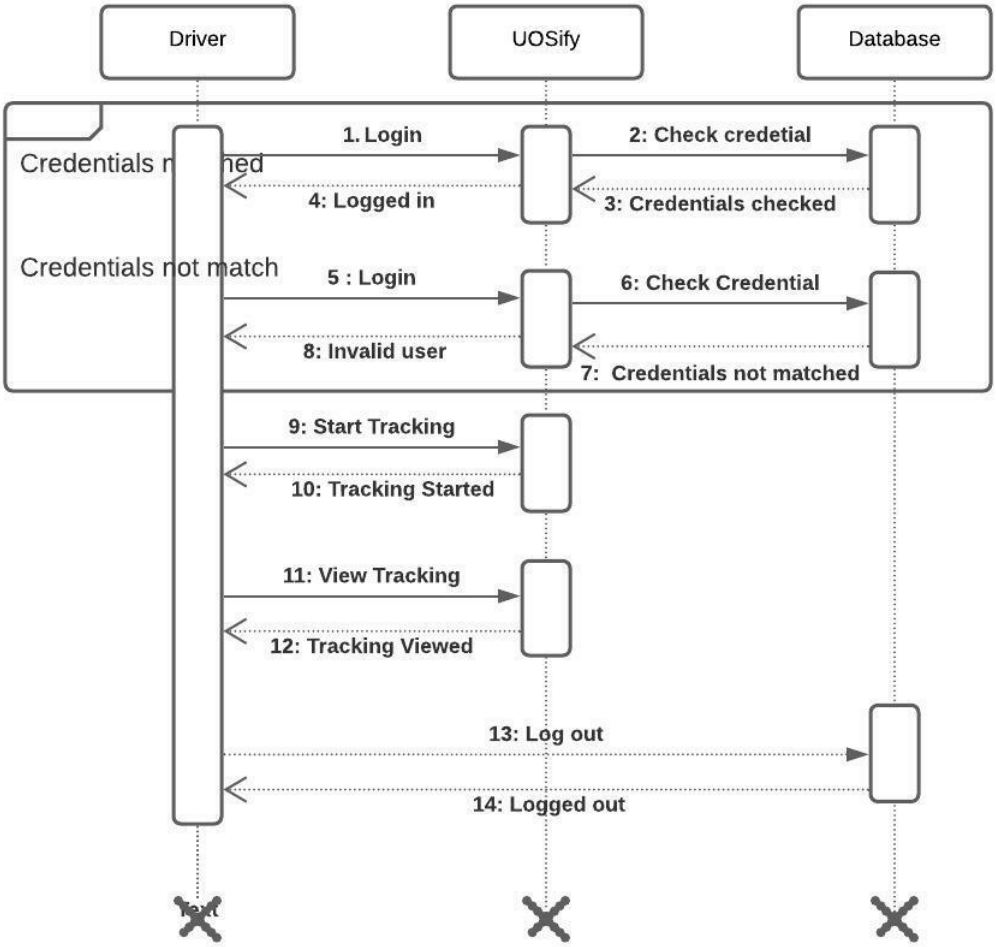


Figure 9: Sequence Diagram of Driver

3.3.2 Class Diagram

- **Classes:**
  - T**eacher Class:** Manage all the operations of teacher
  - **Student Class:** Manage all the operations of Student and sub class Cr/Gr that is inherited from student class
  - **Admin Class:** Manage all the operations of Admin
  - **Driver Class:** Manage all the operations of Driver
  - **Subject Class:** Manage all the operations of Subject
  - **Class Class:** Manage all the operations of Class
  - **Bus incharge Class:** Manage all the operations of Bus incharge
  - **Room Class:** Manage all the operations of Room

- **Associated Classes and their Attributes:**
  - **Teacher-Student Attributes:** TCNIC, SCNIC, and role
  - **Teacher-Subject Attributes:** TCNIC, Course_code, and role
  - **Subject-Student Attributes:** Course code, SCNIC
  - **Subject-Class Attributes:** Course code, Class_id

- **Classes Attributes:**
  - **Teacher Attributes:** TCNIC, TName
  - **Student Attributes:** SCNIC, SName, Roll_no
  - **Admin Attributes:** ACNIC, AName
  - **Driver Class Attributes:** DCNIC, DName, Phone_number
  - **Subject Attributes:** Course_code, Subject_title
  - **Class Attributes:** Class_id, Class_nam
  - **Bus incharge Attributes:** BCNIC, BName, Phone_number
  - **Room Attributes:** room_id, room_type and status

- **Methods of each Class:**
  - **Teacher Methods:** View_timetable (), reschedule_class (), cancel_class (), login ()
  - **Student Methods:** view_timetable (), login ()
  - **Cr/Gr Methods:** reschedule_class ()
  - **Admin Methods:** login (), update_timetable (), add_info (), view_timetable ()
  - **Driver Methods:** login (), view_timetable, start_trackig ()
  - **Bus incharge Methods:** login (), view_businfo (), update_businfo (), view_tracking ()
  - **Room Methods:** check_freeRoom ()

