**CAR RENTAL SYSTEM**



Session: 2022 – 2026

**Submitted by:**

Muhammad Hamza Waseem (2022-CS-54)

**Supervised by:**

Sir. Laeeq Khan Niazi

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

# CONTENTS

[Project Description 3](file:///C:\Users\Ibrahim\Downloads\semester-1-code-main%20(1)\semester-1-code-main\PROJECT_MIDTERM\Project%20Documentation.docx#_gjdgxs)

[Users of Application](file:///C:\Users\Ibrahim\Downloads\semester-1-code-main%20(1)\semester-1-code-main\PROJECT_MIDTERM\Project%20Documentation.docx#_30j0zll) 3

[Functional Requirements](file:///C:\Users\Ibrahim\Downloads\semester-1-code-main%20(1)\semester-1-code-main\PROJECT_MIDTERM\Project%20Documentation.docx#_1fob9te) 4

Wireframes of Application5

OOP Concepts 19

Comparison between procedural Programming and OOP 20

Design Pattern Implementation 21

Class Details: 21

Class Diagram 23

Conclusion 24

* **Short Description of my project**

The Referred Project is a Car Rental System. The Business Application is GUI based and it solves many Purposes that a Car Rental System usually does. The Transition from Console base to GUI based has been an intuitive and user-friendly experience. The best thing about this System is that it provide User Control over the system which was lacking in Console based System.

This application is very user-friendly as it provides a separate menu for **Admin** as well as **Customers**.

* **Users of Application**
  + Following are the users of my application:
  + **Customers or Clients**: Customers can Get Benefits from the System and use the system according to their needs.
  + **Admin**: Admin can access the portal anytime and can see how much many users rented the cars and other tasks performed by the customer. Admins has the access to the backbone of the system.
* **Functional Requirements**
  + Following are the functional requirements of my application

|  |  |  |  |
| --- | --- | --- | --- |
| ***User Story ID*** | ***As a*** | ***I want to perform*** | ***So that I can*** |
| ***1*** | Admin | View Costumers | Admin can view the costumers that accessed the system. |
|  |  | Add New Cars | Add new Cars to enhance the business. |
|  |  | View Cars | View Cars that are Currently available in the Garage. |
|  |  | Check Availability | Add the name of Car And check its Availability in the system/garage. |
|  |  | Add Promo | Promo code can be added by the Admin to facilitate the costumers and give them discounts. |
|  |  | Reviews | See the Reviews that the Costumers have given about their previous experience with the cars they took on Rent. |
|  |  | Verify Costumer | Verify the documents of the Costumers. |
|  |  | Rented Cars | Display the Cars that are currently taken on Rent by the costumers. |
|  |  | Modify Cars | Admin can do modifications to the cars to make the Cars better. |
|  |  | Notify Costumers | Make new notifications so that the costumers know about the updates and up gradations to the system. |
|  |  | Remove Cars | Remove cars from the system if they are no longer required by the Car Rental System. |
|  |  | View Notifications | Admins can view the notifications by himself/herself |
|  |  | Exit | Exit from the system if no further actions are required. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***2*** | Customer |  | Add Information so that I can keep my information available to Admin. |
|  |  | Book Ticket | Book ticket so that I can travel from one place to another. |
|  |  | Cancel Ticket | Cancel Ticket as I can change my mind. |
|  |  | Enter Promocode | Enter promocode to get discount. |
|  |  | Feedback | Feedback to improve trains condition. |
|  |  | DM to Admin | DM to Admin to get immediate response. |
|  |  | View Announcements | View Announcements put by the Admin. |
|  |  | Log Out | Log out my account so that no one could access my account. |

* **Wireframes**
  + Following are the wireframes of my application

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Sign Up

2. Sign In

3. Exit

Enter one option :

**Figure: Main Menu**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter your username: mudassir

Enter your password: 123

Enter your role: ADMIN

Username already exists

Press any key to continue

**Figure: Sign Up**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter your username: mudassir

Enter your password: 123

Enter your role: ADMIN

Login Successful

Press any key to continue

**Figure: Sign In**

**Admin Menus:**

**Figure: Main Admin Form**

**1st Menu of Admin:**

**Figure: Add Cars Form**

**2nd Menu of Admin:**

**Figure: Verify Costumer Form**

**3rd Menu of Admin:**

**Figure: View Cars Form**

**4th Menu of Admin:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Train Time

Attari Express 11:59am

Press any key to continue

**Figure: View Schedule Menu**

**5th Menu of Admin:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Announcement:

You train Attari Express has been delayed

Press any key to continue

**Figure: Announcement Menu**

**6th Menu of Admin:**



**Figure:**

**7th Menu of Admin:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter name of driver: Amjad Mehran

Enter number of driver: 03099507890

Press any key to continue

**Figure: Add Drivers Menu**

**8th Menu of Admin:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Drivers

Amjad Mehran

Press any key to continue

**Figure: View Drivers Menu**

**9th Menu of Admin:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Drivers Numbers

Amjad Mehran 03099507890

Press any key to continue

**Figure: Contact Drivers Menu**

**10th Menu of Admin:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FeedBack

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Your services are very good and I really enjoyed it and 5 stars stars

Press any key to continue

**Figure: View Feedback Menu**

**Customer Menus:**

**Figure: Customer Main Menu**

**1st Menu of Customer:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter Your name: Mudassir Naveed

Enter your phone number: 03099603075

Press any key to continue

**Figure: Add Information Menu**

**2nd Menu of Customer:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*Following are the available routes\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1-Karachi to Lahore................

