

BSc SE-CIS-6003- ST20261236.docx

by Rideegammana Gedara Pramod Sandakelum

Submission date: 14-Mar-2025 11:07AM (UTC+0000)

Submission ID: 252624530

File name: 119969_Rideegammana_Gedara_Pramod_Sandakelum_BSc_SE-CIS-6003-ST20261236_2517131_51066609.docx (7.05M)

Word count: 2772

Character count: 14890



Cardiff
Metropolitan
University

Prifysgol
Metropolitan
Caerdydd

School of Technologies

Assessment Brief

Module Code	Module Title
CIS6003	Advanced Programming
Academic Year	Semester
2024	1

Module Leader email

priyanga@icbtcampus.edu.lk

Content

Assessment Details	2
Submission Details.....	3
Assessment Criteria	4
Further Information	6
Who can answer questions about my assessment?	6
Referencing.....	6
Submission problems	6
Unfair academic practice.....	7
How is my work graded?	7

Assessment Details

Assessment title	Abr.	Weighting
Online vehicle reservation System	WRIT1	100%
Pass marks are 40% for undergraduate work and 50% for postgraduate work unless stated otherwise.		

Task/assessment brief:

Scenario

Mega City Cab is a popular Cab service in Colombo City. Thousands of customers use cabs monthly, currently manages customer orders and details manually. They need a computerized system to streamline their operations, including maintaining customer bookings, managing booking, and calculating bills.

Every customer transaction is assigned a booking number. New customers are registered by the system, and it should capture information such as customer registration number, name, address, NIC, etc.

1. The program should do the following: Proper authentication to the system (Login) - Use username and password to login
2. Add new customer booking: Record order number, customer name, address, telephone number, destination details, etc.
3. Display booking details: View details of customer orders
4. Calculate and print bill: Calculate the total amount based on the booking number and any applicable taxes or discounts
5. Manager car information and driver information
6. Help: Provide system usage guidelines for new users
7. Exit from the system: Properly log out and close the system

(More functionalities can be included)

Create an error free, effective one with user friendly interfaces, appropriate messages, outputs and menu driven application using Java programming language. Students can use appropriate data structures and text files to store information.

Students are free to make necessary assumptions on system design & granting access permissions other than those mentioned within the scenario, but all suggestions must be well explained with valid reasons.

Guidelines for the report format

Paper A4 | Margins 1.5" left, 1" right, top and bottom

Page numbers – bottom, right | Line spacing 1.5

Font size

Headings 14pt, Bold | Normal 12pt

Font face- Times New Roman

Use Harvard reference to acknowledge all the external sources you use properly.

Tasks:

Tasks A:

Provide the UML diagrams for the given problem with clear explanations of the design decisions. Derive detailed Use Case diagram, Class diagram and sequence diagram. Whenever necessary document the relevant assumptions you made. **(LO I) (20 marks)**

Tasks B:

Develop an interactive System with set of interfaces to get the necessary user inputs. Make sure to implement proper validation mechanisms in order to restrict invalid entries to the system. Come up with a suitable set of reports, which you think add more value to your system

- i. Your program must be a distributed application with web services
- ii. Appropriate design patterns must be implemented in your system
- iii. Your program should make use of a proper database to store information

(LO II) (40 marks)

Tasks C:

Document the test plan and explain how you used test-driven development in this scenario and do a test automation to achieve that. This includes test rationale, test plan, test data and proper application of the test plan **(LO II) (20 marks)**

Tasks D:

Create your own Git/ GitHub repository which is public to access and upload /deploy the changes of the software project you have developed in it. Share the report link within the documentation. Update it with several versions where modifications are applied each day, that you have applied the new features into which were initially uploaded. Version control techniques you have used throughout the development should be highlighted and documented properly. Demonstrate workflows deployed with the Git repository. **(LO III)(20 marks)**

Word count (or equivalent):

4000

This a reflection of the effort required for the assessment. Word counts will normally include source code, any text, tables, calculations, figures, subtitles and citations. Reference lists and contents of

Assignment Cover Sheet

Qualification	Module Number and Title
Top up - BSc in Software Engineering (CMU)	CIS6003 Advanced Programming
1 Student Name & No.	Assessor
R.G Pramod Sandakelum Registration Number GM/BSCSD/04/08 University Number ST20261236	Mrs. Vijini Mekala
Hand out date	Submission Date
	14.03.2025 – before 2.00pm
Assessment type WRIT1-Coursework	Duration/Length of Assessment Type Weighting of Assessment 100%

Learner declaration
I, R.G Pramod Sandakelum GM/BSCSD/04/08 - ST20261236 certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Marks Awarded	
First assessor	
IV marks	
Agreed grade	
Signature of the assessor	Date

Contents

Mega City Cabs.....	7
Class Diagram For the System.....	7
Use Case Diagram.....	8
Sequence Diagram	9
Justification For the System Design	11
System Nature Processes and Operations Justification	13
Coding Screenshots for the System	20
GitHub Repository of the System	22
User Manual for The System	24
Future Recommendations For the system.....	33
Test Plan.....	34
References.....	40

4

Figure 1 class diagram	7
Figure 2 Login Sequence	9
Figure 3 user add Sequence	9
Figure 4 add Vehicle Sequence.....	10
Figure 5 booking and billing sequence	10
Figure 6 login screen.....	13
Figure 7 signup screen	14
Figure 8 admin panel	15
Figure 9 driver panel	15
Figure 10 customer panel.....	15
Figure 11 user management	16
Figure 12 vehicle management	16
Figure 13 Destinations Management	17
Figure 14 distance and fare calculation	17
Figure 15 book a trip	18
Figure 16 driver and vehicle selection	18
Figure 17 locked out driver.....	19
Figure 18 locked out vehicle	19
Figure 19 Backend Structure	20
Figure 20 Front End Structure	21
Figure 21 Git hub Repository	22
Figure 22 main login	24
Figure 23 user role selection	24
Figure 24 booking	25
Figure 25 Trips Screen.....	25

Figure 26 bill table	26
Figure 27 bill View	26
Figure 28 bill print	27
Figure 29 driver view	28
Figure 30 trip end.....	28
Figure 31 bill generated	28
Figure 32 Admin View	29
Figure 33 User Management.....	29
Figure 34 Add New User	30
Figure 35 edit user	30
Figure 36 manage vehicle	31
Figure 37 edit vehicle.....	32
Figure 38 Destination Manage	32