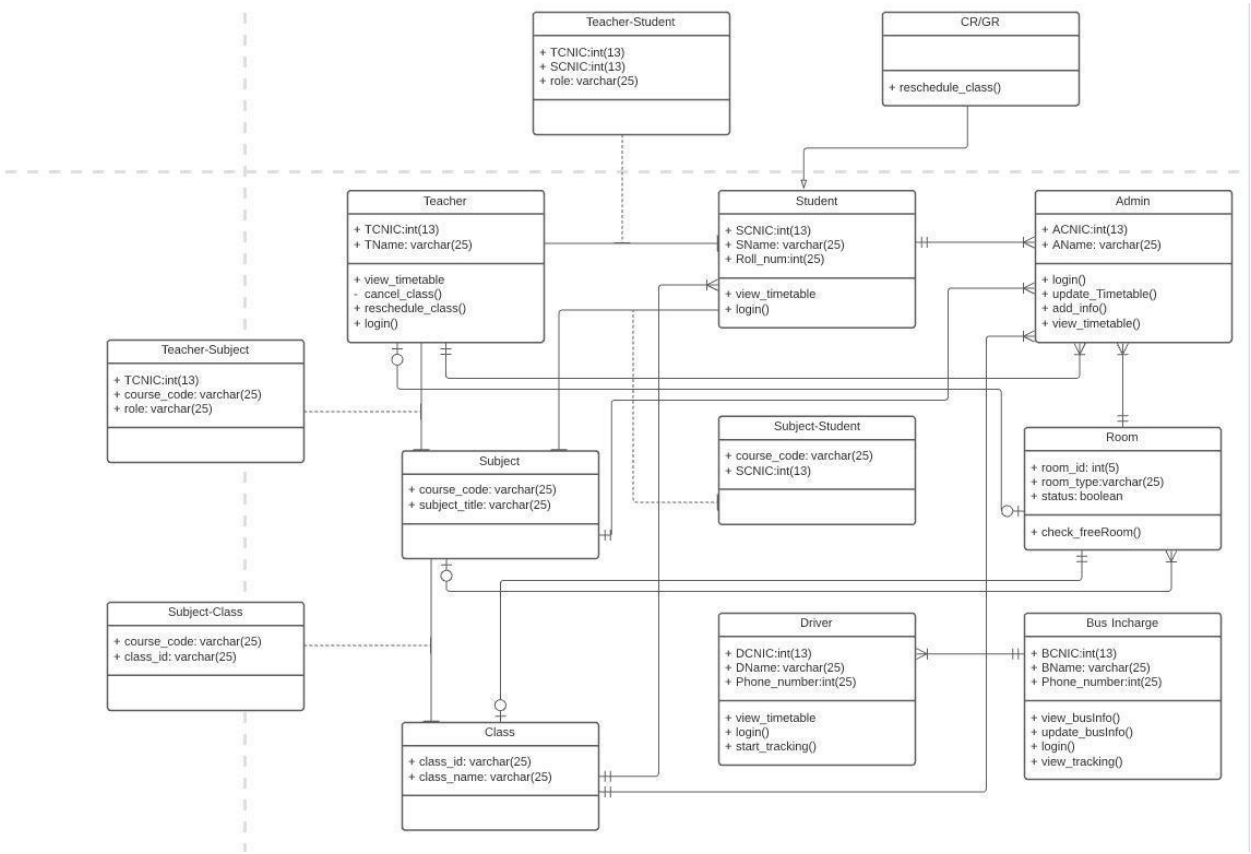


Figure 10: Class Diagram of System

Chapter 4

Functional Requirements

4.1 System Architecture

4.1.1 Sub level / Architecture

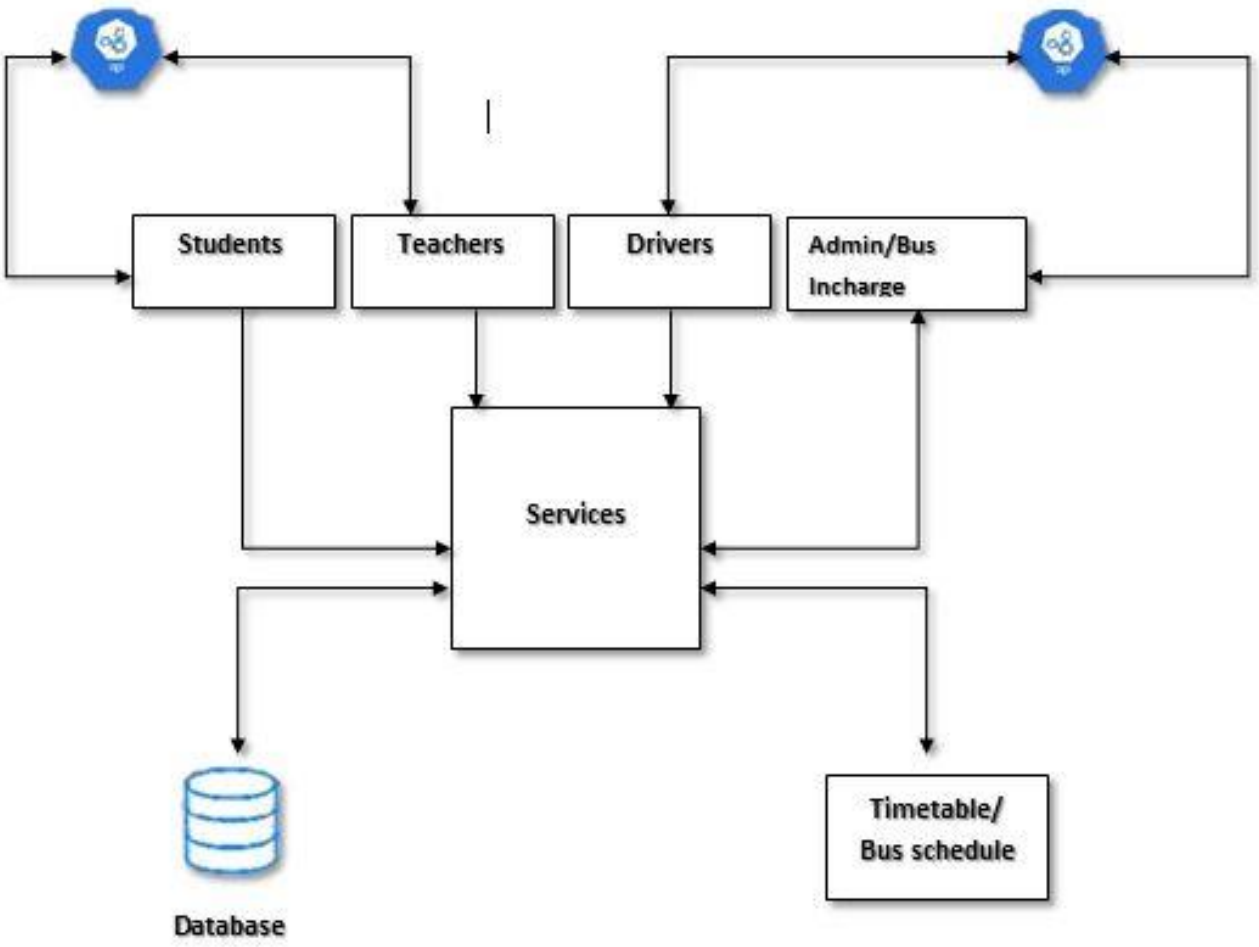
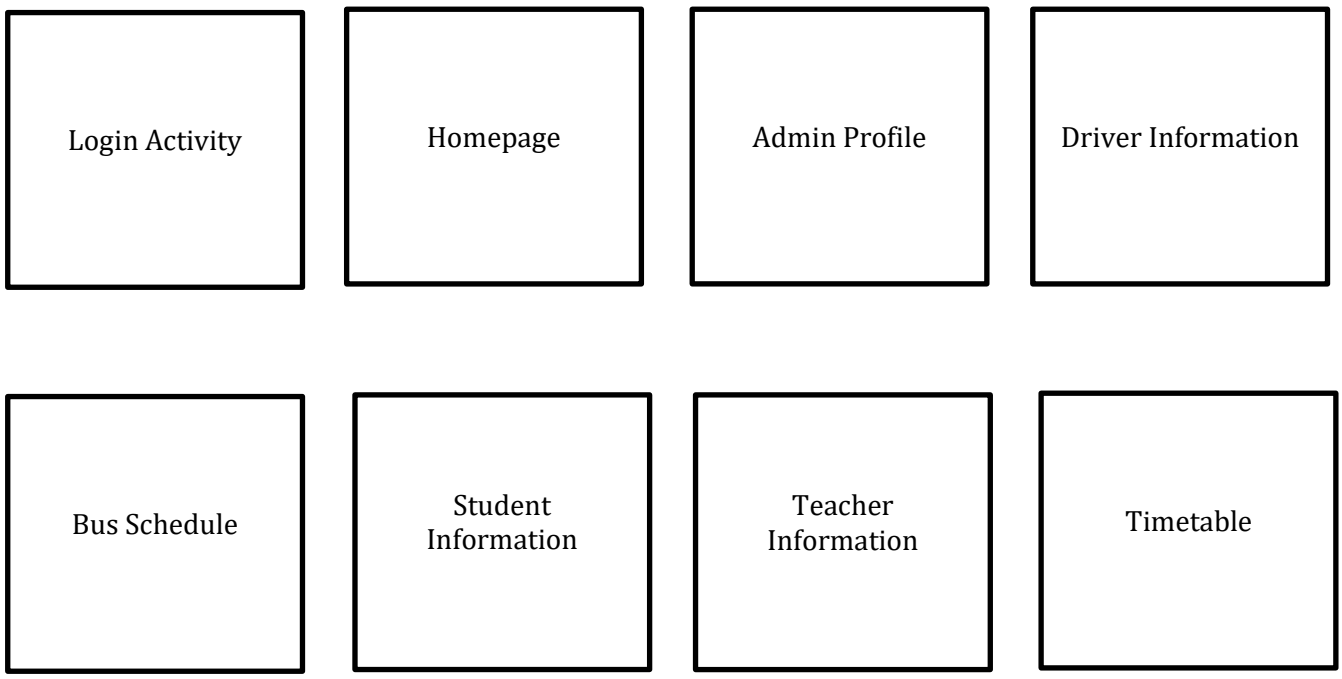


Figure 11: System Architecture Diagram of System

4.1.2 Sub-System / Component / Module Level Architecture



4.1.3 Data Flow Diagram

- **High Level Entities and Process Flow of DFD for UOSify:**
  - Managing all the Timetable
  - Managing all the Bus schedule
  - Managing all the Admin
  - Managing all the Students
  - Managing all the Teachers
  - Managing all the Classes
  - Managing all the Drivers
  - Managing all the Bus Tracking

- **Low Level System Entities of DFD for UOSify:**
  - Admin logins to the system and manage all the updating of Timetable and Bus info
  - Teacher logins to the system and view all the Timetable and Bus info
  - Teacher can reschedule and cancel the class
  - Student logins to the system and view all the Timetable and Bus info
  - Student can also view Tracking
  - Driver logins to the system and start the Bus Tracking
  - Driver can also View all the Bus Tracking

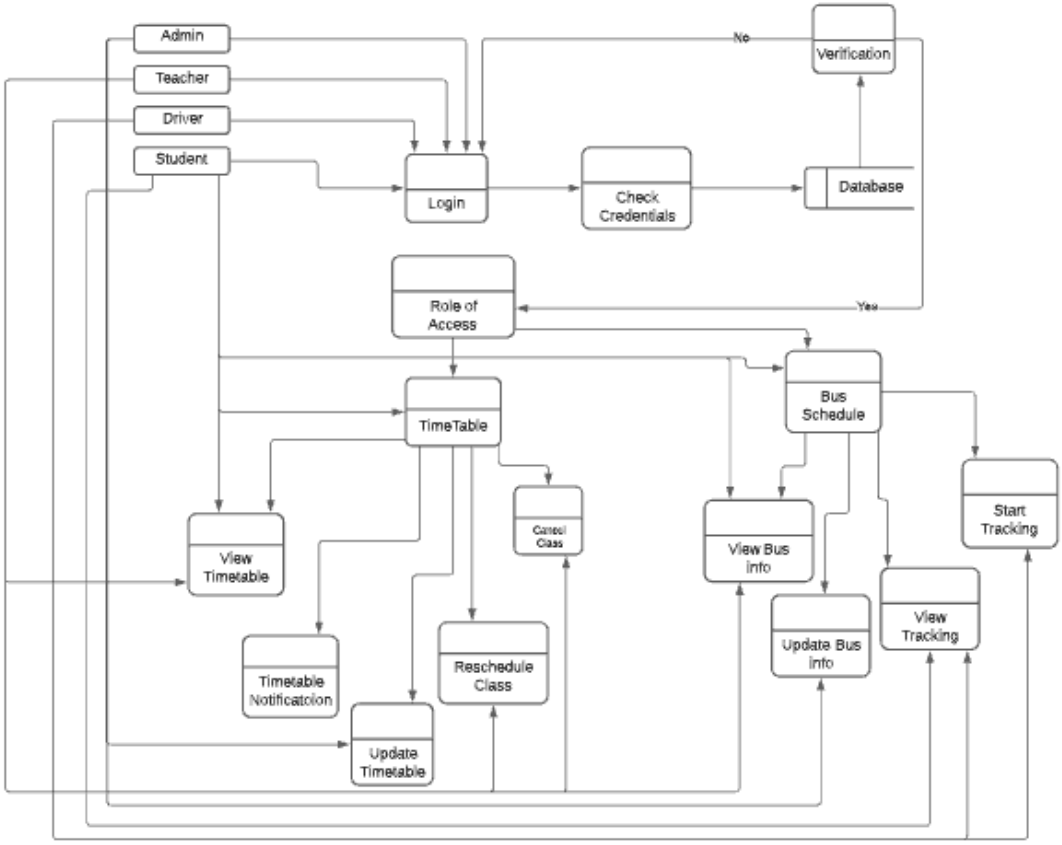


Figure 12 Dataflow Diagram of System

4.2 Use Cases

4.2.1 Use case Diagram

The main actors of UOSify in the below diagram are student, teacher, bus driver, admin, and bus incharge. All the actors need login into the system and can perform their respective functionality after credentials verification that are already store in the database.

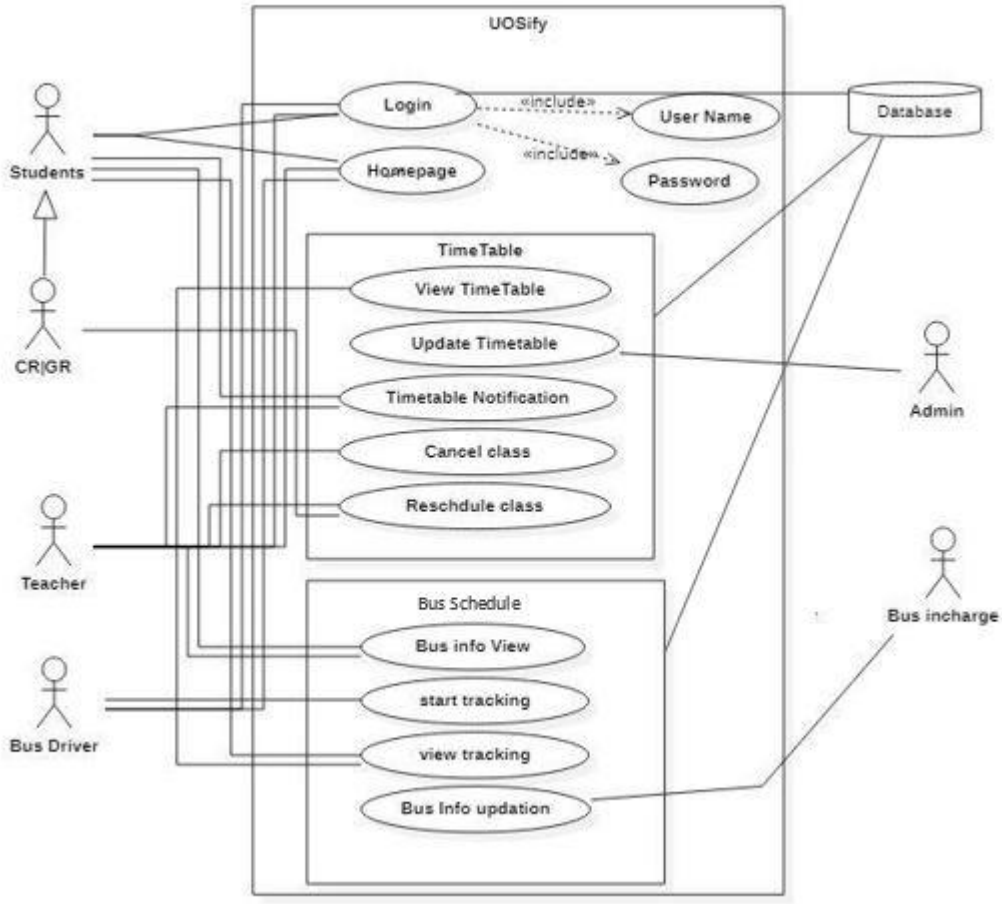


Figure 13 Use case Diagram

4.2.2 Use Case Description

Each Use Case has a description, which describes the functionality that will be built in the proposed system. The template for Use Case description is given below:

1: User Login		
Actors: Admin, Teacher, Student, Driver, Bus Incharge		
Feature: Login		
Use case Id:	1	
Pre-condition:	You must have a running app and strong internet connection.	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	Open login page	Render registration page
2.	Fill the form	Check User in database
3.	Submit form	Show submission status
Alternate Scenarios: **1a:** There is an error in data submission **2a:** System not working properly		
Post Conditions		
Step #	**Description**	
1.	User logged in	
2.	System allows their roles, i.e., Admin, teacher, student, driver	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	Figure 2 GUI 1.2	

2: View Timetable		
Actors: Teacher, Student		
Feature: view timetable		
Use case Id:	2	
Pre-condition:	1. Teachers and students must be logged in according to their roles. 2. Timetable module must be opened.	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	Select Timetable module	Open Timetable module
2.	Select day.	Selected Day Frame will open.
3.	View timetable.	Show Timetable
Alternate Scenarios: **1.** Error in Viewing Timetable. 2. System not working properly.		
Post Conditions		
Step #	**Description**	
1.	User Timetable will be shown	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	Figure 3 GUI 1.5	
Concurrency and Response Give an estimate of the following ♦ Actors: Teacher and student ♦ Expected response time is 10s.		

