**Project Log Book**

**Group Members:**

|  |  |
| --- | --- |
| **Entry Date** | **Work Done** |
| FEB  8TH, 2021 | Discussed the basic plan to build the prototype for TRM in class, noting down all constraints to be taken care of. Furthermore, we decided our next group meeting would be on 8 FEB, 2021 . |
| September 15th, 2000 | Meeting at Zaida's Place: We discussed about the project objective. Using the Software Management Plan template printed from the web site, we stepped through each section and discussed what was required and what resources were available to us. We also discussed how this prototype should be flexible for other countries. There was constant reference to the "Chinese Railway Passenger Reservation System" and other related articles. |
| September 16th, 2000 | Finished a rough draft prototype and set it up on the online account. |
| September 19th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan, and she made some correction marking the corrections in red. |
| September 20th, 2000 | The mistakes were corrected on the web site, and email was sent to Zaida M. morales to check the document for any more mistakes |
| September 20th, 2000 | The document was checked by Zaida M. Morales and few more mistakes were found. These mistakes were corrected and put on the web. |
| September 22th, 2000 | Meeting at Zaida's Place: We discussed the Reservation System in more detail and added more information to the SPMP document. |
| September 25th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan, and she made some corrections. |
| September 27th, 2000 | The mistakes were corrected on the web site, and email was sent to Zaida M. morales to check the document for any more mistakes. |
| September 29th, 2000 | Meeting at Zaida's Place: We discussed parts 4 and 5 of the Software Project Management Plan in more detail and decided to update some information in the SPMP document. The different parts of the document were divided between the team for updates. |
| October 3th, 2000 | Finished updating the rough draft prototype and set it up on the online account. Sent all team members email with link to latest copy of the document. |
| October 4th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan. The mistakes were corrected on the web site. The latest version of the document is available online. |

**Software Requirements Specification**

**For**

# TRANSPORT MANAGEMENT SYSTEM

**Prepared by**

**Marium Jahan  
Rida Usman  
Sakina Marium  
Aqsa Khan  
Nabiya Afzal  
Nimra Jamil**

**8th Feb 2021**

|  |  |  |
| --- | --- | --- |
| 1.0 | TRM Review Version | 08/02/2021 |
|  | | |

## Table of Contents

### 1. Introduction

1.1. Purpose

1.2. Product Scope

1.3. References

### 2. Overall Description

2.1. Product Perspective

2.2. Product Functions

2.3. User Classes and Characteristics

2.4. Operating Environment

### 3. Functional Requirements

3.1. User Story

3.2. Use-Cases

### 4. Flow of Events

4.1. Flow of Events for TICKET BOOKING use-case

4.2. Flow of Events for MAINTAIN PASSENGER INFO use-case

4.3. Flow of Events for MAINTAIN BUS INFO use-case

4.4. Flow of Events for MAINTAIN EXPENSES use-case

4.5. Flow of Events for MAINTAIN ROUTES use-case

4.6. Flow of Events for MAINTAIN DRIVER INFO use-case

4.7. Flow of Events for MAINTAIN SCHEDULE use-case

### 5. Non-Functional Requirements

5.1. Safety Requirements

5.2. Security Requirements

### 6. Diagrams

6.1. Use-Case Diagram

6.2. Use-Case Realization

6.3. Sequence Diagram

6.4. Activity Diagram

6.5. Class Diagram

6.6 State Chart Diagram

### 7. User Interface

7.1. Login Form

7.2. Sign up Form

## 1. Introduction

### 1.1 Purpose

This document is meant to delineate the features of the Transport Management System.

It will explain the purpose and features of the software and what the software will do. This document is intended for users of the software and also potential developers.

### 1.2 Product Scope

Transport management system (TMS) is used to book a ticket easily along with viewing all travel details. The user can save information in the database allowing the admin to add, update and delete information.

**1.3 References**

• SRS Template for Gephi

## 2.Overall Description

### 2.1 Product Perspective

The Transport management system application enables passengers to get online Bookings, and administrator to maintain information. Also, the developer is designing this to help users book tickets without having to go through much trouble.

### 2.2 Product Functions

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be -

#### 2.2.1 Registration

If a person wants to Book a ticket or become a driver, then he/she would have to register.

##### 2.2.2 Login

Passenger/admin/driver logins to the system by entering valid email and password in order to book a ticket, maintain information or check schedule.

##### 2.2.3 Ticket Bookings/Cancelations 2.2.4 Maintain Information

The Passenger after login can book tickets as well as cancel tickets.

The Admin has the ability to maintain all information which includes passenger, driver, bus, expenses schedules and routes.

### 2.3 User Classes and Characteristics

#### 2.3.1 Administrator

Administrators, after logging in are allowed to view information and maintain information, meaning they can add, update or delete information regarding passengers, drivers, schedules.

#### 2.3.2 Passenger

Passengers can book tickets as well as cancel any previously booked ticket. He/she can check all travel detail as well.

#### 2.3.3 Driver

A driver can register himself into the system, this would allow him to check or print schedules

### 2.4 Operating Environment

* Windows 8
* Windows 10
* MySQL

## 3.Functional Requirements

### 3.1 User Story

Suppose that a customer enters the transport office wanting to book a ticket for himself for Islamabad one week from today. The receptionist checks the availability of buses on the day you insist on traveling. If a bus is unavailable the receptionist requests to consider another day of travel or another time on the same day or maybe another, and incase a bus is available the receptionist asks the customer for his personal details for example; First name, Last name, CNIC, Date of Birth etc. After entering passenger information, the receptionist discusses the travel details which includes the kind of bus, the driver name, the route that will be followed, the stops etc. After this is discussed and decided by the customer, the receptionist requests to reconfirm all the personal information. Once that is confirmed the receptionist saves the info and prints out the ticket which includes all the info as well as the travel fee, seat number and bus number. As soon as the save button is pressed the passenger information is saved in the database.

If a person comes in stating that he wants to register as a driver, then the receptionist requests to hand in all the documentation for example; education certificate, medical certificate, driver license etc. Once he has handed in the requested documentation, the receptionist asks him to fill in the form for registration which includes all of his personal information. After the form is submitted along with the certificates, the receptionist informs that he will be contacted via message if chosen. If he is qualified, then he is requested to come in again to pick up his schedule that will be already made. Later on, the schedule will be sent to him via mail or text. The details of the driver will be saved in the database

If a particular bus is selected for travel then that bus will go through necessary check-ups and then made available on the system. The system shows how many seats are there in total in that bus and how many are remaining or have been booked. Each bus will have its own specific number. this information will also be saved in the database.

