



# RouteEase Project



**Submitted by:**

Faiza Ishfaq (2020CS633)

**Submitted to:** Mr Waqas Ali

Department of Computer Science

**University of Engineering and Technology Lahore**

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>MILESTONE 1</b>   | <b>1</b>  |
| 1.1      | Abstract . . . . .   | 1         |
| 1.2      | Introduction . . . . .                                       | 1         |
| 1.3      | Approch . . . . .  | 1         |
| 1.4      | Requirements . . . . .                                       | 2         |
| 1.4.1    | Functional Requirements . . . . .                            | 2         |
| 1.4.1.1  | User . . . . .   | 2         |
| 1.4.2    | Non-Functional Requirements . . . . .                        | 2         |
|          | Here system's non-functional requirements are given below:   | 2         |
| 1.4.2.1  | Responsive . . . . .   | 2         |
| 1.4.2.2  | Performance . . . . .  | 2         |
| 1.4.2.3  | Security . . . . .   | 2         |
| 1.4.2.4  | Capacity . . . . .   | 3         |
| 1.4.2.5  | Scalability . . . . .  | 3         |
| 1.4.2.6  | Availability . . . . .                                       | 3         |
| 1.4.2.7  | Reliability . . . . .  | 3         |
| 1.4.2.8  | Usability . . . . .  | 3         |
| 1.4.2.9  | Recoverability . . . . .                                     | 3         |
| 1.4.2.10 | Manageability . . . . .                                      | 3         |
| 1.4.2.11 | Data Integrity . . . . .                                     | 3         |
| 1.5      | Features . . . . .   | 3         |
| 1.6      | Tools and Technology . . . . .                               | 4         |
| 1.7      | Project Workflow . . . . .                                   | 4         |
| 1.8      | References . . . . .   | 4         |
| 1.9      | Feasibility Report . . . . .                                 | 5         |
| 1.9.1    | The task to be under taken . . . . .                         | 5         |
| 1.9.2    | Benefits . . . . .   | 5         |
| 1.9.3    | A Preliminary Requirements Analysis . . . . .                | 5         |
| 1.9.4    | Technical Requirements . . . . .                             | 6         |
| 1.9.5    | Suggested Deliverables: . . . . .                            | 6         |
| 1.9.6    | Software Development Process: . . . . .                      | 7         |
| 1.9.7    | Outline Plan (Principal activities and Milestones) . . . . . | 7         |
| 1.9.8    | Visibility Plan: . . . . .                                   | 7         |
| 1.9.9    | Risk Analysis . . . . .                                      | 8         |
| 1.9.10   | Conclusion: . . . . .  | 9         |
| <b>2</b> | <b>MILESTONE 2</b>   | <b>10</b> |

---

|        |   |    |
|--------|---|----|
| 2.1    | Use Cases . . . . .                               | 10 |
| 2.1.1  | Student Log-In . . . . .                          | 10 |
| 2.1.2  | Admin Log-In . . . . .                            | 11 |
| 2.1.3  | Search Route . . . . .                            | 11 |
| 2.1.4  | View Stops . . . . .                              | 12 |
| 2.1.5  | View current location of routes . . . . .         | 12 |
| 2.1.6  | Add Buses Routes . . . . .                        | 13 |
| 2.1.7  | Update Buses Routes . . . . .                     | 13 |
| 2.1.8  | Delete Buses Routes . . . . .                     | 14 |
| 2.1.9  | Add Bus Stops . . . . .                           | 14 |
| 2.1.10 | Update Buses Routes . . . . .                     | 15 |
| 2.1.11 | Delete Buses Stops . . . . .                      | 15 |
| 2.2    | Use Case Diagram . . . . .                        | 16 |
| 2.3    | Entity Relationship Diagram: Case Study . . . . . | 17 |
| 2.4    | Schema Diagram . . . . .                          | 18 |
| 2.5    | Class Diagram . . . . .                           | 18 |
| 2.6    | Activity Diagram . . . . .                        | 19 |
| 2.7    | Wireframes . . . . .                              | 22 |
| 2.8    | UI/UX . . . . .                                   | 30 |

# Chapter 1

## MILESTONE 1

### 1.1 Abstract

Student bus services are widely deployed in universities around the world because they provide cost-effective and economic student transportation. However, from a passenger point of view urban bus systems can be complex and difficult to navigate in case of person accidentally miss the bus or in case of bus late. The objective of this project is to build Transport Navigation System to provide ease to university students and faculty to locate the bus. We present REase (RouteEase), a reality-aware Student bus navigation system for bus students with the ability to recognize and track the physical transport i.e buses

### 1.2 Introduction

In Student bus navigation system, the student will just sign up the system make account, log in the system and view all routes current location. Most of the time, buses are susceptible to delay and temporary rerouting due to traffic congestion and construction work. Because of this buses take longer route and take long time than usual. And student on their stops unknowingly wait that bus and their important waste in waiting. This system provide a huge ease to the student in this aspect and student get all this information from the system and then take decision accordingly.

### 1.3 Approach

Agile methodology will be adopted in our project. As Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments.

The Key Values and Principles of the Agile Manifesto

- Individuals and interactions over processes and tools.

- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

## 1.4 Requirements

Requirements for REase project both functional and non-functional will be accommodate.

### 1.4.1 Functional Requirements

#### 1.4.1.1 User

Here we have functional requirements given below in user perspective:

- System shall allow the user to create his her Account and Sign-in.
- System shall allow admin to Add buses and assign routes to the buses their starting and destination point.
- System shall allow admin to enter buses GPS addresses for their tracking.
- System shall allow user to search their required route.
- System shall allow user to view the current location of their required route.
- System shall allow the user to logout from the system.

Optional

- Our system display the time how much route has currently far from the user stop.

### 1.4.2 Non-Functional Requirements

Here system's non-functional requirements are given below:

#### 1.4.2.1 Responsive

Our system must be responsive as each request will be accommodate within 2 seconds from backend side.

#### 1.4.2.2 Performance

Our system must perform well as it handles ten thousand users per second.

#### 1.4.2.3 Security

Our system must be secure as credentials of its users accounts. Not even the admin can see any password or any other details.

#### **1.4.2.4 Capacity**

Our system must handle ten thousand users at a time without any blockage.

#### **1.4.2.5 Scalability**

Our system must be horizontally scalable, must handle one million users in future as demand for this application is rapidly increasing with the passage of time.

#### **1.4.2.6 Availability**

Our system must be available twenty four hours to its users. It must never give downtime for any operation.

#### **1.4.2.7 Reliability**

Our system must show correct information to the user (i.e, bus route number, bus current location).

#### **1.4.2.8 Usability**

System must be usable as it displays text, images and easy to understand GUI as it will be user-friendly that any lay man can use it with just having the knowledge of typing. It must be user-friendly as it is interactive, good contrast of text color with background attractive layout.

#### **1.4.2.9 Recoverability**

Our system must be able to recover itself in case of any unforeseen circumstances.

#### **1.4.2.10 Manageability**

Our system must be manageable as there will be a management team always to look up or monitor each and everything concerning critical health issues. In case of any failure the team will debug, analyze and understand the root cause of failure for improving its performance..

#### **1.4.2.11 Data Integrity**

Our system must provide data integrity.

### **1.5 Features**

Here we have some features that we will accommodate in our Chat App.

- Simple log-in. Most of the apps today require some kind of sign-up and sign-in
- Data security and encryption
- Dark and light modes
- expanding and contracting map
- mobile app

- Platform Availability
- Performance

## 1.6 Tools and Technology

Tools and Technologies that are gonna be use in this projects are:

- HTML,CSS
- React JS
- Adobe XD
- WAMP Server
- UI/Ux

## 1.7 Project Workflow

- Use cases and models
- UI/UX
- ERD
- Functionality Implementation
- Testing
- Documentation Report
- POC

## 1.8 References

Google Map



## 1.9 Feasibility Report

### 1.9.1 The task to be under taken

The project is to develop a real time platform that provide students their required bus route current location and developing an interactive user interface. We present REase(Route Ease), a reality-aware bus navigation system for students with the ability to recognize and track the physical public transport i.e buses.