Mega City Cabs

Class Diagram For the System

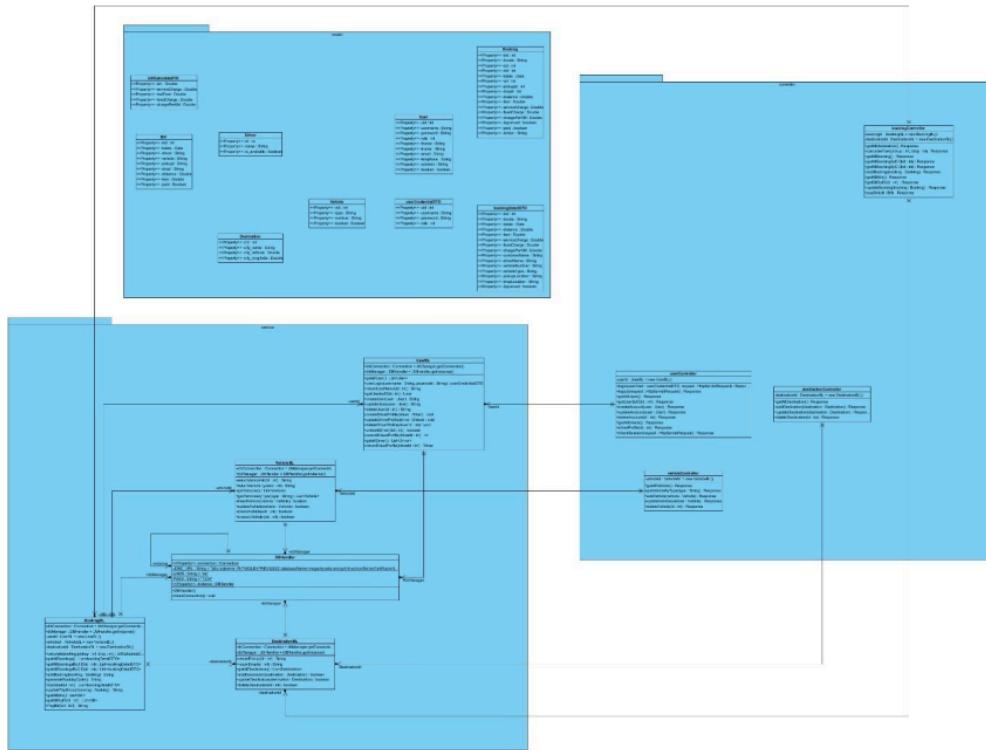
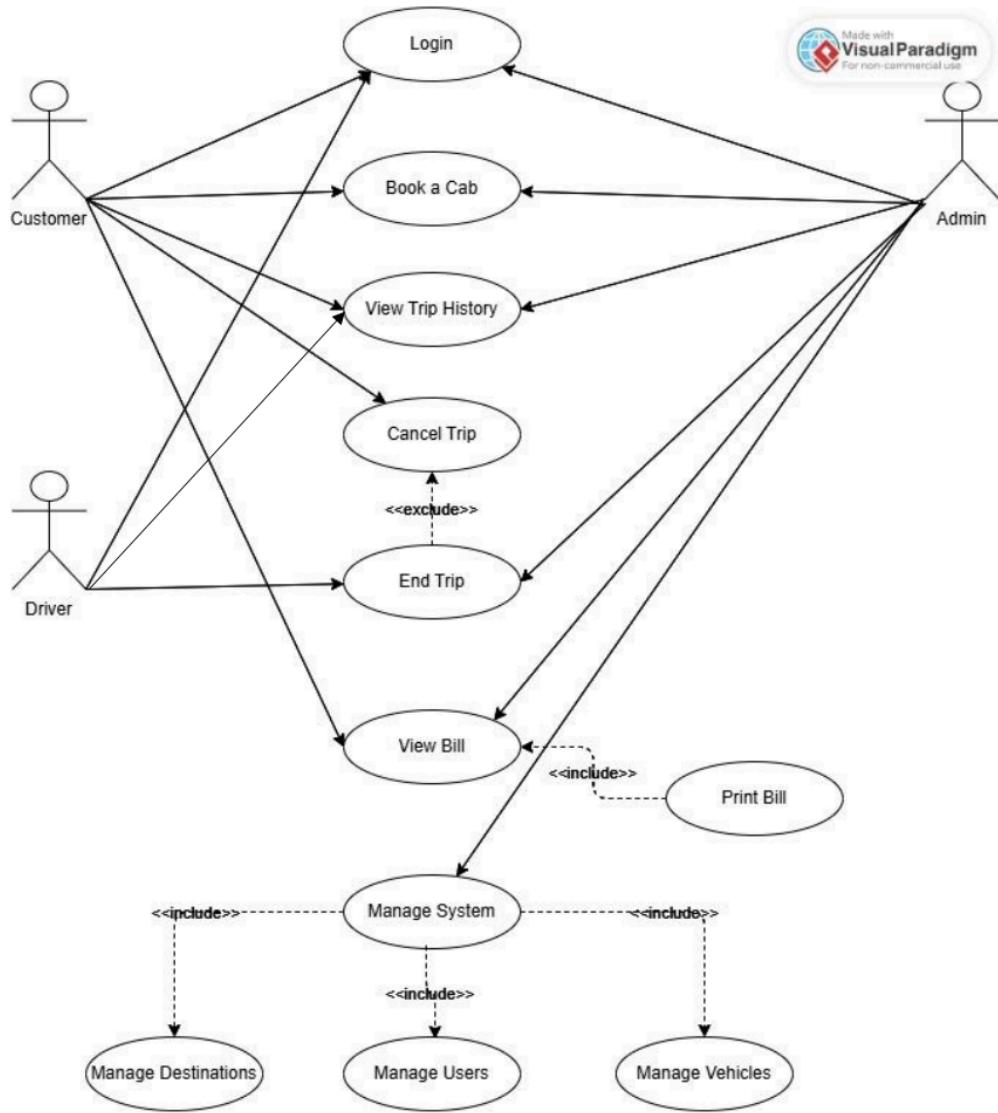


Figure 1 class diagram

(Original Image can be found [Here](#)

<https://github.com/pramodsandakelum/vehiclesystem/blob/main/Diagrams/Class%20Diagram1.jpg>