If at any moment someone wants to print the information of the buses or drivers or the passengers, they can print it since all is save in the database.

### 3.2 Use-Cases

* Login (Admin/Passenger/Driver)
* Ticket Book (Passenger)
* Availability of bus (Passenger)
* Availability of ticket (Passenger)
* Add passenger (Passenger/Admin)
* Update passenger (Admin)
* Delete passenger (Admin)
* Check travel details (Passenger/Admin)
* Save information (Passenger)
* Print Ticket (Passenger)
* Payment (Passenger)
* Register driver (Driver)
* Delete Driver (Admin)
* Update driver information (Admin)
* Save driver information (Admin)
* Send Message / Email (Admin)
* Create Schedule (Admin)
* Update schedule (Admin)
* Delete schedule (Admin)
* Expenses (Admin)
* Add bus (Admin)
* Delete bus (Admin)
* Save buses information (Admin)
* Add travel routes (Admin)
* Update travel routes (Admin)
* Delete travel routes (Admin)
* Select bus types (Passenger)
* Check schedule (driver)
* View buses (Admin)
* Vies Passenger (Admin)
* Check Routes (Admin)
* View Driver (Admin)
* Cancel Ticket (Admin/Passenger)

## 4. FLOW OF EVENTS

### 4.1 Flow of Events for TICKET BOOKING Use-Case

**4.1.1 Preconditions**

The Passenger should be logged onto the system for this use-case to begin.

#### 4.1.2 Main Flow

This use-case begins when a passenger logs into the system wishing to book a ticket. The passenger enters his/her email and password, the system verifies it. The system them prompts the passenger to select the desired activity: BOOK, CANCEL or PRINT.

If the activity selected is BOOK, the S-1: Book a ticket sub-flow is performed.

If the activity selected is CANCEL, the S-2: Cancel a ticket sub-flow is performed. If the activity selected is PRINT, the S-3: Print a ticket sub-flow is performed.

#### 4.1.3 Sub-Flows

##### S-1: BOOK a ticket

The system displays the ticket booking form containing a field of source and destination. The passenger enters his/her source and destination of travel. After the passenger is done, he submits this successfully booking a ticket. The use-case starts again.

##### S-2: CANCEL a ticket

The system retrieves and displays the previously booked ticket. The passenger verifies the cancelation of the ticket. The system cancels the previously booked ticket.

***S-3: PRINT a ticket***

The system prints the ticket that has been booked by the passenger.

#### 4.1.4 Alternate Flows

***E-1*:** An invalid email or password is entered. The passenger can re-enter the email or password.

***E-2:*** No route might be present to or from the desired source or destination. So, the ticket cannot be booked.

***E-3:*** The previously booked ticket cannot be canceled; the passenger is informed that the option might not be available at the moment. Or, the passenger decides to not cancel the ticket, so the cancelation withdraws.

***E-4:*** The ticket cannot be printed. The passenger is informed that the option may not be present at the moment.

### 4.2 Flow of Events for MAINTAIN PASSENGER INFORMATION Use-Case

**4.2.1 Preconditions**

The Admin should be logged onto the system for this use-case to begin.

#### 4.2.2 Main Flow

This use-case begins when the admin logs into the system providing the admin with the following activities:

ADD, UPDATE and DELETE.

If the activity selected is ADD, the S-1: Add passenger sub-flow is performed.

If the activity selected is UPDATE, the S-2: Update passenger sub-flow is performed. If the activity selected is DELETE, the S-3: Delete passenger sub-flow is performed.

#### 4.2.3 Sub-Flows

##### S-1: ADD a passenger

The system requests that the admin enter the passenger information, which includes:

* First Name
* Last Name
* Contact
* CNIC

Once the admin provides the requested information, the system generates and assigns a unique id to the passenger. The passenger information is saved in the system.

##### S-2: UPDATE a passenger

The system requests to enter the passenger id or first name of the passenger to retrieve the information of that particular passenger. The admin makes the desired changes in the information including the one provided during the **Add a passenger** sub-flow.

Once the information is successfully changed, the system saves the updated information.

##### S-3: DELETE a passenger

The system retrieves a particular passenger based on his/her passenger id or first name. The system verifies the deletion of the passenger information. The passenger information is then deleted from the system.

#### 4.2.4 Alternate Flows

***E-1*:** An invalid email or password is entered. The passenger can re-enter the email or password.

***E-2***: In the Delete a passenger and Update a passenger sub-flows, the passenger id or first name does not exist. In this case, the admin can search the passenger regarding his/her last name

***E-3***: If the admin decides not to delete the passenger information, the deletion is terminated and this use case begins again.

### 4.3 Flow of Events for MAINTAIN BUS INFORMATION Use-Case

**4.3.1 Preconditions**

The Admin should be logged onto the system for this use-case to begin.

#### 4.3.2 Main Flow

This use-case begins when the admin logs into the system providing the admin with the following activities:

ADD and DELETE.

If the activity selected is ADD, the S-1: Add bus sub-flow is performed. If the activity selected is DELETE, the S-2: Delete passenger sub-flow is performed.

#### 4.3.3 Sub-Flows

##### S-1: ADD bus

The system requests to enter some basic information about the bus, which includes:

* Registration plate number
* Model Number

Once the admin provides the requested information, the system generates and assigns a unique id to the bus. The bus information is saved in the system.

##### S-2: DELETE bus

The system retrieves the information of the desired bus based on the bus id. The system verifies the deletion of the bus information. The bus information is then completely erased from the system.

#### 4.3.4 Alternate Flows

***E-1:*** An invalid email or password is entered. The passenger can re-enter the email or password.

***E-2:*** the total limit of buses owned has been fulfilled, so the admin does not add new bus information rather puts it in the pending list.

***E-3:*** During the monthly check-up, a particular bus is not certified for further use. The admin wishes to delete that bus’s information from the system but is unable to do so.

