Department of Computer Science
American International University-Bangladesh
Final Term Project

Course Code : CSC2209
Course Name : Object Oriented Analysis and Design
Section : H
Department : Computer Science & Engineering

Prepared & submitted by:

Name	ID
Sriman Dhar	20-43774-2
Sumayea Khatun	20-43837-2
Md. Shariar Mahmud Sachcha	20-43772-2
Mubasshar-Ul-Ishraq Tamin	20-43814-2
Nobonita Nonde	20-43819-2

Submitted To:

Course Instructor: Dr. Md Alamgir Kabir

AIUB

Date of Submission: 27 November 2021

Project Name: Railway Planner

Table of contents:

Abstract

Bangladesh is a densely populated country. People from all over the country uses public transport must. As railway system is an essential addition to the public transport sector. So, to remove the hassle of reserving tickets from physical stations and long waiting ticket lines, our project will help them on the process.

Background Information:

Online based reservation management system in railways is a reservation system which is online based and where user can login to the user portal system by using their actual user id and password and book a rail (like international rail/metro rail/sub rail) tickets from anywhere in the world. Online based reservation system is now a useful and new addition to our daily life in public transport medium. It can be easier public's life. This system is a full fledge online based system and the most important part of the system is reserving tickets for travel, now in this project we are only concentrating on the following part.

Problem Statement:

People to travel from home to their workstations and people must spend their valuable time by waiting in line to get on a bus to go to work also because of the narrow roads and rickshaws traffic jam is meant to happen which doubles the stress and kills more time. Due to hassle in front of ticket counters people may not get or make reservation as they desire when buying ticket in person.

Objectives:

- ➢ Use mobile application as a method of booking tickets.
- ➢ Achieve the complete functionality of reservation system (get update of timing, to manage reservation etc.) with the mobile application.
- ➢ Eventually replace the in-person ticket booking method.

Scope:

Our project is targeted at all the people who uses railway as a method of transportation.

Features & functionalities of our project include:

a. The requirements are analyzed and refined which enables the end users to
efficiently use the online Railway Reservation System.

b. The main scope for this project is the user/passenger should reserve for the
train ticket.

c. First the user has to login to the database after that the person wants to fill
their details.

d. User must input time and destination. If tickets are available, they can make
reservation.

e. They must complete the payment and then they will get a confirmation
message.

Proposed Solutions

The android application we aim to develop, contains most of the project's fundamental capabilities and data. There are a few basic components in our system, which are mentioned and described below:

Mobile Application:

In this project, three basic tasks necessitate the use of the application. Which are

1. Provide an efficient user interface.

2. To access the system user logged in using proper user id and password or to

   register with appropriate information.

3. To check or update information regarding their reservation.

After registration or logging in user can check & cancel their reservations, make payment, get updates about the timing and location.

Online Database:

This the place where all the data about the users and their reservations are stored. It contains, the Username, User ID, Password, Email, Phone number, Reservations (Time, destination, train number, train name), payment status. Every user can only access his own data using his user ID and password.

Technical Feasibility:

In terms of technology and infrastructure, I believe our proposed solution is totally doable. Our main purpose is to use a mobile application instead of in person ticket booking. So, we don’t need an additional hardware or infrastructure establishment. We just need to develop the mobile application and gain access to the user information database. Our main target audiences are the people who use railway as a method of transportation. We are confident enough that it will be very beneficial for them because it saves their valuable time, they may be able to make their desired reservations, the application is very easy to use. By making an online based reservation management system for the public transport people will find easy to travel through railway transport.

Comparison:

Reasons why our software system is better:

Reasons	**Rail Sheba Application services**	**Our application services**
1.Cancelation Services	Rail Sheba doesn't have any cancelation services. If one person wants to cancel, they need to go to the counter and then apply	But in our services, Customer can cancel the ticket and will get refund the full ticket price.

	cancelation and for refund. It is not assured that they will get the cancelation and even if they get the cancelation confirmed they will not get appropriate refunds.	
2. Tracking Services	Rail Sheba do not have appropriate service for tracking trains if customer wants to track. Also, there are some strict instructions if they want to track. Such as charges applicable for this service, they are required to use a specific operator, they get messages about the information's.	In our application no charges are required if one wants to use tracking system. They also do not require using any particular operator. We will provide the location in the application via GPS.
3.Contact Services	The rail Sheba application only provides contact information regarding particular stations such as Dhaka, Chittagong, Khulna, Rajshahi. So, if customer requires information regarding other stations, they are unable to do so.	In our application we provide contact information of any station situated in Bangladesh both email and phone number. So customer can easily contact with the stations they need.

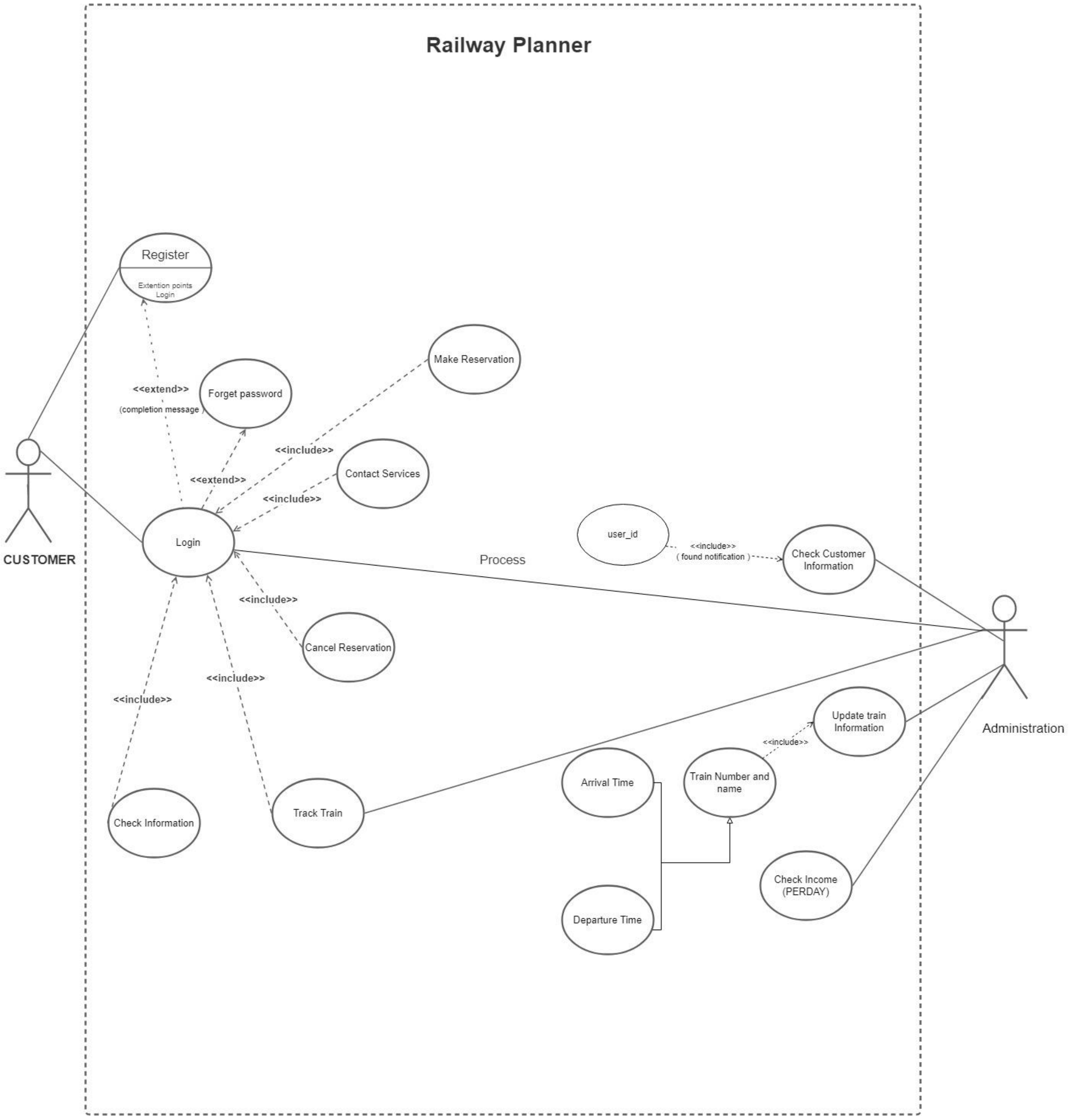
Constraints:

1. As it is a smart phone-based application the analog phone user can't use or get help of this application.
2. Some people may not know how to use these mobile applications even though they do possess smart phones.

Risks:

People might not provide valid information during registration or reservation. Though it's an online service, but emergency booking system is not available.

Use-Case Diagram:



Use-Case Specification:

////////////////// REGISTER ///////////////////

Use Case Name: Register

Actor(s): Customer

Description: The use case diagram describes the process of a customer to register for the application.

Reference ID: E_TICKET 1.1

Typical course of events:	**Actor Action**	**System Response**
	Step 1: This use case is initiated when customer select Register option.	
		Step 2: The system will then ask to fill some details such as Name, Email, phone number and Password.
	Step 3: Then the customer needs to fill their valid information (Name, Email, phone number and Password.) as required.	
		Step 4: The system will authorize the information and send a completion

		message.
		Step 5: Then the system will move the user to login page.

Alternative course of events: **STEP 1.1:** For unsuccessful registration scenario user data will not be saved in the online database.

Postcondition: Customer must input valid email id and phone number.

////////////////////// **LOGIN** ////////////////////////////

Use Case Name: Login

Actor(s): Customer, Administration

Description: The use case diagram describes the process of a customer/administration to login into the application.

Reference ID: E_TICKET 2.0

Typical course of events:	**Actor Action**	**System Response**
	Step 1: This use case is initiated when customer/administration select login option.	

		Step 2: The system will then ask to fill some details such as Email/user_id and Password.
	Step 3: Then the customer needs to fill their valid information (user_id or Email, Password.) as required.	
		Step 4: The system will authorize the information password and (user_id/email) message.
		Step 5: The system will then send a successful notification if it is authorized.
		Step 6: Then system will move the customer to the next segment.
Alternative course of events:	**STEP 2.1:** For wrong password or user_id/email user will get a warning notification and will not be allowed to enter the system as a user.	

STEP 2.2: if user forgets password, they can select forgotten password option.

Precondition: Customer/Administration must need to be registered.

Postcondition: Customer/Administration must input valid email/ user_id and password.

//////////////////// CHECK CUSTOMER INFORMATION //////////////////

Use Case Name: Check customer information

Actor(s): Administration

Description: The use case diagram describes the process of Check customer information.

Reference ID: E_TICKET 3.0

Typical course of events:	**Actor Action**	**System Response**
	Step 1: This use case is initiated when administration select check customer information option.	

		Step 2: The system will then ask to fill some details such as Customer’s Email or user_id .
	Step 3: Then the administration needs to fill valid information (user_id or Email) as required.	
		Step 4: The system will check the information (user_id/email) is available or not.
		Step 5: The system will then show the details of customer after showing user found notification.
Alternative course of events:	**STEP 3.1:** For wrong user_id/email administration will get notified.	

STEP 3.2: If the user is not found administration will get no match found notification.

Precondition: Administration must need to login successfully.

Postcondition: Administration must input valid email/ user_id and password.