Use Case Diagram



Sequence Diagram

Sequence Diagram - Login Process

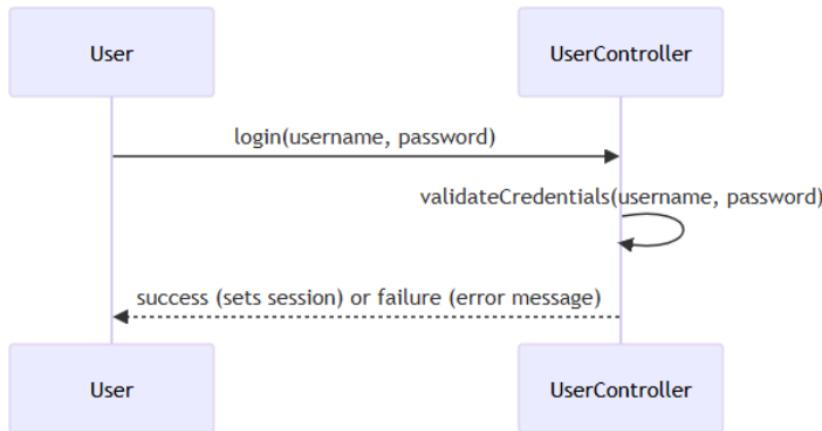


Figure 2 Login Sequence

Sequence Diagram - User Adding Process

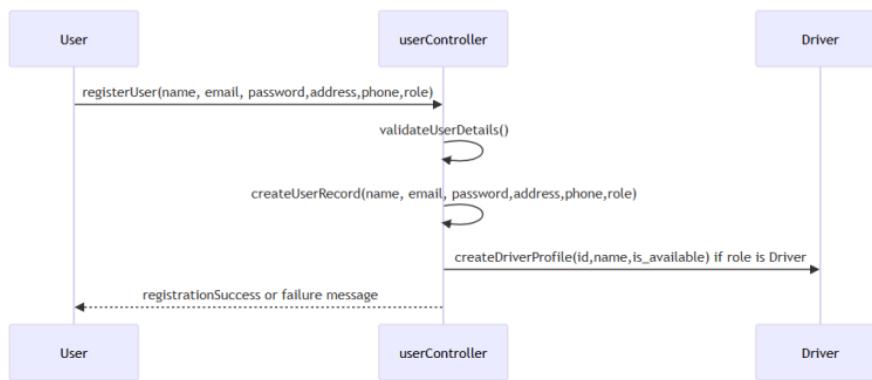


Figure 3 user add Sequence

Sequence Diagram - Add Vehicle

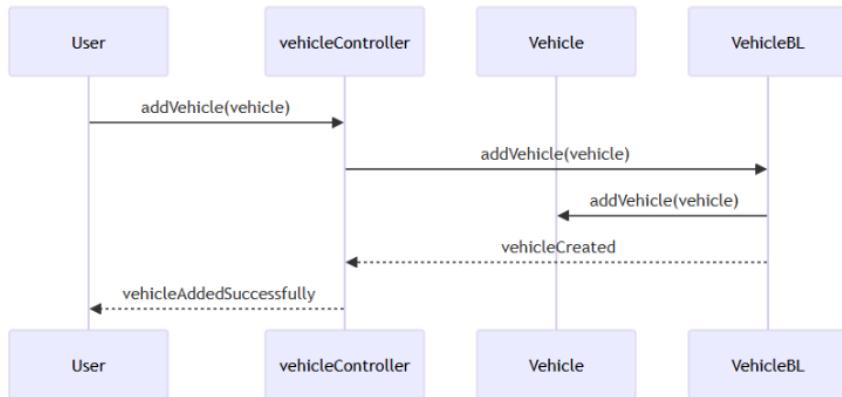


Figure 4 add Vehicle Sequence

Sequence Diagram - Booking a Cab and Billing Process

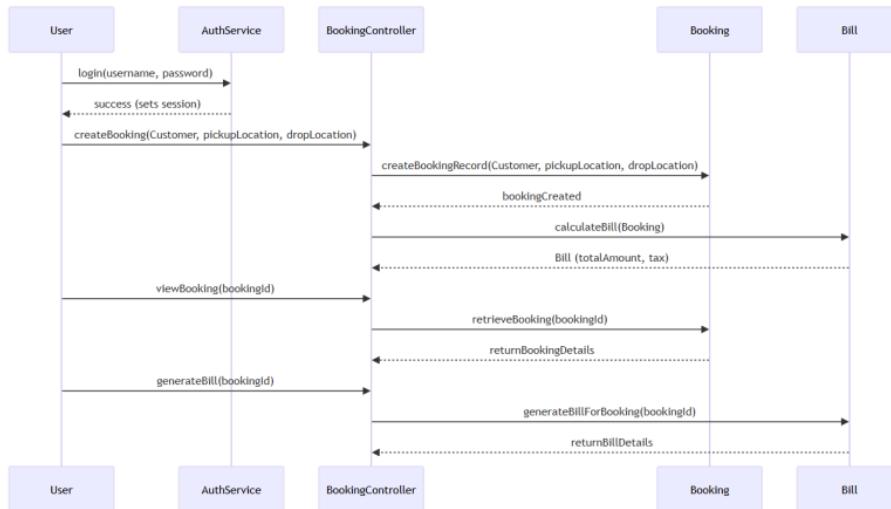


Figure 5 booking and billing sequence

Justification For the System Design

The system is designed following the MVC (Model-View-Controller) architecture and Singleton Design Pattern with a layered approach to separate concerns, improve scalability, and ensure modular design. Below is the justification for each component:

Controller Layer

The four controller classes

1. UserController
2. DriverController
3. VehicleController
4. BookingController

act as intermediaries between the client (Front end) and the business logic layer. Their primary responsibilities are:

- Handling HTTP requests (e.g., user authentication, booking creation, insert update delete operations for the whole system).
- Communicating with the Business Logic Layer (BL) to process data and return responses.
- Ensuring data validation and input sanitization before passing it to the BL.

Reason

- This ensures that all business rules are implemented in the BL, keeping controllers lightweight.
- Improves maintainability by making controllers independent of database operations.

Business Logic (BL) Layer

The four BL classes

1. UserBL
2. DriverBL
3. VehicleBL
4. BookingBL

serve as the core processing units of the system. Their primary functions include:

- Processing data received from the controllers.
- Applying business rules such as fare calculation, trip assignment, and status updates.
- Interfacing with the DBHandler for database operations.

Reason

- Separates business logic from controllers, making it easier to modify logic without affecting external APIs.
- Encapsulates core operations, allowing future enhancements like Google maps based location feeding to the system and driver allocation using gps eg.- Uber.

Model Layer

The nine model classes

1. User
2. Driver
3. Vehicle
4. Booking
5. Bill
6. Destination
7. userCredentialDTO
8. bookingdetailDTO
9. billcalculateDTO

represent real-world entities and database structures. Their purpose is to:

- Define data attributes and enforce object structure.
- Facilitate communication between the BL and DB.
- Ensure encapsulation by providing getters and setters.

Reason

- Promotes reusability across multiple system layers.
- Enhances data consistency by enforcing well-defined attributes.
- Faster Transactions rather than a direct database connection

Database Handler (DBHandler)

The DBHandler class is a **singleton** that manages database connectivity in a Java application. It ensures that only **one instance of the database connection exists** throughout the application lifecycle.

- Uses a **singleton pattern** to ensure that only **one connection instance** exists.
- Providing database connectivity and handling transactions.
- Executing CRUD operations for all entities.

Reason

- Prevents direct DB access from multiple classes, reducing redundancy.
- Enhances security and scalability by centralizing query execution.

System Nature Processes and Operations Justification

This system Operates as a Backend API there are API end points for each operation
And the Frontend Part Operates Using JSP Pages and JavaScript functions using JSON for in between data transactions

Users can Sign up and Login to the System
And the Based on the User Roles the functionality is different
Below are Some Screenshots

Login



Figure 6 login screen

Signup

Sign Up

Username

First Name

Last Name

Email

Password

Telephone

Address

Role

Already have an account? [Login here](#)

Figure 7 signup screen

Admin Panel

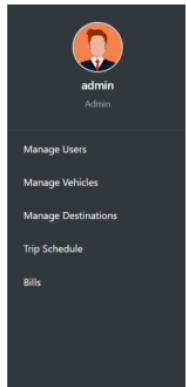


Figure 8 admin panel

Driver

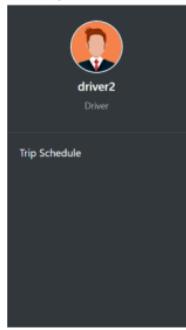


Figure 9 driver panel

Customer

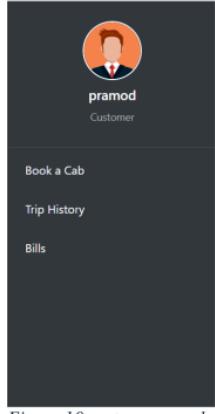


Figure 10 customer panel

Registration Process

System Already Has a built-in Super Admin Account

Rest of the users can register into the system using sign up portal as customers or drivers
And if the User is a driver, he doesn't need to create a separate driver profile based on the role selection the system will automatically generate the driver profile.