3: Update Timetable		
Actors: Admin		
Feature: Timetable information update		
Use case Id:	3	
Pre-condition:	1. Admin must have access to update timetable. 2.Admin must have data to make timetable	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	Login as admin	Check User in database
2.	Click on update timetable	Open update interface
3.	Edit any information	Information will be updated
Alternate Scenarios:		
1a: There is an error in updating submission **2a:** System not working properly		
Post Conditions		
Step #	**Description**	
1.	Timetable updated	
2.	Notification of updated class will be shown	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	GUI 4.3, GUI 4.4, GUI 4.5, GUI 4.6	
Concurrency and Response Give an estimate of the following ♦ Actor: Admin ♦ Expected response time of the use case is 5s.		

4: Timetable Notification		
Actors: Student		
Feature: Timetable Notification		
Use case Id:	5	
Pre-condition:	1. Students must be logged in according to their roles. 2. Timetable module must be opened.	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	Select Timetable module	Open Timetable module.
2.		Check User in database
3.	Submit form	Show submission status
Alternate Scenarios: **1a:** There may be delay in receiving notification. **2a:** System not working properly		
Post Conditions		
Step #	**Description**	
1.	Timetable notification received.	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	Figure 4 GUI 1.5	
Concurrency and Response *Give an estimate of the following* ♦ *Actor: students* ♦ *Expected response time 10s.*		

5: Cancel class		
Actors: Teachers.		
Feature: Cancel Class		
Use case Id:	6	
Pre-condition:	1. Teachers and students must be logged in according to their roles. 2. Timetable module must be opened.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select the Timetable module.	Open Timetable module.
2.	Select class.	Class selected.
3.	select cancel class	Class canceled.
Alternate Scenarios: **1a:** Class may not be canceled due to weak internet connection **2a:** System not working properly		
Post Conditions		
Step#	**Description**	
1.	Class will be canceled	
2.	Show notification of cancel class	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	Figure 5 GUI 2.6	
Concurrency and Response *Give an estimate of the following* ♦ *Actor: Teacher* ♦ *Expected response time of the use case is 5s.*		

6: Reschedule class		
Actors: Teachers, CR/GR		
Feature: Reschedule class		
Use case Id:	7	
Pre-condition:	1. Teachers and students must be logged in according to their roles. 2. Timetable module must be opened.	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Select the Timetable module.	Open Timetable module.
2.	Select class.	Class selected.
3.	select Reschedule class	Class Rescheduled.
Alternate Scenarios:		
1a: Class may not be Rescheduled due to weak internet connection **2a:** System not working properly		
Post Conditions		
Step#	**Description**	
1.	Class will be Rescheduled	
2.	Show notification of Reschedule class	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	Figure 6 GUI 2.6, Figure 7 GUI 1.6	
Concurrency and Response *Give an estimate of the following* ♦ *Actor: Teacher, CR/GR* ♦ *Expected response time of the use case is 5s.*		

7. Bus Info View		
Actors: Student		
Feature: Bus Info View		
Use case Id:	8	
Pre-condition:	1 Students must be logged in according to their roles. 2. Bus Info view module must be opened.	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	Select Bus info view module	Open Bus Info module
2.	Select Bus number.	Bus number will be selected
3.	View bus Info	Bus info will be opened.
Alternate Scenarios:		
1. Error in Viewing Bus Info. 2. System not working properly.		
Post Conditions		
Step #	**Description**	
1.	Bus info is shown.	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	Figure 17 GUI 3.4, Figure GUI 3.5	
Concurrency and Response Give an estimate of the following Actor: Students Expected Response time 10s.		

8: Tracking		
Actors: Bus Incharge, Teacher, Student, Driver		
Feature: Tracking of busses		
Use case Id:	9	
Pre-condition:	Driver must start tracking	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	Start tracking	System will start tracking of bus using Google Map.
2.	View tracking	Show tracking
Alternate Scenarios		
1a: There is an error in Starting tracking **2a:** System not working properly		
Post Conditions		
Step #	**Description**	
1.	Tracking will start	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	Figure 17 GUI 3.4, Figure 18 GUI 3.5	
Concurrency and Response *Give an estimate of the following* ♦ Actors: Bus Incharge, Teacher, Student, Driver ♦ Expected response time of the use case is 5s.		

9: Update Bus info		
Actors: Bus Incharge		
Feature: Login		
Use case Id:	10	
Pre-condition:	1. Bus incharge must be logged in with his role access. 2.Bus incharge must have data to update Information	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	Login as bus incharge	Check User in database
2.	Click on update bus info	Open update interface
3.	Edit any information	Information will be updated
Alternate Scenarios		
1a: There is an error in Updation. **2a:** System not working properly		
Post Conditions		
Step #	**Description**	
1.	Information updated	
2.	Show notification of updated information	
Use Case Cross referenced	Figure 13 Use case Diagram	
User Interface reference	GUI 4.8, GUI 4.9	

4.3 Graphical User Interfaces

1. Students' interfaces for Timetable (App based)

Splash Screen for Students	
Interface Id.	1.1
Use case Reference	Figure 13 Use case Diagram
Snapshot	

FIGURE 2 GUI 1.1

Description

When App will Start Splash Screen will be shown to Students.

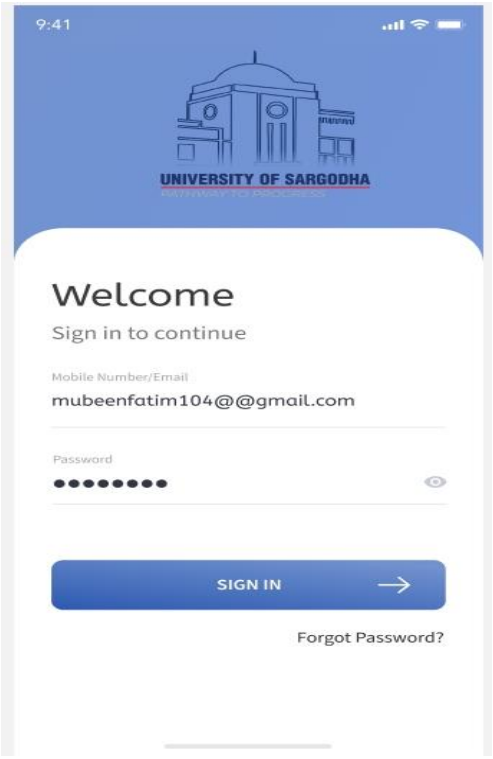
Login for Students	
Interface Id.	1.2
Use case Reference	Figure 13 Use case Diagram
Snapshot	

FIGURE 7 GUI 1.2

Description

After Splash Screen Login Screen Will Display to Students. They can Login in App using Email and Password.

Reset password for Students	
Interface Id.	1.3
Use case Reference	Figure 13 Use case Diagram

Snapshot

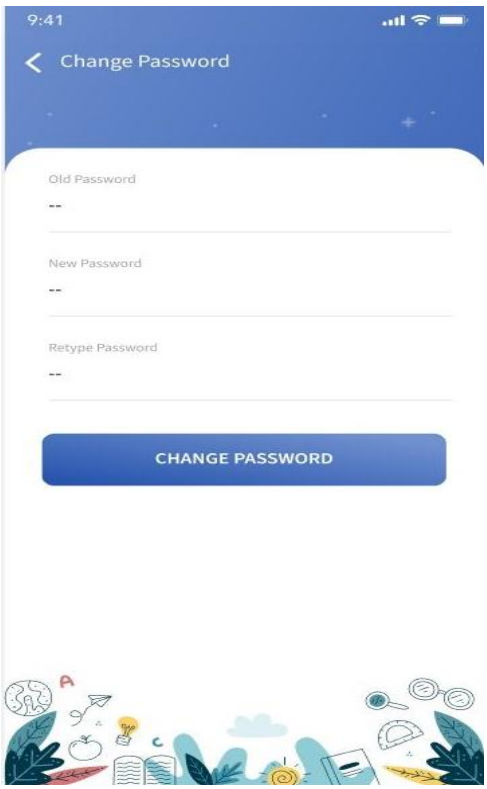


FIGURE 8 GUI 1.3

Description

When students will Login, they can reset their Password.

Modules Interface for Students	
Interface Id.	1.4
Use case Reference	Figure 13 Use case Diagram

Snapshot

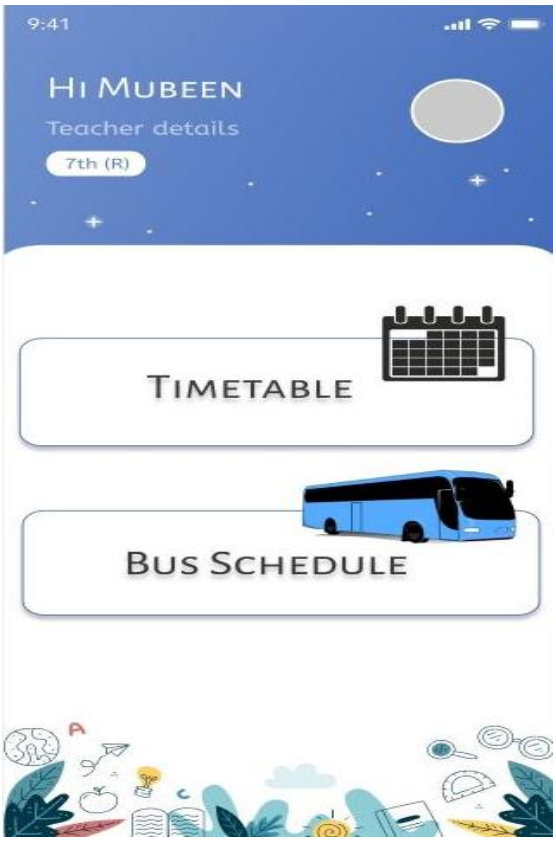


FIGURE 9 GUI 1.4

Description

When Student successfully login, module interface will display to students.