/////////////////////// UPDATE TRAIN INFORMATION ////////////////////////////

Use Case Name: Update Train information

Actor(s): Administration

Description: The use case diagram describes the process of an administration updating train information.

Reference ID: E_TICKET 4.0

<u>Typical course of events</u>:	**<u>Actor Action</u>**	**<u>System Response</u>**
	Step 1: This use case is initiated when administration select update train information option.	

Step 2: The system will then ask to fill the train name & number.

Step 3: Then the administration needs to input the train number and name.

Step 4: The system will then search for that train.

Step 5: The system will then send a notification if the train number and name match is found.

Step 6: Then system will ask the administration to change the arrival time / departure time for that train.

Step 7: Then the administration needs to update the time of departure and arrival.

		Step 8: After administrations updates the timing the system will show a notification that timing is updated.
Alternative course of events:	**STEP 4.1:** For wrong train number and name they will get a notification that the train number or name does not match.	
Precondition:	Administration must login into the system.	
Postcondition:	Administration must input train name and number.	

////////////////////// TRACK TRAIN ///////////////////////////

Use Case Name:	Track Train	
Actor(s):	Administration	
Description:	The use case diagram describes the process of an administration wants to track train.	
Reference ID:	E_TICKET 5.0	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: This use case is initiated when administration select track	

	train option.	
		Step 2: The system will then ask to fill the train name, number, destination, time.
	Step 3: Then the administration needs to input the train number and name, destination, time.	
		Step 4: The system will then search for that train.
		Step 5: The system will then send a notification if the train is found.
		Step 6: Then system will show the administration the GPS location of that train.
Alternative course of events:	**STEP 5.1:** For wrong information of train, they will get a notification that the train number or name does not match.	

Precondition: Administration must login into the system.

Postcondition: Administration must input valid train information.

//////////////////////// CHECK INCOME ////////////////////////////

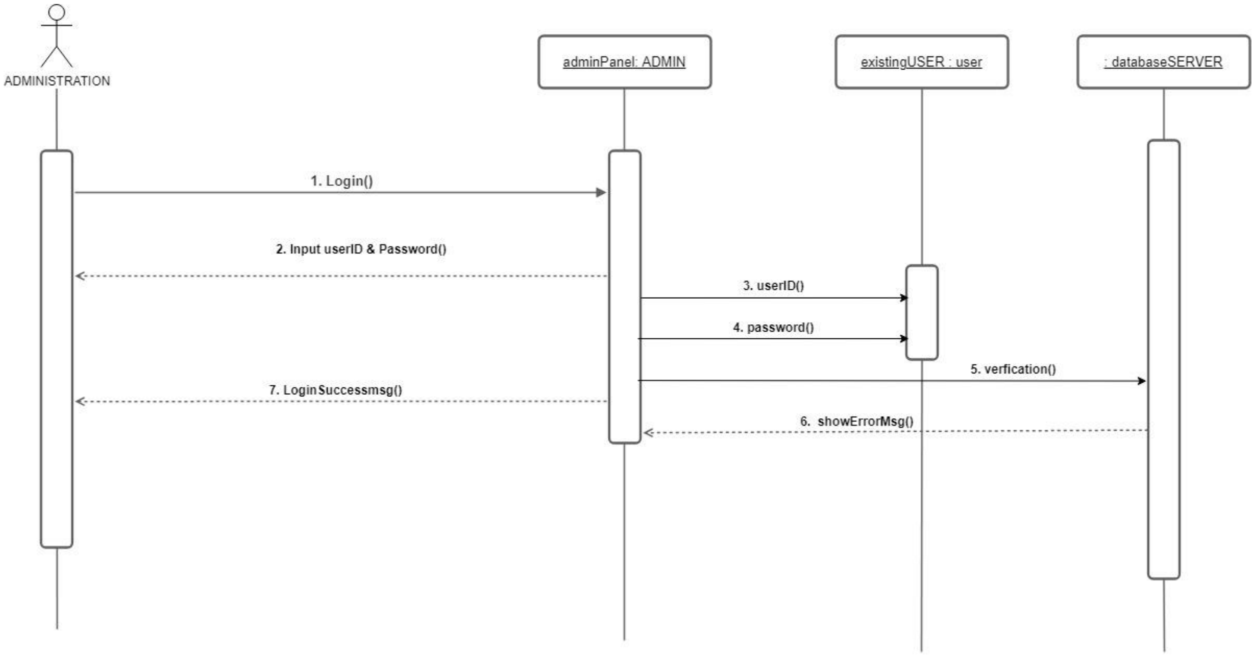
Use Case Name:	Check Income	
Actor(s):	Administration	
Description:	The use case diagram describes the process of an administration wants to Check Income.	
Reference ID:	E_TICKET 6.0	
<u>Typical course of events</u>:	**<u>Actor Action</u>**	**<u>System Response</u>**
	Step 1: This use case is initiated when administration select check income option.	
		Step 2: The system will then calculate the income for that day (24 hours).

Step3: System will show the total amount to the administration.

Precondition: Administration must login into the system.

Sequence Diagram:

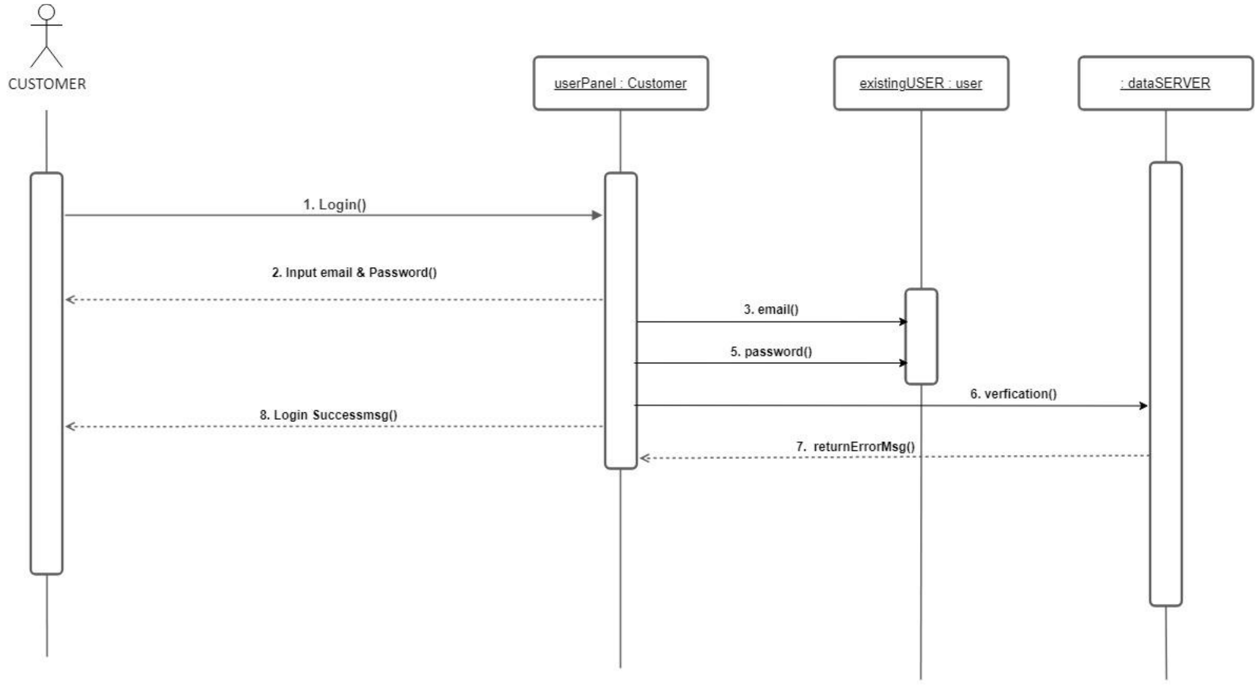
Sequance #1



Log In Admin:

The sequence diagram depicts that admin requires to login into the system. Firstly, admin select login option from admin panel. Then admin will get a message from system to provide userID and password. After that, the system will create an object for storing the login information that admin has provided to the system via userID and Password().Therefore, the system will enquire the database for verification, to check the account whether exist or not. If the target account does not exist, the system will return the error message to admin and system may remind the customer to register an account or “forget password?” Finally, the database will retrieve other information to the system and if verification is successful and account does exist, then system will return successful message to admin.

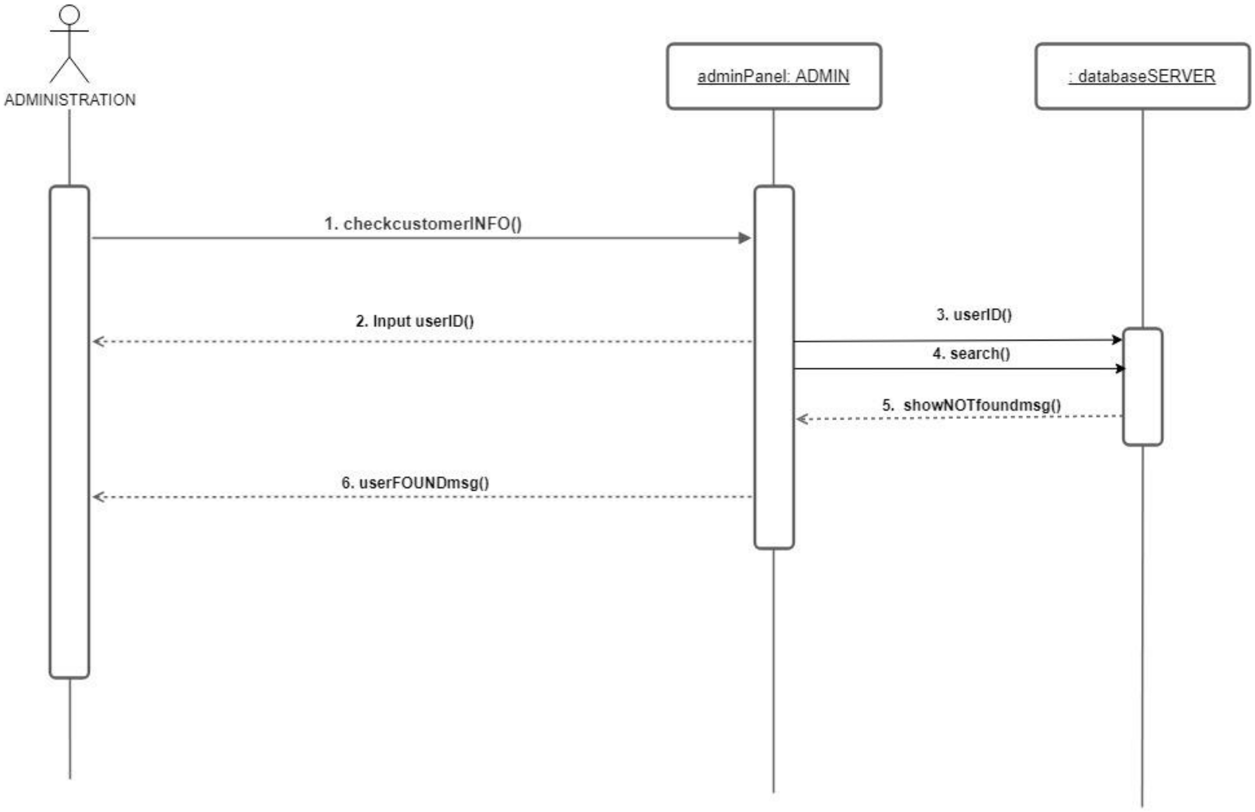
Sequance #2



Log In Customer:

The sequence diagram represents that customer requires to login into the system. The customer selects the login option from customer panel. Then customer will get a message from system to provide email and password. After that, the system will create an object for storing the login information that customer has provided to the system via email and Password(). Therefore, the system will enquire the database for verification, to check the account whether exist or not. If the target account does not exist, the system will return the error message to customer and system may remind the customer to register an account or “forget password?” Finally, the database will retrieve other information to the system and if verification is successful and account does exist, then system will return successful message to customer.