Other than that, the administrator can change the user profiles to admin customer or user
Below is the Admin Panel for User Management

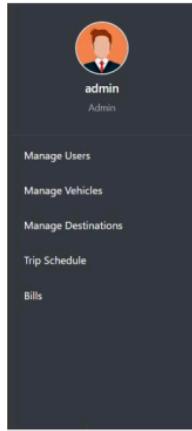


Figure 11 user management

Like wise the admin can manage Vehicles and Destinations also

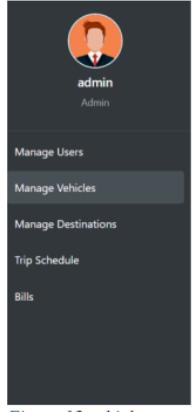


Figure 12 vehicle management

ID	Name	Latitude	Longitude	Actions
1	Colombo	6.9271	79.8612	Edit Delete
2	Kandy	7.2906	80.6337	Edit Delete
3	Galle	6.0326	80.2168	Edit Delete
4	Jaffna	9.6615	80.0255	Edit Delete
5	Anuradhapura	8.3122	80.4131	Edit Delete

Figure 13 Destinations Management

Destinations Management

When inserting Destinations the admin can enter the corresponding location latitude and longitude

Reason

This is Because the system uses the Haversine Formula to calculate the distance between the two points of pickup and drop location providing the user with accurate information like taxi fare and the total distance

3

The Haversine formula is used to calculate the great-circle distance (shortest distance) between two points on a sphere, given their latitudes and longitudes. It is commonly used in geographical applications like GPS and navigation systems. (SimonKettle, 2017)

Figure 14 distance and fare calculation

Booking a Trip

The Customer can easily book a trip using below simple interface and after the booking is done the related driver and the car is locked out in the system until the relevant trip is ended or cancelled this prevents accidental allocation of a unavailable driver or a car for another ride

Book a Ride

Customer ID: 3
Customer Name: **Pramod Sandakelum**

Pickup Location:	Booking Date:
Select Pickup	mm/dd/yyyy
Drop Location:	Vehicle Type:
Select Drop	Select Vehicle Type
Total Distance	Vehicle:
	Select Vehicle
Price (LKR)	Driver:
	Select Driver

Calculate Fare **Confirm Booking**

Figure 15 book a trip

Even the administrator is locked out from editing or deleting the driver or car from a active trip

This is an example booking demo to show the system capability

Vehicle Type:

Car

Vehicle:

PV-9370

Driver:

Saman Priyantha

Confirm Booking

Figure 16 driver and vehicle selection

View for the Admin locked out user and car

Manage Users

Manage Users								
ID	Username	First Name	Last Name	Email	Telephone	Role	Actions	
1	admin	Ramod	Sandakelum	pramodexample.com	1234567890	Admin	<button>Edit</button>	<button>Delete</button>
3	pramod	Pramod	Sandakelum	pramodexample.com	1234567890	Customer	<button>Edit</button>	<button>Delete</button>
9	driver	Pramod	Sanda	pramodsandakelum@gmail.com	0702718771	Driver	<button>Edit</button>	<button>Delete</button>
13	driver2	Saman	Priyantha	saman@mail.com	07012345678	Driver	<button>Edit</button>	<button>Delete</button>

Figure 17 locked out driver

Car

Vehicle Number:

Save Vehicle

ID	Type	Number	Booked	Actions	
1	Van	HV-5641	No	<button>Edit</button>	<button>Delete</button>
26	Van	PV-3452	No	<button>Edit</button>	<button>Delete</button>
28	Car	PV-9370	Yes	<button>Edit</button>	<button>Delete</button>
29	Car	PV-9371	No	<button>Edit</button>	<button>Delete</button>

Figure 18 locked out vehicle

Automatic Logout functionality

The System will automatically log out the user after 15 minutes of inactivity to prevent the wastage of server resources and for extra security.