Student can view timetable	
Interface Id.	1.5
Use case Reference	Figure 13 Use case Diagram

Snapshot

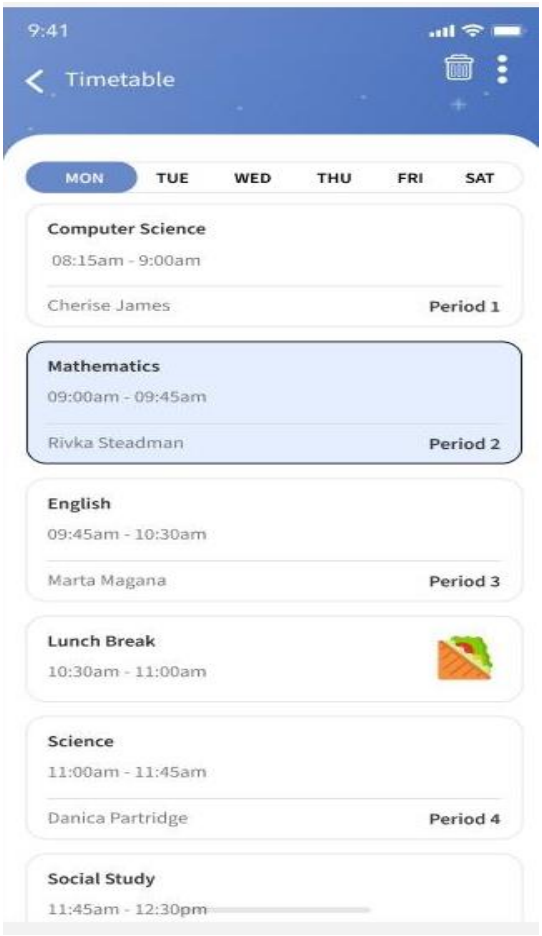


FIGURE 10 GUI 1.5

Description

When Students will Open Timetable module, they can view only their timetable of any day of week.

Reschedule classes Interface.	
Interface Id.	1.6
Use case Reference	Figure 13 Use case Diagram

Snapshot

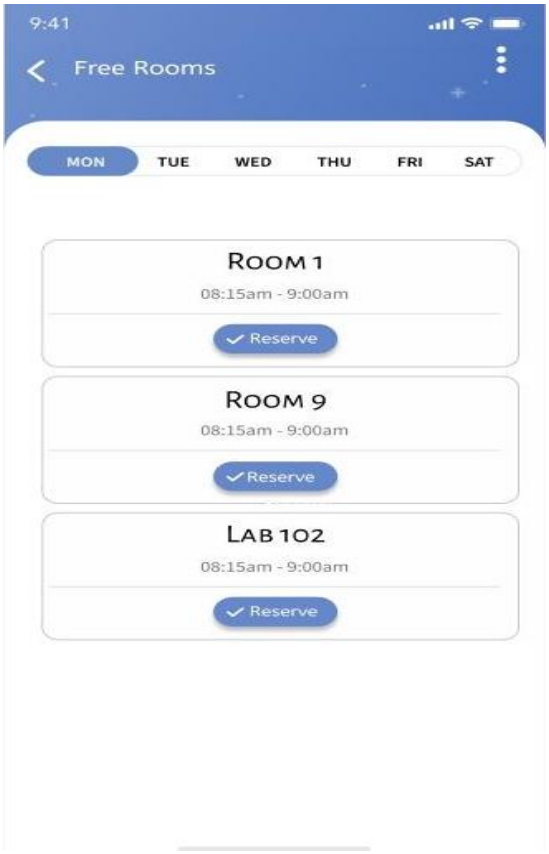


FIGURE 11 GUI 1.6

Description

CR/GR can view free rooms and can arrange classes.

2. Teacher interfaces for Timetable (App based)

Splash Screen for Teachers	
Interface Id.	2.1
Use case Reference	Figure 13 Use case Diagram
Snapshot	

FIGURE 12 GUI 2.1

Description

When App will Start Splash Screen will be shown to Teachers.

Login for Teachers	
Interface Id.	2.2
Use case Reference	Figure 13 Use case Diagram

Snapshot

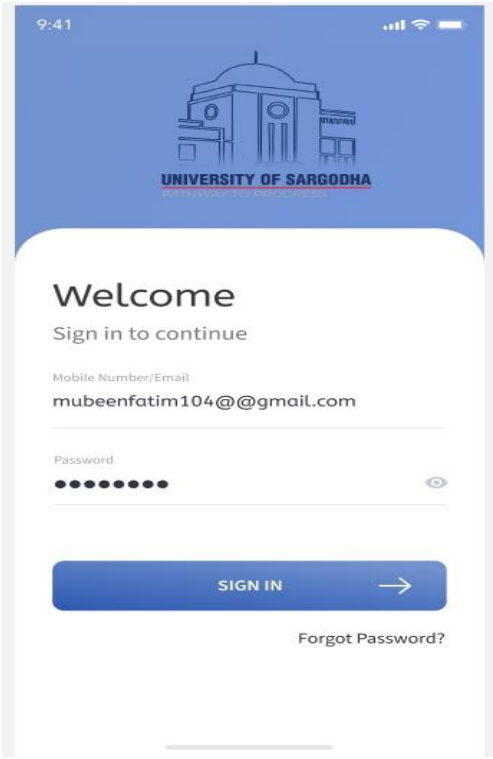


FIGURE 13 GUI 2.2

Description

After Splash Screen Login Screen Will Display to Teachers. They can Login in App using Email and Password.

Reset password for Teachers	
Interface Id.	2.3
Use case Reference	Figure 13 Use case Diagram

Snapshot

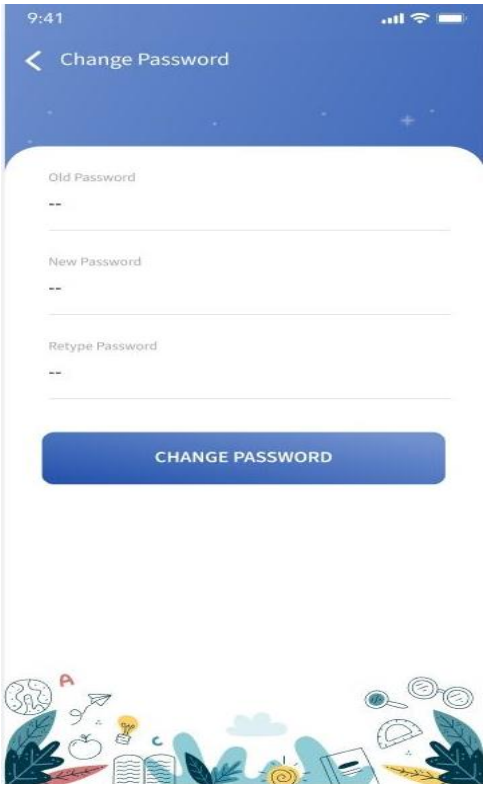


FIGURE 14 GUI 2.3

Description

When Teachers will Login, they can reset their Password.

Modules Interface for Students	
Interface Id.	1.4
Use case Reference	Figure 13 Use case Diagram

Snapshot

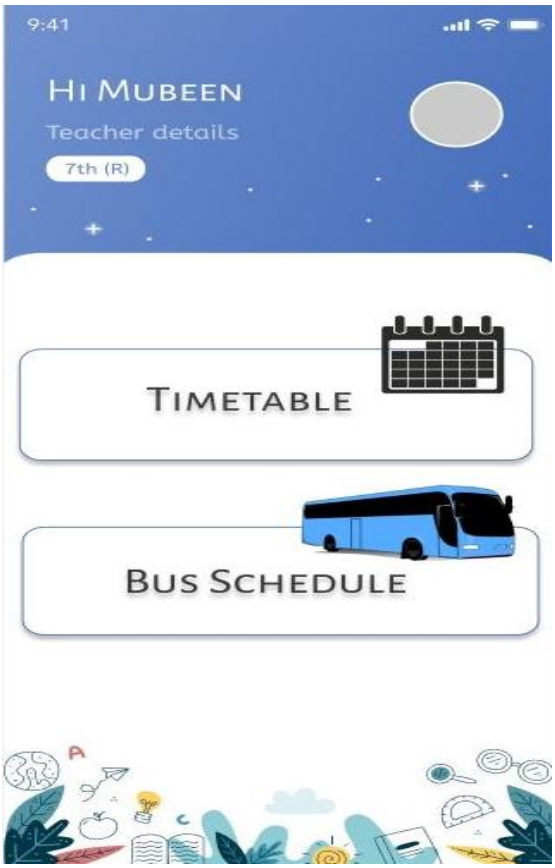


FIGURE 15 GUI 2.4

Description

When Teachers successfully login, module interface will display to Them.

Teachers can view timetable	
Interface Id.	2.5
Use case Reference	Figure 13 Use case Diagram

Snapshot

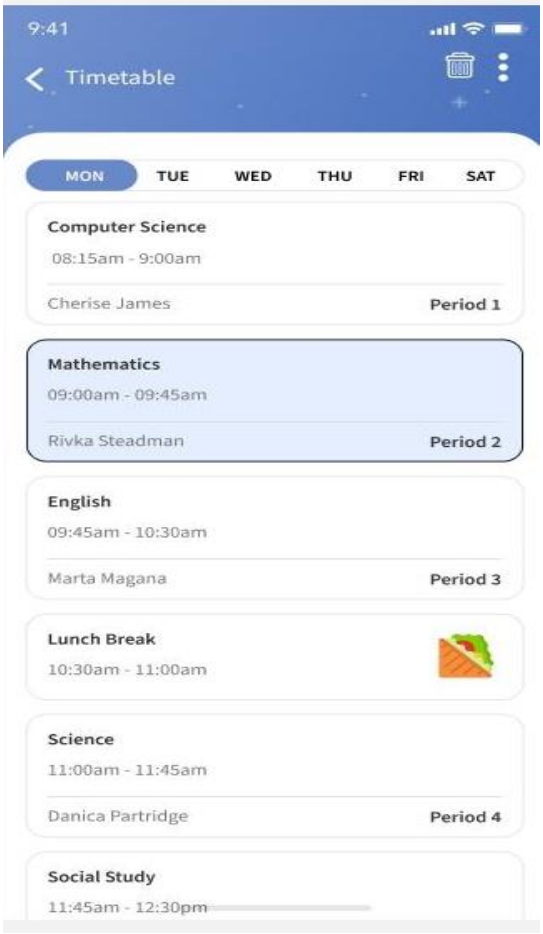


FIGURE 16 GUI 2.5

Description

When Teachers will Open Timetable module, they can view only their timetable of any day of week.

Student can view timetable	
Interface Id.	2.6
Use case Reference	Figure 13 Use case Diagram

Snapshot

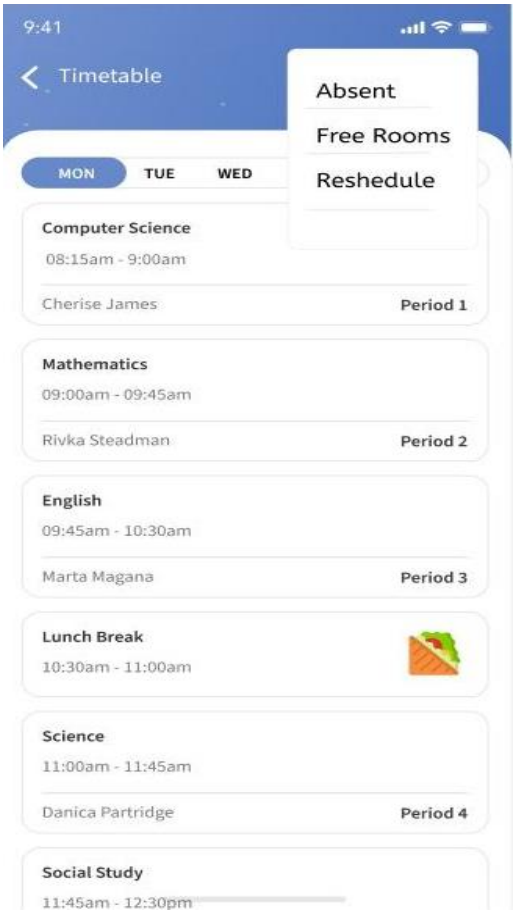


FIGURE 17 GUI 2.6

Description

Teachers can cancel and Reschedule Class form Drop down menu.

