

CS 385 T Software Engineering Term Project

Title: BusMap

Section: 5C2

Group no. 4

No.	Student Name	Student Number
1		
2		
3	مجد محمد العمري	443007585
4		
5		

First Semester 2023

Write your name next to section(s) you worked on, or state you all shared the work by ticking on the following statement box. (You will still share responsibilities even if you state your contribution)

☒ **We all share the work and responsibility of the project.**

PLO	Student Name	FOR INSTRUCTOR'S USE ONLY		
		Max. Mark	Mark Obtain	Notes
Cover and Table of Contents		0.5		
Project Description				
Functional Requirements (user and system requirements)		2		
Non-Functional Requirement				
Context Diagram with description		0.5		
Use-case Diagram (for the whole system)		1		
Scenario (MSS & Extension) <i>For Use Case 1:</i> create passenger account.		1		
Scenario (MSS & Extension) <i>For Use Case 2:</i> provide info.				
Scenario (MSS & Extension) <i>For Use Case 3:</i> assign buses.				
Scenario (MSS & Extension) <i>For Use Case 4:</i> Track preferences.				
Class Diagram (for the whole system)		1		
Sequence Diagram <i>For Function 1:</i> change house location.		1		
Sequence Diagram <i>For Function 2:</i> view attendance sheet.				
Sequence Diagram <i>For Function 3:</i> creating passenger account.				
Sequence Diagram <i>For Function 4:</i> view user info.				
Activity Diagram <i>For Function 1:</i> [Name the function]		1		
Activity Diagram <i>For Function 2:</i> [Name the function]				
Activity Diagram <i>For Function 3:</i> [Name the function]				
Activity Diagram <i>For Function 4 :</i> [Name the function]				
State Diagram (If any) <i>For Object 1 :</i> [Name the object]		1		
System Architecture				
Total Score				
		9		

Contents

1. Chapter 1	5
System descriptions	5
The development Process model used	5
System Requirements.....	5
Functional Requirements.....	5
Non-Functional Requirements.....	7
2. Chapter 2.....	8
Context Diagram	8
System Use-case Diagram with description	9
Use case scenario (MSS +Extensions) (one for each student).....	16
Use case 1: create passenger account.....	16
Goal: login application.	16
Actor: passenger.	16
Precondition: choose to create a passenger account.	16
Main success scenario:	16
1-passenger enters their email and password.	16
2. System will verify if the account is already registered.	16
3. System will validate email and password.	16
4. System will authorize the passenger to use their account.....	16
Post condition: successful login.....	16
Extensions:	16
1a. Passenger enters wrong password or email.....	16
1-The system fails validation.	16
Use case 2: provide info.....	16
Use case 3: assign buses.	17
Goal: to assign each buss to a driver.	17
Actor: manager.	17
Main success scenario:	17
1- the manager should login to their managing system account.....	17
2- view the driver information.....	17
3- assign each bus to a particular driver.....	17
Post condition: The system should notify each driver with their assigned bus.	17
Extension:	17
3a. If the provided bus is not available.....	17

1. The driver cannot use that bus.....	17
Use case 4: Track preferences.	17
Goal: check the driver’s preferences.....	17
Precondition: having access to the live location feature.	17
Main success scenario:	17
1- the manager should login to their managing system account.....	17
2- the manager should open the driver live location.	17
3-manager should check if the driver is arriving at time and driving safely.	17
Post condition: a preference report should be sent to the driver.	17
Extension:	17
2a. the live location cannot be accessed.....	17
1. The driver’s preferences cannot be checked.	17
3. Chapter 3	18
System Class Diagram	18
Sequence Diagrams:	19
Sequence diagram for change house location.	19
Sequence diagram for view attendance sheet.....	20
Sequence diagram for creating passenger account.	20
Sequence diagram for view user info.	21
4. Chapter 4	21
Activity Diagram (for at least 4 functions)].....	21
Activity Diagram for [insert a title]	21
Activity Diagram for [insert a title]	21
Activity Diagram for [insert a title]	21
Activity Diagram for [insert a title]	22
State Diagram	22
State Diagram for [insert a title]	22
5. Chapter 5.....	21
Architecture Diagram	22

1. Chapter 1

System descriptions

The bus map is an application that government agencies such as the Ministry of Transport benefit from, as it makes it easier for the driver to accurately arrange passenger locations with the help of artificial intelligence and geolocation algorithms. It mainly targets university students and also school students, and can be linked to the platforms of these parties. Users must create an account with their ID or email used on the Madrasati platform, then enter the home location, the student's name and their phone number.