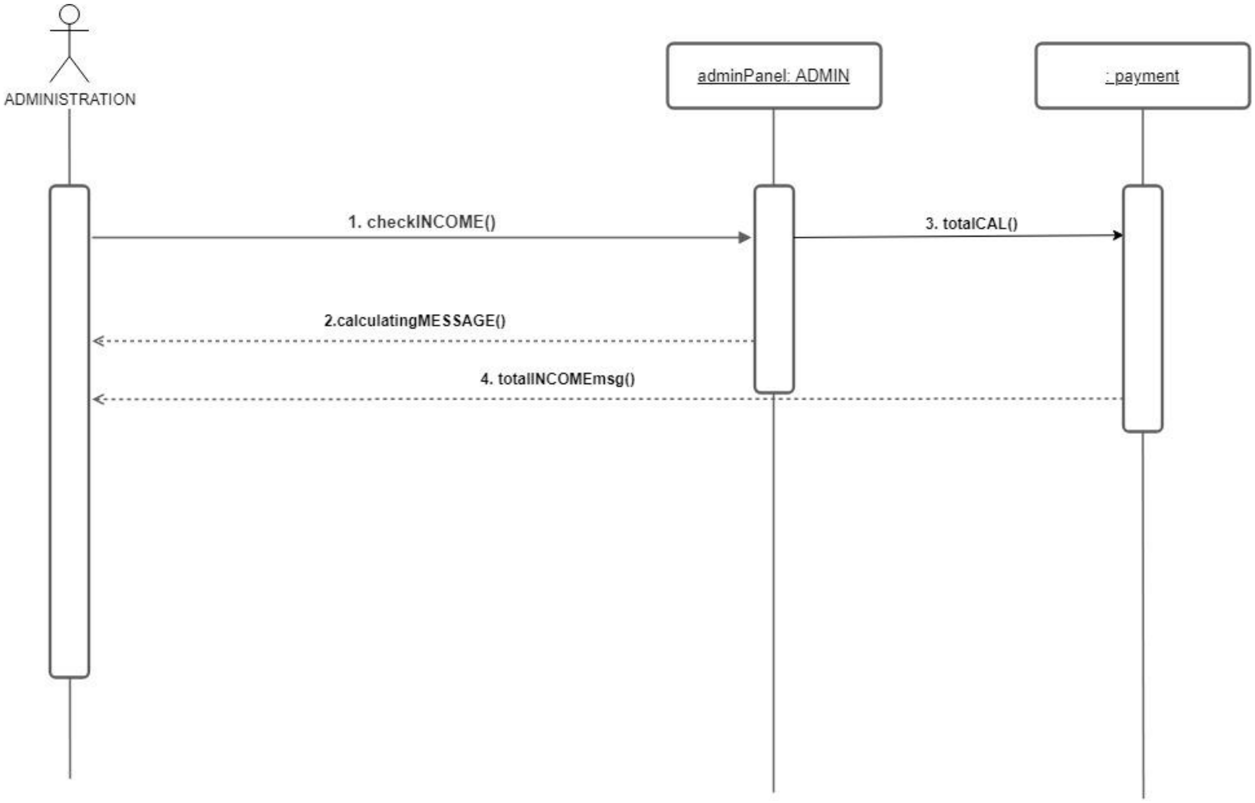
Sequence #3



Check Customer Information:

The sequence diagram represents that check customer information into the system via administration. To check the customer information administration need to select the checkcustomerINFO() from administration panel. Then administration will get a message from system to provide userID. After that the admin will provide the customer userID and system will send it to database server for searching. If the UserID doesn’t match with the stored data then system will show showNOTfoundmsg() message and if the UserID does match with the stored data then system will show userfoundmsg() message.

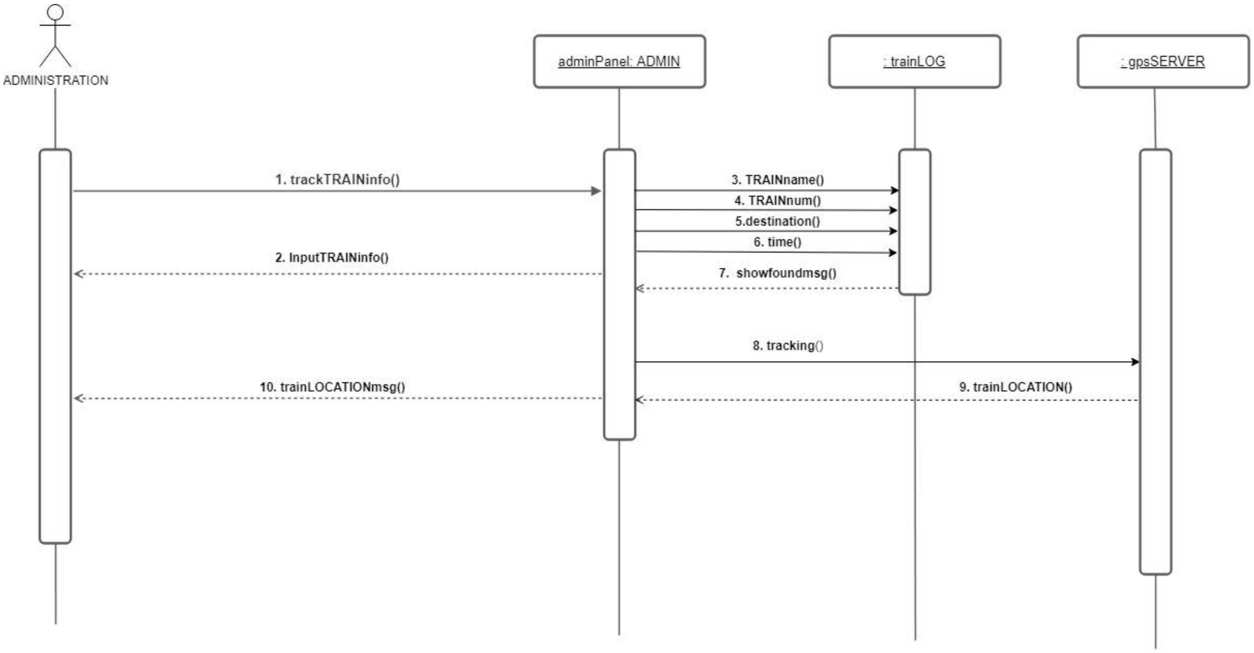
Sequence #4:



CheckIncome

The sequence diagram represents that administration requires to CheckIncome from the system. If the administration wants to see the per day total income, administration need to select checkINCOME option from the admin panel. Then administration will get a message from the system that it is calculating the amount. After that the system will show the totalINCOMEmsg message.

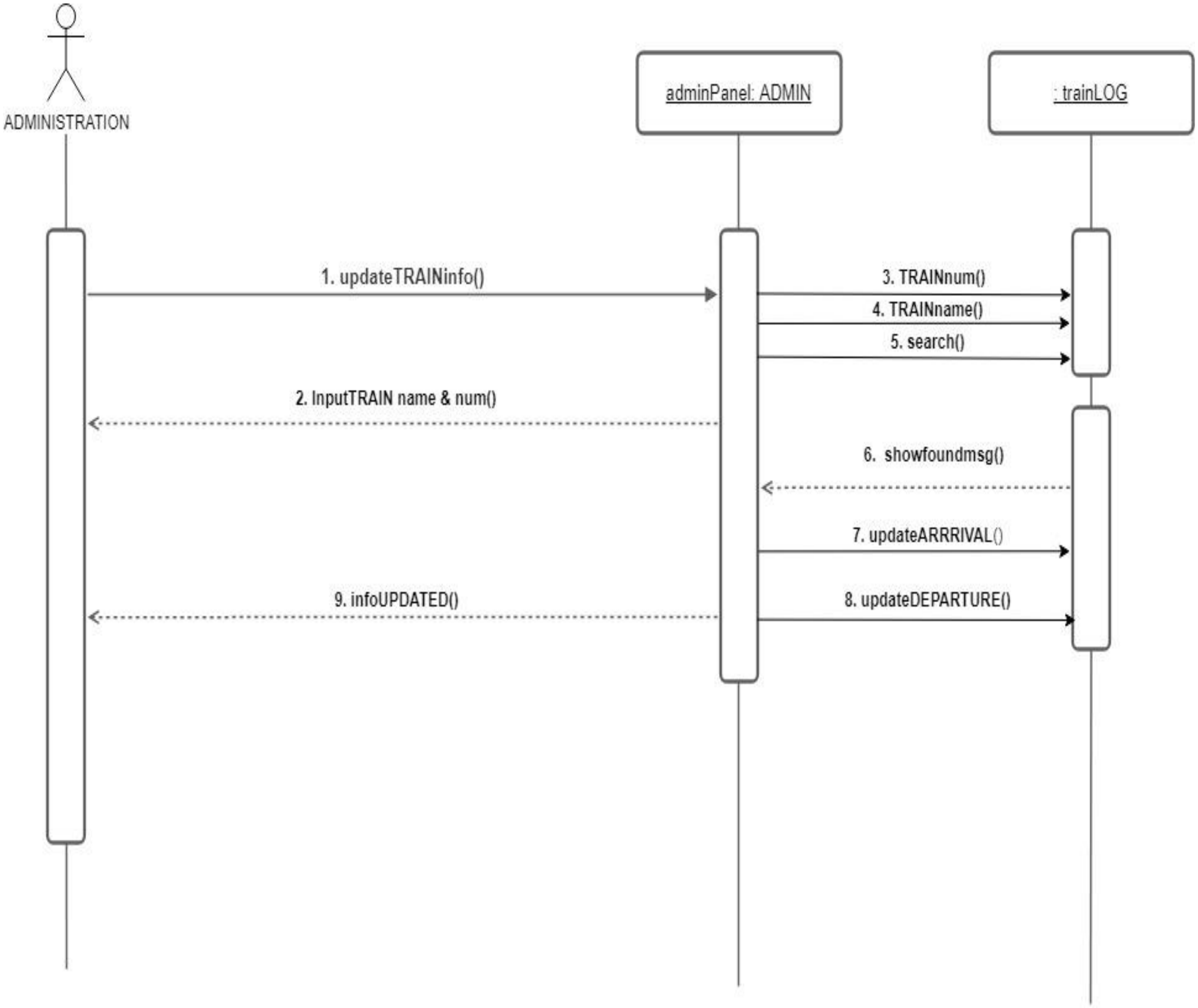
Sequence #5



Track Train

The sequence diagram represents that administration requires to Track Train using the system. To track the train information administration need to select the trackTRAINinfo() from admin panel. Then administration will get a message from system to provide train information. After that admin will provide train name, train umber, destination, and time. If the train information match, admin will get showfoundmsg() message from system. Then admin will track the train from gpsSERVER and system will sent trainlocation message to administration through the admin panel.