3. Driver interfaces for Bus Schedule (App based)

Splash Screen for Drivers	
Interface Id.	3.1
Use case Reference	Figure 13 Use case Diagram

Snapshot

FIGURE 18 GUI 3.1

Description

When App will Start Splash Screen will be shown to Drivers.

Login for Drivers	
Interface Id.	3.2
Use case Reference	Figure 13 Use case Diagram

Snapshot

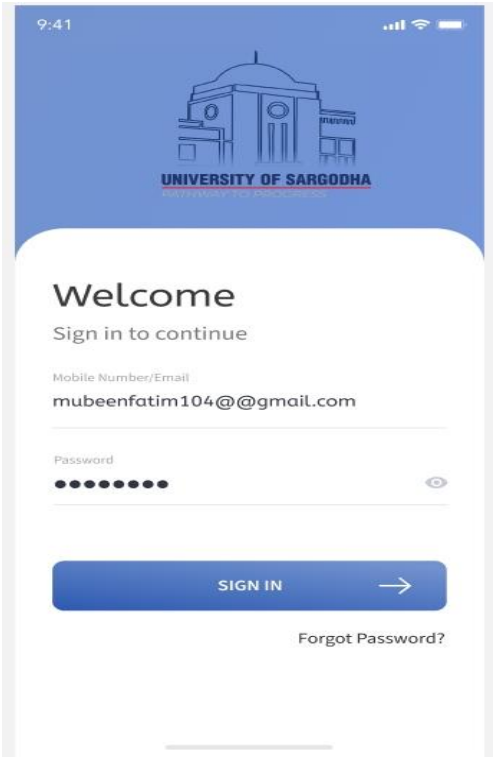


FIGURE 19 GUI 3.2

Description

After Splash Screen Login Screen Will Display to Drivers. They can Login in App using Email and Password.

Reset password for Drivers	
Interface Id.	3.3
Use case Reference	Figure 13 Use case Diagram

Snapshot

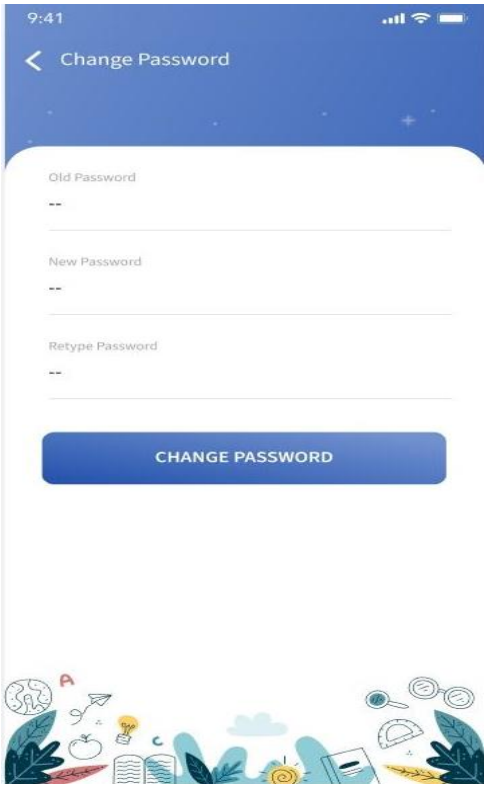


FIGURE 20 GUI 3.3

Description

When Drivers will Login, they can reset their Password.

Driver Interface for Bus Schedule	
Interface Id.	3.4
Use case Reference	Figure 13 Use case Diagram

Snapshot

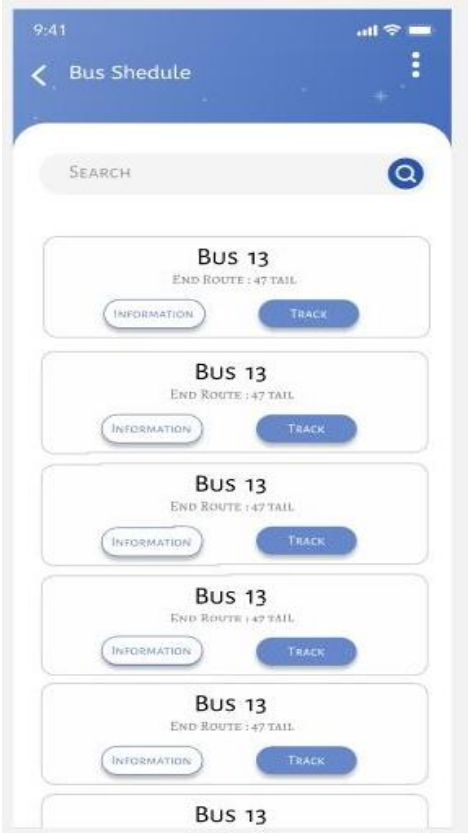


FIGURE 21 GUI 3.4

Description

When Driver will login, Bus Schedule interface will be display to Drivers. Driver can start tracking by pressing track button. Driver can also view information about buses.

Teachers and Students Interface for Bus Schedule	
Interface Id.	3.5
Use case Reference	Figure 13 Use case Diagram

Snapshot

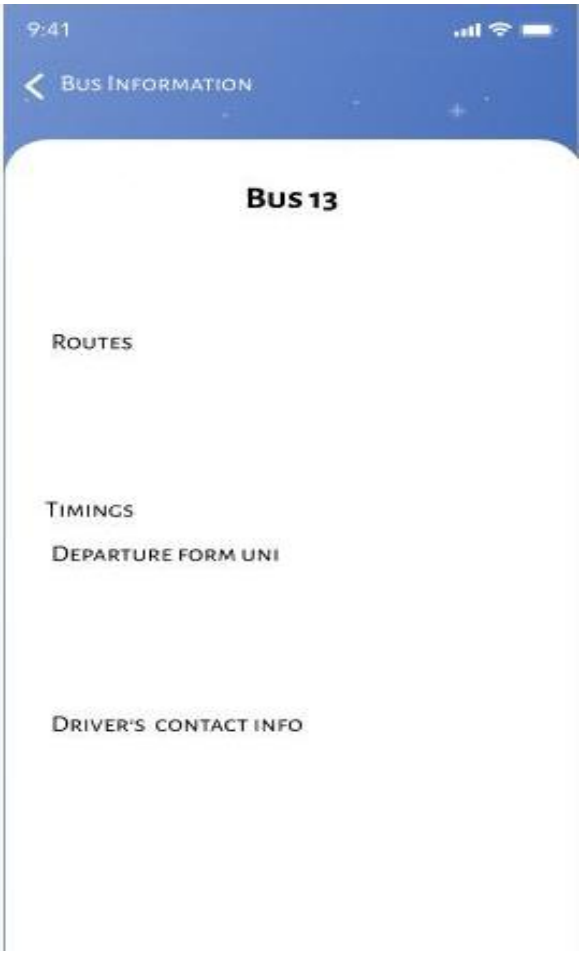


FIGURE 18 GUI 3.5

Description

When Teachers and students click on Bus Schedule module, they can view buses, their routes, Timings Drivers Contact Info etc.

4. Admin Interface for Timetable Module (Web based)

Admin Login.	
Interface Id.	4.1
Use case Reference	Figure 13 Use case Diagram

Snapshot

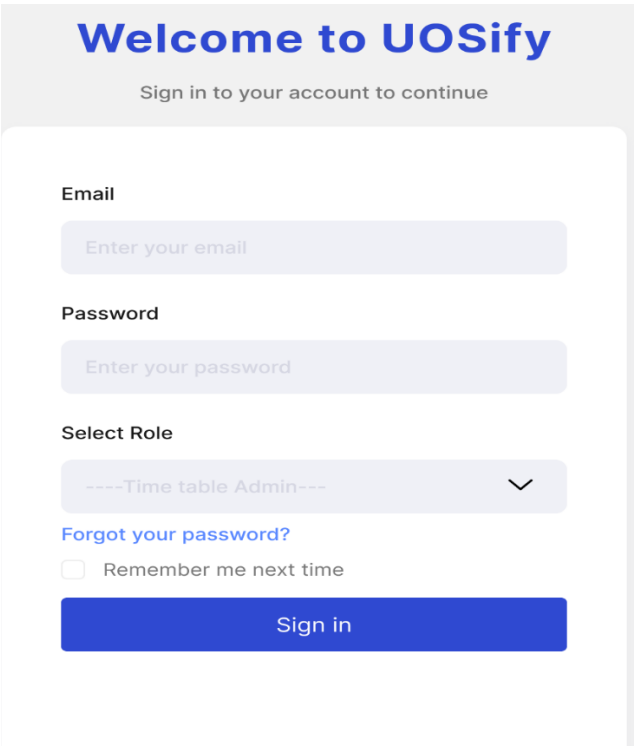


FIGURE 19 GUI 4.1

Description

Admin will login through email and password.

Admin Dashboard	
Interface Id.	4.2
Use case Reference	Figure 13 Use case Diagram

Snapshot

FIGURE 220 GUI 4.2

Description

When Admin Successfully login, Admin dashboard will appear.