Coding Screenshots for the System

Project Structure

Backend

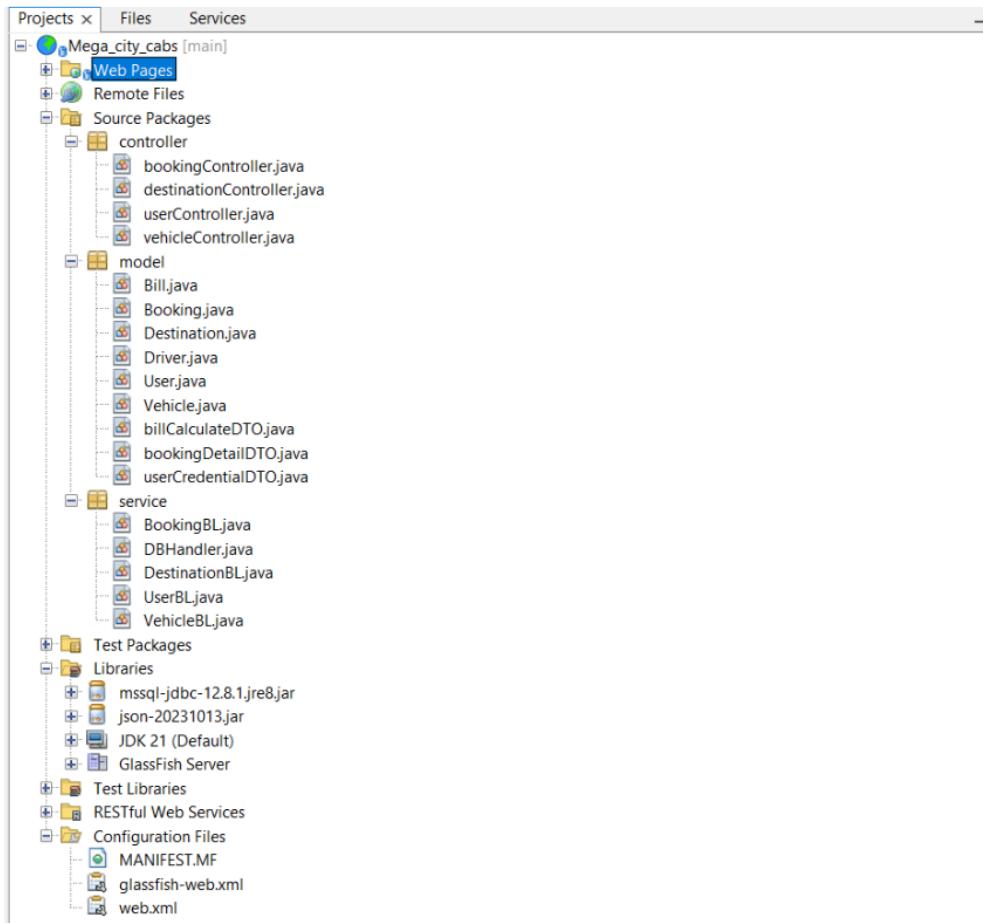


Figure 19 Backend Structure

Front End Structure

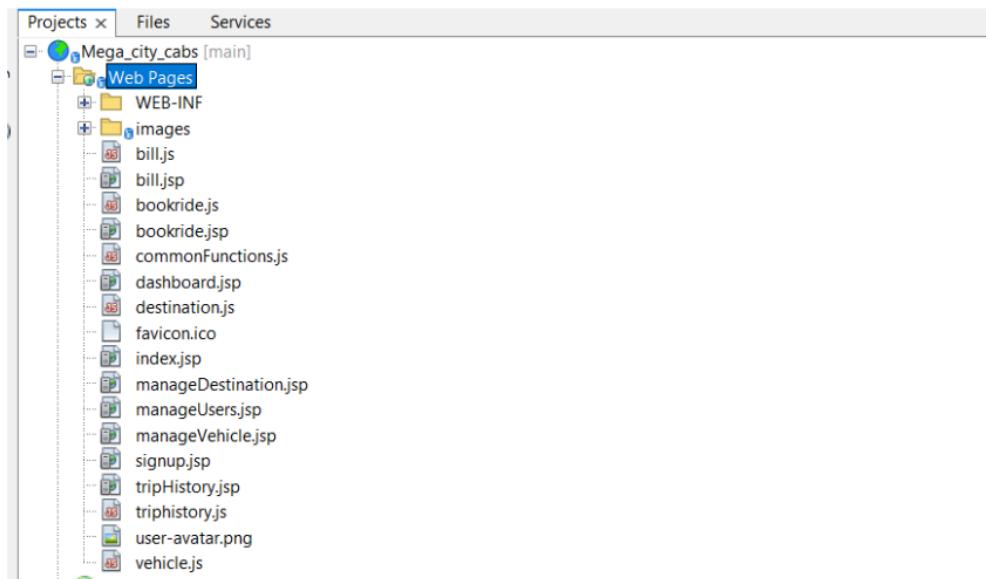


Figure 20 Front End Structure

Backend has three main packages

1. Controller
JAX-RS API End points to communicate with the front end
2. Models
Model Classes For the entities inside the system that are mapped with the database tables to maintain inter communication
3. Service
Business Logic Layer of the System That will perform CRUD Operations for entire system

Front End

Front end consists of JSP Pages and JavaScript files that are linked them JSP Pages will provide the view for the user while JavaScript files will maintain the transactions between frontend and the backend using JSON objects for increased reliability JS files and JSP files are coded separately for better troubleshooting.

GitHub Repository of the System

Version Control and GitHub Repository

To ensure efficient tracking of project changes, a public GitHub repository has been created for version control and deployment. The repository is updated daily with new features, bug fixes, and enhancements. Below is the link to the repository:

GitHub Desktop Client Was Used for Version Control and To Push the Code into Server

🔗 GitHub Repository: <https://github.com/pramodsandakelum/vehiclesystem>

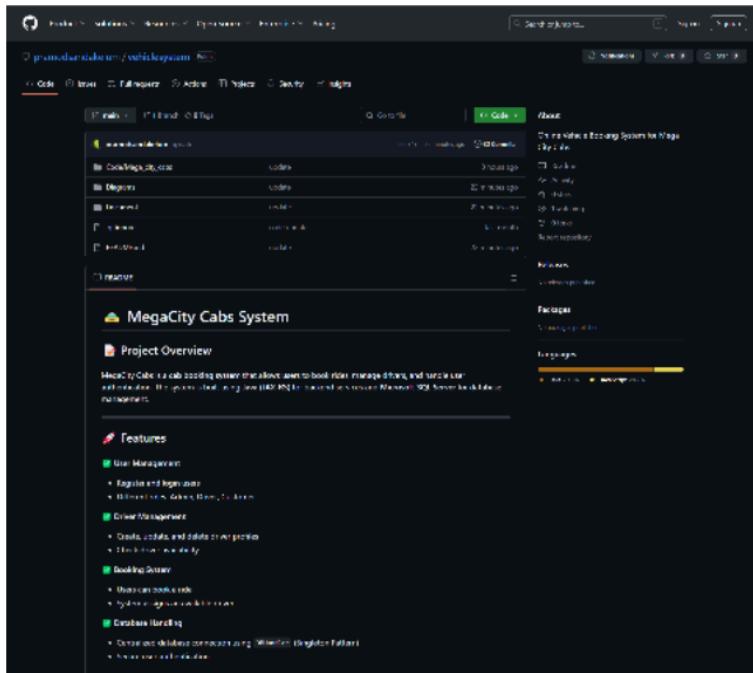


Figure 21 Git hub Repository

Version 8 Control Practices Implemented:

- **Commit Messages:** Clear and descriptive commit messages are used for tracking changes.
- **Pull Requests & Code Reviews:** Changes are reviewed before merging to maintain code quality.
- **Tagging and Releases:** Each major update is tagged for easy tracking.
- **Continuous Integration:** Automated builds and tests are triggered upon every push to the repository.
-

Workflow Demonstration:

1. Clone the Repository:

```
git clone [https://github.com/pramodsandakelum/vehiclesystem]
```

2. Create a New Branch for Features:

```
git checkout -b feature-branch
```

3. Stage and Commit Changes:

```
git add  
git commit -m "Added new feature"
```

4. Push Changes and Create a Pull Request:

```
git push origin feature-branch
```

5. Merge Approved Changes:

```
git checkout main  
git merge feature-branch
```

User Manual for The System

Login and Registration Process

If The User already has an account user can login directly using the username and password



Figure 22 main login

Otherwise, the user has to register in the system during registration user must select the correct role Customer or Driver if the user failed to do so they will have to contact the administrator to change the user role to relevant one.

A screenshot of a user registration form. It includes fields for 'Password', 'Telephone', and 'Address', each with an input box. Below these is a 'Role' section with a dropdown menu titled 'Select Role'. The dropdown menu is open, showing three options: 'Select Role' (which is highlighted in dark grey), 'Driver', and 'Customer'.

Role
Select Role
Driver
Customer

Figure 23 user role selection

Customer – Book A Trip

To add a booking user must click on the book a cab link on the side bar then on the booking screen user must fill the details correctly after selecting the drop and pickup location the user can calculate the actual fare and distance to travel between the two points by clicking calculate fare button all fields are validated user can't submit without filling the necessary details

Book a Ride

All fields are required!

Customer ID: 3
Customer Name: Pramod Sandakelum

Pickup Location: <input type="text" value="Select Pickup"/>	Booking Date: <input type="text" value="mm/dd/yyyy"/> <input type="button" value=""/>
Drop Location: <input type="text" value="Select Drop"/> ! Please select an item in the list.	Vehicle Type: <input type="text" value="Select Vehicle Type"/>
Total Distance <input type="text"/>	Vehicle: <input type="text" value="Select Vehicle"/>
Price (LKR) <input type="text"/>	Driver: <input type="text" value="Select Driver"/>
<input type="button" value="Calculate Fare"/> <input type="button" value="Confirm Booking"/>	

Figure 24 booking

Customer – View Trips

Customer can view all the trips under his id by visiting the Trip History page ongoing trip can be cancelled by clicking the Cancel Trip button

Trip Details

Code	Date	Customer	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Ongoing	Action
MCB-1741894461310-CA7D	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Matara	132.07	16348.69	No	<input type="button" value="Cancel Trip"/>
MCB-1741894558472-270F	3/20/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<input type="button" value="Cancel Trip"/>
MCB-1741894740841-419A	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<input type="button" value="Cancel Trip"/>
MCB-1741894887542-7125	3/21/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<input type="button" value="Cancel Trip"/>
MCB-1741895734843-C482	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	Yes	<input type="button" value="Cancel Trip"/>

Figure 25 Trips Screen

Customer -Billing

From Bills Screen customer can view the bills and print them and if the customer has any promo codes, they can apply it prior to billing to obtain a discount by applying the code to the bill.

Billing Details

Bill No	Date	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Paid	Action
2	3/12/2025	Saman Priyantha	PV-9370	Colombo	Kandy	94	10648.8	Yes	<button>View Bill</button>

Figure 26 bill table

Bill Details

Bill No	2
Date	3/12/2025
Driver	Saman Priyantha
Vehicle	PV-9370
Pickup	Colombo
Drop	Kandy
Distance (km)	94
Fare	10648.8
Paid	Yes
Promo Code	<input type="text" value="Enter promo code"/> <button>Apply</button>
	<button>Print Bill</button> <button>Pay Bill</button>

Figure 27 bill View

Bill Print View

By clicking on the print bill button, the customer can print or save the bill as a pdf document.