The application will save the driver time and confusion, it's safer for all. It will make sure that the passengers arrive to their destinations in the shortest time possible.

The development Process model used

Waterfall, The system requirement are clear and fixed, the system cannot be partially delivered to provide a faster delivery

System Requirements

Functional Requirements

User Requirements

1. the system should ask the user to create an account for driver or passenger.
2. The driver should put their information in their account.
3. The passenger should put their information in their account.
4. The system should take all the passengers locations and arrange them from closest to farthest.
5. The system should provide a live location.

System Requirements

1. The system should ask the user to create an account for drivers or passengers.

1.1 if the user has an account ask for their name, ID, and password.

1.2 if the user doesn't have an account, they should create a new one.

2. The driver should put their information in their account.

2.1 include the required information, their Name, ID, and phone number.

2.2 the bus number must be provided by the driver; it's Uniquely associated to them.

2.3 list the shift hours that the manager assigned to them.

2.4 list the residential area they are assigned to by the manager.

2.5 if the driver faced some issues like the bus idle or the driver is not working today the system should automatically put the passenger with alternative bus.

3. The passenger should put their information in their account.

3.1 include the required information, their Name, ID and phone number.

3.2 must provide the period that the bus is needed.

3.3 must provide their house address.

3.4 the passenger should attach the system with their house location.

3.5 If the passenger wanted to change their house location the system should be able to assign them with new bus.

4. The system should take all the passengers' locations and arrange them from closest to farthest.

4.1 passengers put their attendants through the system to let the driver know if they are using the bus that day.

4.2 the system arrange the location from the closest to the farthest from the driver current location.

4.3 When the bus arrives at the passenger location, a notification is sent to the user and also the arrival time.

4.4 in the passengers home, the bus should wait for 3 minutes and leaves after that time regardless of the passenger arrival.

4.5 When returning home, the passenger will have to go to the designated bus location.

5. The system should provide a live location.

5.1 the live location should be shared with all users at all times.

5.2 the live location should provide the expected arrival time.

5.3 the manager should have their own copy of the live location to make sure that the driver is on schedule.

Non-Functional Requirements

1. Reliability

2. System maintenance takes place outside working hours and takes a maximum of two hours.

3. Each page must load within 2 seconds.

4.the user shall authenticate themselves using the code that was sent to their phone number so that it's secure.

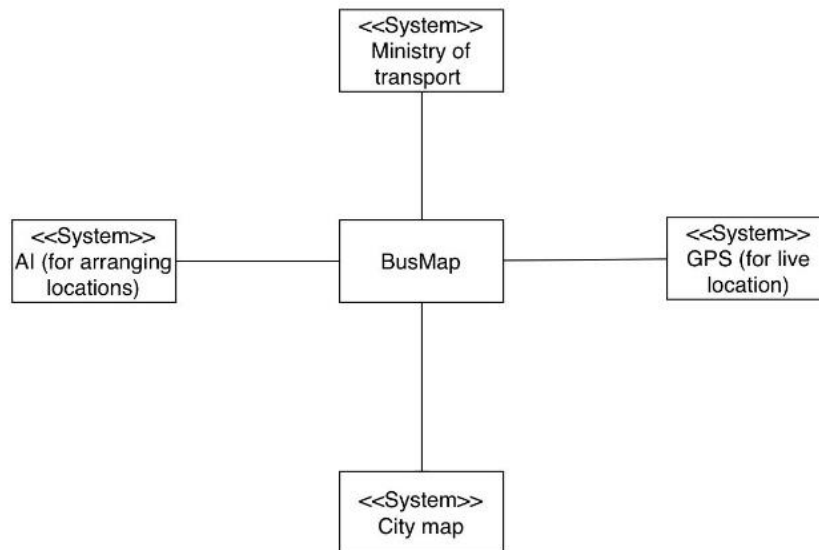
6.Data should be encrypted to protect the customer information.