### 1.9.2 Benefits

- The Students bus Navigator "REase" track the location of buses and provide real-time updates to the students. The system allow students to track buses in real-time, making sure they know exactly when the bus reached currently.

tracking systems are a great way to increase efficiency and make sure your buses run on time

- Students, Workers that used university transport daily anxious about when will bus come or may the bus passes from the stop. They have a great ease from this system they know the current location of the bus and have estimate when will bus arrived and manage all his activities accordingly.
- In case of bus pass from their stop ,students get information from the app and will wait for the next possible route if available.

### 1.9.3 A Preliminary Requirements Analysis

The system needs to meet the following functional requirements:

#### 1. Mobile App Interface

#### 3. Administrator Side

- Allow admin add buses and assign routes to the buses their starting and destination point.
- Allow admin to enter buses GPS addresses for their tracking

#### 4. Public Side

- allow the user to create his her Account and Sign-in.
- Display interactive User interface
- Allow user to search their required route.
- Allow user to view the current location of their required route.

Optional

- Our system display the time how much route has currently far from the user stop.

#### 1.9.4 Technical Requirements

Tools and Technologies that are gonna be use in this projects are:

- HTML,CSS
- React JS
- Adobe XD
- WAMP Server
- UI/Ux

#### 1.9.5 Suggested Deliverables:

##### 1. Requirements Analysis:

a document and a presentation to go over the formal requirements of the project, both functional and non-functional. This deliverable ensures that the Group is working on a system that closely matches to the wishes of the Client. This deliverable gives the Client a chance to modify and correct items that were mis-communicated or missed out before allowing the Group to proceed further in the design.

##### 2. Design Document:

a document and a presentation to go over the design of the system. This is the Group's opportunity to go over how the project is to be implemented to the Client. This deliverable is done by the more technical and experienced in the Group, based on the understanding of the requirements established in the previous deliverable.

##### 3. Source Code:

a document, presentation along with the source code of the final completed project. This final deliverable wraps up and concludes the project. In this deliverable, the Group delivers the final implementation based on the requirements specified and the design developed in previous stages. The system would have been tested thoroughly with unit tests and with a final acceptance test and would be ready for deployment to the production system.

### 1.9.6 Software Development Process:

The project will undertake Agile methodology as there is a well-defined set of requirements. As the Client has very specific needs for the system but changes in our system features expected during the development so, Agile methodology is suited for our system. Agile methods are based on the following principles:

- Rapid response to any changes with the assistance of adaptive planning.
- Joint development of requirements.
- Rationalization of duties performed by the members.
- Step-by-step software development.

### 1.9.7 Outline Plan (Principal activities and Milestones)

Milestone 1 -Requirement Analysis (Draft): An initial draft of the requirements analysis should be done as Milestone 1.

Milestone 2 -Requirement Analysis (Final): The final draft of the requirements analysis as well as feasibility study should be done for Milestone 2. In addition, a presentation will be prepared as a part of this milestone.

Milestone 3 –Use cases and scenarios: Use cases will be made in Milestone 3.

Milestone 4 –UI/UX: In this milestone user interface will be designed.

Milestone 5 –UMLUML and implementation will be done in Milestone 5.

Milestone 6 –POC: Coding and functionality implementation i.e., Proof of Claim(POC) will be done in Milestone 6.

Milestone 7 –Testing: Testing, Debugging and Integration of project will be done in this Milestone.

Milestone 8 –Project Demonstration(Deployment): The project source code should be handed over to the Client for the final milestone. A presentation will also be delivered.

### 1.9.8 Visibility Plan:

External:The group will conduct Weekly meeting with Client (i.e., Teacher) in lab . If situations arise or if a problem needs to be addressed between the meetings, the Group will conduct any further necessary communication via email. Because agile model will be used, a report will be issued to the Client at the end of every step to ensure that both parties are in-sync and to minimize any miscommunication in the requirements.

Internal: The Group will meet daily to discuss progress and problems. Meeting minutes will be kept track of and sent to all members of the Group for reference. Any additional communication will be done via email or through other collaboration tools such as document sharing. In addition, the source code will be stored on drive as repository. All source code will be documented carefully before being submitted to the repository. The progress of the principal activities and major milestones will be closely monitored and compared with the schedule. During the regular meetings, progress will be compared to a Gantt chart, which will be drafted after the completion of the requirements study when the tasks are identified and assigned.

### **1.9.9 Risk Analysis**

Changing Requirements: Risk: The Client may have different ideas about the system during the course of the project. Depending on the situation, the changes that the Client wishes to have implemented may require little or major changes to the architecture. Solution: To reduce the possibility of this occurring, the Group needs to establish a clear visibility plan with the Client.

Incomplete Requirements: Risk: It is possible that requirements may be implied but not discussed or misunderstood. This frequently occurs after meetings. Solution: The Group's interpretation of the Client's requirements will be presented back to the Client to get a confirmation on whether the Group has understood the Client. Frequent client updates and a high level of visibility will also help call attention to any misunderstandings.

System Integration: Depending on the level of access to the servers that the Group receives, the Group may need to work on the system offline and eventually integrate with the production system when it is ready and thoroughly tested. Due to different software configuration, there may be unpredictable obstacles.

Solution: To ensure a smooth system integration, the Group needs to be aware of as much about the configuration as early as possible.

Non-functional Requirements: Risk: Similar to incomplete requirements, non-functional requirements is something that has not been brought up in the initial meeting with the Client. These include requirements on the number of users that the system expects to support concurrently, and the response time of the database lookup. Solution: A follow up meeting is needed to specify the non-functional requirements.

Human resources: Risk: The Group is relatively small consisting of only 3 members, some members are not technically oriented and almost all members have limited knowledge of relevant web-technologies. Solution: For these reason the Group acknowledges that a slow design and implementation phase may be inevitable, and are planning accordingly.

**1.9.10 Conclusion:**

Based on the analysis of this feasibility study, the Group has collectively agreed that this project IS FEASIBLE and the group is WILLING to take on the afore mentioned project. The benefits are significant enough to justify the development effort required

## Chapter 2

# MILESTONE 2

### 2.1 Use Cases

#### 2.1.1 Student Log-In

|                     |   |
|---------------------|---|
| Name                | Log-In  |
| Participating Actor | Student   |
| Goals               | Student can successfully log-in to the system   |
| Triggers            | Request to log in   |
| Pre-Condition       | Student have authentic ID and password  |
| Post-Condition      | Student access the system and its functionalities   |
| Basic Flow          | <ul style="list-style-type: none"><li>• enter name</li><li>• enter password</li><li>• Request to log in</li></ul> |
| Alternate flow      | Reset Password or contact to admin  |
| Qualities           | System must respond within 3 to 5 seconds.<br>System validate correct ID and password.                            |
| Exceptions          | wrong ID or password  |

**2.1.2 Admin Log-In**

|                     |   |
|---------------------|---|
| Name                | Log-In  |
| Participating Actor | Admin   |
| Goals               | Admin can successfully log-in to the system   |
| Triggers            | Request to log in   |
| Pre-Condition       | Admin have authentic ID and password  |
| Post-Condition      | Student access the system and its functionalities   |
| Basic Flow          | <ul style="list-style-type: none"> <li>• enter name</li> <li>• enter password</li> <li>• Request to log in</li> </ul> |
| Alternate flow      | Reset Password  |
| Qualities           | System must respond within 3 to 5 seconds.<br>System validate correct ID and password.                                |
| Exceptions          | wrong ID or password  |

**2.1.3 Search Route**

|                     |   |
|---------------------|---|
| Name                | Search Route  |
| Participating Actor | Student, Admin  |
| Goals               | Search the particular Route   |
| Triggers            | Search the route by scrolling drop down and select your route   |
| Pre-Condition       | log into the system   |
| Post-Condition      | View your route bus location and stops  |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Search route from drop down menu</li> <li>• Select Route</li> </ul> |
| Alternate flow      | Manually Search by expanding map  |
| Qualities           | System must respond within 3 to 5 seconds.<br>show correct routes name  |
| Exceptions          | Search route not available.   |

**2.1.4 View Stops**

|                     |   |
|---------------------|---|
| Name                | View Stops  |
| Participating Actor | Student, Admin  |
| Goals               | View stops of the selected routes   |
| Triggers            | Search the route by scrolling drop down and select your route   |
| Pre-Condition       | log into the system, Search Route, Select route   |
| Post-Condition      | View the stops of selected routes   |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Search route from drop down menu</li> <li>• Select Route</li> </ul> |
| Alternate flow      | N/A   |
| Qualities           | System must respond within 3 to 5 seconds.<br>show correct routes name  |
| Exceptions          | Search route not available.   |