Timetable view	
Interface Id.	4.3
Use case Reference	Figure 13 Use case Diagram

Snapshot

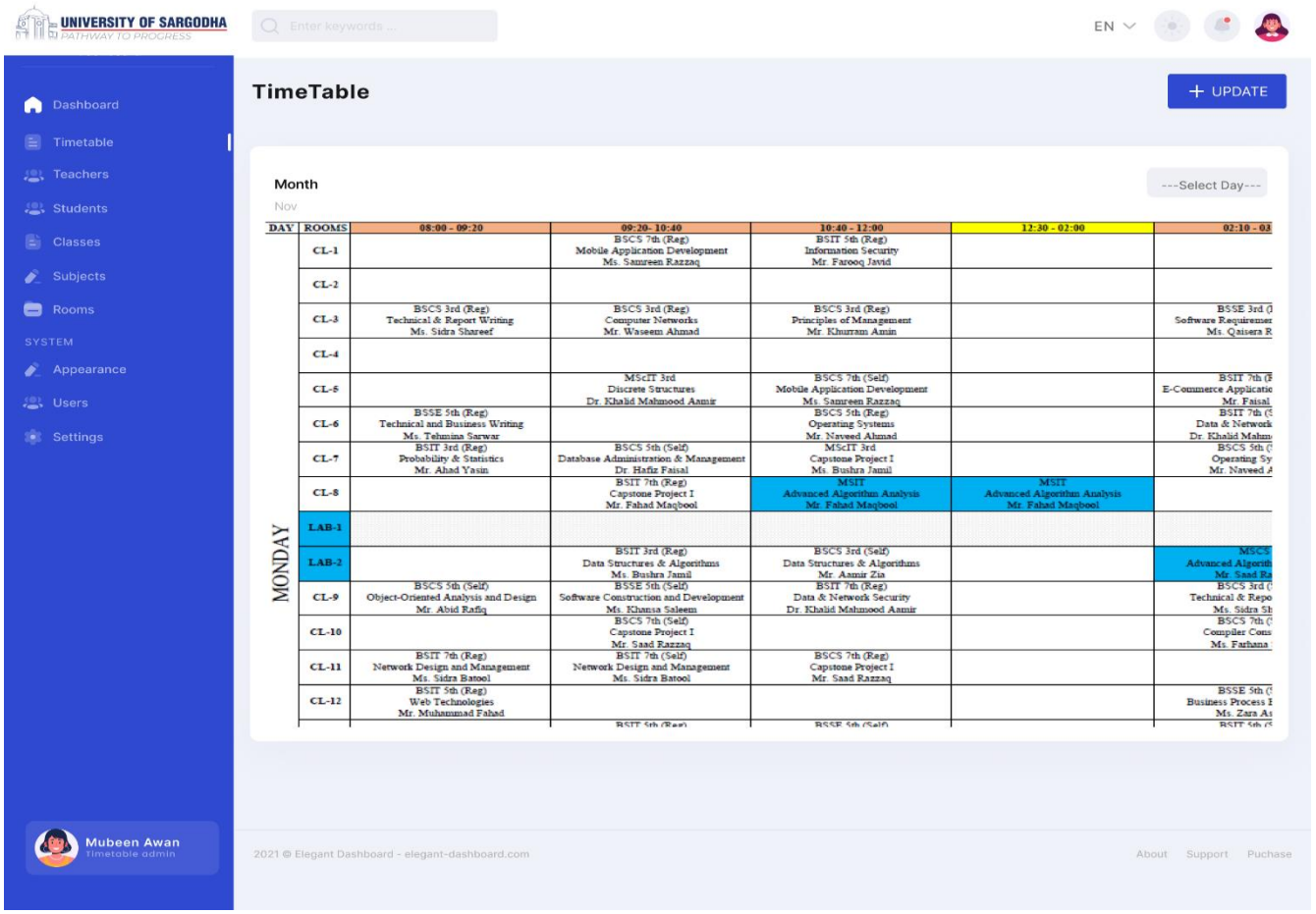


FIGURE 231 GUI 4.3

Description

Admin will click on timetable; Timetable dashboard will appear. Admin can view and assign time, room no, teacher and subject to a particular class.

Teachers' information	
Interface Id.	4.4
Use case Reference	Figure 13 Use case Diagram

Snapshot

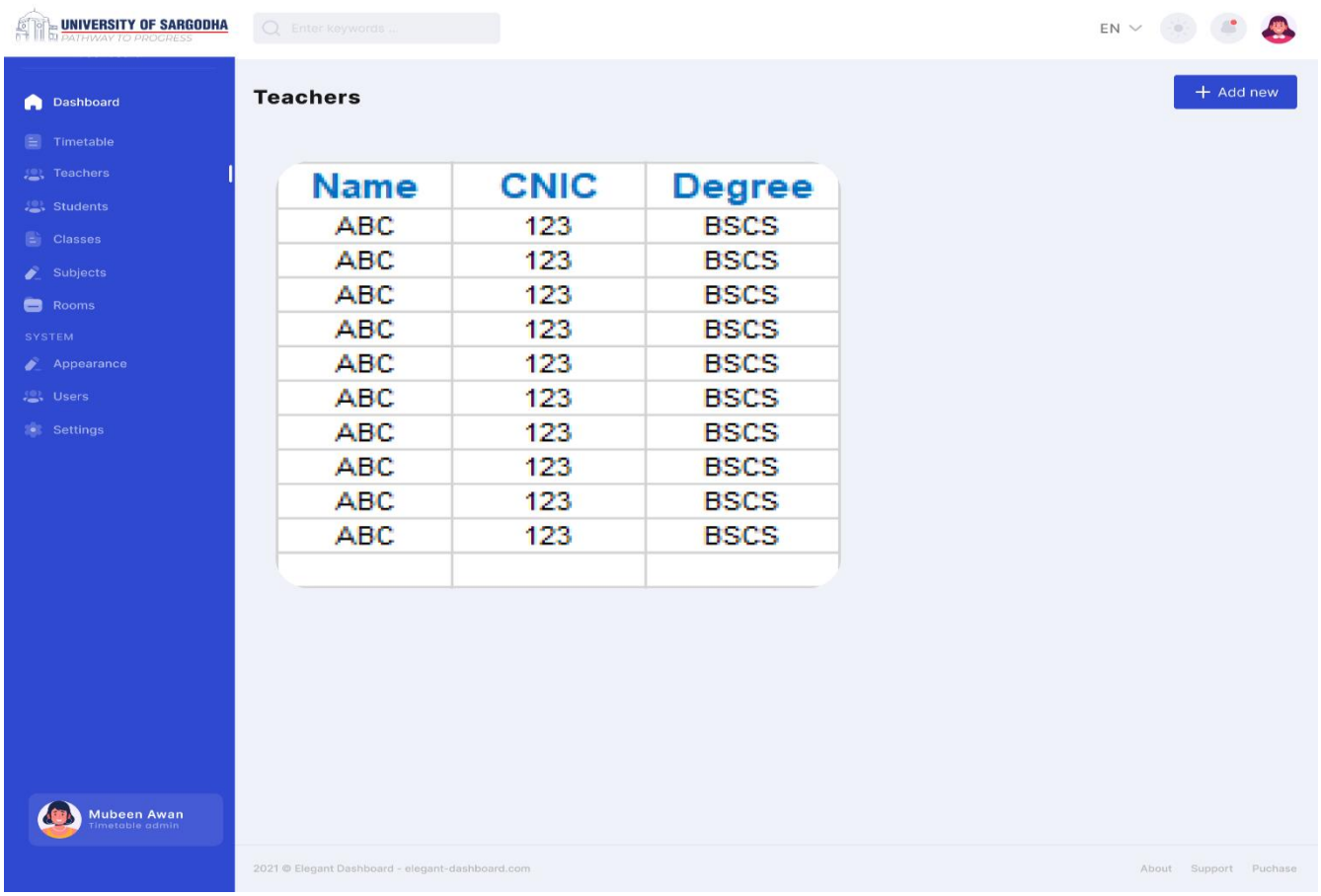


FIGURE 242 GUI 4.4

Description

Admin can view which teacher are available for a particular subject and can add teachers to assign subjects to them.

Subject Information	
Interface Id.	4.5
Use case Reference	Figure 13 Use case Diagram

Snapshot

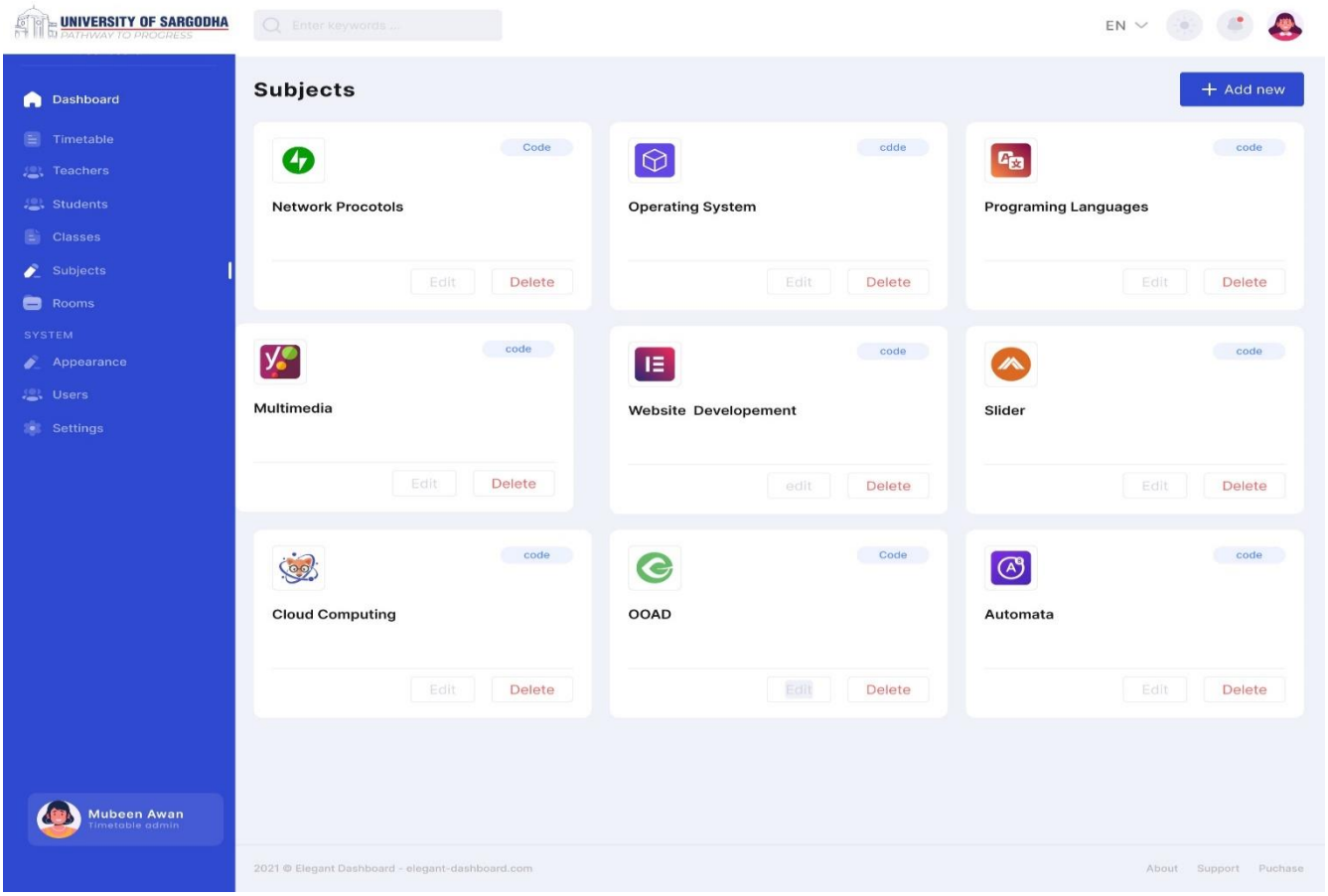


FIGURE 253 GUI 4.5

Description

Admin can view, add, update, and delete Subjects.