7.the location should start working at the beginning of the shift until it arrives at the last destination.

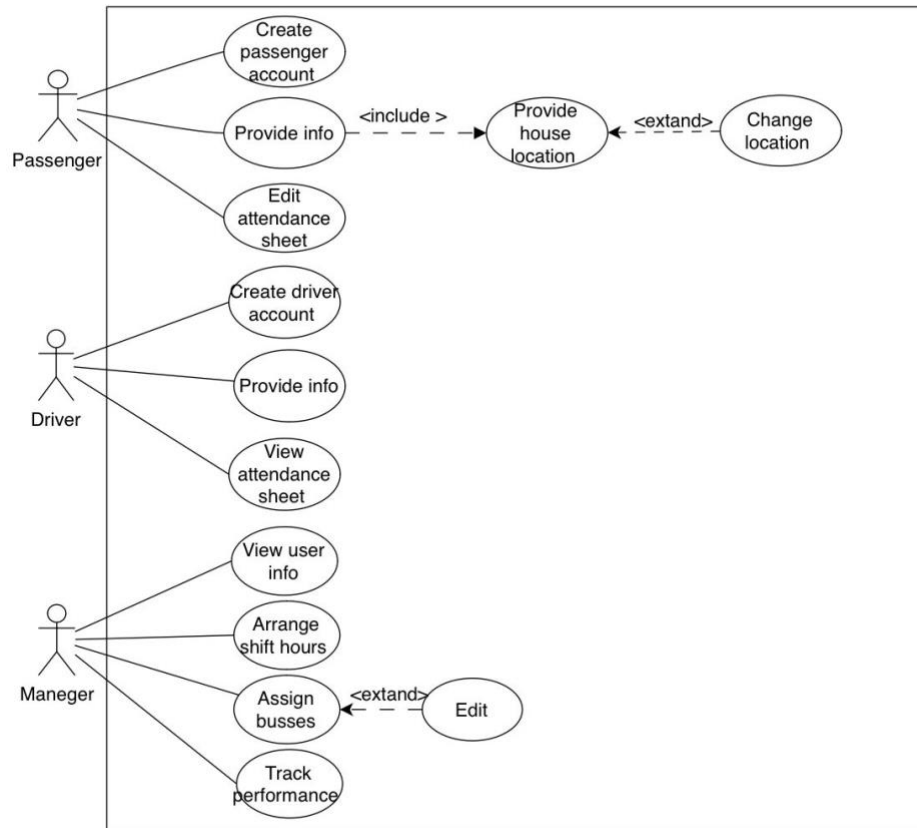
8.If the shift started no one can use the attendance sheet.

2. Chapter 2

Context Diagram



System Use-case Diagram with description



-Use case1 [Create passenger account]

Use case name	Create passenger account.
Actor	Passenger.
Description	Passenger will create account to be able to use the system.
Precondition	choose to create a passenger account
Event flow	<p>1-passenger choices to create passenger account.</p> <p>2-passenger enters their email and password.</p> <p>3- System will verify if the account is already registered.</p> <p>4-System will validate email and password. 5-</p>

	System will authorize the passenger to use the system.
Postcondition	successful login.

-Use case2 [Provide info]

Use case name	Provide info.
Actor	Passenger, Driver.
Description	users will provide important information like name and ID.
Precondition	users created an account.
Event flow	<p>1.users enter their email and password.</p> <p>2. System will verify if the account is already registered.</p> <p>3. System will validate email and password.</p> <p>4. System will authorize the users to use their account.</p> <p>5.the user will go to “provide info” page.</p> <p>6.If the user is “passenger” they must enter their location, house number, phone number, and period needed, if the user is “driver” they must enter phone number, and bus number.</p>
Postcondition	The user’s information will be saved successfully in the database.

-Use case3 [Provide house location]

Use case name	Provide house location.
---------------	-------------------------

Actor	Passenger.
Description	passenger will attach their house location link.
Precondition	passenger created an account and provided info.
Event flow	1.passenger log in. 2. Passenger will go to “provide info” page. 3.passenger will attach their house location link.
Postcondition	The passenger information will be saved successfully in the database.