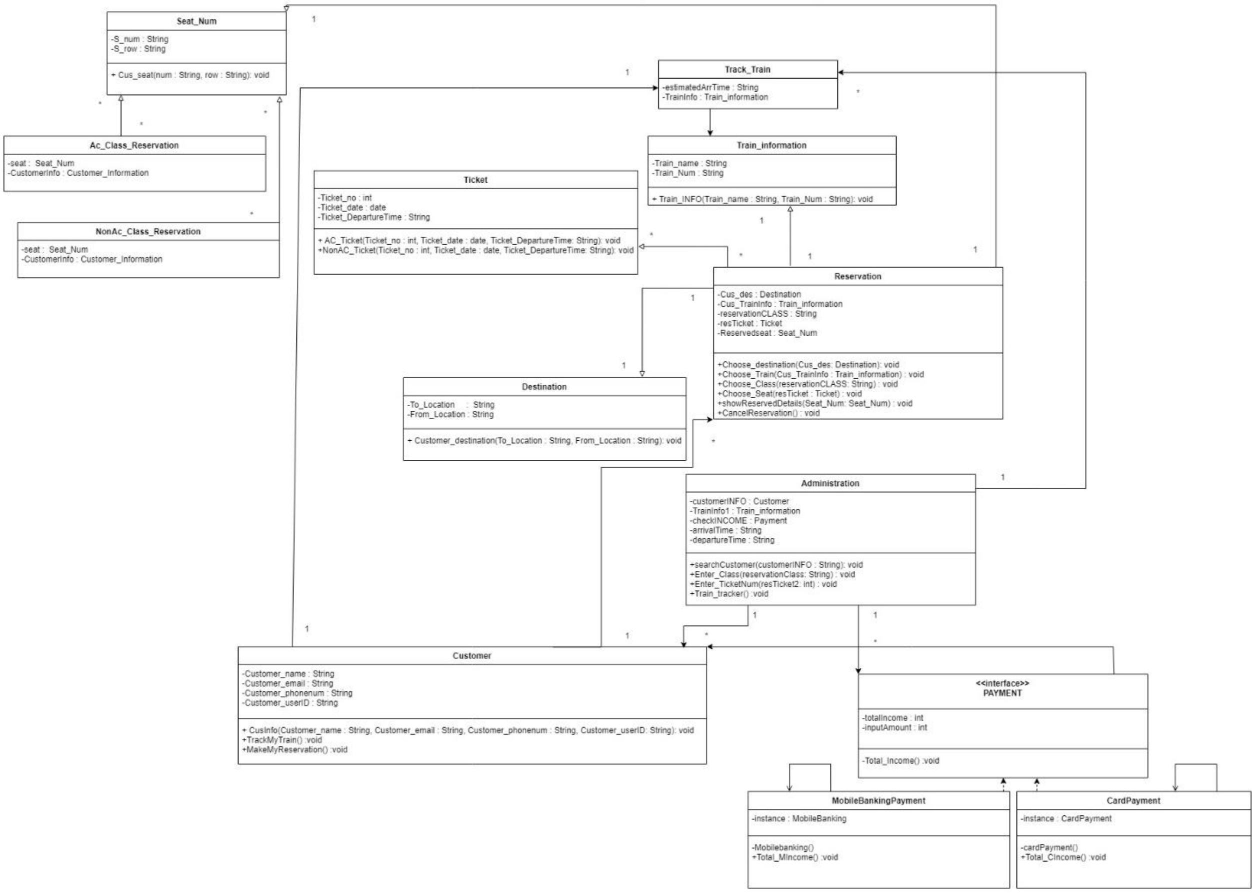
Sequence #6



Update Train Information:

The sequence diagram represents that administration requires to update train information using the system. To update the train information, administration need to select updateTRAINinfo. Then administration will get a message from system to provide train name and train number. After that admin will provide train name, train number. Admin will search the train according to train name and number. If the name and number does match admin will get a showfoundmsg(). Then administration need to update updateARRIVAL() and updateDEPARTURE(). After updating administration will get infoUPDATED() message from system.

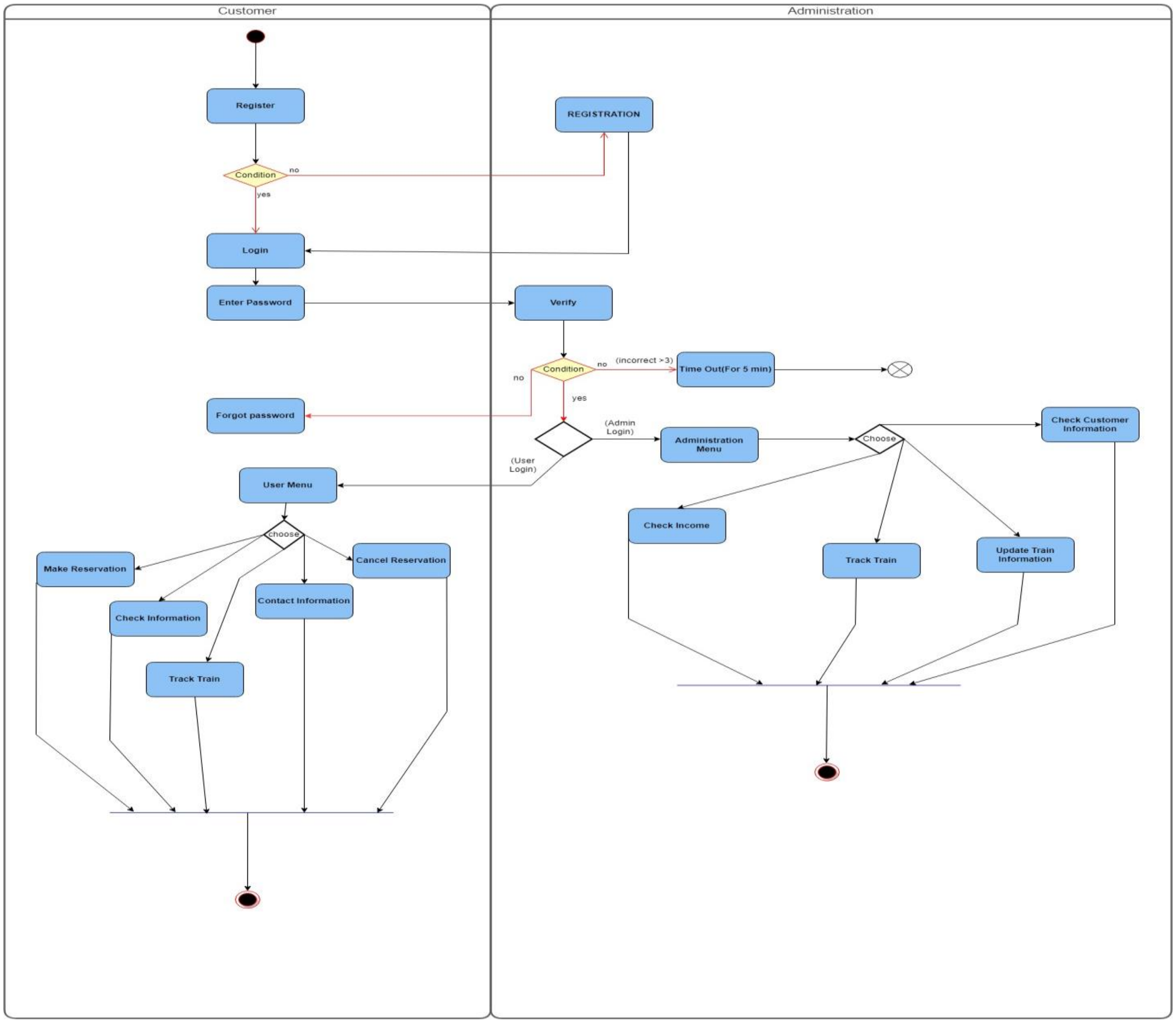
Class Diagram:


Seat_Num
-S_num : String
-S_row : String
+ Cus_seat(num : String, row : String): void
Ac_Class_Reservation
-seat : Seat_Num
-CustomerInfo : Customer_Information
NonAc_Class_Reservation
-seat : Seat_Num
-CustomerInfo : Customer_Information
Ticket
-Ticket_no : int
-Ticket_date : date
-Ticket_DepartureTime : String
+ AC_Ticket(Ticket_no : int, Ticket_date : date, Ticket_DepartureTime: String): void
+NonAC_Ticket(Ticket_no : int, Ticket_date : date, Ticket_DepartureTime: String): void
Track_Train
-estimatedArrTime : String
-TrainInfo : Train_information
Train_information
-Train_name : String
-Train_Num : String
+ Train_INFO(Train_name : String, Train_Num : String): void
Reservation
-Cus_des : Destination
-Cus_TrainInfo : Train_information
-reservationCLASS : String
-resTicket : Ticket
-Reservedseat : Seat_Num
+Choose_destination(Cus_des: Destination): void
+Choose_Train(Cus_TrainInfo : Train_information) : void
+Choose_Class(reservationCLASS: String) : void
+Choose_Seat(resTicket : Ticket) : void
+showReservedDetails(Seat_Num: Seat_Num) : void
+CancelReservation() : void
Destination
-To_Location : String
-From_Location : String
+ Customer_destination(To_Location : String, From_Location : String): void
Administration
-customerINFO : Customer
-TrainInfo1 : Train_information
-checkINCOME : Payment
-arrivalTime : String
-departureTime : String
+searchCustomer(customerINFO : String): void
+Enter_Class(reservationClass: String) : void
+Enter_TicketNum(resTicket2: int) : void
+Train_tracker() :void
Customer
-Customer_name : String
-Customer_email : String
-Customer_phonenum : String
-Customer_userID : String
+ CusInfo(Customer_name : String, Customer_email : String, Customer_phonenum : String, Customer_userID: String): void
+TrackMyTrain() :void
+MakeMyReservation() :void
<<interface>>
PAYMENT
-totalIncome : int
-inputAmount : int
-Total_Income() :void
MobileBankingPayment
-instance : MobileBanking
-Mobilebanking()
+Total_MIncome() :void
CardPayment
-instance : CardPayment
-cardPayment()
+Total_CIncome() :void
1
1
*
*
*
*
1
*
1
1
*
1
1
1
1
*
1
1
1
1
*
1
*

Description Of Class Diagram:

For all customers, there is an administration. A customer can make multiple reservation. To make reservation customer can buy multiple ticket. One reservation has at most one destination. A customer will be able to know the information of the train in which the reservation will make and track the train. Administration can be able to know the of all train information. For one reservation there will be a one seat number. There are two types of class for reservation. AC class and non-AC class. For the payment method, there are only two kind of payment method which are pay by mobile banking and credit card. A customer can pay via mobile banking and card with the help of interface name payment. Admin can be able to see total income per day.