Student information	
Interface Id.	4.6
Use case Reference	Figure 13 Use case Diagram

Snapshot

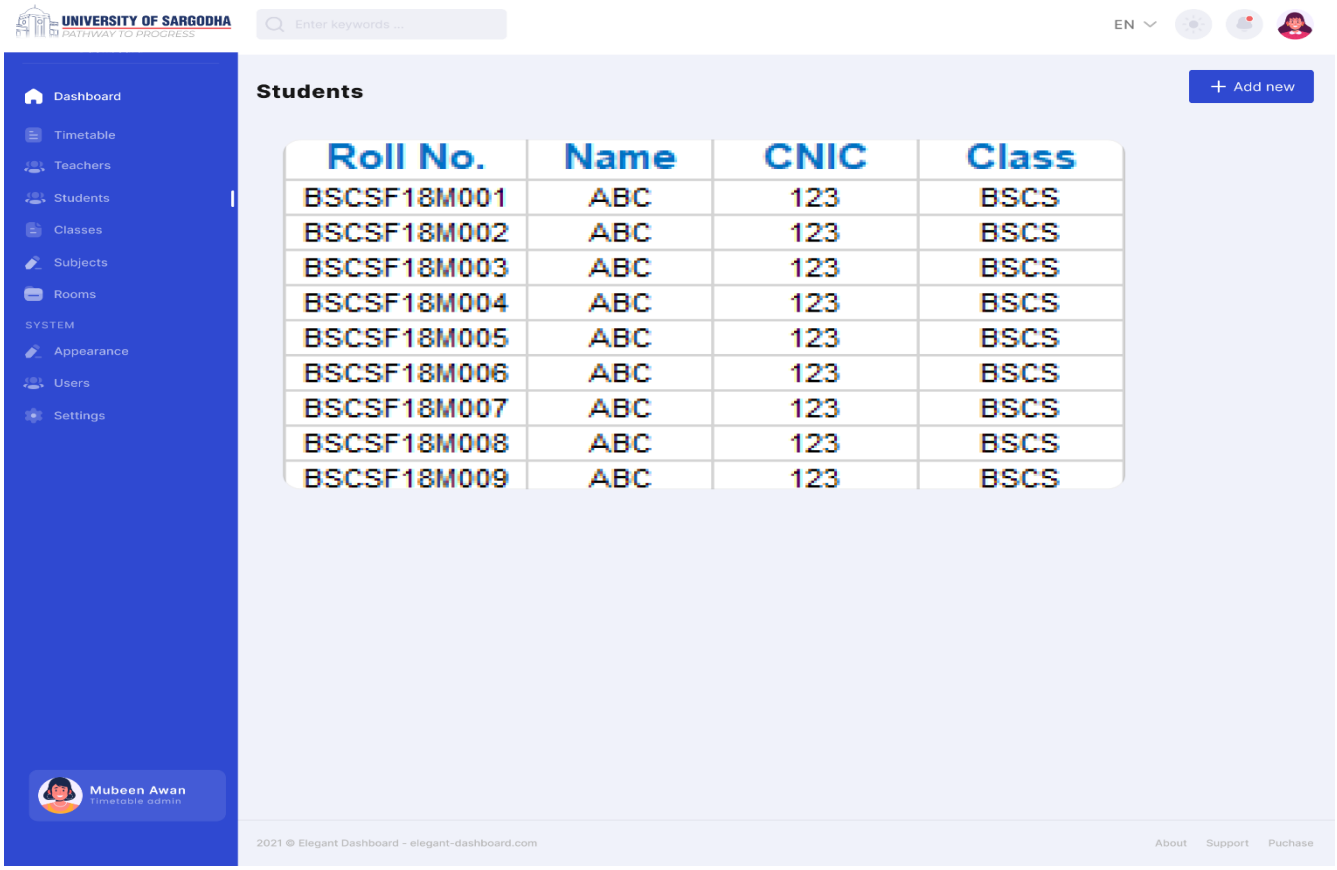


FIGURE 264 GUI 4.6

Description

Admin can view, add, update, and delete Students

Bus Incharge Interface for Bus Schedule Module (web based)

1: Bus Incharge Login

Login for Bus Incharge	
Interface Id.	4.7
Use case Reference	Figure 13 Use case Diagram

Snapshot

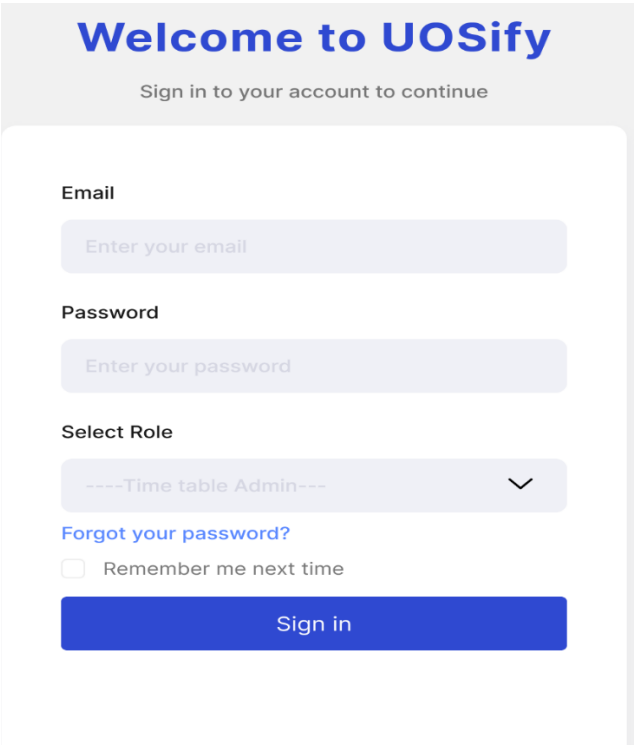


FIGURE 25 GUI 4.7

Description

Admin will login through email and password.

Dashboard for Bus Schedule module	
Interface Id.	4.8
Use case Reference	Figure 13 Use case Diagram

Snapshot

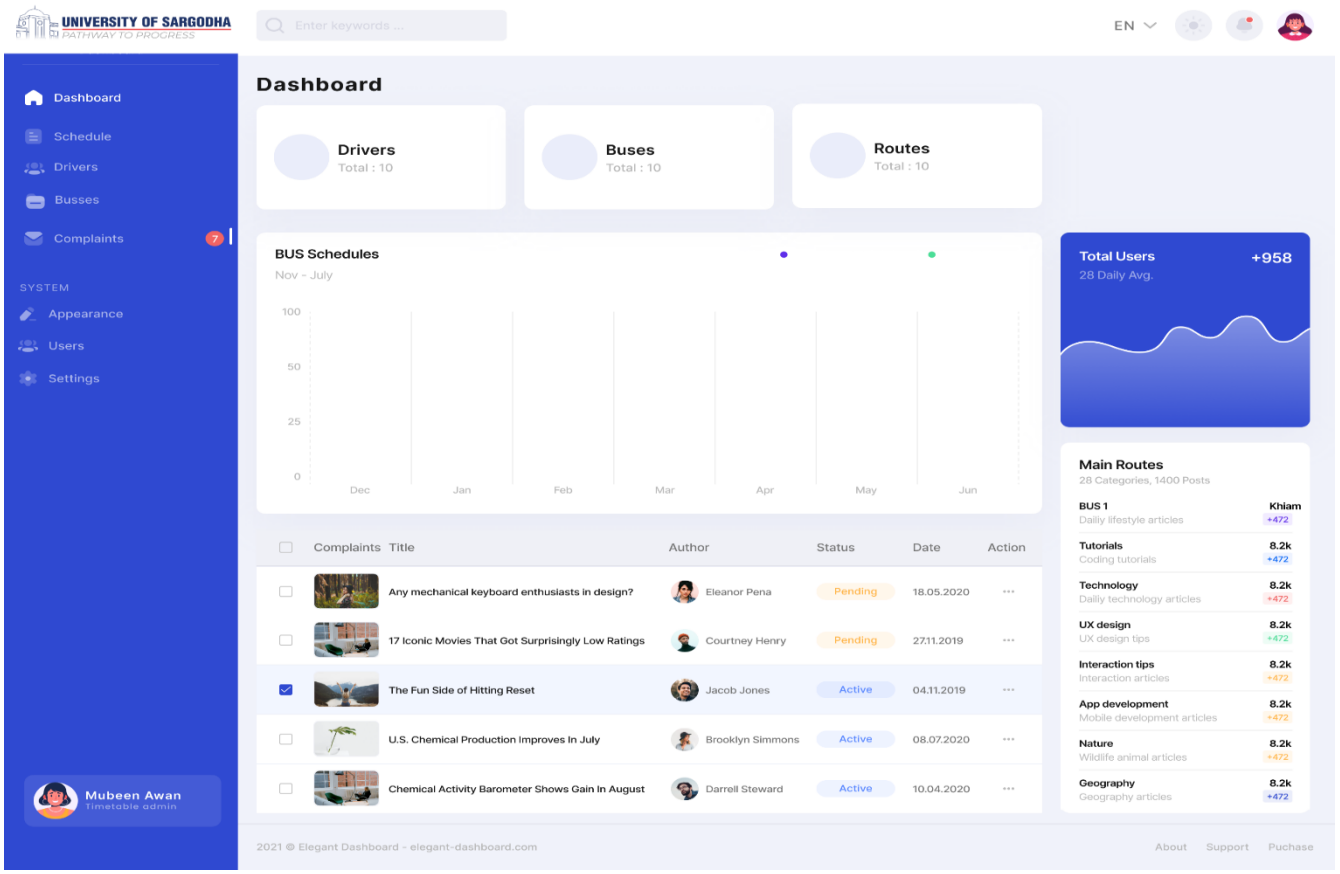


FIGURE 26 GUI 4.8

Description

Bus schedule dashboard will appear when bus incharge will successfully login.

Driver information	
Interface Id.	4.9
Use case Reference	Figure 13 Use case Diagram

Snapshot

FIGURE 27 GUI 4.9

Description

Bus incharge can view driver information and can add new driver if necessary.

Complaints	
Interface Id.	4.10
Use case Reference	Figure 13 Use case Diagram

Snapshot

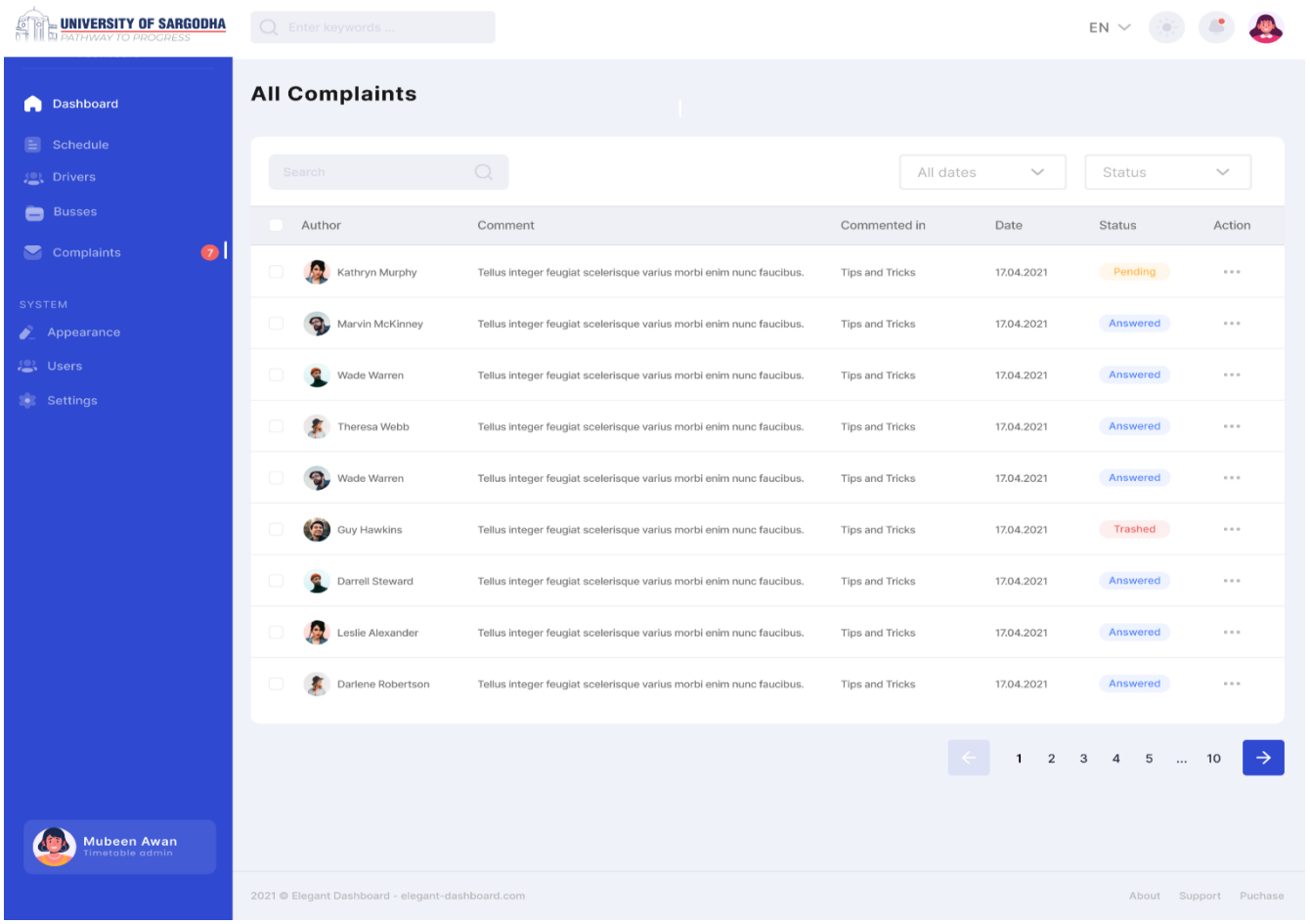


FIGURE 28 GUI 4.10

Description

Bus incharge can view complaints and can take proper action against those complaints.