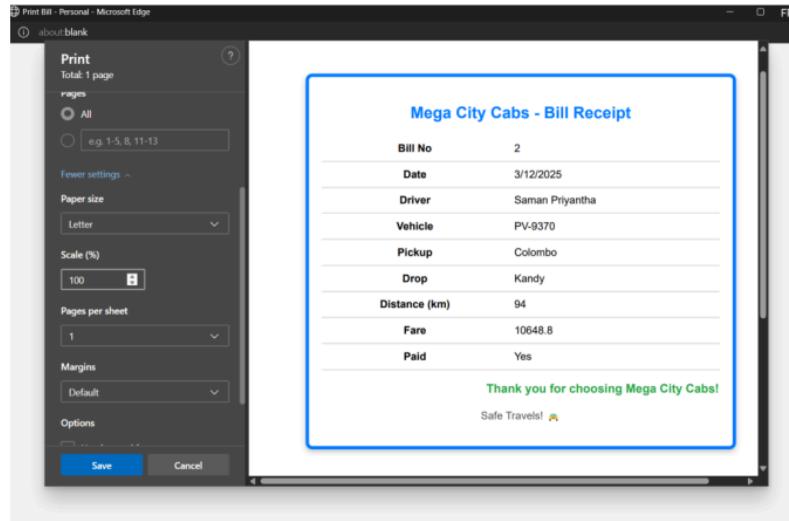


Figure 28 bill print

Driver – Trip History

Login process is same for the driver after logging into the system a driver can see this screen

Code	Date	Customer	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Ongoing	Action
MCB-1741894461310-CA7D	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Matara	132.07	16348.69	No	<button>End Trip</button>
MCB-1741894558472-270F	3/20/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<button>End Trip</button>
MCB-1741894740841-419A	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<button>End Trip</button>
MCB-1741894887542-7125	3/21/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<button>End Trip</button>
MCB-1741895734843-C482	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<button>End Trip</button>
MCB-1741896350870-3A22	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<button>End Trip</button>
MCB-1741896787904-BBAE	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Badulla	131.82	16317.8	No	<button>End Trip</button>

Figure 29 driver view

Finish a Trip

Driver Can end the trip after arriving at the destination after ending the trip the bill will be generated automatically for the customer so he can pay it after

Code	Date	Customer	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Ongoing	Action
MCB-1741897393750-BFE1	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Mullaitivu	272.46	33194.81	Yes	<button>End Trip</button>

Figure 30 trip end

Generated Bill

Bill Details	
Bill No	3
Date	3/14/2025
Driver	Saman Priyantha
Vehicle	PV-9370
Pickup	Colombo
Drop	Mullaitivu
Distance (km)	272
Fare	33194
Paid	Yes
Promo Code	<input type="text"/> Enter promo code
	<button>Print Bill</button> <button>Pay Bill</button>

Figure 31 bill generated

Admin – Managing User Vehicles Destinations

Admin uses the same login for the system this is the view for an admin

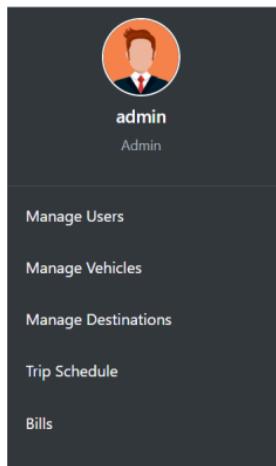


Figure 32 Admin View

Managing Users

Admin can manage users in this screen add update delete or change the role of a user
If a driver is on a booked trip the driver is not available to edit or delete until the trip ends

Manage Users

Manage Users							
ID	Username	First Name	Last Name	Email	Telephone	Role	Actions
1	admin	Ramod	Sandakelum	pramodexample.com	1234567890	Admin	<button>Edit</button> <button>Delete</button>
3	pramod	Pramod	Sandakelum	pramodexample.com	1234567890	Customer	<button>Edit</button> <button>Delete</button>
9	driver	Pramod	Sanda	pramodsandakelum@gmail.com	0702718771	Driver	<button>Edit</button> <button>Delete</button>
13	driver2	Saman	Priyantha	saman@mail.com	07012345678	Driver	<button>Edit</button> <button>Delete</button>

Figure 33 User Management

Add new User

The screenshot shows a modal dialog titled 'Add User'. It contains fields for 'Username' (with placeholder 'Last'), 'Password' (with placeholder 'Sandakelum'), 'First Name' (with placeholder 'Sar'), 'Last Name' (with placeholder 'Priy'), 'Email' (with placeholder 'Email'), 'Telephone' (with placeholder 'Address'), 'Role' (set to 'Customer'), and two buttons at the bottom: 'Cancel' and 'Save'.

Figure 34 Add New User

Edit Existing User

Admin can click Edit Button in front of each user to edit them

The screenshot shows a modal dialog for editing a user. The fields are pre-filled with the values from Figure 34: 'First Name' (Ramod), 'Last Name' (Sandakelum), 'Email' (pramodexample.com), 'Telephone' (1234567890), 'Address' (123 Main St, City), and 'Role' (Admin). The 'Cancel' and 'Save' buttons are at the bottom.

Figure 35 edit user

Delete User

To delete a user simply click delete button and confirm the message

The screenshot shows a web application interface for managing users. A confirmation dialog box is overlaid on the page, asking "Are you sure you want to delete this user?". The dialog has two buttons: "OK" and "Cancel". Below the dialog, the main content area is titled "Manage User". It contains a table with the following data:

ID	Username	First Name	Last Name	Email	Telephone	Role	Actions
1	admin	Ramod	Sandakelum	pramodexample.com	1234567890	Admin	<button>Edit</button> <button>Delete</button>
3	pramod	Pramod	Sandakelum	pramodexample.com	1234567890	Customer	<button>Edit</button> <button>Delete</button>
9	driver	Pramod	Sanda	pramodsandakelum@gmail.com	0702718771	Driver	<button>Edit</button> <button>Delete</button>
13	driver2	Saman	Priyantha	saman@mail.com	07012345678	Driver	<button>Edit</button> <button>Delete</button>

Admin – Manage Vehicles

Managing vehicles also same as above process Vehicles that are already booked cannot be edited or deleted until the trip is finished.

Manage Vehicles

Vehicle Type:

Car

Vehicle Number:

PV-9370

Save Vehicle

ID	Type	Number	Booked	Actions
1	Van	HV-5641	No	<button>Edit</button> <button>Delete</button>
26	Van	PV-3452	No	<button>Edit</button> <button>Delete</button>
28	Car	PV-9370	Yes	<button>Edit</button> <button>Delete</button>
29	Car	PV-9371	No	<button>Edit</button> <button>Delete</button>

Figure 36 manage vehicle

Edit Vehicle

When editing a vehicle, the booking status cannot be edited because it is decided by the booking generation because if mistakenly a vehicle state was changed it will be unavailable for bookings

The screenshot shows the 'Manage Vehicles' page. On the left, there is a table with columns: ID, Type, Number, and Actions (Edit, Delete). One row is highlighted in brown. On the right, a modal dialog titled 'Edit Vehicle' is open. It has fields for 'Vehicle Type' (set to 'Van') and 'Vehicle Number' (set to 'HV-5641'). Below these fields is a green 'Update Vehicle' button. At the bottom of the modal is a blue 'Save Vehicle' button.