2-Sialkot to Lahore................

3-Kashmir to Karachi...............

4-Peshawar to Sargodha.............

5-Narowal To Faisalabad............

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter your name: mudassir

Enter your destination(format: source to destination): Sialkot to Lahore

Enter the number of tickets: 2

Enter your pin: 123

Ticket booked successfully!

Press any key to continue

**Figure: Book Ticket Menu**

**3rd Menu of Customer:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter your ID to delete: 123

Identity 123 not found.

Press any key to continue

**Figure: Cancel Ticket Menu**

**4th Menu of Customer:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter promocode(within 4 digit): 1234

Congurations You get 10 percent discount!!

Press any key to continue

**Figure: Enter Promocode Menu**

**5th Menu of Customer:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FeedBack

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Your services are very good and I really enjoyed it and 5 stars

Press any key to continue

**Figure: Feedback Menu**

**6th Menu of Customer:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter what you want to tell to ADMIN:

I just want to tell you that Your services are very good but your servants disrespect to me

Press any key to continue

**Figure: DM to Admin Menu**

**7th Menu of Customer:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* RAILWAY MANAGEMENT SYSTEM \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

You train Attari Express has been delayed

Press any key to continue

**Figure: View Announcement Menu**

* **OOP CONCEPTS:**

I have completed my project using oop concepts. Following are the concepts that I have use in my application.

* **ASSOCIATION:**

In my project, I have utilized the concept of association to establish relationships between different classes. Association is a fundamental object-oriented programming concept that allows classes to interact and collaborate with each other.

In particular, I have employed association between the Ticket class and the Train class. Each Ticket object is associated with a specific Train object, representing the train for which the ticket is booked. This association enables the retrieval of relevant train information, such as the train number, origin, destination, fare, and departure time, through the property of the Ticket class.

By using the concept of association, I have achieved a logical and cohesive relationship between the Ticket and Train classes, enhancing the functionality and usability of my project. This association allows for seamless data retrieval and manipulation, enabling efficient ticket booking and management within the train management system.

Overall, the incorporation of association in my project exemplifies the power of object-oriented programming principles in establishing meaningful connections between classes and promoting modular and extensible code design.

* **INHERITANCE:**

In the development of my project, I have successfully employed the concept of inheritance to enhance code reusability and promote a hierarchical relationship between classes. Inheritance is a key feature of object-oriented programming that allows a class to inherit properties and behaviors from a parent class, known as the base or superclass.

In my project, I have utilized inheritance by creating a derived class called Ticket and Train that inherits from the base class Management. By doing so, the Ticket class inherits all the attributes and methods defined in the Management class, such as train number, origin, destination, fare, and departure time. This inheritance relationship allows the Ticket and Train class to inherit the existing functionality of the Management class, eliminating the need to redefine common attributes and behaviors.

Through inheritance, I have achieved a clear and logical structure within my project. The Ticket and Train class inherits the fundamental characteristics of a management from the Management class, while also introducing additional properties specific to a ticket, such as ticket number and passenger name. This hierarchical arrangement ensures a streamlined and efficient design, as well as simplifying the implementation and maintenance of the project.

By utilizing inheritance, I have significantly reduced code duplication, enhanced code readability, and facilitated future scalability. The concept of inheritance has enabled me to leverage existing functionality and extend it further, contributing to the overall effectiveness and flexibility of my project.

* **COMPARISON:**

Following is the comparison between PROCEDURAL and OBJECT ORIENTED PROGRAMMING.

* **Focus**:

Procedural programming focuses on writing procedures or functions that perform specific tasks. Whereas, Object-oriented programming focuses on objects and their interactions to solve problems.

* **Structure**:

Programs are structured around functions, where data is passed between functions as parameters. Whereas in Object Oriented Programming Programs are structured around classes and objects, where objects encapsulate data and behavior.

* **Data and Behavior**:

Data and behavior are separate entities, and functions manipulate the data.

Whereas in Object Oriented Programming Data and behavior are combined into objects, which interact with each other through methods.

* **Code Reusability**:

Code reusability is limited as functions are often specific to a particular task.

Whereas in Object Oriented Programming Code reusability is promoted through the concept of inheritance, allowing classes to inherit properties and behaviors from parent classes.

* **Modifiability**:

Modifying code can be challenging as changes may require modifications in multiple functions. Whereas in Object Oriented Programming Modifying code is easier as changes can be localized to specific classes and objects.

* **Encapsulation**:

Encapsulation of data is limited as functions have access to shared data.

Whereas in Object Oriented Programming Encapsulation of data is emphasized, with objects controlling access to their data through encapsulation mechanisms like access modifiers.