**2.1.5 View current location of routes**

|                     |   |
|---------------------|---|
| Name                | view current location of routes   |
| Participating Actor | Student, Admin  |
| Goals               | View current location of routes   |
| Triggers            | Search the route by scrolling drop down and select your route   |
| Pre-Condition       | log into the system   |
| Post-Condition      | View your route bus location  |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Search route from drop down menu</li> <li>• Select Route</li> </ul> |
| Alternate flow      | None  |
| Qualities           | System must respond within 3 to 5 seconds.<br>show correct location of route on map   |
| Exceptions          | Search route not available.   |



**2.1.6 Add Buses Routes**

|                     |  |
|---------------------|--|
| Name                | Add Buses Routes   |
| Participating Actor | Admin  |
| Goals               | Add bus routes in the system   |
| Triggers            | Request to Add Bus   |
| Pre-Condition       | log into the system, have bus tracker ID   |
| Post-Condition      | Bus show in the available routes   |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Enter buses number and tracker ID</li> <li>• Click add to add the bus</li> </ul> |
| Alternate flow      | N/A  |
| Qualities           | System must respond within 3 to 5 seconds.<br>Enter correct bus no and tracker ID  |
| Exceptions          | Search route not available.  |

**2.1.7 Update Buses Routes**

|                     |  |
|---------------------|--|
| Name                | Update Buses Routes  |
| Participating Actor | Admin  |
| Goals               | Update bus routes in the system  |
| Triggers            | Request to update Bus  |
| Pre-Condition       | log into the system, have bus number ,tracker ID   |
| Post-Condition      | Bus show on the available routes   |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Enter buses number</li> <li>• Click to Update route</li> </ul> |
| Alternate flow      | N/A  |
| Qualities           | System must respond within 3 to 5 seconds.<br>Enter correct bus no and tracker ID  |
| Exceptions          | N/A  |

**2.1.8 Delete Buses Routes**

|                     |  |
|---------------------|--|
| Name                | Delete Buses Routes  |
| Participating Actor | Admin  |
| Goals               | Delete bus routes from the system  |
| Triggers            | Request to delete Bus  |
| Pre-Condition       | log into the system, have bus number ,tracker ID   |
| Post-Condition      | Bus not show on the available routes   |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Enter buses number</li> <li>• Click on delete route</li> </ul> |
| Alternate flow      | N/A  |
| Qualities           | System must respond within 3 to 5 seconds.<br>System validates bus number  |
| Exceptions          | N/A  |

**2.1.9 Add Bus Stops**

|                     |  |
|---------------------|--|
| Name                | Add Bus Stops  |
| Participating Actor | Admin  |
| Goals               | Add stops in the route   |
| Triggers            | Request to Add stops   |
| Pre-Condition       | log into the system, have bus number   |
| Post-Condition      | Bus Stops of the corresponding route show on map   |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Enter buses number</li> <li>• Enter their stops and highlight it.</li> </ul> |
| Alternate flow      | N/A  |
| Qualities           | show correct bus stops .<br>System validates bus number  |
| Exceptions          | N/A  |

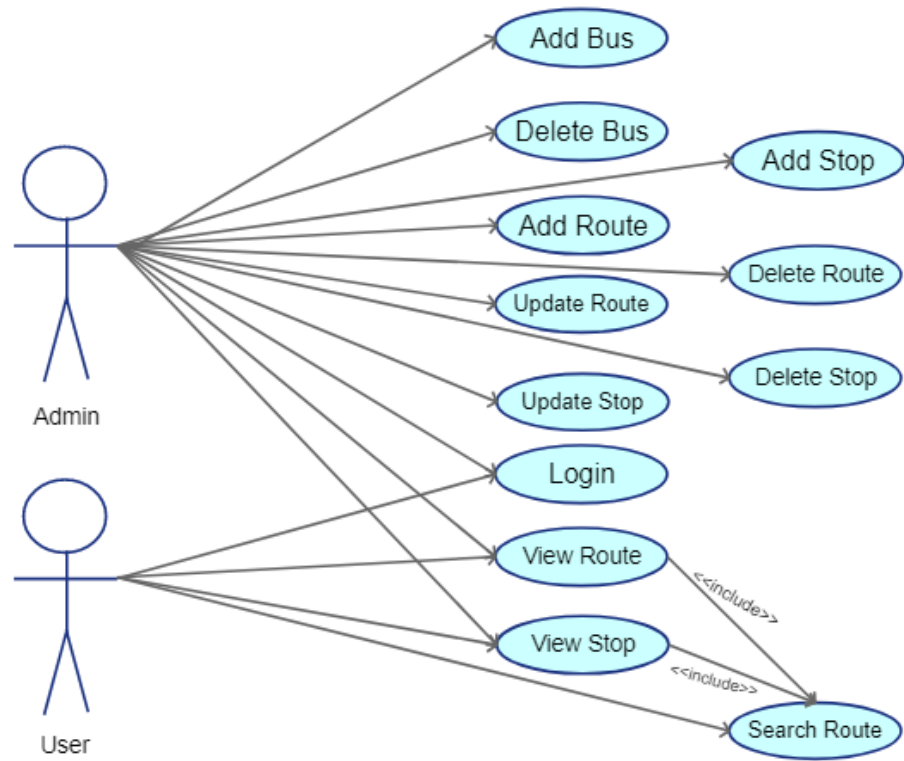
**2.1.10 Update Buses Routes**

|                     |   |
|---------------------|---|
| Name                | Update Bus Stops  |
| Participating Actor | Admin   |
| Goals               | Update stops of the route   |
| Triggers            | Request to Update stops   |
| Pre-Condition       | log into the system, have bus number  |
| Post-Condition      | Updated Bus Stops of the corresponding route show on map  |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Enter buses number</li> <li>• Update stops</li> </ul> |
| Alternate flow      | N/A   |
| Qualities           | show correct updated bus stops .<br>System validates bus number   |
| Exceptions          | N/A   |

**2.1.11 Delete Buses Stops**

|                     |   |
|---------------------|---|
| Name                | Delete Bus Stops  |
| Participating Actor | Admin   |
| Goals               | Delete stops from the route   |
| Triggers            | Request to delete stops   |
| Pre-Condition       | log into the system, have bus number  |
| Post-Condition      | Bus Stops of the corresponding route not show on map  |
| Basic Flow          | <ul style="list-style-type: none"> <li>• log into the system</li> <li>• Enter buses number</li> <li>• Enter the stops</li> <li>• delete stop</li> </ul> |
| Alternate flow      | N/A   |
| Qualities           | show correct bus stops .<br>System validates bus number   |
| Exceptions          | N/A   |