-Use case4 [Change location]

Use case name	Change location.
Actor	Passenger.
Description	passenger will Change location if needed.
Precondition	passenger created an account and provided info.
Event flow	1.passenger log in. 2. Passenger will go to “provide info” page. 3.passenger will update their location.
Postcondition	The passenger information will be saved successfully in the database.

-Use case5 [Edit attendance sheet]

Use case name	Edit attendance sheet.
Actor	Passenger.
Description	Passenger will edit the attendance sheet to let the

	driver know if they will come that day or not.
Precondition	Has a passenger account.
Event flow	1-Click on attendance sheets. 2-If the passenger wants to attend, they will enter their name.
Postcondition	Their name and location will appear on the attendance sheet.

-Use case6 [create driver account]

Use case name	create driver account.
Actor	Driver.
Description	Driver will create account to be able to use the system.
Precondition	choose to create a driver account.
Event flow	1-Driver choices to create driver account. 2-driver enters their email and password. 3- System will verify if the account is already registered. 4-System will validate email and password. 5-System will authorize the driver to use the system.
Postcondition	successful login.

-Use case7 [View attendance sheet]

Use case name	View attendance sheet.
Actor	Driver.

Description	Driver will look at the attendance sheet to check who will be coming today.
Precondition	The passengers filled out the attendance sheet.
Event flow	1.log in. 2.checks attendance sheet.
Postcondition	The system will arrange locations based on attendees sheet.

-Use case8 [View user info]

Use case name	View user info.
Actor	Manger.
Description	Manger will check every account to make sure they are putting the right info.
Precondition	Manger entered the managing system.
Event flow	1.log in. 2.cheaking attendee's sheet. 3.cheaking each user's account. 4.making sure that the info made sense.
Postcondition	every user account is Checked.

-Use case9 [Arrange shift hours]

Use case name	Arrange shift hours.
Actor	Manger.
Description	Manger will arrange shift hours among the available drivers.

Precondition	Has each driver's info that is currently working.
Event flow	1-Click on drivers list button. 2-Show info for every driver. 3-Give every driver shift hours.
Postcondition	Every driver knows their shift.

-Use case10 [assign busses]

Use case name	assign busses.
Actor	Manger.
Description	Manger will assign busses for the working drivers.
Precondition	Know the drivers that will be working each week.
Event flow	1- the manager should login to their managing system account. 2- view the driver's information. 3- assignee each buss to a particular driver.
Postcondition	The system should notify each driver with their assigned buss.

-Use case11 [Edit]

Use case name	Edit
Actor	Manger
Description	Manger will edit busses if needed.
Precondition	manger assigned busses.
Event flow	1- the manager should login to their managing system account.

	<p>2- view the driver's information.</p> <p>3- edit the bus that need an alternative bus to the particular driver.</p>
Postcondition	The system should notify each driver with their new assigned bus.

-Use case12 [Track performance]

Use case name	Track performance.
Actor	Manger.
Description	Manger will check the driver's performance using live location feature.
Precondition	Having access to the live location feature.
Event flow	<p>1- the manager should login to their managing system account.</p> <p>2- the manager should open the driver's live location.</p> <p>3-manager should check if the driver is arriving at time and driving safely.</p>
Postcondition	Giving Each driver their feedback according to their performance.

Use case scenario (MSS +Extensions) (one for each student)

Use case 1: create passenger account.

Goal: login application.

Actor: passenger.

Precondition: choose to create a passenger account.

Main success scenario:

- 1-passenger enters their email and password.
2. System will verify if the account is already registered.
3. System will validate email and password.
4. System will authorize the passenger to use their account.

Post condition: successful login.

Extensions:

- 1a. Passenger enters wrong password or email.
 - 1-The system fails validation.

Use case 2: provide info.

Goal: passengers upload their information.

Actor: passenger.

Precondition: passenger created an account.

Main success scenario:

- 1-passenger enters their email and password.
2. System will verify if the account is already registered.
3. System will validate email and password.
4. System will authorize the passenger to use their account.
- 5.Passenger enter their location and house number and period needed.

Post condition: The passenger information was saved successfully in the database.

Extension:

- 5a. The location that was provided by the passenger is invalid.
 - 1- the system rejects the location.

Use case 3: assign buses.

Goal: to assign each buss to a driver.

Actor: manager.