4.4 High Level Design

4.5.1 ER Diagram

This Entity Relationship Diagram of UOSify shows all the visual instruments of database tables and the relations between them.

- **Description of ERD for UOSify:**

  The main entities of UOSify are Admin, Teacher, Student, Driver, Class, Room, Subject and Bus incharge. Each entity contains a primary key as shown in diagram. The associative entities are teacher-subject, teacher-student, subject-student, and subject-class. There are One-to-One, One-to-Many, One-to-Optional and Many-to-many relationships between the main and associative entities that are shown in diagram.

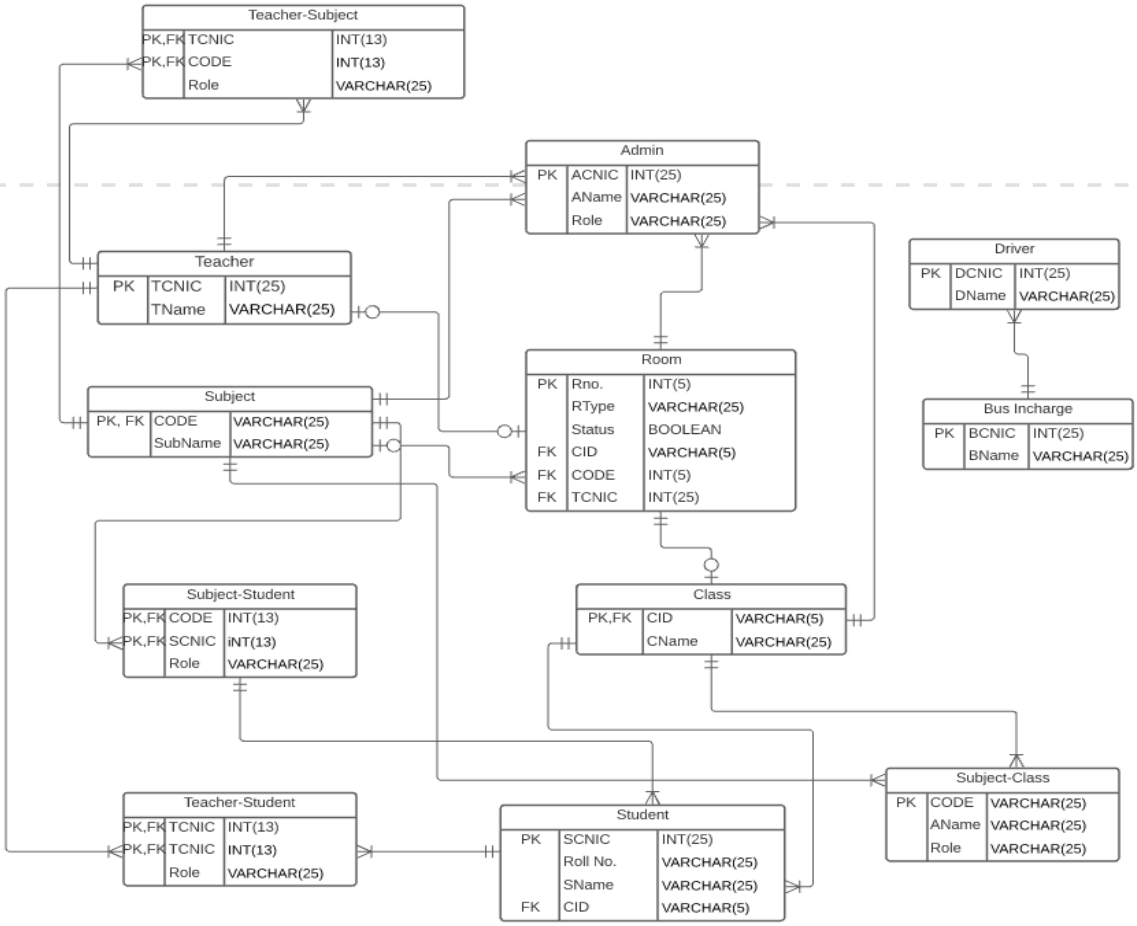


Figure 14 ERD of System

4.5 Data Dictionary

4.6.1 UOSify

Data 1	
Name	User login
Alias	Login page
Where-used/how-used	User provide its information and system validate that data and store into database
Content description	There user describe about himself/herself either he/she teacher, student, or admin.
Supplementary information	-

Data 2	
Name	View information
Alias	Timetable and bus information
Where-used/how-used	Users can view timetable and bus information. Drivers and bus incharge can only view bus information.
Content description	All users can view required data only if they enter valid credentials.
Supplementary information	-

Data 3	
Name	Notifications
Alias	Notification to users
Where-used/how-used	In the case of any updation in timetable (cancel and Reschedule classes) and bus schedule users will get instant notifications.

Content description	For notifications it's necessary to login.
Supplementary information	-

Data 4	
Name	Tracking
Alias	Bus tracking
Where-used/how-used	Drivers will start tracking and users can view tracking through google maps.
Content description	For tracking it's necessary to login.
Supplementary information	-

.

Data 5	
Name	Update information
Alias	Timetable and bus information
Where-used/how-used	Admin/Bus incharge can update timetable or bus schedule information.
Content description	For updating it's necessary to login.
Supplementary information	-

4.6 Requirements Traceability Matrix

Assuming priority scale to be 5.0

Sr. #Feature	Features	Use case ID	UI ID	Priority	Build Number	Use Case Cross reference (Related Use Cases)
1	User login	1	1	4.8/5.0	1	-
2	View timetable	2	2	4.8/5.0	2	-
3	Update timetable	3	3	4.5/5.0	3	-
4	Timetable notification	4	4	4.3/5.0	4	-
5	Cancel class	5	5	4.2/5.0	5	-
6	Reschedule class	6	6	4.2/5.0	6	-
7	Bus Info view	7	7	4.8/5.0	7	-
8	Tracking	8	8	4.6/5.0	8	-
9	Update bus info	9	9	4.3/5.0	9	-

4.7 Risk Analysis

Risks of the project in different scenarios:

- Risk Identification
  - Overload of database
  - Response time can be delay due to server etc.
- Risk Drivers
  - Large number of fake accounts.
  - Too much traffic on system which will decrease the response time.
  - System will not verify teachers, students, and drivers.
- Percentage Impact of Risk Drivers
  - 25 % chances of overloading of database
  - 20 % chances of late response
- Risk Mitigation Plan
  - For database risk we will increase the size of storage.
  - For increasing response rate, we will use latest technology and fast response server etc.

4.8 Cost Estimation Sheet

COCOMO Model:

This project "UOSify" is an average complexity and fair complexity as it is going to be used for timetable & bus management and tracking via google maps; thus, it is classified as the first class of the COCOMO model i.e., the Organic Projects. Following is the formula for the cost estimation for the COCOMO Organic projects:

Size = the number of lines of code for completed product.

It is measured in KLOC (thousands of lines of code)

Time = total number of months

Basic COCOMO Model:

Effort:

Effort = 2.4 * (Size) ^ 1.05
Effort = 2.4*12^1.05
Effort= 32.61

Time:

Time = 2.5 * (Effort) ^3.08
Time = 2.5*32.61^0.38
Time= 9 months

Persons Required:

Persons=Effort/Time
Persons=32.61/9
4

Intermediate COCOMO Model:

The EAF (Effort Adjustment Facto r) is the product of the 15 adjustment parameters.

EAF = the effort adjustment factor

Each of these adjustments is categorized as very low, low, nominal, high, or very high. The following table shows all the EAF's along with the corresponding ranges.

EFFORT ADJUSTMENT FACTOR	RANGE
Required Reliability	0.75-1.40
Database Size	0.94-1.16
Product Complexity	0.70-1.65
Execution Time Constraint	1.00-1.66
Analyst Capability	1.46-0.71
Application Experience	1.29-0.82
Programmer Capability	1.42-0.70
Language Experience	1.14-0.95
Use of Modern Practices	1.24-0.82
Use of Software Tools	1.24-0.83
Required Development Schedule	1.23-1.10

Classified the adjustment factors in the following way:

- Required reliability as low and a value of 1.0.
- Database size as low and a value of 1.0
- Product complexity as normal and a value of 1.08
- Execution time constraint as normal and a value of 1.2
- Analyst capability as high and a value of 1.1
- Applications experience as normal and a value of 1.0
- Programmer capability as normal and a value of 1.0
- Language experience as normal and a value of 1.0
- Use of modern practices as normal and a value of 0.82
- Use of software tools as high and a value of 0.83
- Required development schedule as normal and a value of 1.10

EAF

EAF= (1.0+1.0+1.08+1.2+1.1+1.0+1.0+1.0+0.82+0.83+1.10+0.95+0.9+1.0+1.1)/15
EAF=0.9

The EAF value evaluated to 1.0

Size:

We estimated the **size to be 2000 LOC.**

Effort:

Effort = 3.2 * EAF * (Size) ^ 1.05
Effort = 3.2*1.0*12.0^1.05
Effort= 39.13

Time:

Time = 2.5 * (Effort) ^3.08
Time = 2.5*29.34^0.38
Time= 10 months

Persons Required:

Persons=Effort/Time
Persons=39.13/10
4

5. References

Ref. No.	Document Title	Date of Release/ Publication	Document Source
1			https://www.kashipara.com/project/idea/java/bus-tracking-application_1830.html
2			https://1000projects.org/college-bus-tracking-system-java-project.html
3			https://www.freeprojectz.com/java-jsp-netbeans-project/timetable-management-system
4			https://myclasscampus.com/home/school-coaching-time-table-management-software
5			https://www.campcodes.com/projects/php/student-timetable-management-system-using-php-mysql-appgini-free-download/
6	Proposal document	8-11-2021	shorturl.at/qxFHP
7		22-11-2021	https://docs.google.com/document/d/1ceQxwiX6vuhFVoAbTp472dUZ8S-noNni/edit?usp=sharing&ouid=106572312992664191811&rtpof=true&sd=true

	Software Requirement Specification		
8	Design Document	15-12-2021	https://docs.google.com/document/d/108rel6Mb6K-yED4vrt4Uzswr9apsUBoj/edit?usp=sharing&ouid=106572312992664191811&rtpof=true&sd=true
9	Functional Specification	20-12-2021	https://docs.google.com/document/d/1VQ1D1m5fWdCZiq4sXnssB4ZoKwob2jiO/edit?usp=sharing&ouid=106572312992664191811&rtpof=true&sd=true