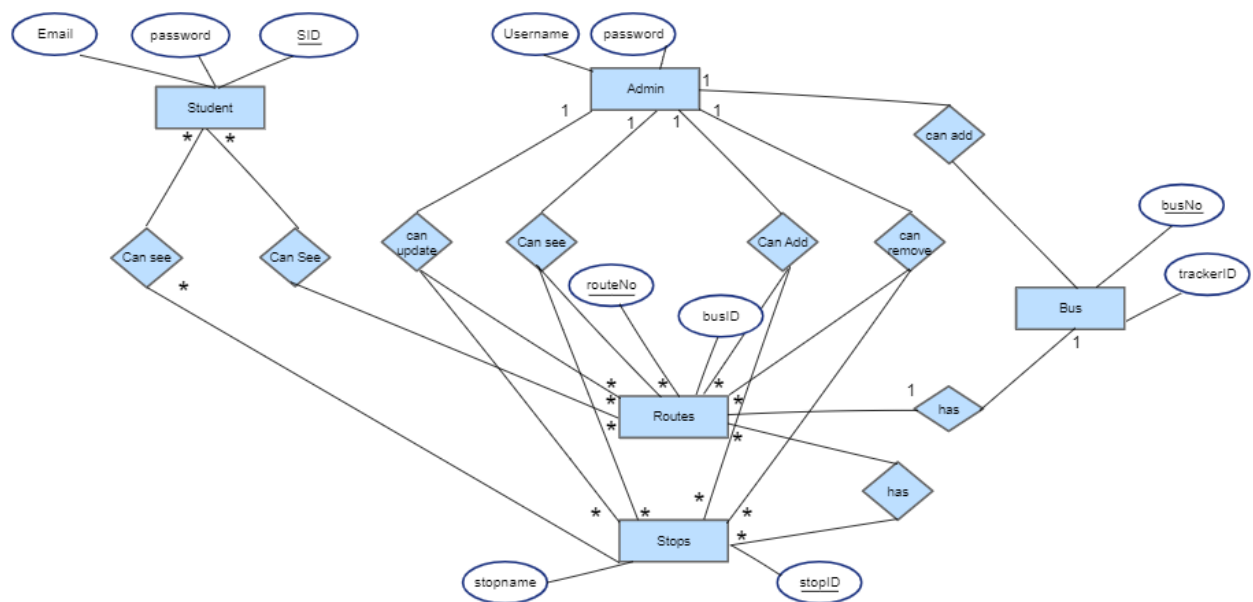
## 2.2 Use Case Diagram



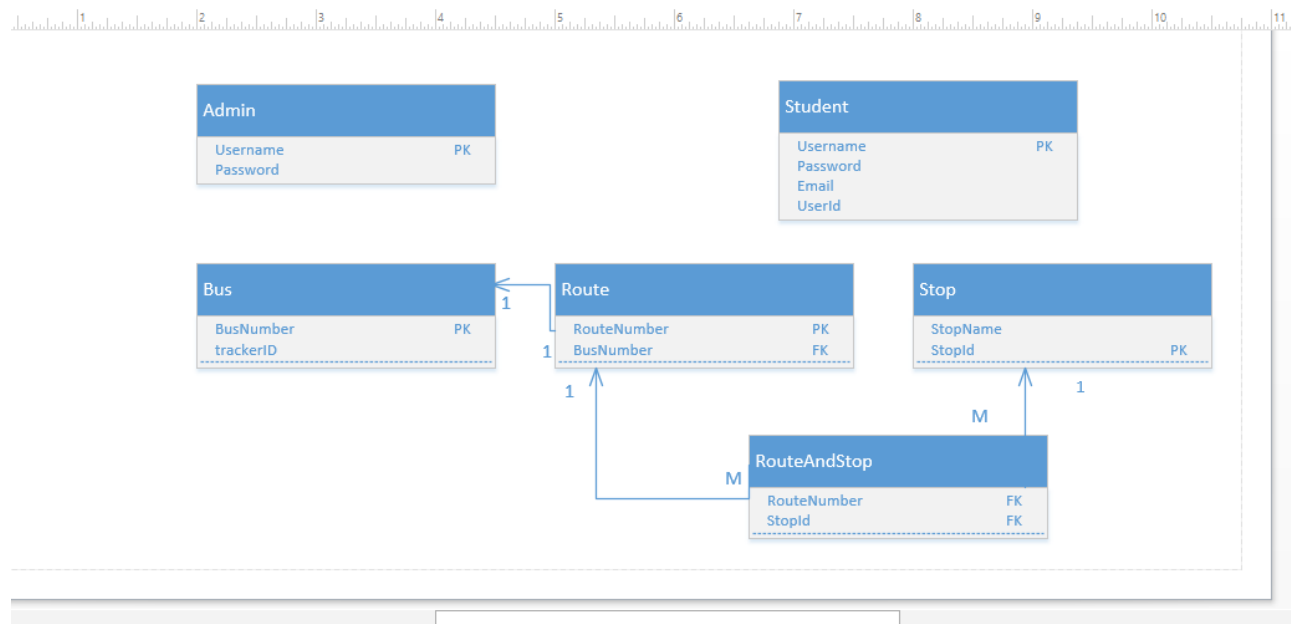
## 2.3 Entity Relationship Diagram: Case Study

Identify the entities and relationships for the following description and draw an ERD diagram:

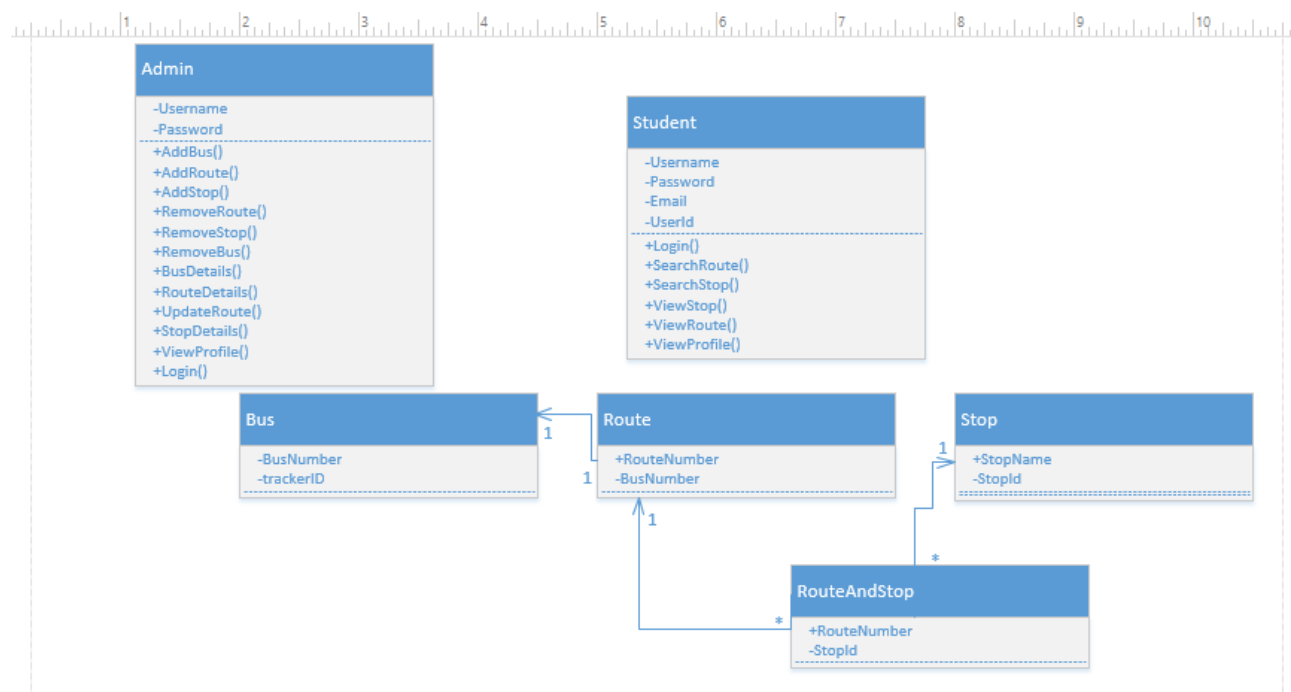
A student can view many routes and stops on the other hand one route or stop can be viewed by many students. Admin can add, update, remove or simply view many routes, buses and stops. One bus can handle one route similarly one route denotes one bus. One route can entertain many stops and one stop can encounter many routes.