### 4.4 Flow of Events for MAINTAIN EXPENSES Use-Case

**4.4.1 Preconditions**

The Admin should be logged onto the system for this use-case to begin.

#### 4.4.2 Main Flow

This use-case begins when the admin logs into the system providing the admin with the following activities:

* MAINTAIN END-OF-MONTH OIL EXPENSES
* MAINTAIN END-OF-MONTH MAINTENANCE EXPENSES
* MAINTAIN END-OF-MONTH SALARIES

If the activity selected is MAINTAIN END-OF-MONTH OIL EXPENSES, the S-1: Maintain end-of-month oil expenses sub-flow is performed.

If the activity selected is MAINTAIN END-OF-MONTH MAINTAINANCE EXPENSES, the S2: Maintain end-of-month maintenance expenses sub-flow is performed.

If the activity selected is MAINTAIN END-OF-MONTH SALARIES, the S-2: Maintain end-ofmonth salaries sub-flow is performed.

#### 4.4.3 Sub-Flow

##### S-1: MAINTAIN END-OF-MONTH OIL EXPENSES

The system retrieves the report of all the oil expenses for the month before forwarding to the accounts department.

##### S-2: MAINTAIN END-OF-MONTH MAINTENANCE EXPENSES

The system retrieves the report of all the maintenance expenses for the month before forwarding to the accounts department

##### S-3: MAINTAIN END-OF-MONTH SALARIES

The system retrieves the report of all the monthly salaries before forwarding to the accounts department

#### 4.4.4 Alternative Flow

***E-1:*** An invalid email or password is entered. The passenger can re-enter the email or password.

### 4.5 Flow of Events for MAINTAIN ROUTES Use-Case

**4.5.1 Preconditions**

The Admin should be logged onto the system for this use-case to begin.

#### 4.5.2 Main Flow

This use-case begins with the admin logging into the system. After verification, the system gives access to the admin to perform the following activities:

ADD, UPDATE AND DELETE.

If the activity selected is ADD, the S-1: Add route sub-flow is performed.

If the activity selected is UPDATE, the S-2: Update routes sub-flow is performed. If the activity selected is DELETE, the S-3: Delete routes sub-flow is performed.

#### 4.5.3 Sub-Flows

***S-1 ADD routes:***

The system displays all possible routes for the passenger containing all details, based on the source and destination. The use case begins again.

***S-2 UPDATE routes:***

The system requests to create a new route based on given source and destination. After creating the new route, the system saves the updated route.

**S-3 DELETE routes:**

The system retrieves the route based on the source and destination. The system verifies the deletion of the route. The route is then completely erased from the system.

#### 4.5.4 Alternative Flows

**E-1:** An invalid email or password is entered. The passenger can re-enter the email or password.

**E-2**: While adding a new route, the source or destination entered might be incorrect. **E-3:** A particular route cannot be deleted.

#### 4.6 Flow of Events for MAINTAIN DRIVER INFORMATION Use-Case

**4.6.1 Preconditions**

The use-case begins when the admin logs onto the system.

##### 4.6.2 Main Flow

This use-case begins when the admin logs into the system providing the admin with the following activities:

ADD, UPDATE and DELETE.

If the activity selected is ADD, the S-1: Add a driver sub-flow is performed.

If the activity selected is UPDATE, the S-2: Update a driver sub-flow is performed. If the activity selected is DELETE, the S-3: Delete a driver sub-flow is performed.

##### 4.6.3 Sub-Flow

###### S-1: ADD a driver

The system requests that the admin enter the driver information, which includes:

* First Name
* Last Name
* Contact
* CNIC

Once the admin provides the requested information, the system generates and assigns a unique id to the driver. The driver information is saved in the system.

###### S-2: UPDATE a driver

The system requests to enter the driver id to retrieve the information of that particular driver. The admin makes the desired changes in the information including the one provided during the **Add a driver** sub-flow.

Once the information is successfully changed, the system saves the updated information.

###### S-3: DELETE a driver

The system retrieves the particular driver based on his/her driver id. The system verifies the deletion of the driver information. The driver information is then deleted from the system.

##### 4.6.4 Alternative Flow

***E-1*:** An invalid email or password is entered. The passenger can re-enter the email or password.

***E-2***: In the Delete a driver and Update a driver sub-flows, the driver id does not exist. In this case, the admin can search the driver regarding his last name

***E-3***: If the admin decides not to delete the driver information, the deletion is terminated and this use case begins again.

### 4.7 Flow of Events for MAINTAIN SCHEDULE Use-Case

**4.7.1 Preconditions**

The use-case begins when the admin logs onto the system.

#### 4.7.2 Main Flow

This use-case begins when the admin enters all necessary information needed, which includes day, time, source destination, bus id and driver name. The system gives access to the admin to perform the following activities:

CREATE, UPDATE, DELETE and SEND.

If the activity selected is CREATE, the S-1: Create schedule sub-flow is performed.

If the activity selected is UPDATE, the S-2: Update schedule sub-flow is performed.

If the activity selected is DELETE, the S-3: Delete schedule sub-flow is performed. If the activity selected is SEND, the S-4: Send schedule sub-flow is performed.

#### 4.7.3 Sub-Flow

##### S-1: CREATE schedule

The route is already created beforehand, the admin assigns a driver to that particular route. A schedule is created.

##### S-2: UPDATE schedule

If a route is updated, the admin updates it the schedule as well after the system retrieves it.

##### S-3: DELETE schedule

*The system retrieves the schedule.* The system verifies the deletion of schedule. The schedule is then deleted from the system.

##### S-4: SEND schedule

After the schedule is created and saved in the system, a copy is sent to the respective driver.