* **Code Organization**:

Programs are organized around functions and flow control. Whereas in Object Oriented Programming Programs are organized around classes, objects, and their relationships, promoting modular and reusable code.

* **DESIGN PATTERN IMPLEMENTATION:**

I have designed my application in such a way using BL, DL, UI.

**BL:**

In BL folder all classes contain attributes with private keyword and getter setter functions are made in this respective folder. This folder basically stands for Business Logic so that all other functions are hide from this folder.

**DL:**

In DL folder all classes contain read and write from file and also read and write from list functions are made in this respective folder. This folder basically stands for Data Layer so that all the functions applied on data are present in this folder.

**UI:**

In UI folder all classes contain User Interacted functions are made in this respective folder. This folder basically stands for User Interface so that all the functions that are visible to human are present in this folder. This can be a console based or windows based as well.

* **CLASS DETAILS:**

Following are the class details that are present in my application.

* **BL FOLDER:**

Following are classes present in BL folder.

**AdminBL**:

This class contains getter setter functions and attributes used for admin.

**CustomerBL**:

This class contains getter setter functions and attributes used for customer.

**ManagementBL**:

This class is a base class of train and ticket. This is also contain common attributes used in child class.

**MUserBL**:

This class contain whole attributes about username, password and role for both admin and customer.

**TicketBL**:

This class inherits from management class because attributes used in this class and ticket class are almost same so we create management class to manage all this.

**TrainBL**:

This class inherits from management class because attributes used in this class and ticket class are almost same so we create management class to manage all this.

* **DL FOLDER:**

Following are the classes present in DL Folder.

**AdminDL**:

This class contain functions like that are used for retaining and extracting admin’s data.

**CustomerDL**:

This class contain functions like that are used for retaining and extracting admin’s data.

**MUserDL**:

This class contain functions like that are used for retaining and extracting data username, password and role for both admin and customer.

**TicketDL**:

This class contain functions like that are used for retaining and extracting ticket’s data.

**TrainDL**:

This class contain functions like that are used for retaining and extracting train’s data.

* **UI FOLDER:**

Following are the classes present in DL Folder.

**AdminUI**:

This class contains the functions that are interacted with admin.

**CustomerUI**:

This class contains the functions that are interacted with customer.

**MenuUI**:

This class contains the functions that are interacted with both users.

**MUserUI**:

This class contains the functions that are interacted while signup and signin menu required for representation of users..

**TicketUI**:

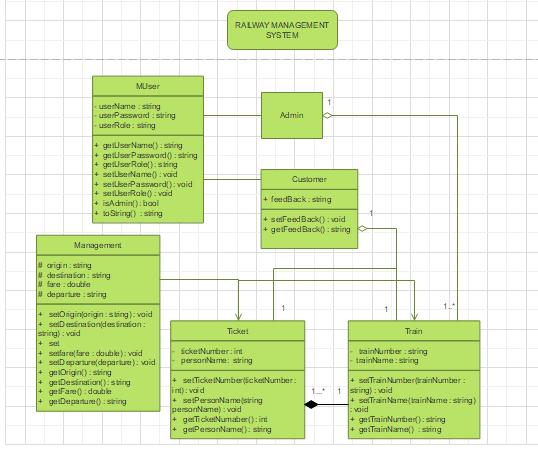
This class contains the functions that are interacted with users related to ticket.

**TrainUI**:

This class contains the functions that are interacted with users related to train.

* **Class Diagram:**

Following is the class diagram of my application



* **Conclusion:**

The train management system project aimed to develop a comprehensive software solution for managing train operations, ticket booking, and passenger information. The project successfully implemented various features including train management, ticket booking, data storage, and user authentication. The system allowed both administrators and customers to interact with the application, perform relevant tasks, and access necessary information.

* **Achievements:**

**Train Management**: The project implemented functionalities for adding, updating, and removing train details, such as train number, name, origin, destination, fare, and departure time. The train data was stored in a file for persistence.

**Ticket Booking**: The system facilitated ticket booking for customers, allowing them to select the train, enter passenger details, and generate tickets. The ticket information was associated with the corresponding train, enabling easy retrieval of train-related data.

**User Authentication**: The project implemented user authentication mechanisms, enabling administrators and customers to log in with their credentials. This ensured secure access to system features and protected sensitive information.

* **Challenges Faced:**

**Data Storage**: One of the challenges encountered was implementing effective data storage and retrieval mechanisms. This required handling file operations, parsing data, and ensuring data integrity. Proper error handling and exception management were crucial to address unexpected situations.

**Input Validation**: Ensuring the validity and integrity of user inputs posed a challenge. Implementing robust input validation techniques helped mitigate issues such as incorrect data entry and potential security vulnerabilities.

**Modular Design**: Designing the project with a modular approach enabled better code organization, improved code reusability, and easier maintenance. Breaking down functionality into smaller, manageable components enhanced the overall structure of the project.

**Error Handling**: Incorporating comprehensive error handling mechanisms helped identify and address potential issues during program execution. Handling exceptions and providing informative error messages improved user experience and system reliability.

**User Interface Design**: Paying attention to user interface design and usability considerations greatly impacted the user experience. Clear and intuitive interfaces enhanced user interaction and minimized confusion.