## 2.4 Schema Diagram

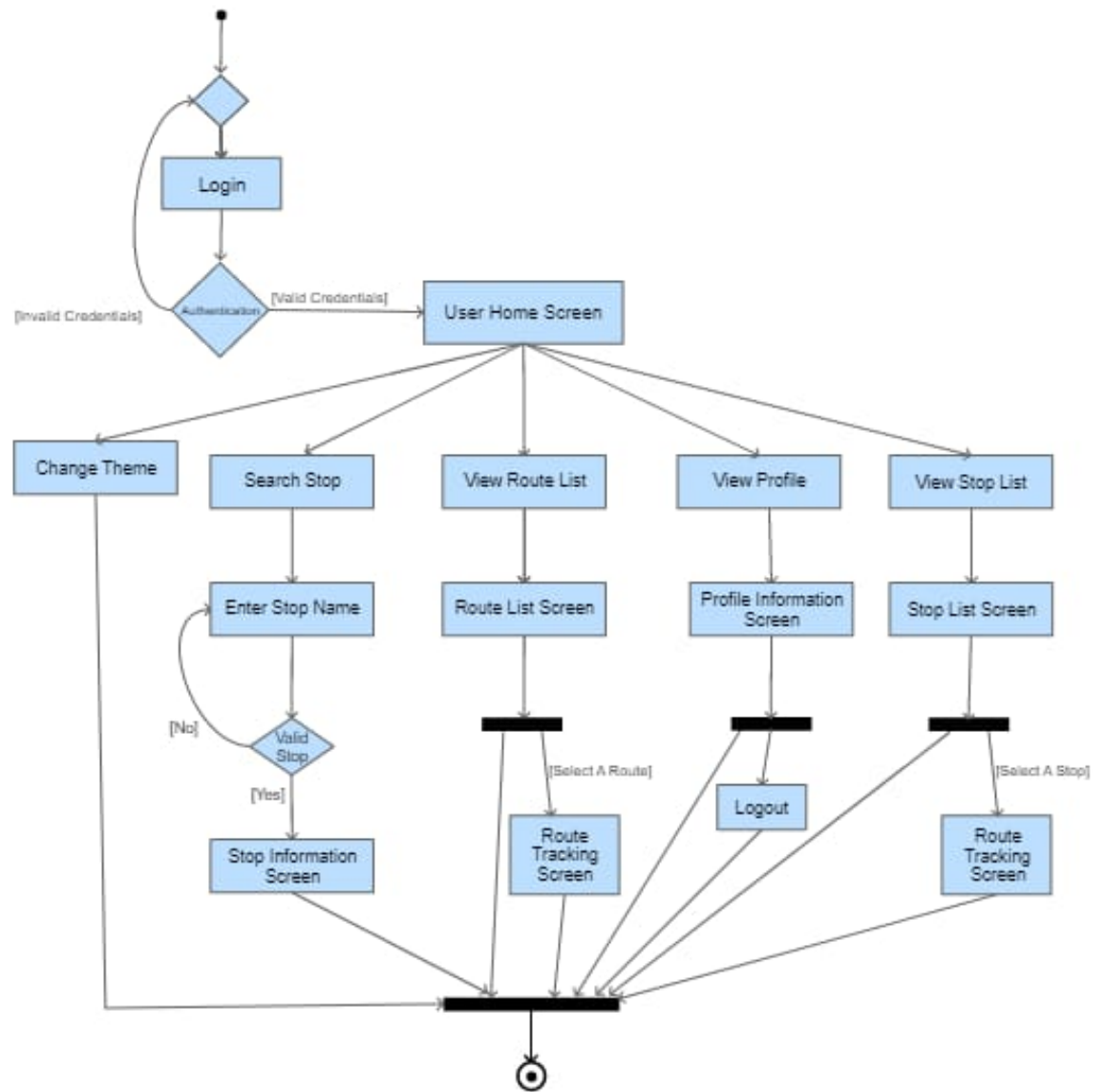


## 2.5 Class Diagram

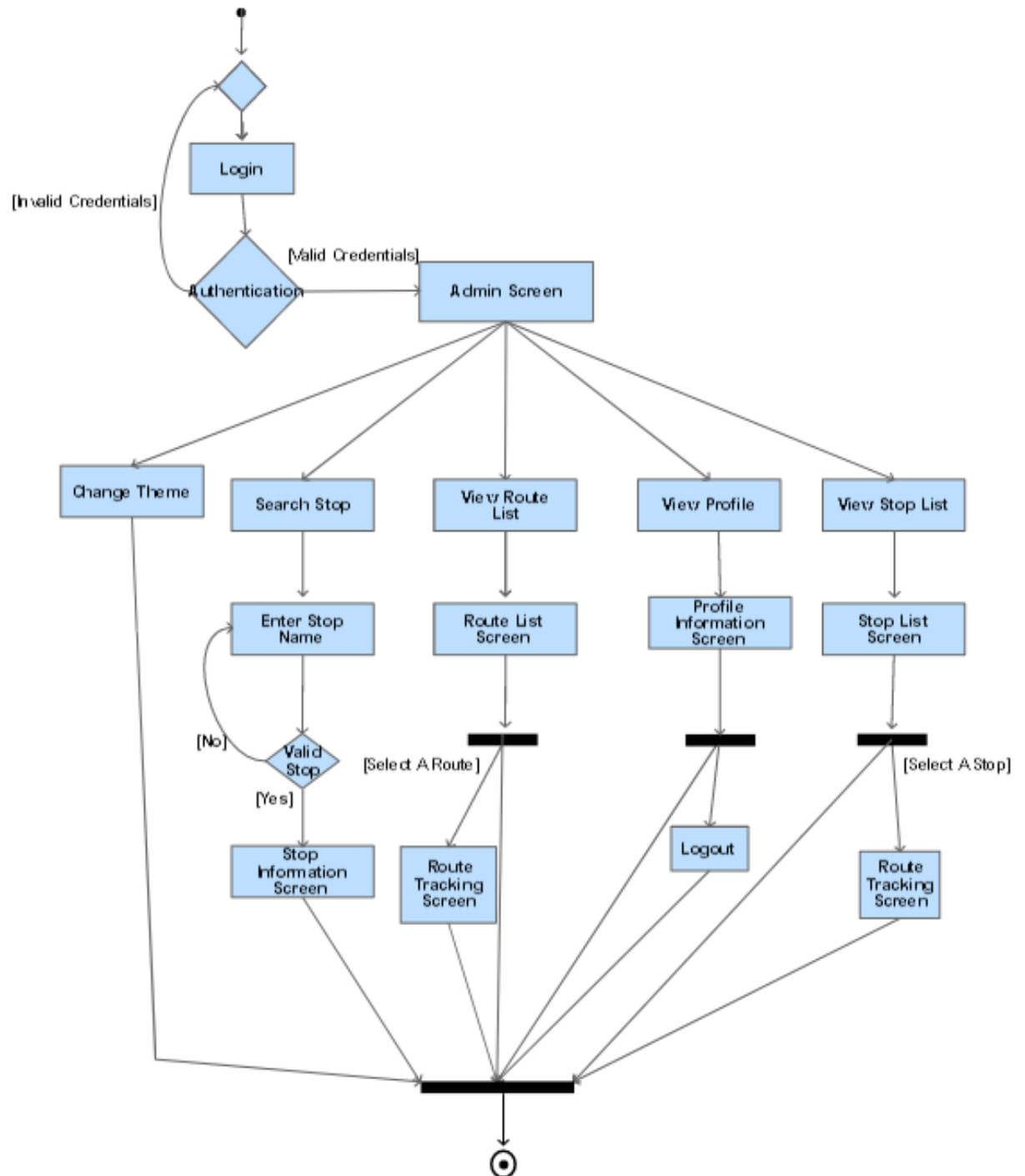


## 2.6 Activity Diagram

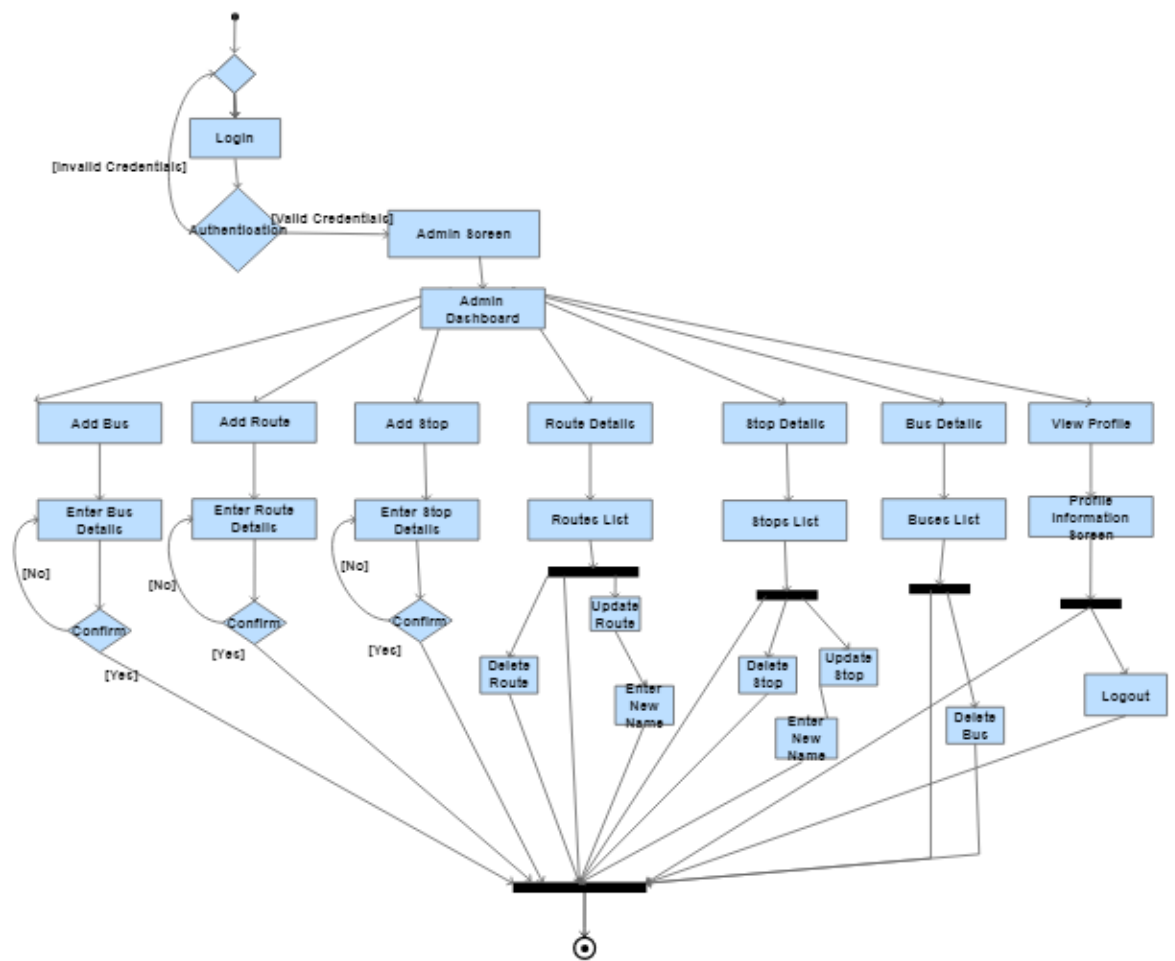
### User Activity Diagram



## Admin Activity Diagram

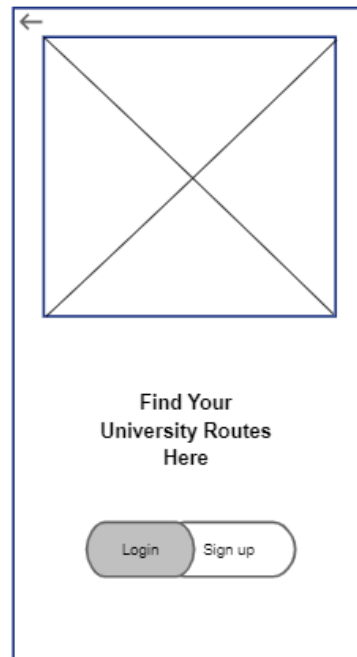