Main success scenario:

- 1- the manager should login to their managing system account.
- 2- view the driver information.
- 3- assign each bus to a particular driver.

Post condition: The system should notify each driver with their assigned bus.

Extension:

- 3a. If the provided bus is not available.
 1. The driver cannot use that bus.

Use case 4: Track preferences.

Goal: check the driver's preferences.

Precondition: having access to the live location feature.

Main success scenario:

- 1- the manager should login to their managing system account.
- 2- the manager should open the driver live location.
- 3-manager should check if the driver is arriving at time and driving safely.

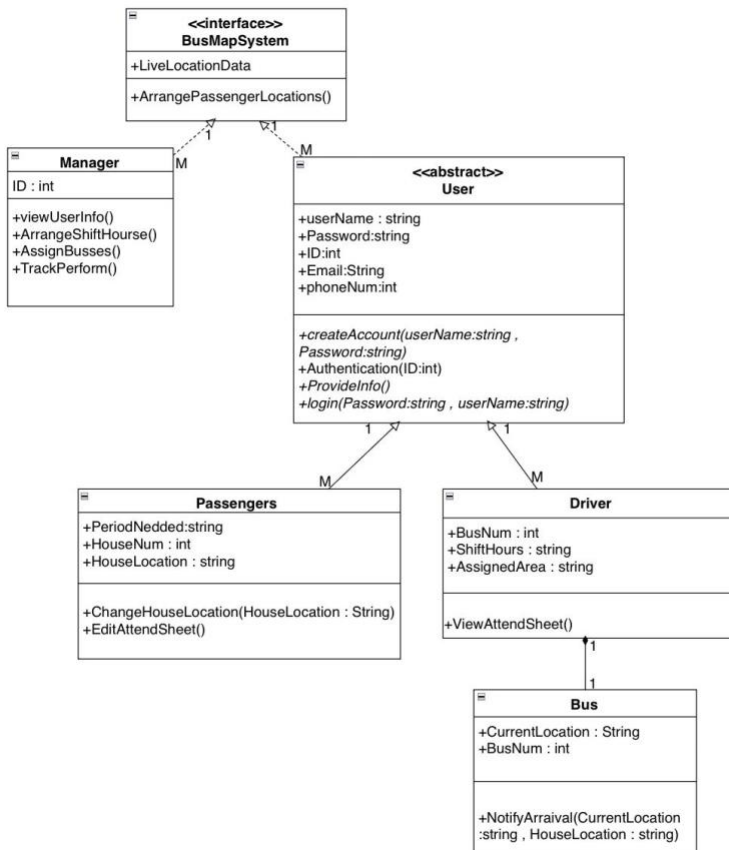
Post condition: a preference report should be sent to the driver.

Extension:

- 2a. the live location cannot be accessed.
 1. The driver's preferences cannot be checked.

3. Chapter 3

System Class Diagram



The BusMapSystem project is represented through a UML class diagram that captures the key classes, attributes, and methods involved. At the core of this system is the BusMapSystem interface, which enforces the implementation of two critical methods: LiveLocationData for handling live location data and ArrangePassengerLocations for the arrangement of passenger locations.

The Manager class, a pivotal figure in the system, is equipped with attributes such as an integer ID and various methods. These methods include viewUserInfo for accessing user information, ArrangeShiftHourse to manage shift hours, AssignBusses for bus assignments, and TrackPerform for performance tracking.

The User class is designated as an abstract class due to its role as the foundation for two essential user types: Passengers and Drivers. Users are characterized by attributes such as UserName, Password, ID, Email, and PhoneNum. Their methods encompass creating an account through createAccount, user authentication via Authentication, providing user information with ProvideInfo, and user login using the login method.