#### 4.7.4 Alternative Flow

***E-1*:** An invalid email or password is entered. The passenger can re-enter the email or password.

***E-2***: If the admin decides not to delete the schedule, the deletion is terminated and this use case begins again.

## 5. Non-Functional Requirements

### 5.1 Safety Requirements

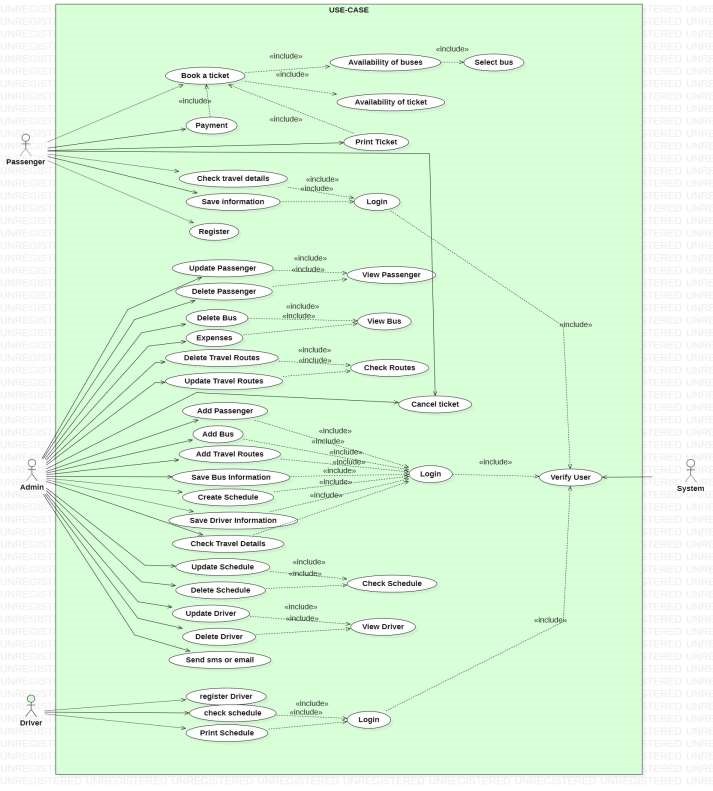
To ensure that there is no data loss (due to a crash or a bug of some kind) the developer team updates the system regularly. There is a bug tracker available where users can report any bugs, that they have encountered so that the developers can fix it in the next release.

### 5.2 Security Requirements

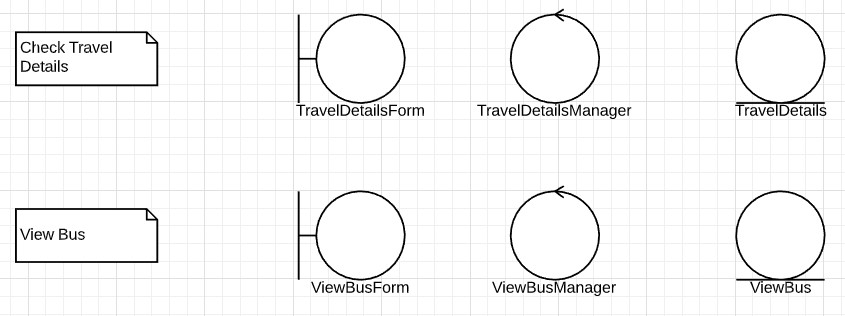
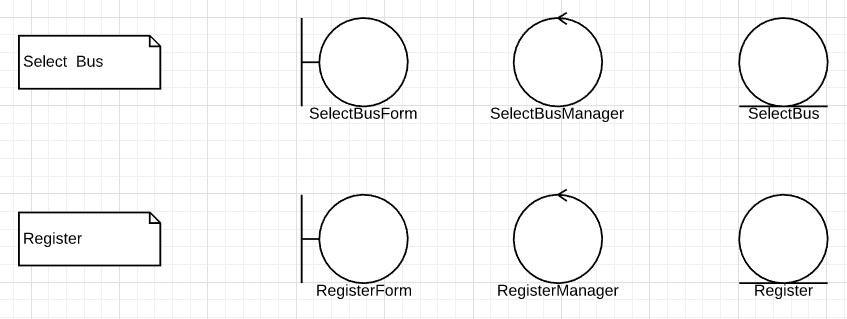
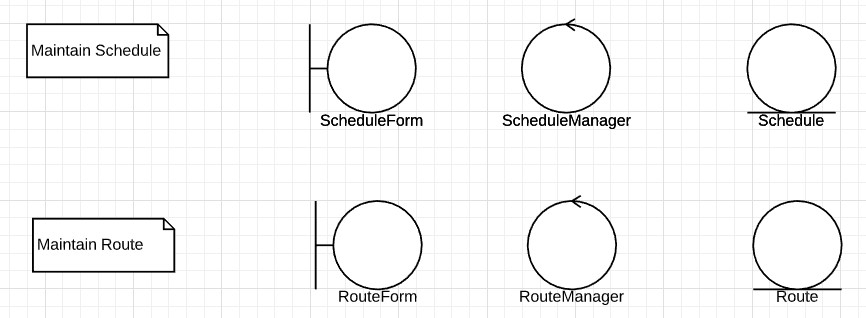
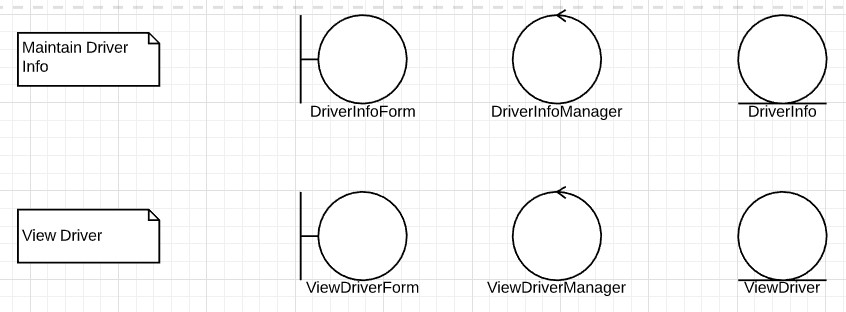
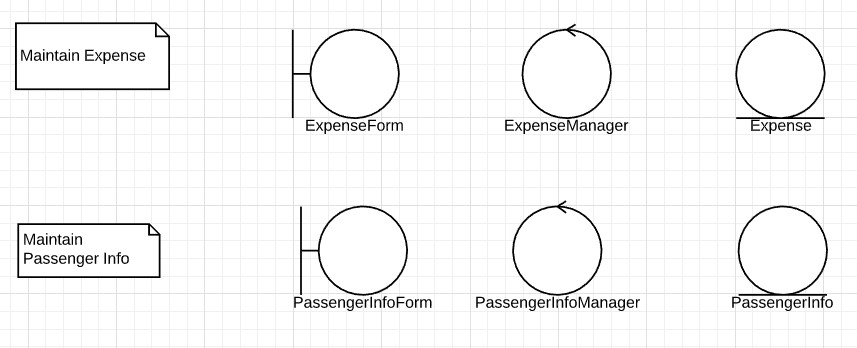
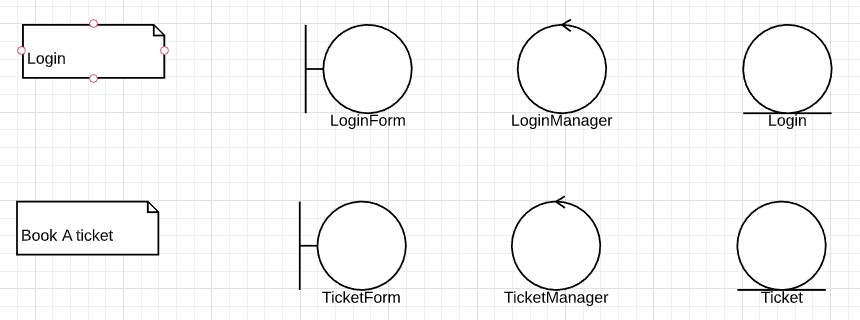
There are not any security requirements so any type of user can use it without any additional privileges