ID	Type	Number	Actions
1	Van	HV-5641	Edit Delete
26	Van	PV-3452	Edit Delete
28	Car	PV-9370	Edit Delete
29	Car	PV-9371	Edit Delete

Figure 37 edit vehicle

Admin – Manage Destinations

Destinations can be added to the system, so the users are able to select pickup and drop locations. When booking a trip when adding the location coordinates for the location should be inserted because these coordinates are used to calculate distance and trip fares

Manage Destination

The screenshot shows the 'Manage Destination' page. On the left, there is a form with fields for 'Destination Name', 'Latitude', and 'Longitude'. Below the form is a blue 'Save Destination' button. On the right, there is a table with columns: CID, Name, Latitude, Longitude, and Actions (Edit, Delete). The table contains five rows of destination data.

CID	Name	Latitude	Longitude	Actions
1	Colombo	6.9271	79.8612	Edit Delete
2	Kandy	7.2906	80.6337	Edit Delete
3	Galle	6.0326	80.2168	Edit Delete
4	Jaffna	9.6615	80.0255	Edit Delete
5	Anuradhapura	8.3122	80.4131	Edit Delete

Figure 38 Destination Manage

Edit And Delete is same process as the users and vehicles.

Other Tasks for Admin

The admin can view trip history and end trips for any user or driver and the admin can view bills for any user.

Future Recommendations For the system

1. Google maps Api can be integrated into the system, so the users don't need to insert the locations into the system
2. Vehicles can be equipped with GPS trackers to monitor the movement Realtime
3. A mobile version can be developed for more compatibility.

Test Plan

This document outlines the test plan for the Mega City Cab system, ensuring all core functionalities are tested systematically. Each test case is structured professionally with clear steps, expected results, and placeholders for screenshots. The focus remains on validating key functionalities to maintain a seamless user experience.

2. Scope

The testing will cover the following key functionalities:

- **User Login Test**
- **Add New User**
- **Display Trip Details**
- **Calculate and Print Bill**
- **Add a Trip Booking**
- **Auto Logout Test**

Test Types and Approach

- **Manual Testing:**
 - Conducted for exploratory, usability, and UI testing scenarios.
 - Each test case in the plan (e.g., login functionality, booking entry) is manually executed to verify functionality and user experience.
- **System Testing:**
 - Comprehensive testing of the entire application to validate overall system behavior.

3. Test Cases

3.1. Test Case: User Login Test - TC/LOGIN/MCB/001

Test Case ID	TC/LOGIN/MCB/001
Description	Validate the login function with both valid and invalid credentials.
Preconditions	The application should be running.
Test Steps	<ol style="list-style-type: none">1. Enter a valid username and password.2. Click the Login button.3. Verify that the system grants access to the dashboard.4. Logout From System5. Enter an invalid username or password.6. Click Login.
Expected Results	<p><input checked="" type="checkbox"/> Valid credentials: Successful login and redirection to the main dashboard.</p> <p><input type="checkbox"/> Invalid credentials: Display of an error message and no access granted.</p>
Screenshot	

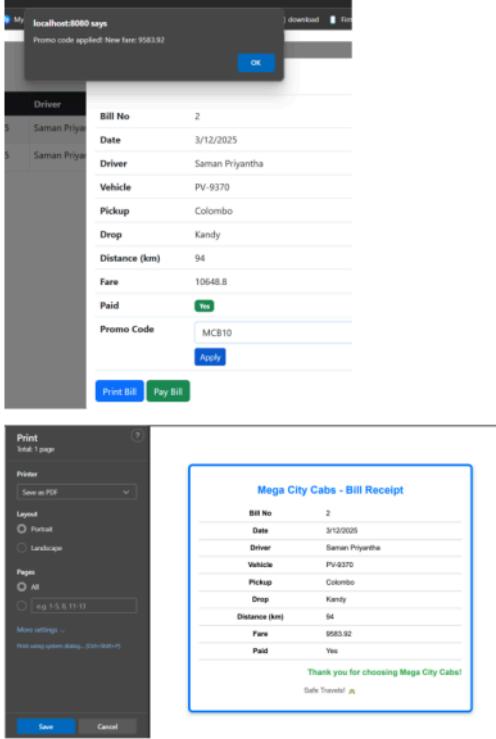
3.2. Test Case: Add New User - TC/USER/MCB/002

Test Case ID	TC/USER/MCB/002
Description	Validate that a new customer can be added, capturing all required details.
Preconditions	Admin is logged into the system.
Test Steps	<p>1. Navigate to the User Management section.</p> <p>2. Enter the following details:</p> <ul style="list-style-type: none"> - username - password - first name - last name - email - phone – address – user role <p>3. save.</p>
Expected Results	<p><input checked="" type="checkbox"/> The booking is saved successfully, and a user is generated.</p> <p><input checked="" type="checkbox"/> A confirmation message is displayed.</p>
Screenshot	<p>The screenshot displays two windows. On the left is the 'Add User' form, which includes fields for Username (pramodsandakelum@gmail.com), Password (redacted), First Name (Pramod), Last Name (Sandakelum), Email (pramodsandakelum@gmail.com), and Telephone (0702718871). On the right is a confirmation dialog box from 'localhost:8080' stating 'Account created successfully!' with an 'OK' button.</p>

3.3. Test Case: Display Trip Details - TC/TRIP/MCB/003

Test Case ID	TC/TRIP/MCB/003																																	
Description	Validate that Trip details are displayed correctly.																																	
Preconditions	At least one Trip detail exists in the system.																																	
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the Trip Details section. 2. View Is loaded with existing Details. 																																	
Expected Results	<input checked="" type="checkbox"/> All details related to the Trip Booking (Code, Date, Customer, Driver etc.) are displayed accurately.																																	
Screenshot	<p style="text-align: center;">Trip Details</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Code</th> <th>Date</th> <th>Customer</th> <th>Driver</th> <th>Vehicle</th> <th>Pickup</th> <th>Drop</th> <th>Distance (km)</th> <th>Fare</th> <th>Ongoing</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>MCB-1741897393750-BFE1</td> <td>3/14/2025</td> <td>Pramod Sandakelum</td> <td>Saman Priyantha</td> <td>Car - PV- 9370</td> <td>Colombo</td> <td>Mullaithivu</td> <td>272.46</td> <td>33194.81</td> <td>No</td> <td><button>End Trip</button></td> </tr> <tr> <td>MCB-1741899210442-DBB2</td> <td>3/13/2025</td> <td>Pramod Sandakelum</td> <td>Saman Priyantha</td> <td>Car - PV- 9370</td> <td>Colombo</td> <td>Mullaithivu</td> <td>272.46</td> <td>33194.81</td> <td>Yes</td> <td><button>End Trip</button></td> </tr> </tbody> </table>	Code	Date	Customer	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Ongoing	Action	MCB-1741897393750-BFE1	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Mullaithivu	272.46	33194.81	No	<button>End Trip</button>	MCB-1741899210442-DBB2	3/13/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Mullaithivu	272.46	33194.81	Yes	<button>End Trip</button>
Code	Date	Customer	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Ongoing	Action																								
MCB-1741897393750-BFE1	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Mullaithivu	272.46	33194.81	No	<button>End Trip</button>																								
MCB-1741899210442-DBB2	3/13/2025	Pramod Sandakelum	Saman Priyantha	Car - PV- 9370	Colombo	Mullaithivu	272.46	33194.81	Yes	<button>End Trip</button>																								