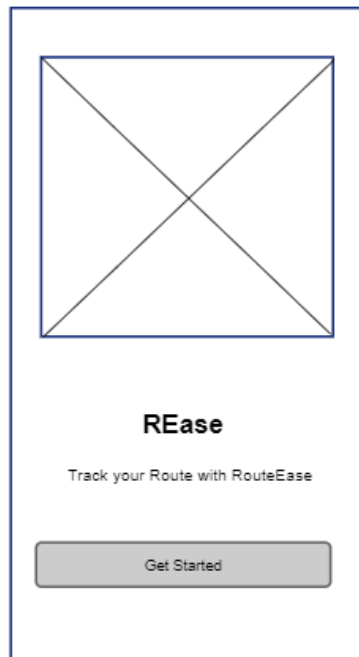




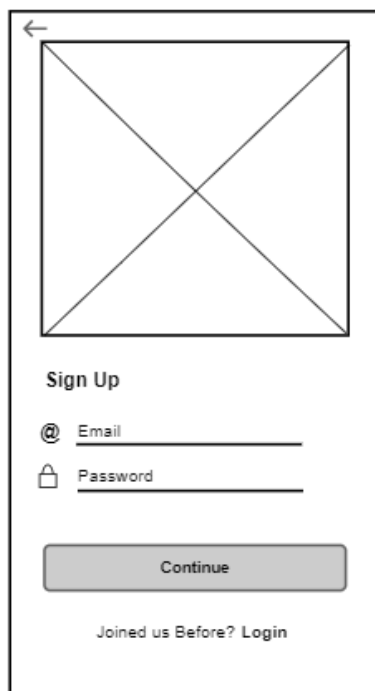


## 2.7 Wireframes

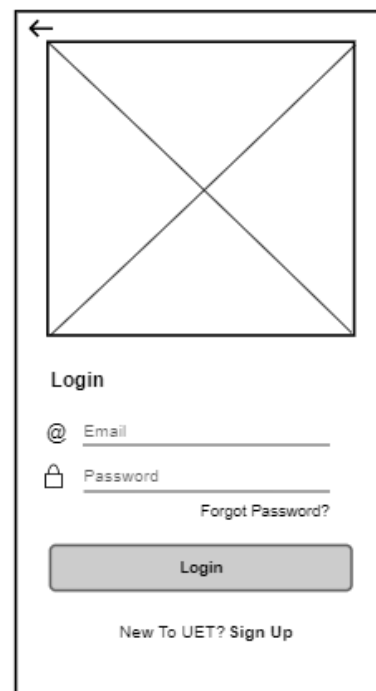
### Get Started



### Sign Up



### Log in



Forget password

Forgot Password?