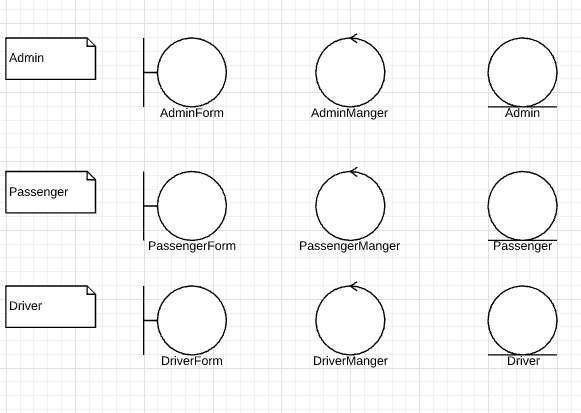
## 6. Diagrams

**6.1 Use-Case Diagram**



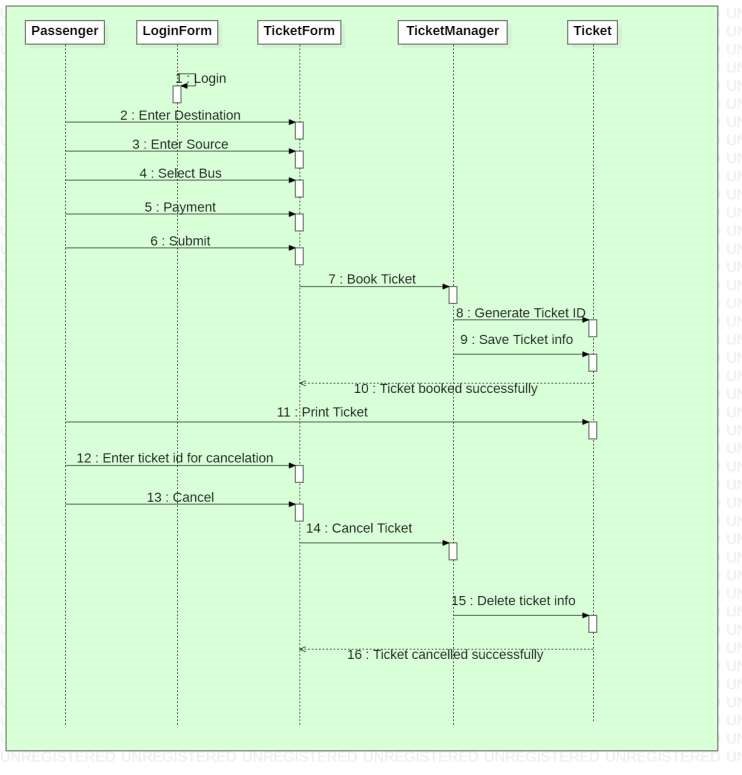
### 6.2 Use-Case Realization



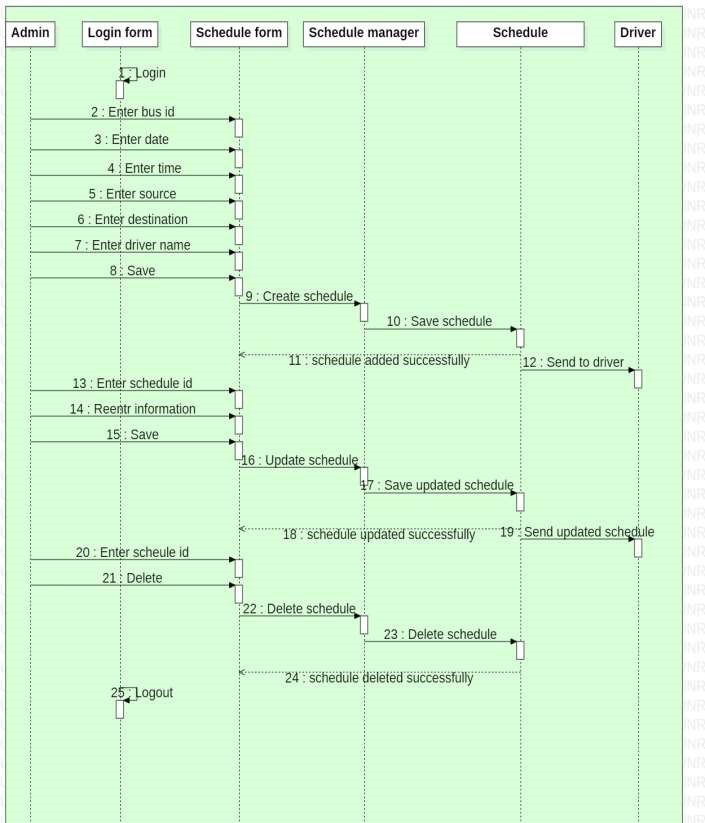


**6.3 Sequence Diagram**

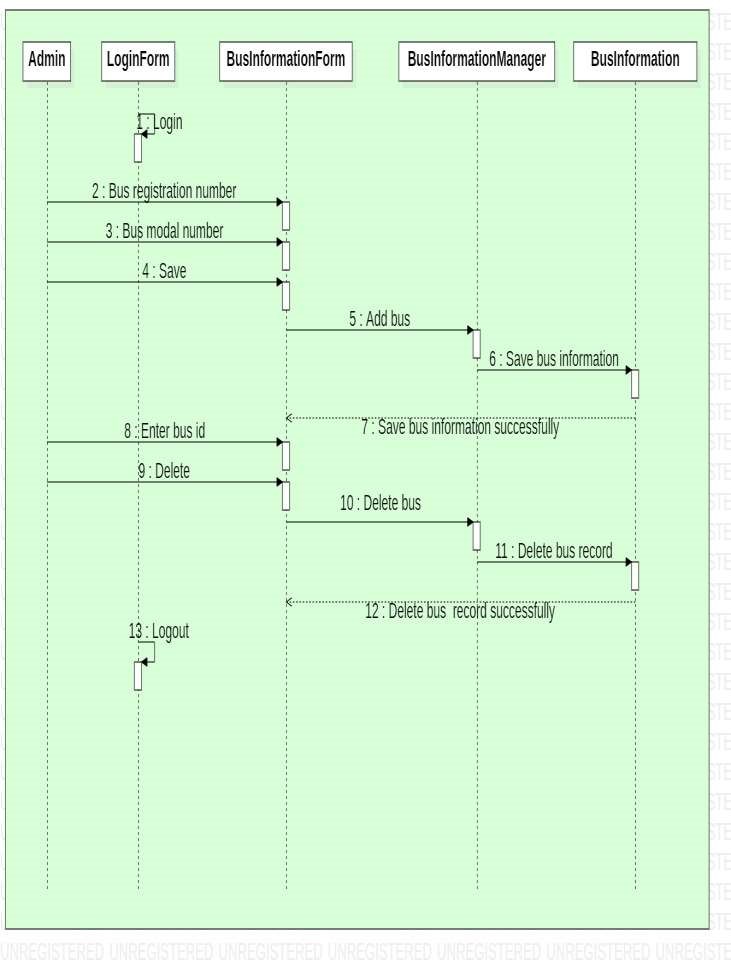