3.4. Test Case: Calculate and Print Bill - TC/BILL/MCB/004

Test Case ID	TC/BILL/MCB/004
Description	Validate the functionality for calculating the bill and printing it.
Preconditions	A valid booking must exist.
Test Steps	<ol style="list-style-type: none"> 1. Open Bill Section. 2. Click the View Bill button. 3. Apply a Promo Code to see discounts are applied correctly pay bill 4. Print Bill
Expected Results	<p><input checked="" type="checkbox"/> The bill amount is calculated correctly with taxes/discounts applied.</p> <p><input checked="" type="checkbox"/> The printed bill output reflects the accurate total.</p>
Screenshot	 <p>The screenshot displays two main windows. The top window is a modal dialog titled 'My localhost:8080 says' with the message 'Promo code applied: New fare: 9583.92'. It has an 'OK' button at the bottom. Below this is a larger form for viewing a bill. The form includes fields for Driver (Saman Priyanka), Vehicle (PV-9370), Pickup (Colombo), Drop (Kandy), Distance (km) (94), Fare (10648.8), Paid (Yes), and Promo Code (MCB10). There are 'Print Bill' and 'Pay Bill' buttons at the bottom. The bottom window is a 'Print' dialog with options for Printer, Layout (Portrait), Pages (All), and a preview area showing a 'Mega City Cabs - Bill Receipt' document. The receipt contains the same bill details as the form above, along with a 'Thank you for choosing Mega City Cabs!' message and a 'Safe Travel!' link.</p>

3.5. Test Case: Add a Trip Booking - TC/BOOK/MCB/005

Test Case ID	TC/BOOK/MCB/005																																													
Description	Validate A Correct Trip Booking is added.																																													
Preconditions	A Customer or Admin is logged in to the system.																																													
Test Steps	<ol style="list-style-type: none"> 1. Open Book a Cab Section. 2. Fill the Form with Correct Details like Pickup And Drop Location. 3. Click Confirm Booking. 4. Check Trip History Section for a Ongoing Trip 																																													
Expected Results	<p><input checked="" type="checkbox"/> The Trip Is Inserted Correctly with A Booking Code.</p> <p><input checked="" type="checkbox"/> The printed bill output reflects the accurate total.</p>																																													
Screenshot	<p>Book a Ride</p> <p>Customer ID: 3 Customer Name: Pramod Sandakelum</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Pickup Location: Colombo</td> <td>Booking Date: 03/15/2025</td> </tr> <tr> <td>Drop Location: Mullaitivu</td> <td>Vehicle Type: Car</td> </tr> <tr> <td>Total Distance: 272.46 Km</td> <td>Vehicle: PV-9371</td> </tr> <tr> <td>Price (LKR) 33194.81</td> <td>Driver: Pramod Sanda</td> </tr> <tr> <td colspan="2">Calculate Fare</td> </tr> <tr> <td colspan="2">Confirm Booking</td> </tr> </table> <p>Trip Details</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Code</th> <th>Date</th> <th>Customer</th> <th>Driver</th> <th>Vehicle</th> <th>Pickup</th> <th>Drop</th> <th>Distance (km)</th> <th>Fare</th> <th>Ongoing</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>MCB-1741897393750-BFE1</td> <td>3/14/2025</td> <td>Pramod Sandakelum</td> <td>Saman Priyantha</td> <td>Car - PV-9370</td> <td>Colombo</td> <td>Mullaitivu</td> <td>272.46</td> <td>33194.81</td> <td>No</td> <td>Cancel Trip</td> </tr> <tr> <td>MCB-1741899210442-DBB2</td> <td>3/13/2025</td> <td>Pramod Sandakelum</td> <td>Saman Priyantha</td> <td>Car - PV-9370</td> <td>Colombo</td> <td>Mullaitivu</td> <td>272.46</td> <td>33194.81</td> <td>Yes</td> <td>Cancel Trip</td> </tr> </tbody> </table>	Pickup Location: Colombo	Booking Date: 03/15/2025	Drop Location: Mullaitivu	Vehicle Type: Car	Total Distance: 272.46 Km	Vehicle: PV-9371	Price (LKR) 33194.81	Driver: Pramod Sanda	Calculate Fare		Confirm Booking		Code	Date	Customer	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Ongoing	Action	MCB-1741897393750-BFE1	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV-9370	Colombo	Mullaitivu	272.46	33194.81	No	Cancel Trip	MCB-1741899210442-DBB2	3/13/2025	Pramod Sandakelum	Saman Priyantha	Car - PV-9370	Colombo	Mullaitivu	272.46	33194.81	Yes	Cancel Trip
Pickup Location: Colombo	Booking Date: 03/15/2025																																													
Drop Location: Mullaitivu	Vehicle Type: Car																																													
Total Distance: 272.46 Km	Vehicle: PV-9371																																													
Price (LKR) 33194.81	Driver: Pramod Sanda																																													
Calculate Fare																																														
Confirm Booking																																														
Code	Date	Customer	Driver	Vehicle	Pickup	Drop	Distance (km)	Fare	Ongoing	Action																																				
MCB-1741897393750-BFE1	3/14/2025	Pramod Sandakelum	Saman Priyantha	Car - PV-9370	Colombo	Mullaitivu	272.46	33194.81	No	Cancel Trip																																				
MCB-1741899210442-DBB2	3/13/2025	Pramod Sandakelum	Saman Priyantha	Car - PV-9370	Colombo	Mullaitivu	272.46	33194.81	Yes	Cancel Trip																																				

3.6. Test Case: Auto Logout Test - TC/LOGIN/MCB/006

Test Case ID	TC/LOGIN/MCB/006
Description	Test Session Timeout and Logout Process.
Preconditions	A Customer or Admin is logged in to the system.
Test Steps	<ol style="list-style-type: none">1. Default Session Timeout is for the System Is 15 Minutes.2. Wait 15 Minutes Without Activity to check
Expected Results	<input checked="" type="checkbox"/> User is Logged out of system after 15 minutes.
Screenshot	 A photograph of a yellow and black taxi sign. Overlaid on the sign is a dark rectangular login dialog box. The dialog box contains the text "Mega City Cabs" at the top. Below this are two input fields: "Username" with the value "pramod" and "Password" with the value "*****". At the bottom of the dialog box is a blue "Login" button. In the bottom right corner of the dialog box, there is very small text that appears to be a link.

References

SimonKettle, 2017. *Distance on a sphere: The Haversine Formula*. [Online] Available at: <https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128> [Accessed 25 January 2025].

ORIGINALITY REPORT



PRIMARY SOURCES

- | | | |
|----------|---|-----------|
| 1 | Submitted to University of Wales Institute, Cardiff | 2% |
| | Student Paper | |
| 2 | Submitted to Queen Mary and Westfield College | 1% |
| | Student Paper | |
| 3 | Submitted to American Community Schools of Athens | 1% |
| | Student Paper | |
| 4 | Submitted to Asia Pacific University College of Technology and Innovation (UCTI) | 1% |
| | Student Paper | |
| 5 | Submitted to Lebanese International University | 1% |
| | Student Paper | |
| 6 | fastercapital.com | 1% |
| | Internet Source | |
| 7 | Submitted to Arab Open University | 1% |
| | Student Paper | |
-

Exclude quotes On

Exclude matches < 1%

Exclude bibliography On