Activity Diagram:

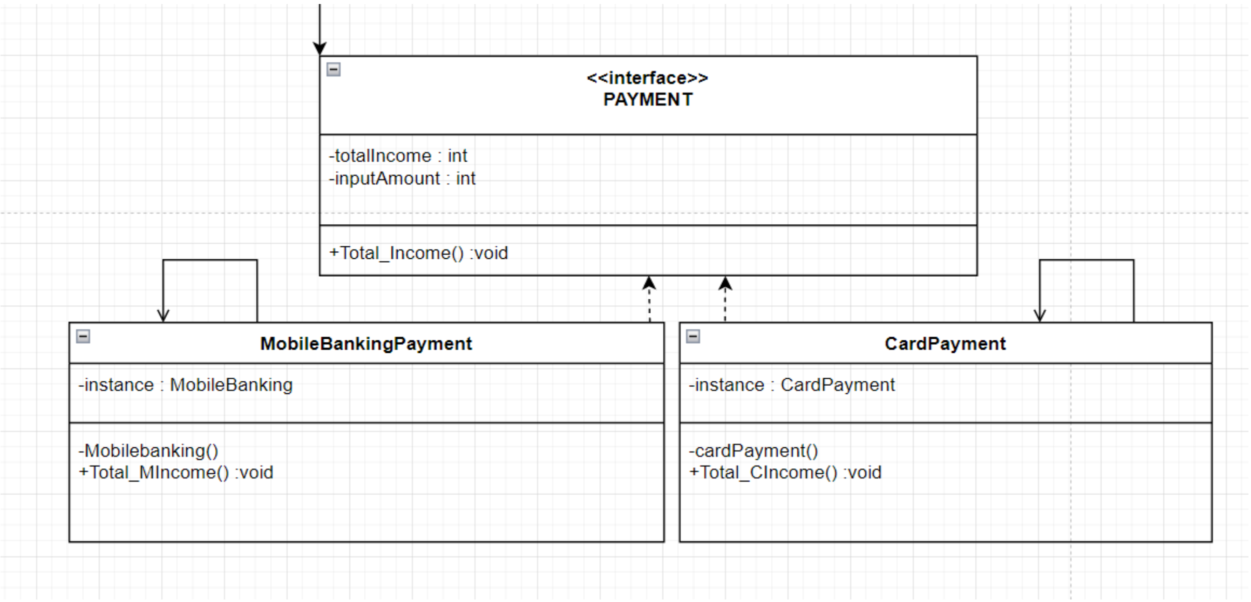


Utilization of Software Design Principles and Patterns:

In our system, we have implemented numbers of principles and patterns. For the principles we implemented, they are Open-Closed Principle (OCP), Liskov Substitution Principle (LSP), Dependency Inversion Principle (DIP) and Law of Demeter (LoD). We were able to implement Singleton Pattern in our project.

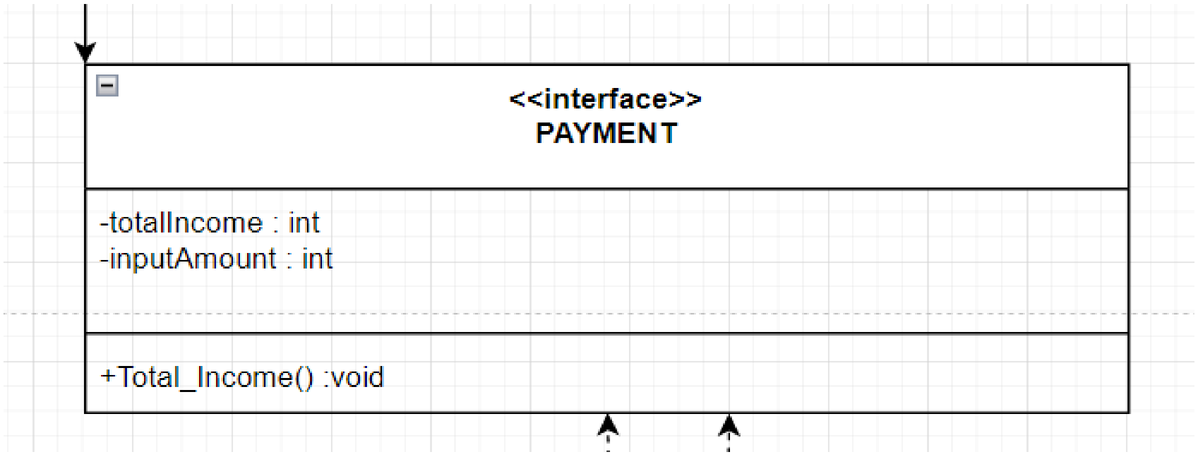
OPEN-CLOSED PRINCIPLE:

The OCP states that the entity is open for extension but closed for modification. The Class Payment is an interface. So, if there is a new type of payment method that can be supported by the system, according to OCP we can extend the payment class and make necessary adjustment. Because it is open for extension but we can’t change the fundamental part of the class.Thus, OCP is applicable in our project system.



LISKOV SUBSTITUTION PRINCIPLE:

LSP states that derived subclasses (sub-types) must be completely substitutable for their parent class (base type). The CardPayment and MobileBanking subclasses are substituable for the parent class Payment.Because the methods stated in the parent class are fully implemented in both child/derived classes. So, if we implement interface class with either dervied classes or system will run properly.Thus, we can say that LSP is applicable in our system.



DEPENDECY INVERSION PRINCIPLE:

In the Administration class, depends on the Interface named Payment rather than any subclass of Payment.With this design, DIP is applicable for our system which provides a better way to extend the function.

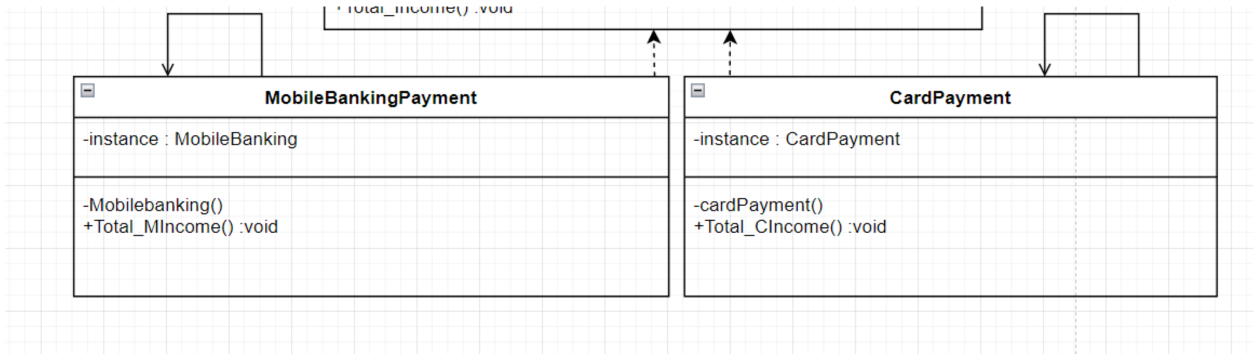
SINGLE RESPONSIVILTY PRINCIPLE:

SRP states this principle states that each class should have one responsibility, one single purpose. This means that a class will do only one job, which leads us to conclude it should have only one reason to change.In our project we made classes following this principle so we devided each of the class and their operations for single purpose only.

DESIGN PATTERNS:

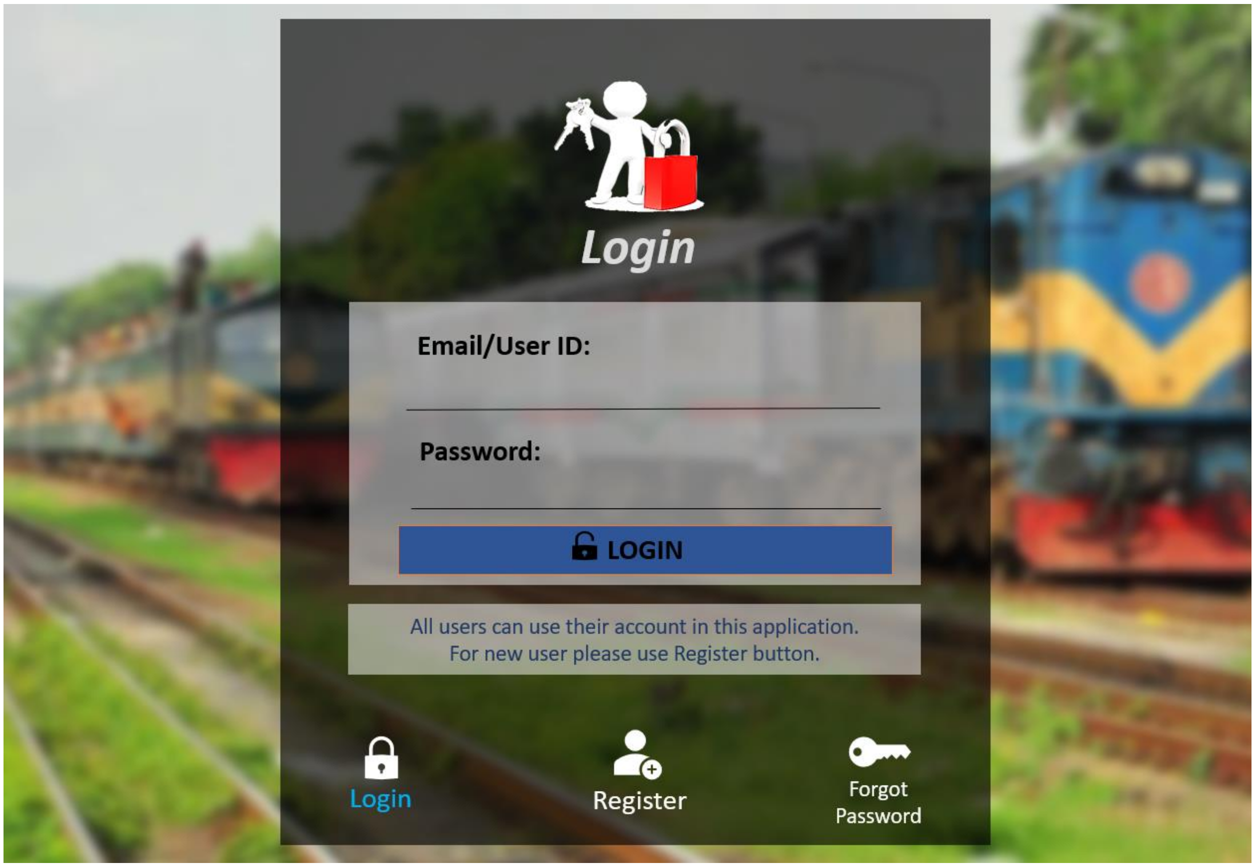
Singleton:

The singleton pattern is one of the design patterns. In our system, we can see that the system restricts the instantiation of a class to one "single" instance. Therefore, same instance can be used from everywhere and invoke directly the constructor each time. Sometimes it's useful to have only one instance for a class. For instance, in a system there should be only one system manager. Singletons are usually used for centralized management of internal or external resources and they provide a global point of access to themselves.