1. **Book ticket**



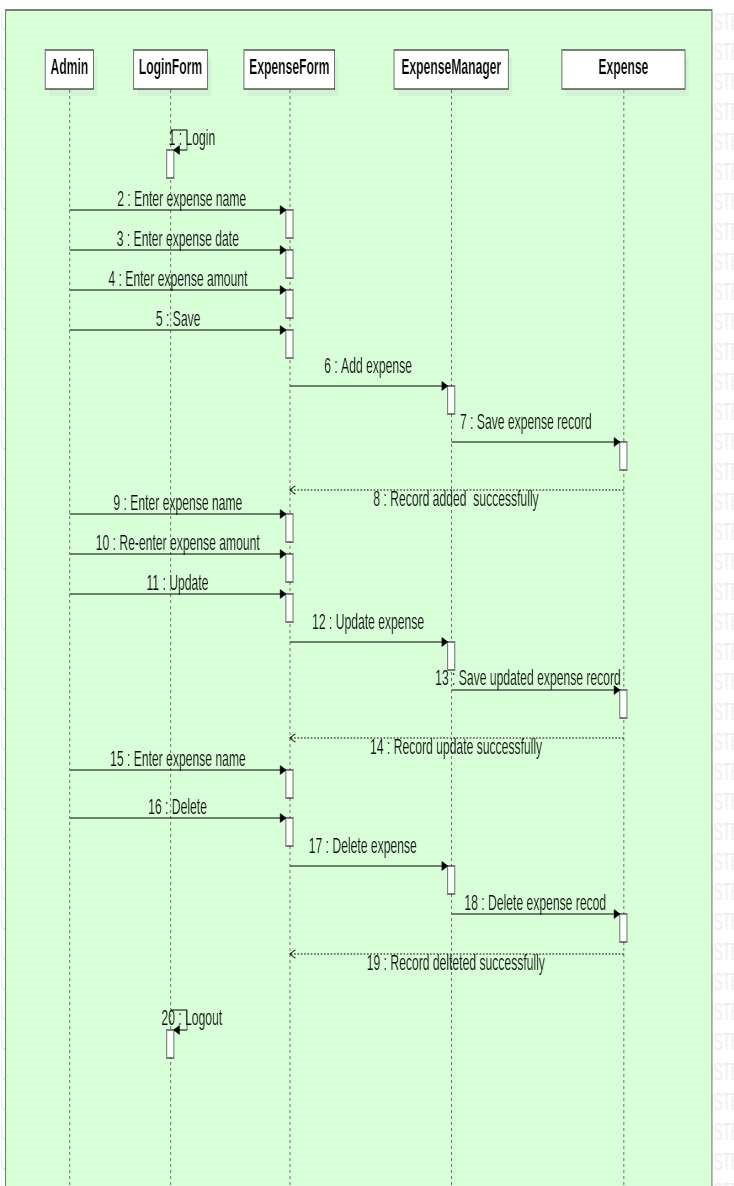
1. **Maintain Schedule**



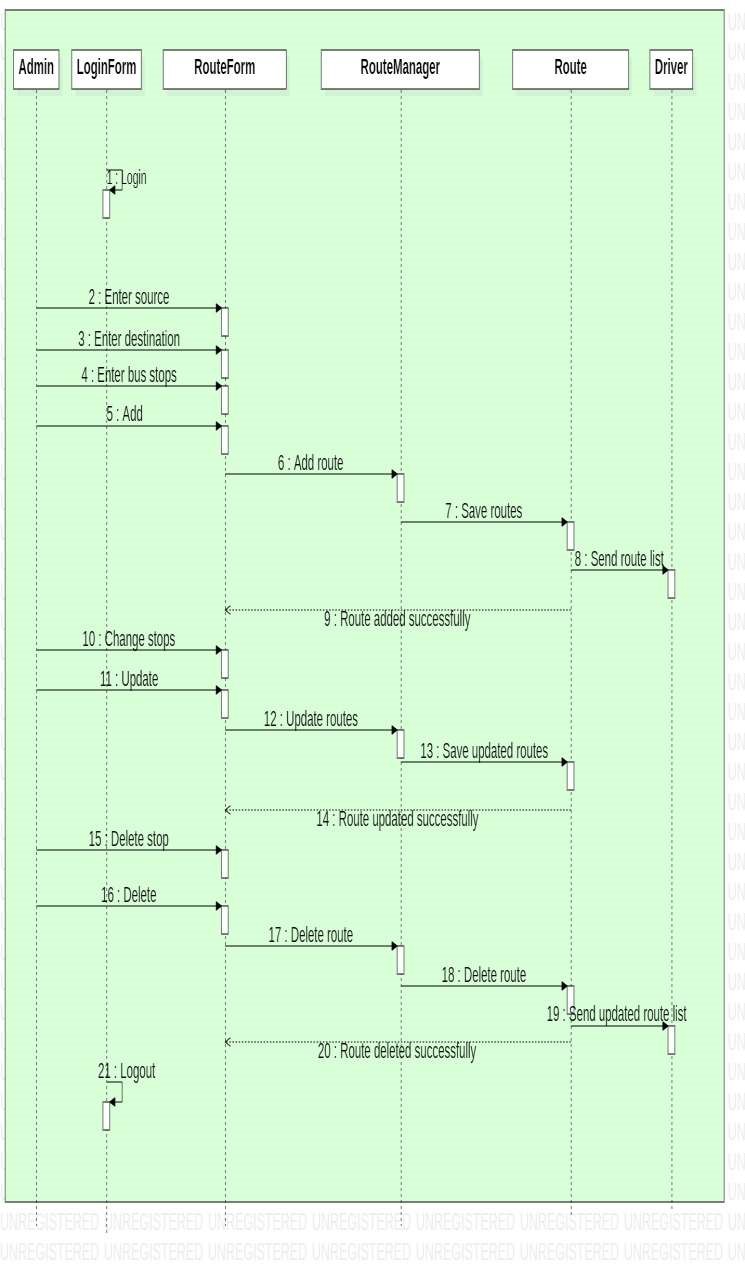
1. **Maintain Bus Info**



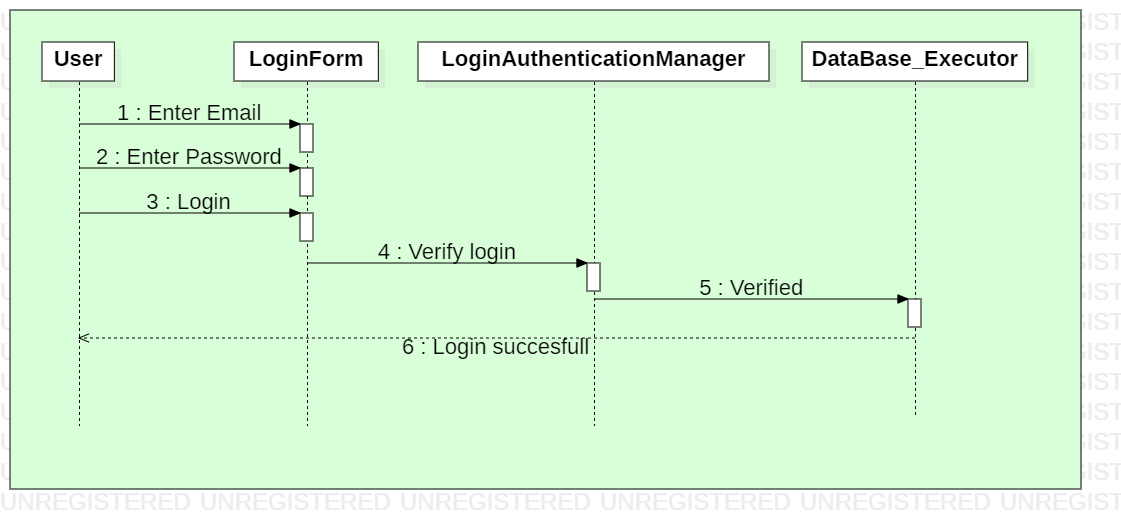