Don't worry it happens. Please enter your Email address

@ Email

Submit

OTP Verification

OTP Verification

Enter OTP Sent to You  
2020\*\*\*\*\*.uet.edu.pk

Don't Receive OTP? Resend OTP

Verify & Proceed

Reset Password

Reset Password

Submit

Home page

Search

R1

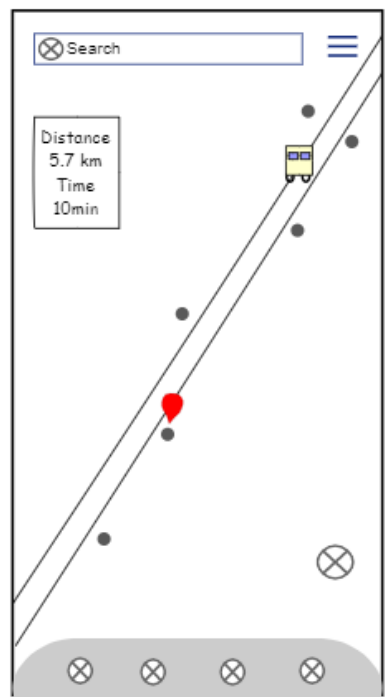
R2

R3

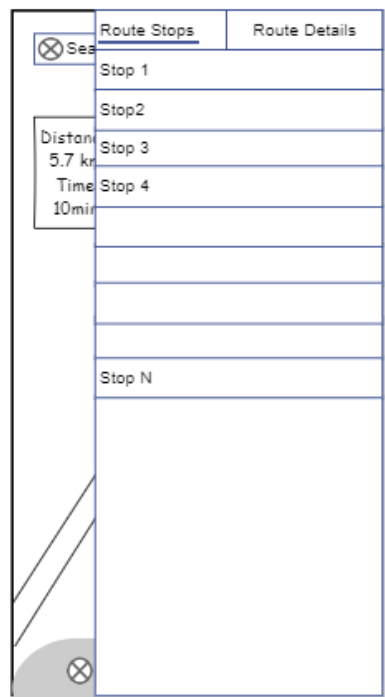
R4

R5

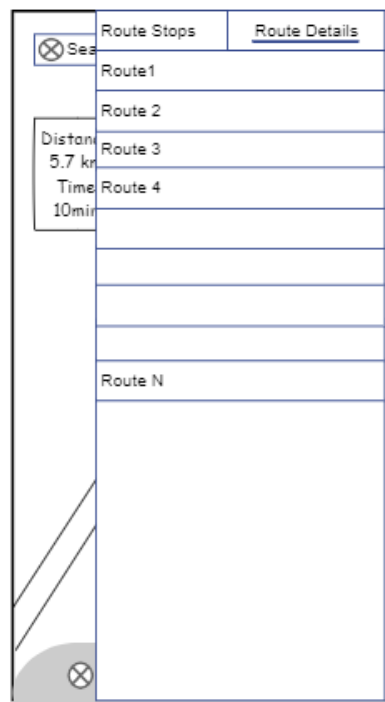
Search Route



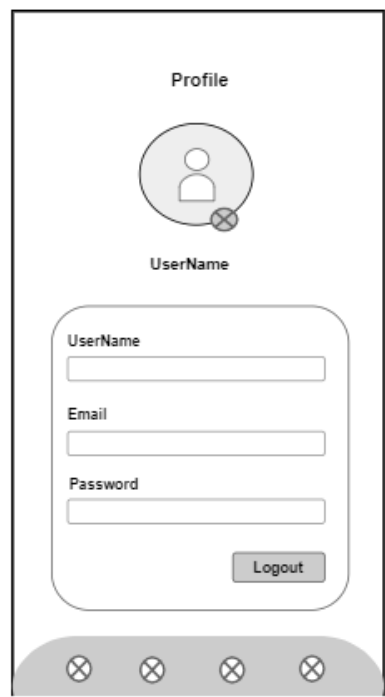
Stops details



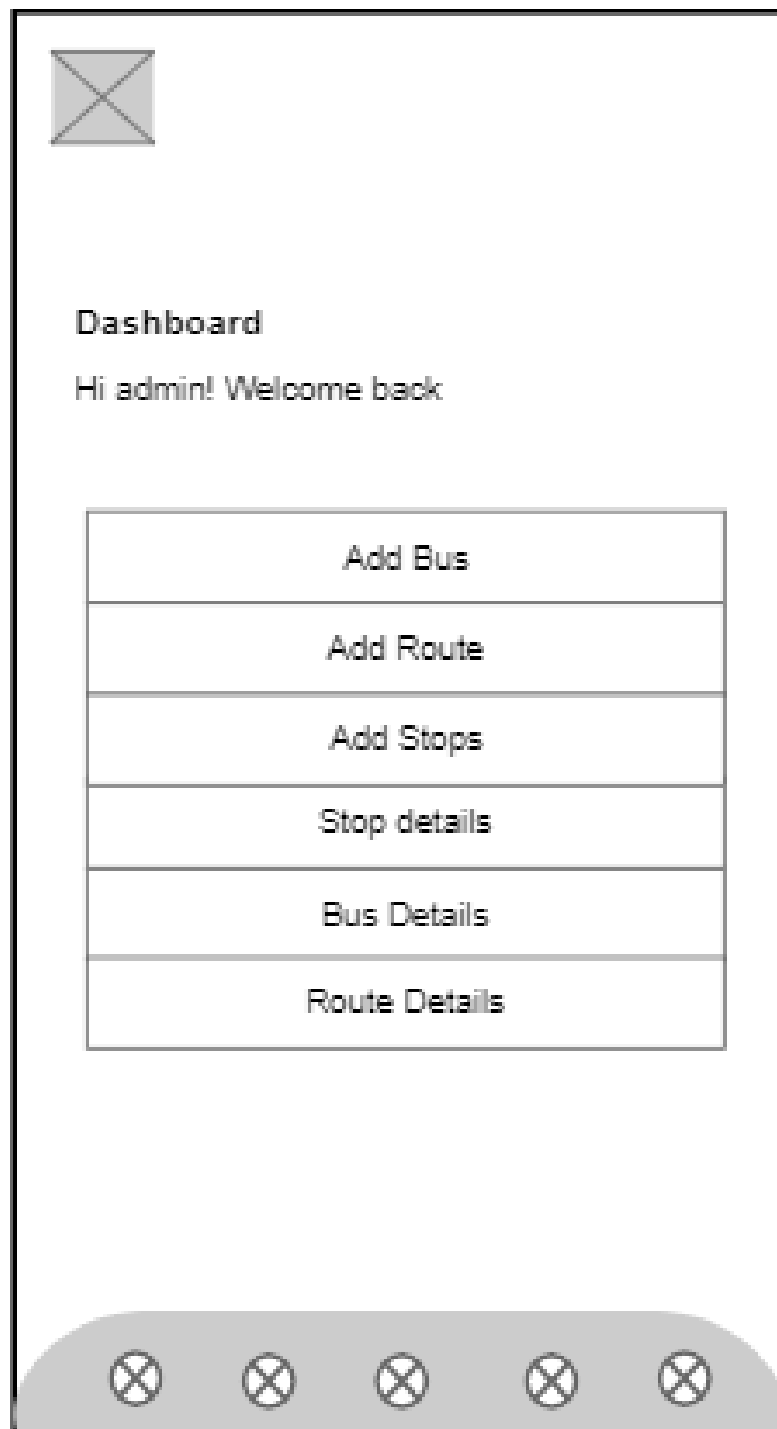
Route Details




Profile



## Dashboard for Admin



Update Route





Update Route



Current Route No


New Route No

Bus No

Update

Add Bus





Add Bus

Bus No

Tracker ID

Add

Add Stop







Add Stop


Route No

Stop Name

Add

Add Route





Add Route

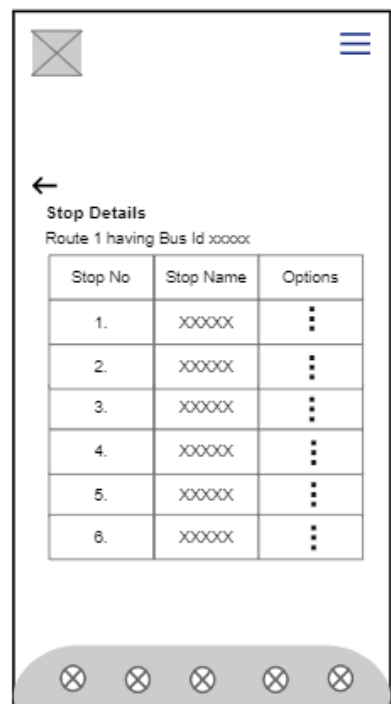
Route No

Bus No

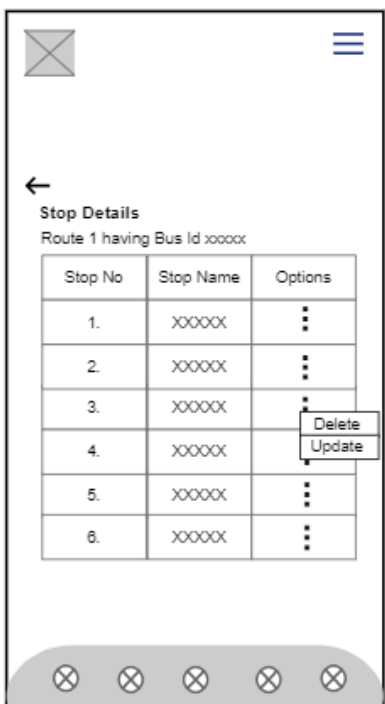
Add



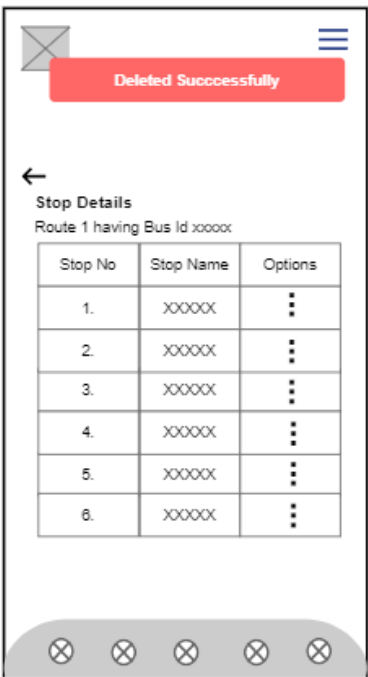
Stop Details



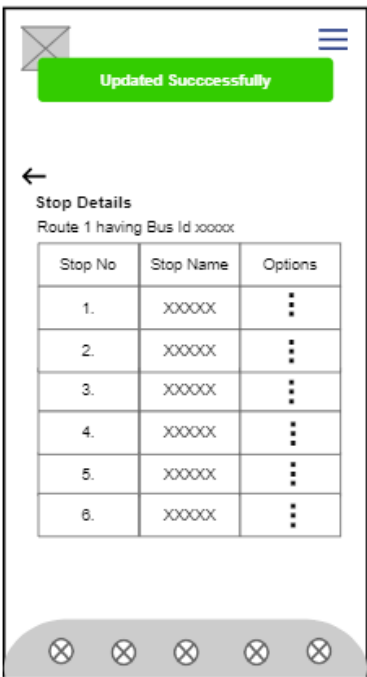
Stop Update/Delete



Stop Delete Successfully

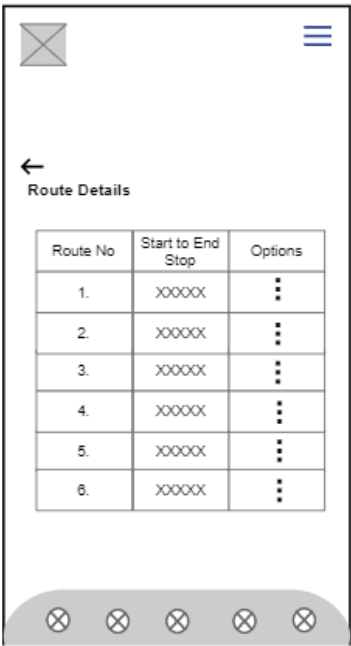


Stop Update Successfully

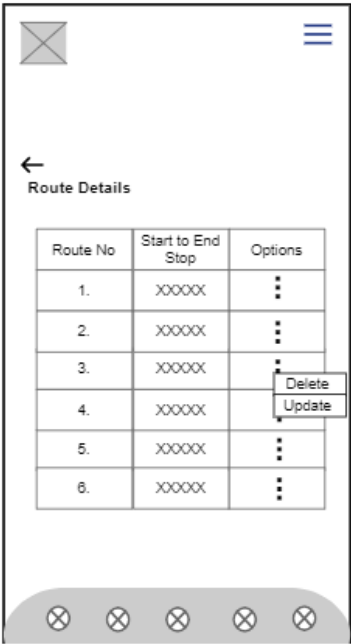