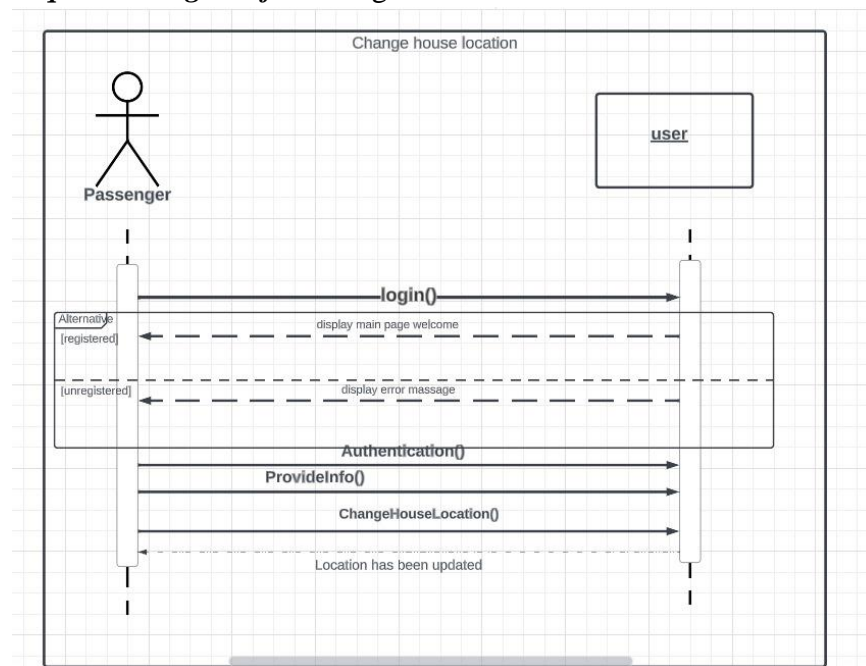
Passengers, a subclass of User, introduce attributes like PeriodNeeded, HouseNum, and HouseLocation. These passengers can edit their attendance sheet with the EditAttendSheet method.

Similarly, Drivers, also a subclass of User, feature attributes including BusNum, ShiftHours, and AssignedArea. They can modify their house location using the ChangeHouseLocation method and view the attendance sheet with ViewAttendSheet.

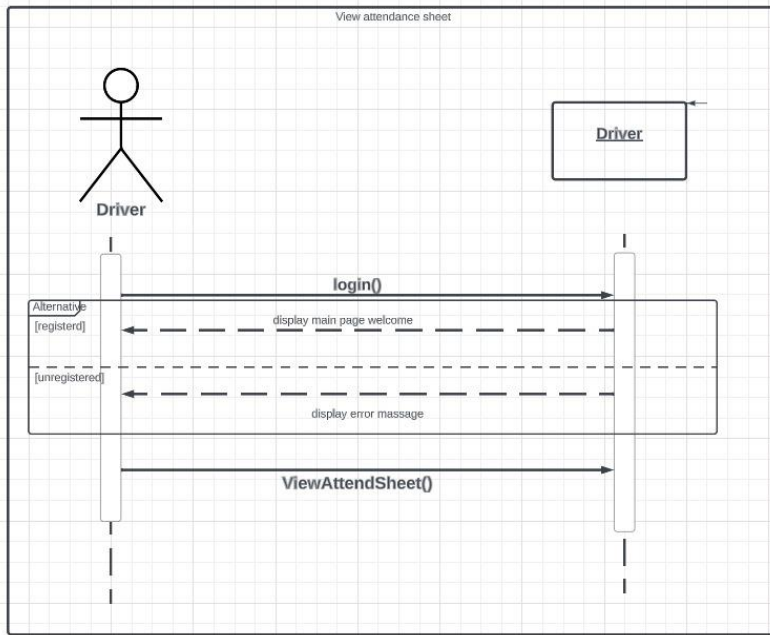
Finally, the Bus class extends from Driver and incorporates CurrentLocation and BusNum attributes. The class supports the NotifyArrival method, allowing drivers to notify passenger arrival based on their current location and the house location.

Sequence Diagrams:

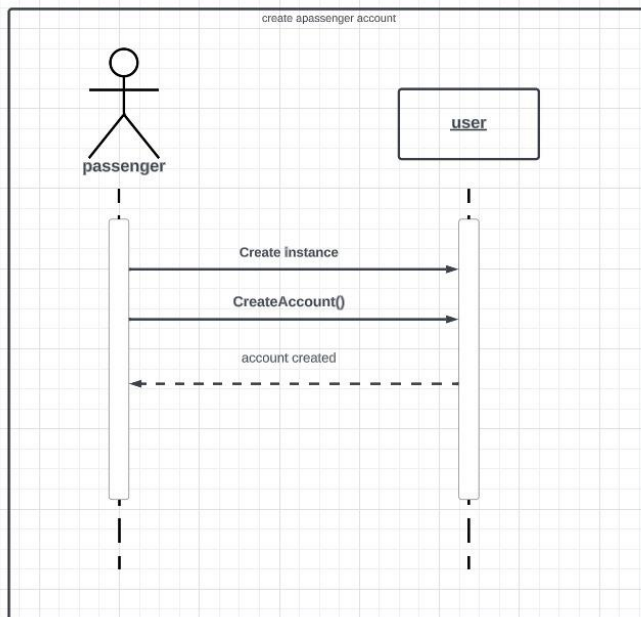
Sequence diagram for change house location.



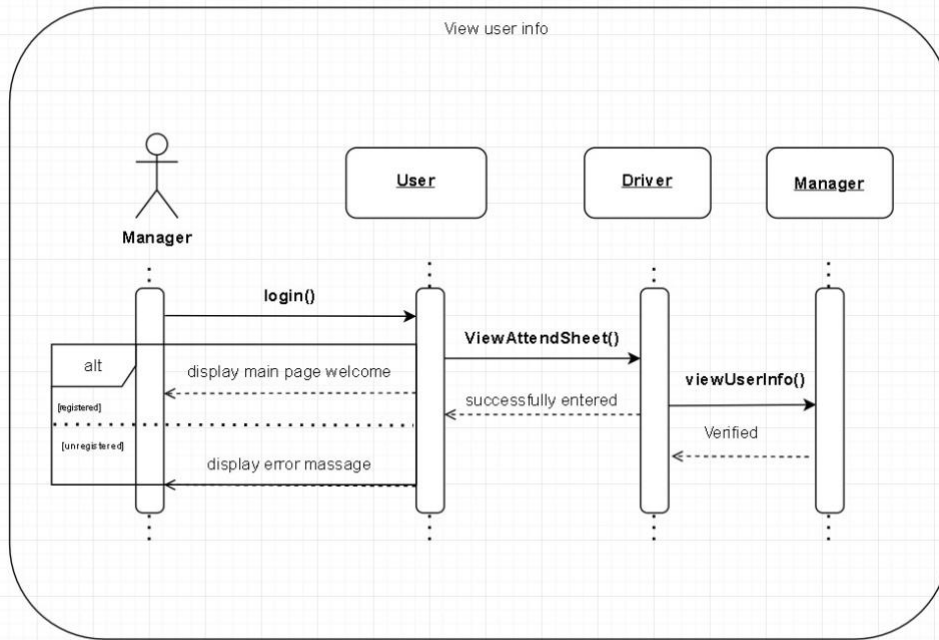
Sequence diagram for view attendance sheet.



Sequence diagram for creating passenger account.



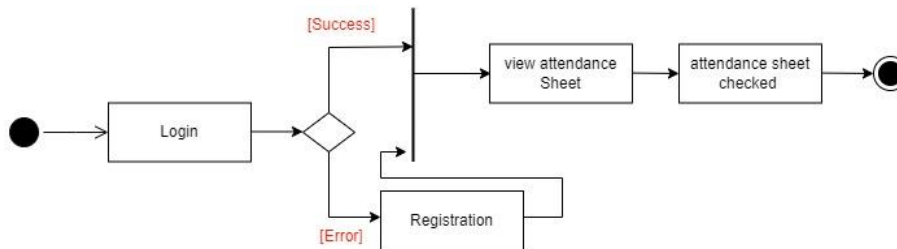
Sequence diagram for view user info.