1. **Maintain Expenses**



1. **Maintain Route**

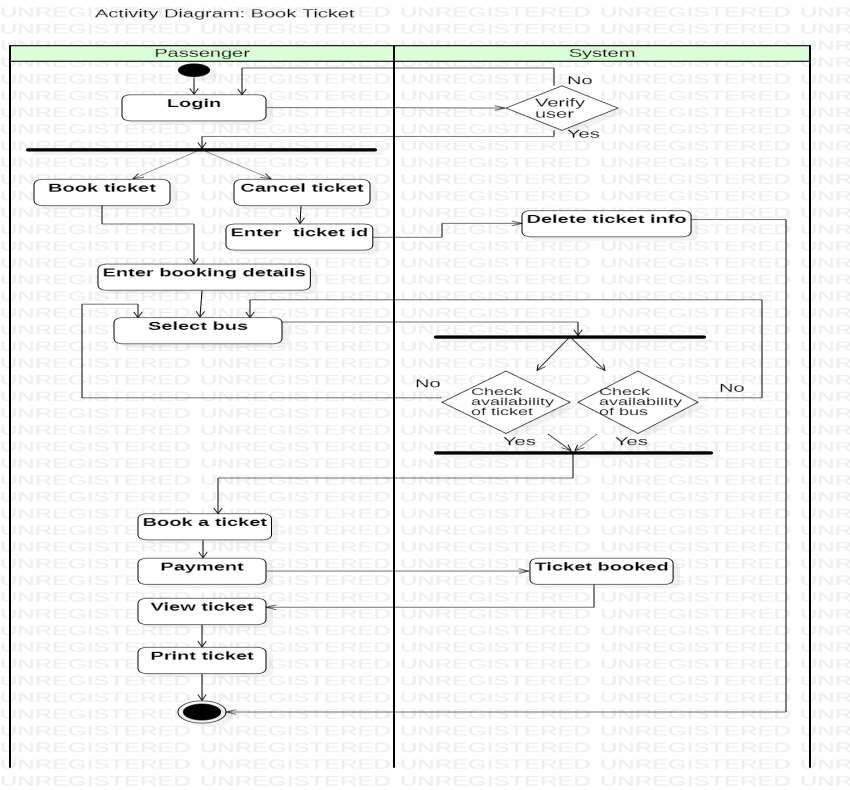


1. **Login**

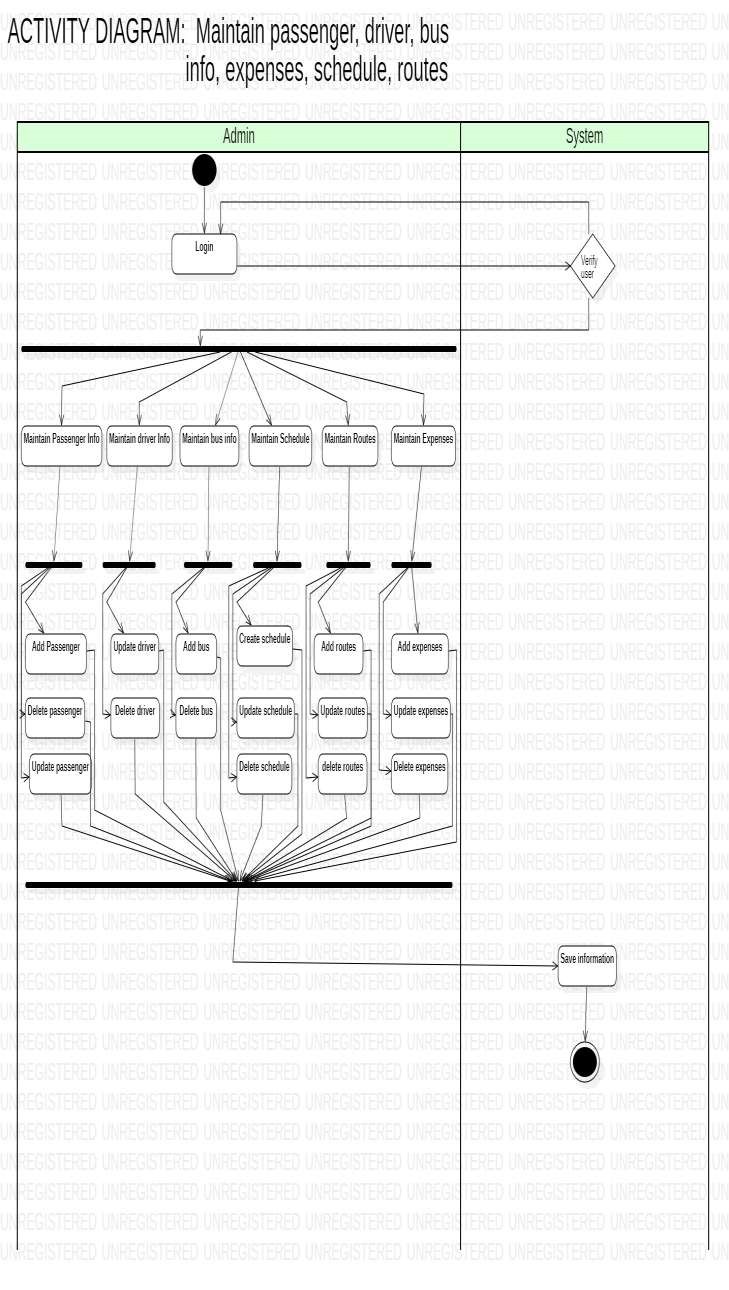


### 6.4 Activity Diagram

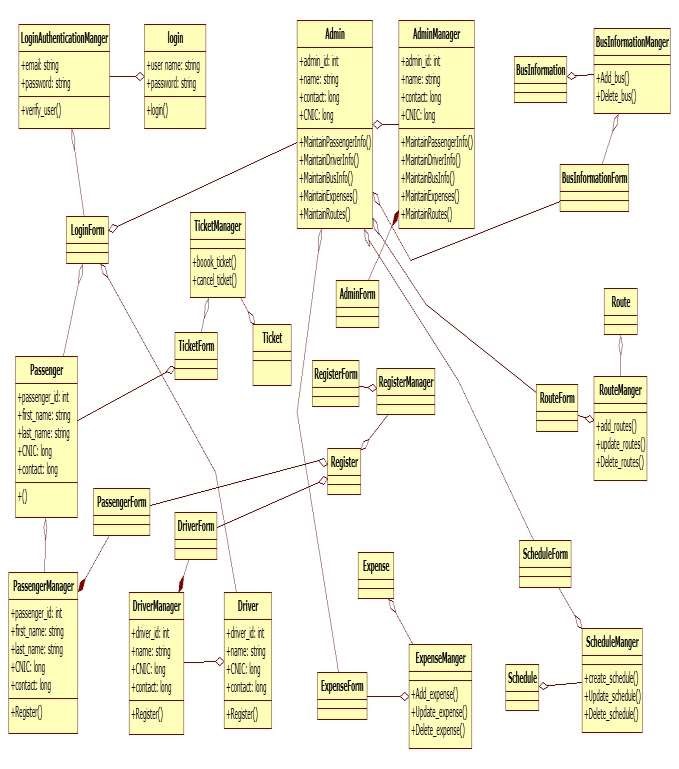
1. **Book a Ticket**



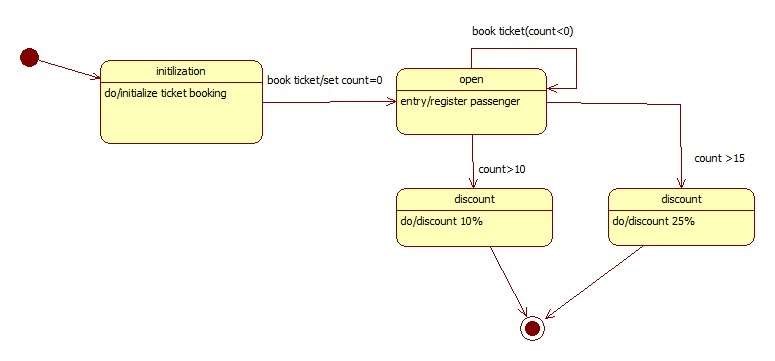
1. **Maintain Passenger, Driver, Bus Info, Expenses, Schedules, routes**



### 6.4 Class Diagram



### 6.5 State Chart Diagram



**7. User Interface**

**7.1. Login Form**

****

**7.2. Sign up Form**

****