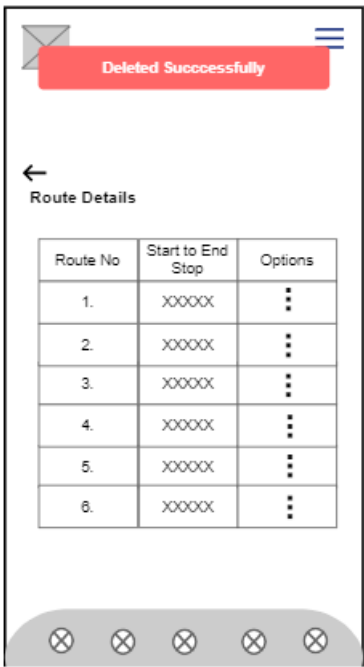
Route Details



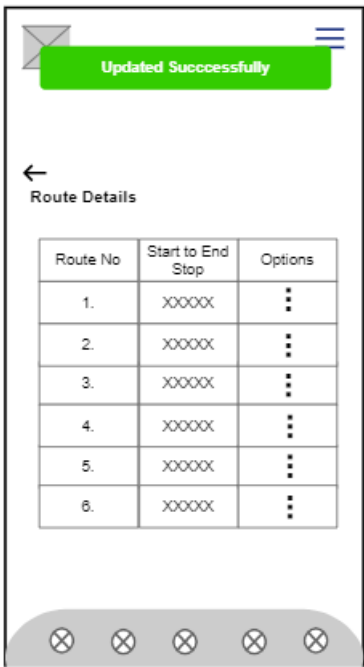
Route Update/Delete



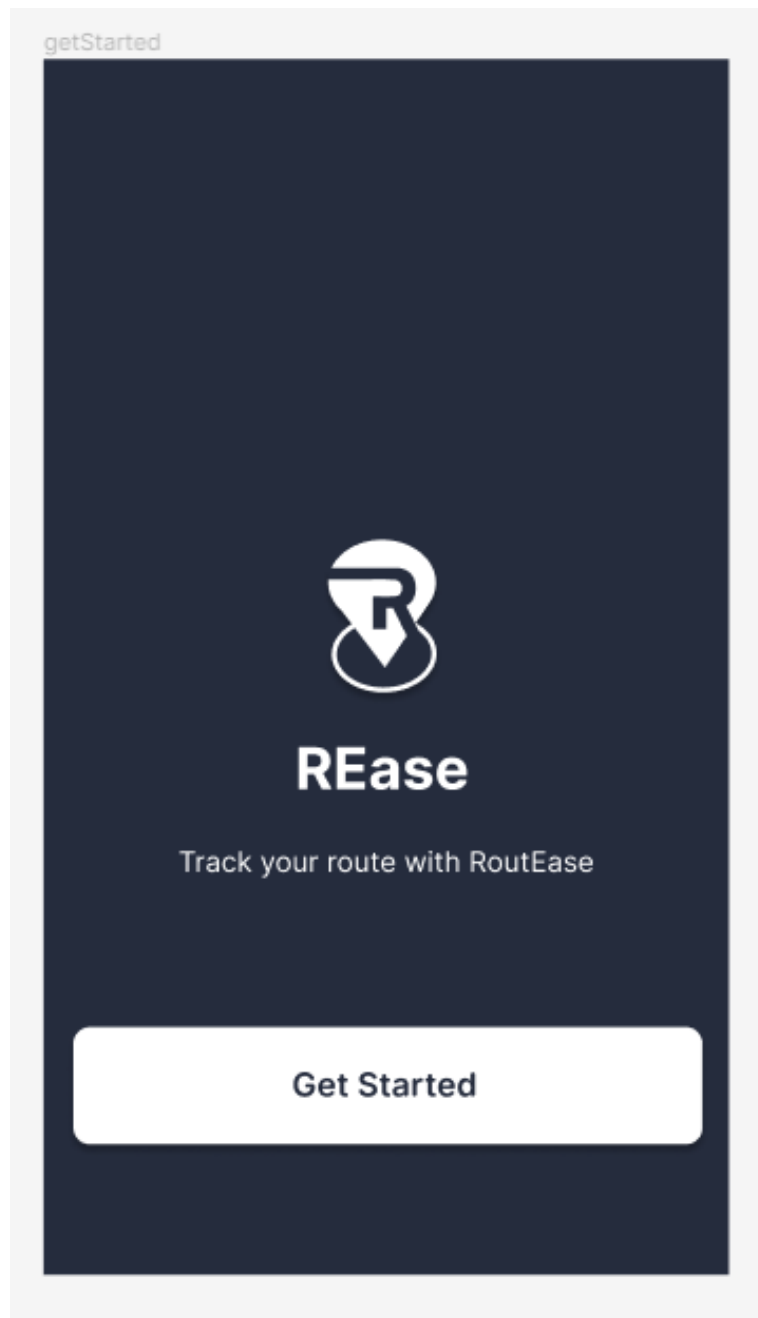
Route Delete Successfully




Route Update Successfully



## 2.8 UI/UX



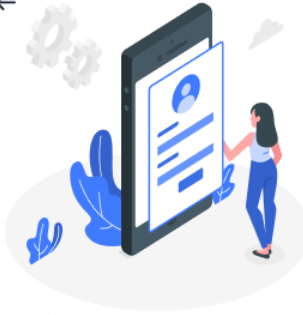
selection



Find Your University route here

Login Sign Up

login



### Login

@ Email


🔒 Password

Forgot Paasword?

Login

New to UET? [Sign Up](#)

forgetpass




### Forgot Password?

Don't worry it happens. Please enter your email address.

@ Email

Submit

forgetpass



### Forgot Password?


Don't worry it happens. Please enter your email address.

@ Email


Invalid Email


Submit

resetpass




**Reset Password**

 New Password


 Confirm Password


Submit

signup



**Sign Up**

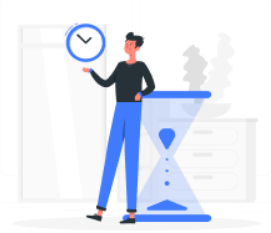
 Email

 Password

Continue

Joined us before? [Login](#)

forgetpass



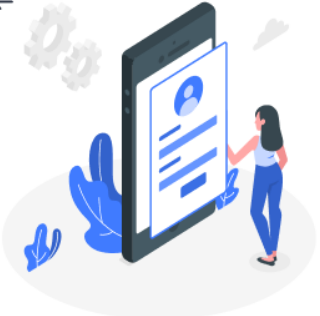
**OTP Verification**

Enter OTP Sent to you at 2020cs\*\*\*\*@uet.edu.pk

Don't receive OTP? [Resend OTP](#)


Verify & Proceed


invalid login



**Login**

Invalid Credentials

 Email

 Password

Forget Password?

Login

New to UET? [Sign Up](#)