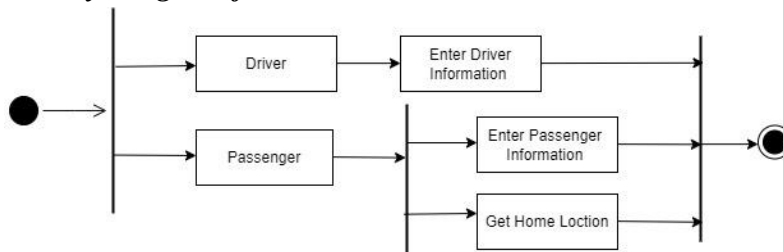
4. Chapter 4

Activity Diagram

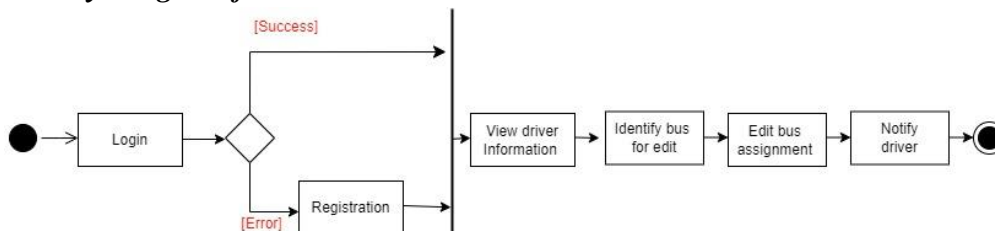
Activity Diagram for View attendance



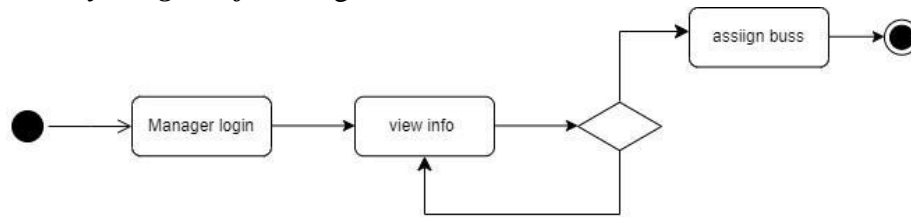
Activity Diagram for create an account.



Activity Diagram for Edit

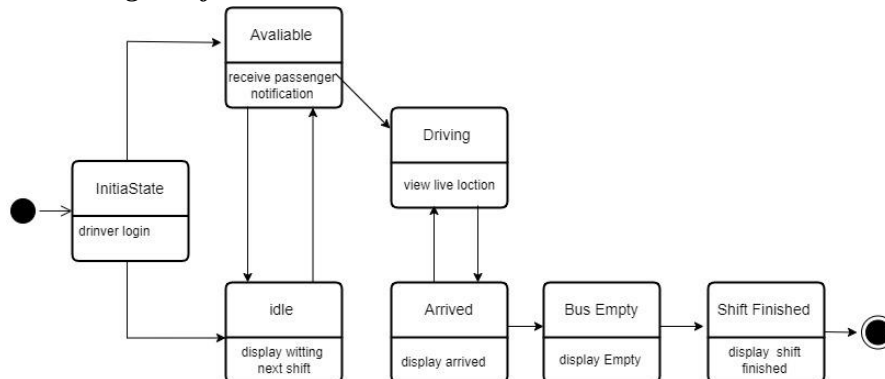


Activity Diagram for assign busses



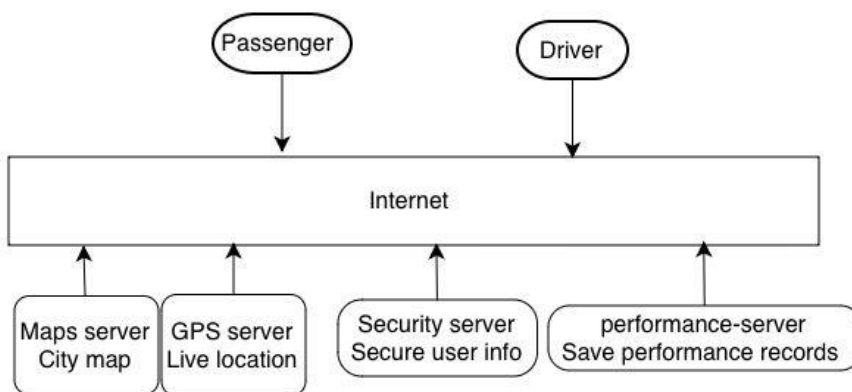
State Diagram

State Diagram for Driver



5. Chapter 5

Architecture Diagram



Why We Chose This Model:

We picked this model because it allows us to control everything from one central place. Since most of our users are students, it's like having a traffic cop to manage the flow of information. It means students can use the system without causing traffic jams, and they can use it anytime and from anywhere they want.