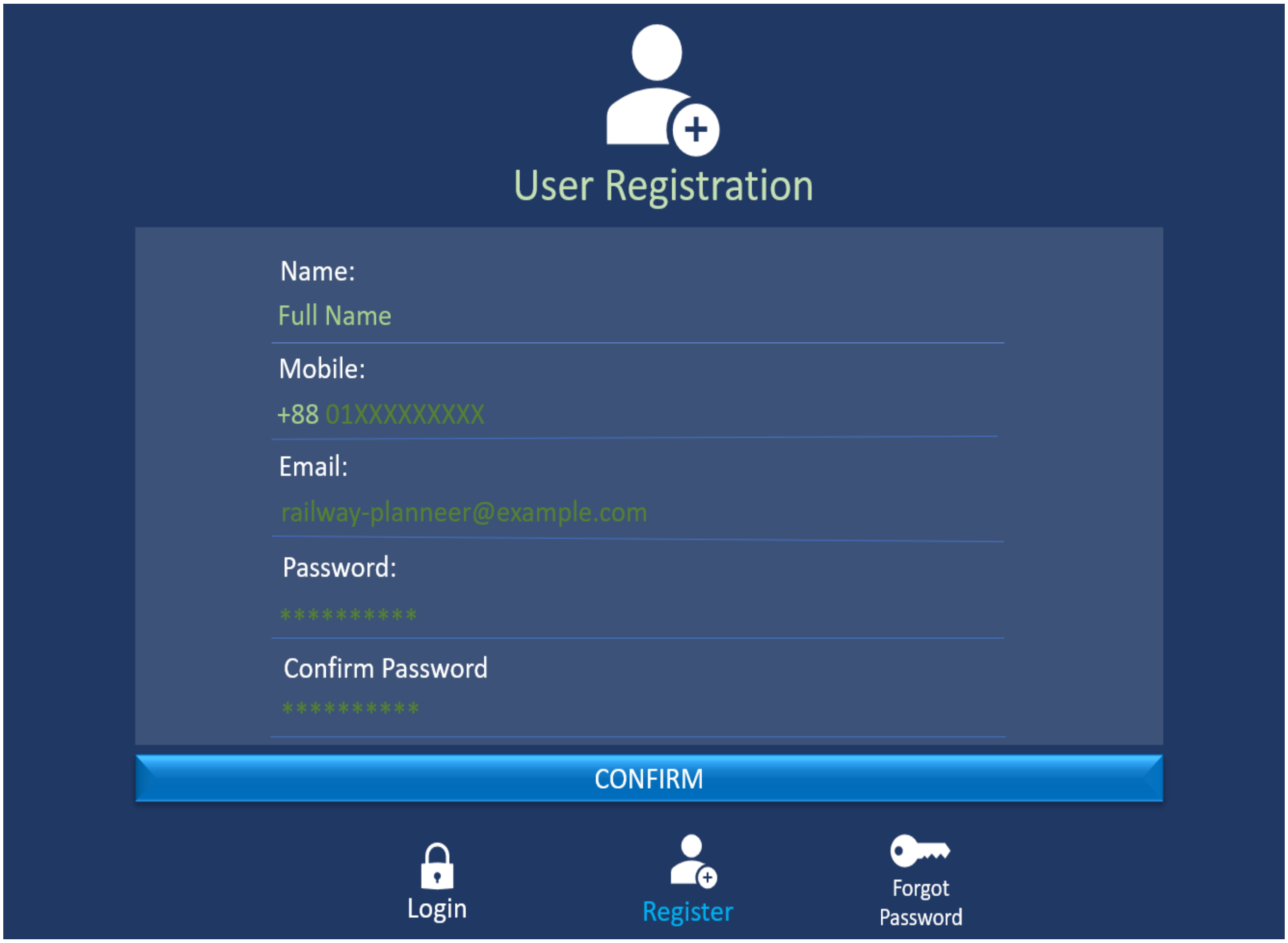
System Prototype:

Login:



All users can use their account in this application.
For new user please use Register button.

User registration:



User Registration

Name:
Full Name

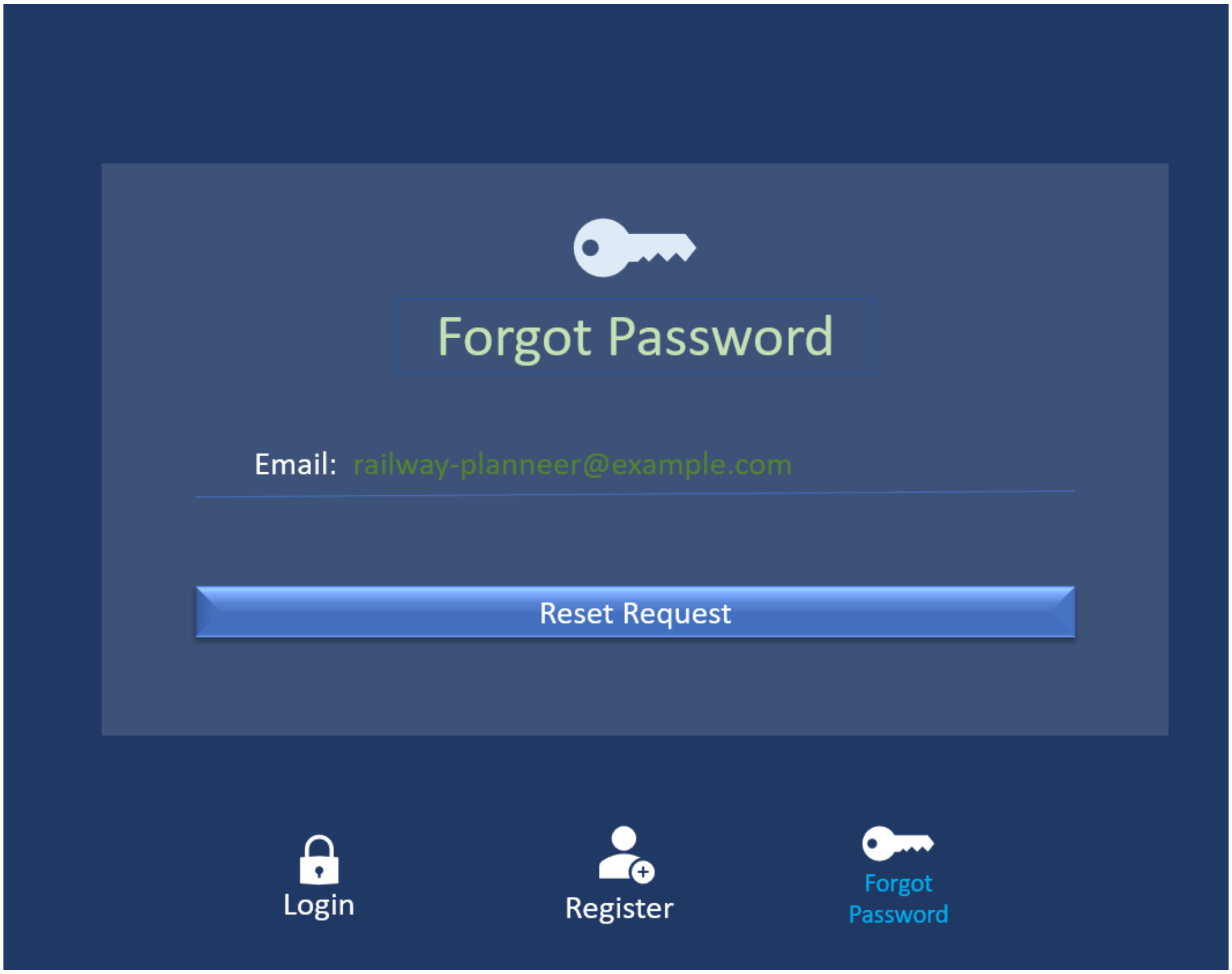
Mobile:
+88 01XXXXXXXXX

Email:
railway-planneer@example.com

Password:
****

Confirm Password
****

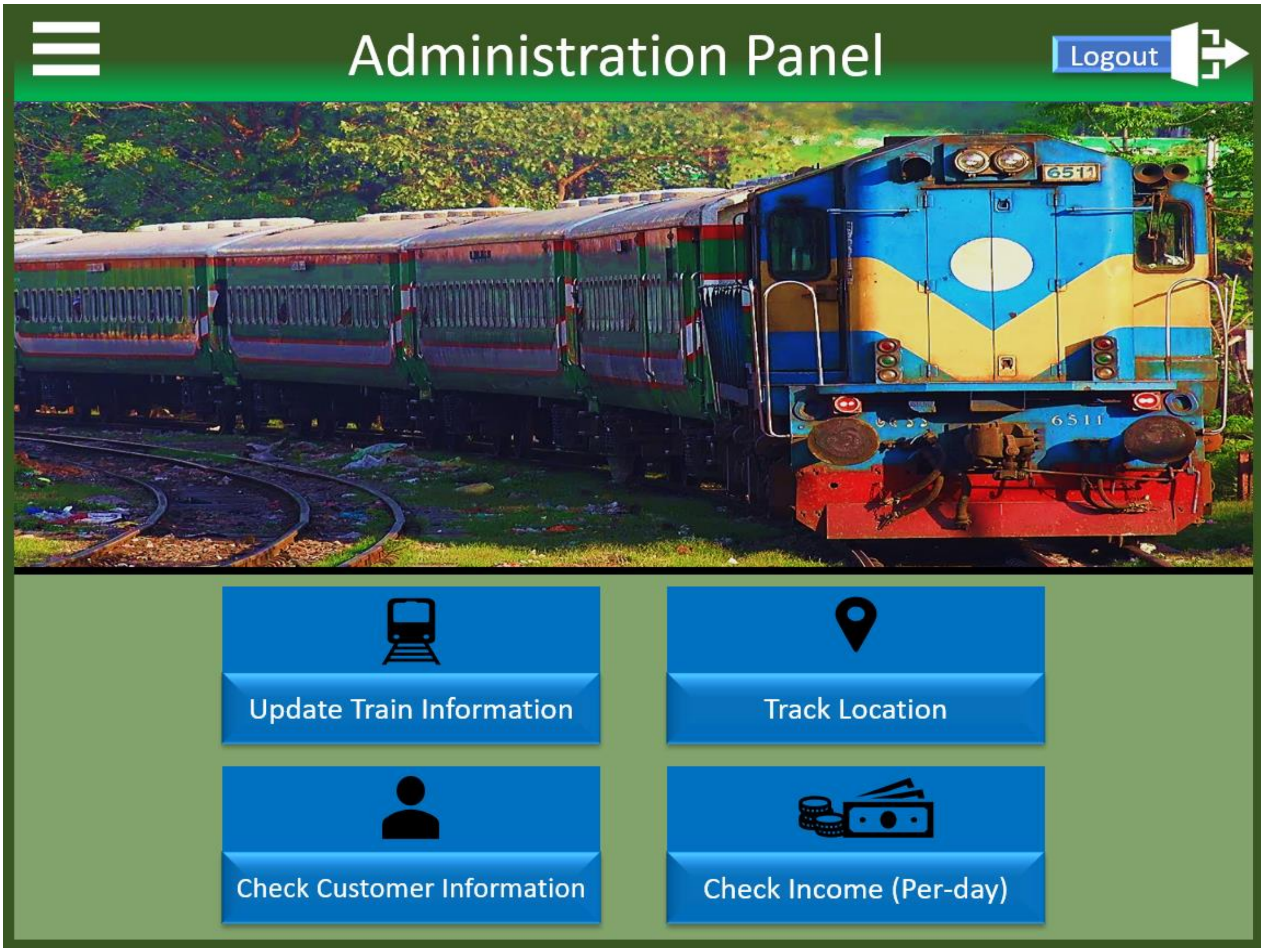
Forgot Password:



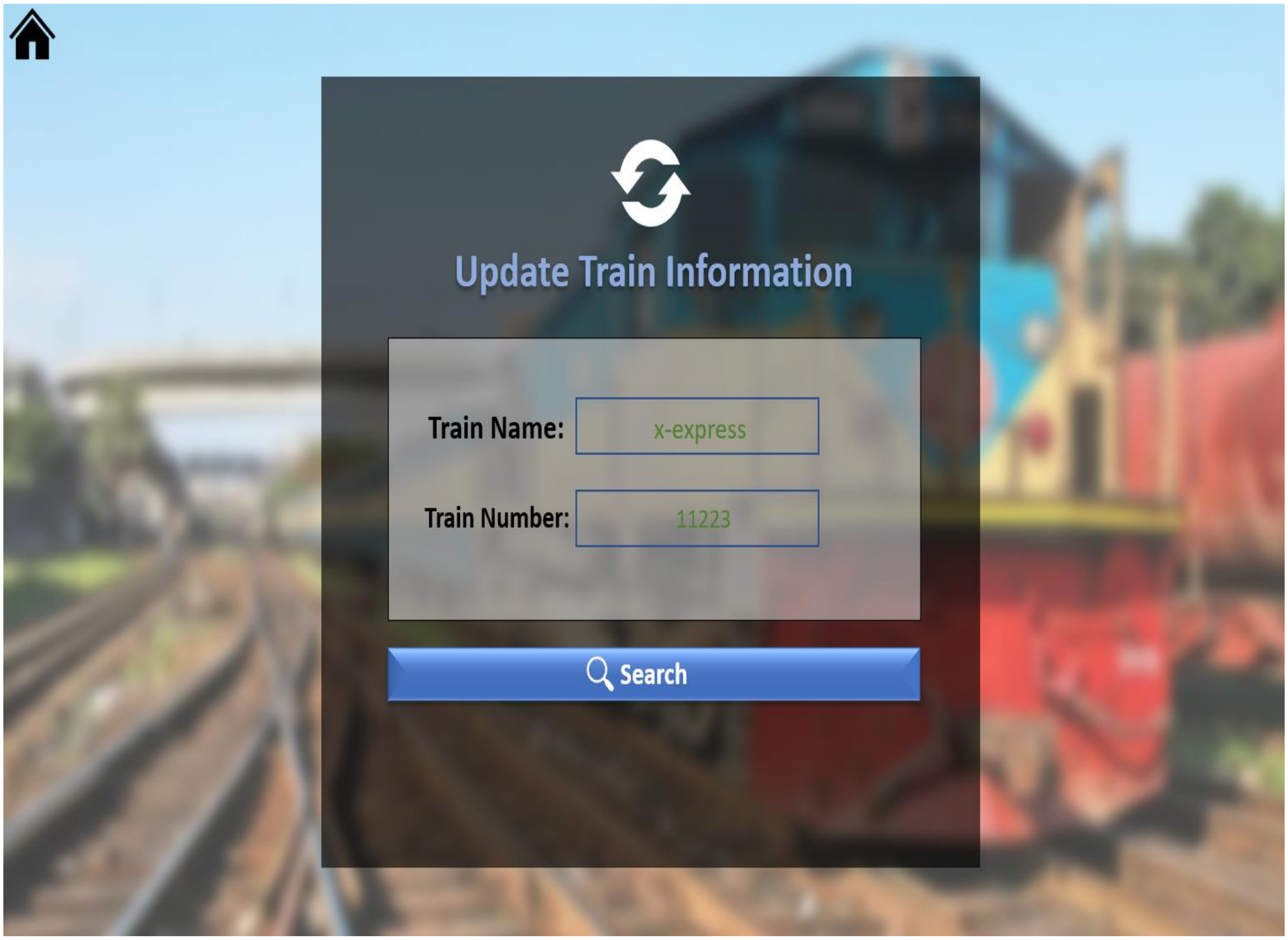
Forgot Password

Email: railway-planneer@example.com

Administration Panel:



Update Train Information:

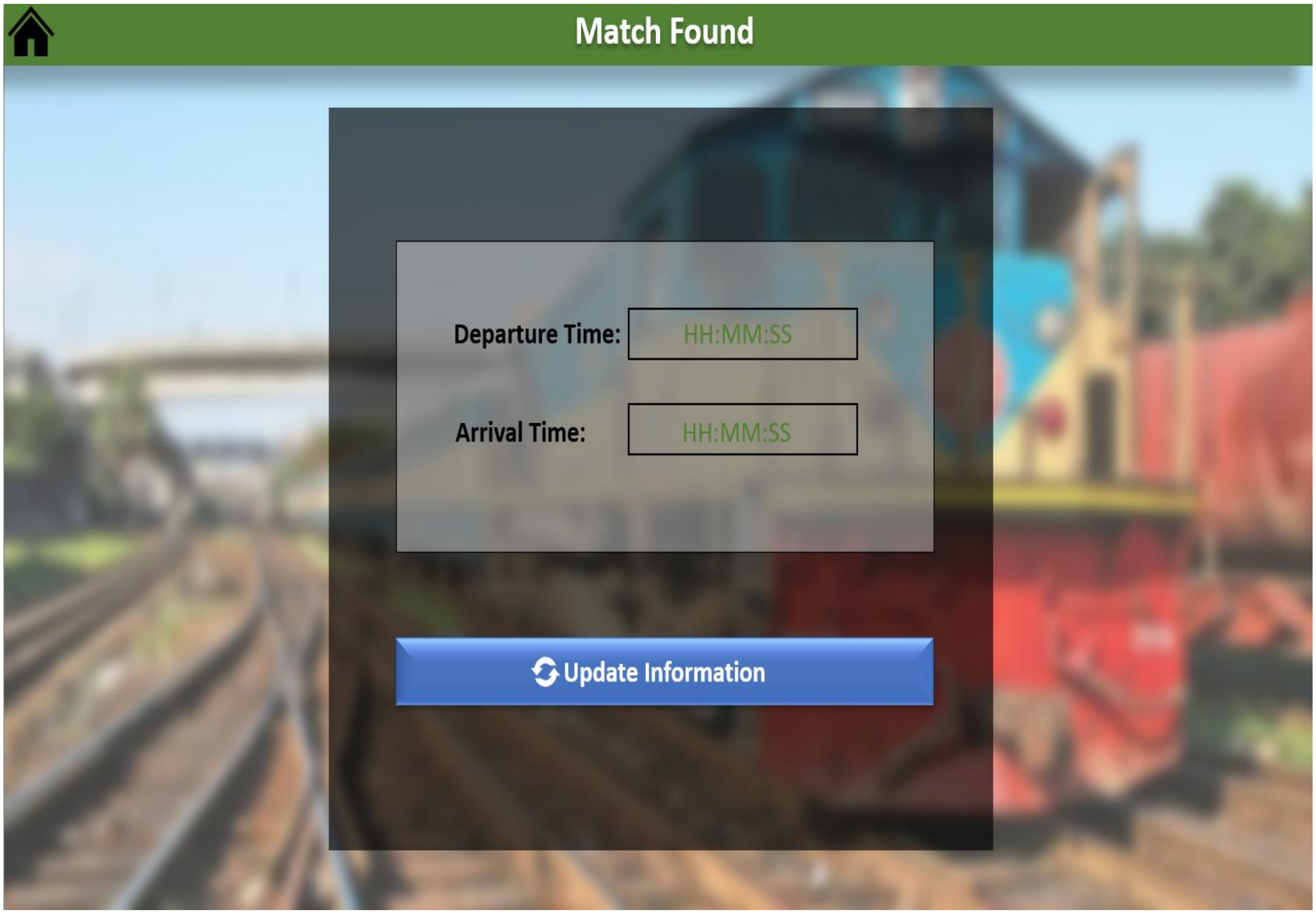


Update Train Information

Train Name: x-express

Train Number: 11223

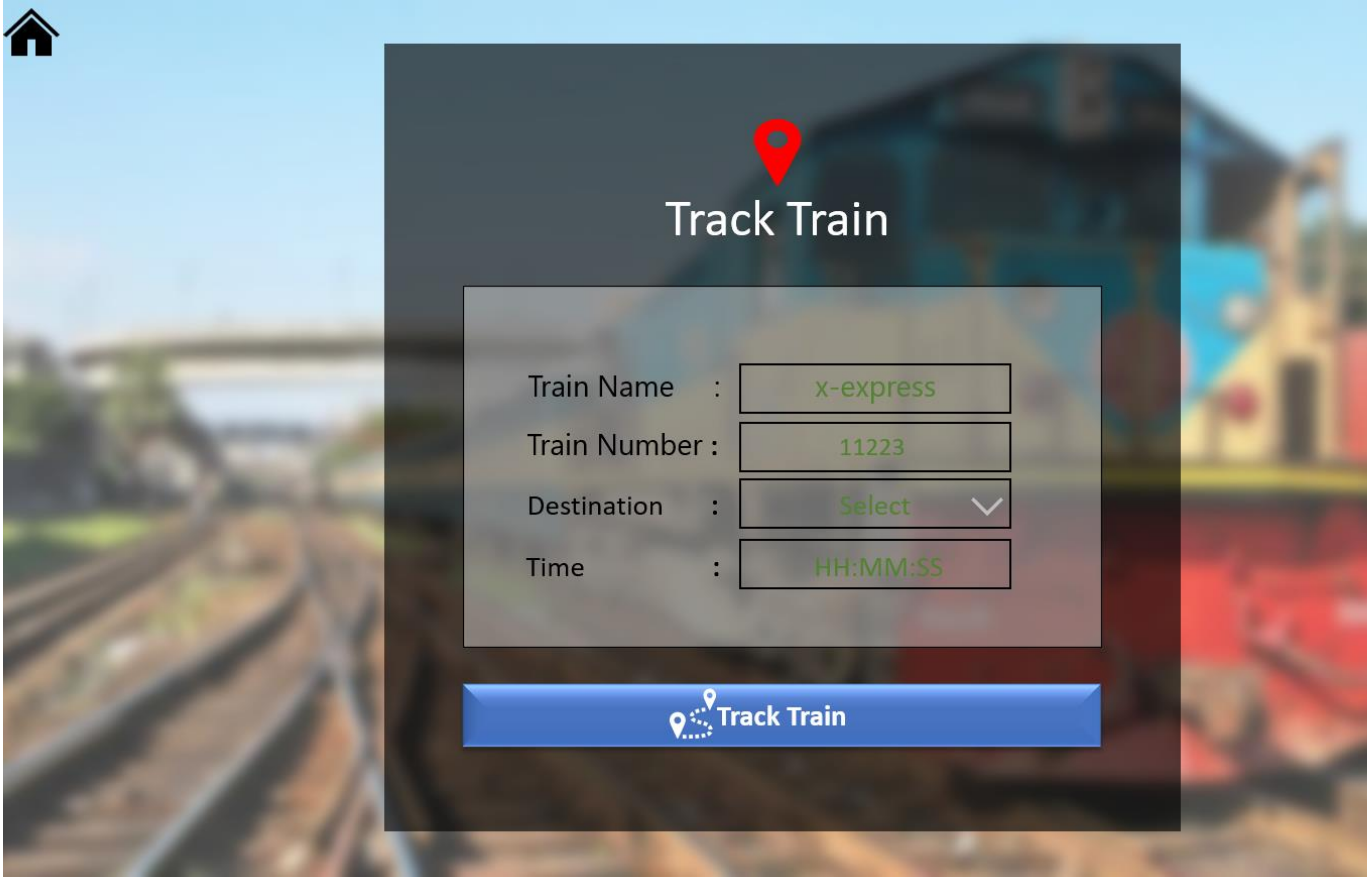
Match Found and updating:


Match Found

Departure Time: HH:MM:SS

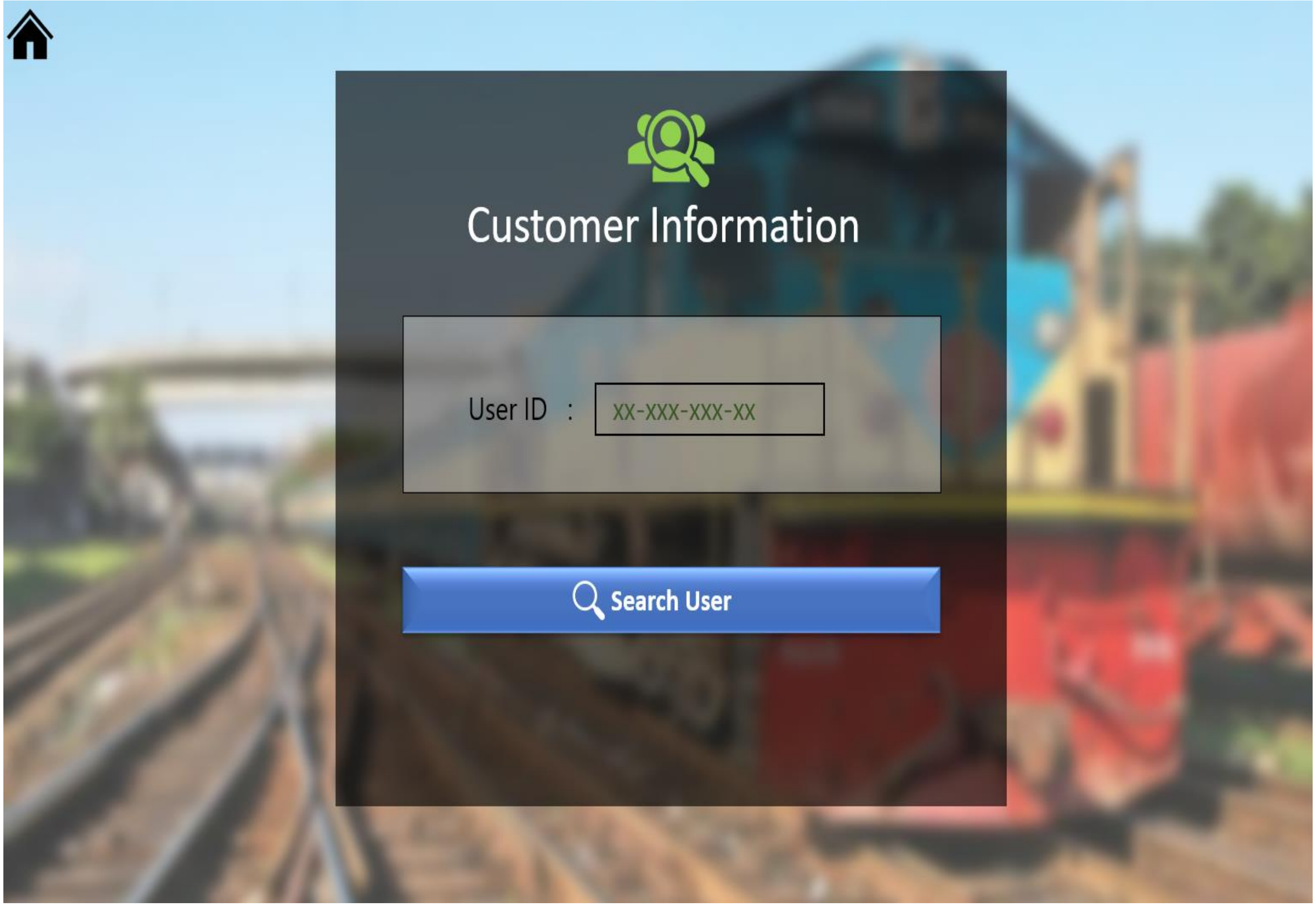
Arrival Time: HH:MM:SS

Track Train:

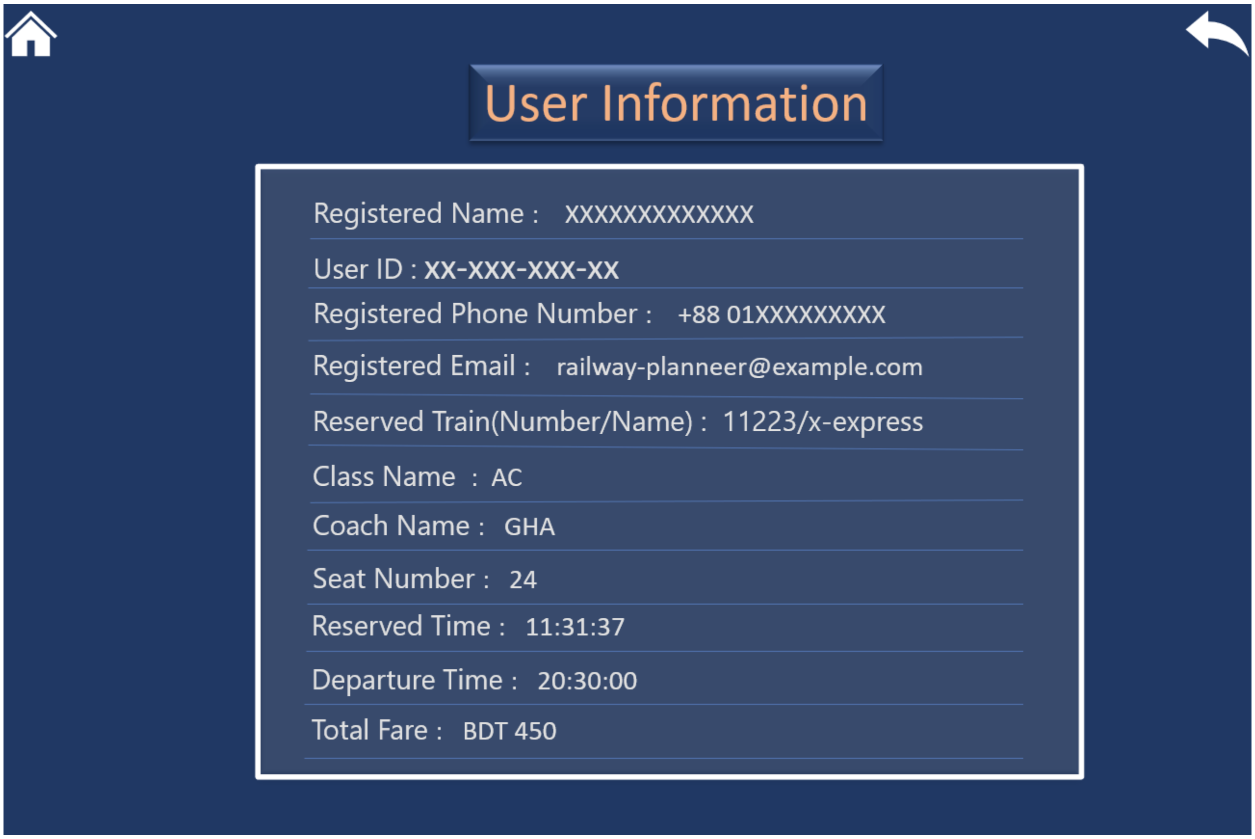


Train Name	:	x-express
Train Number	:	11223
Destination	:	Select
Time	:	HH:MM:SS

Customer Information:



User Information:



User Information

Registered Name : XXXXXXXXXXXXX

User ID : xx-xxx-xxx-xx

Registered Phone Number : +88 01XXXXXXXXX

Registered Email : railway-planneer@example.com

Reserved Train(Number/Name) : 11223/x-express

Class Name : AC

Coach Name : GHA

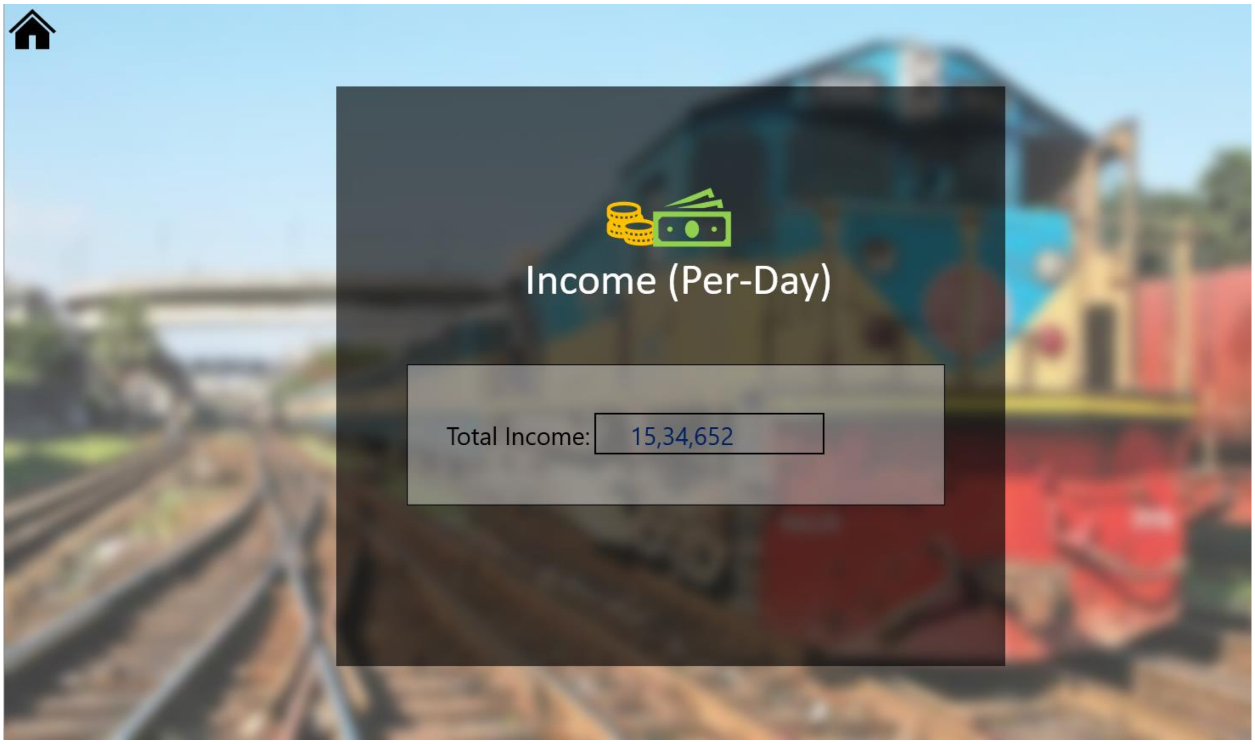
Seat Number : 24

Reserved Time : 11:31:37

Departure Time : 20:30:00

Total Fare : BDT 450

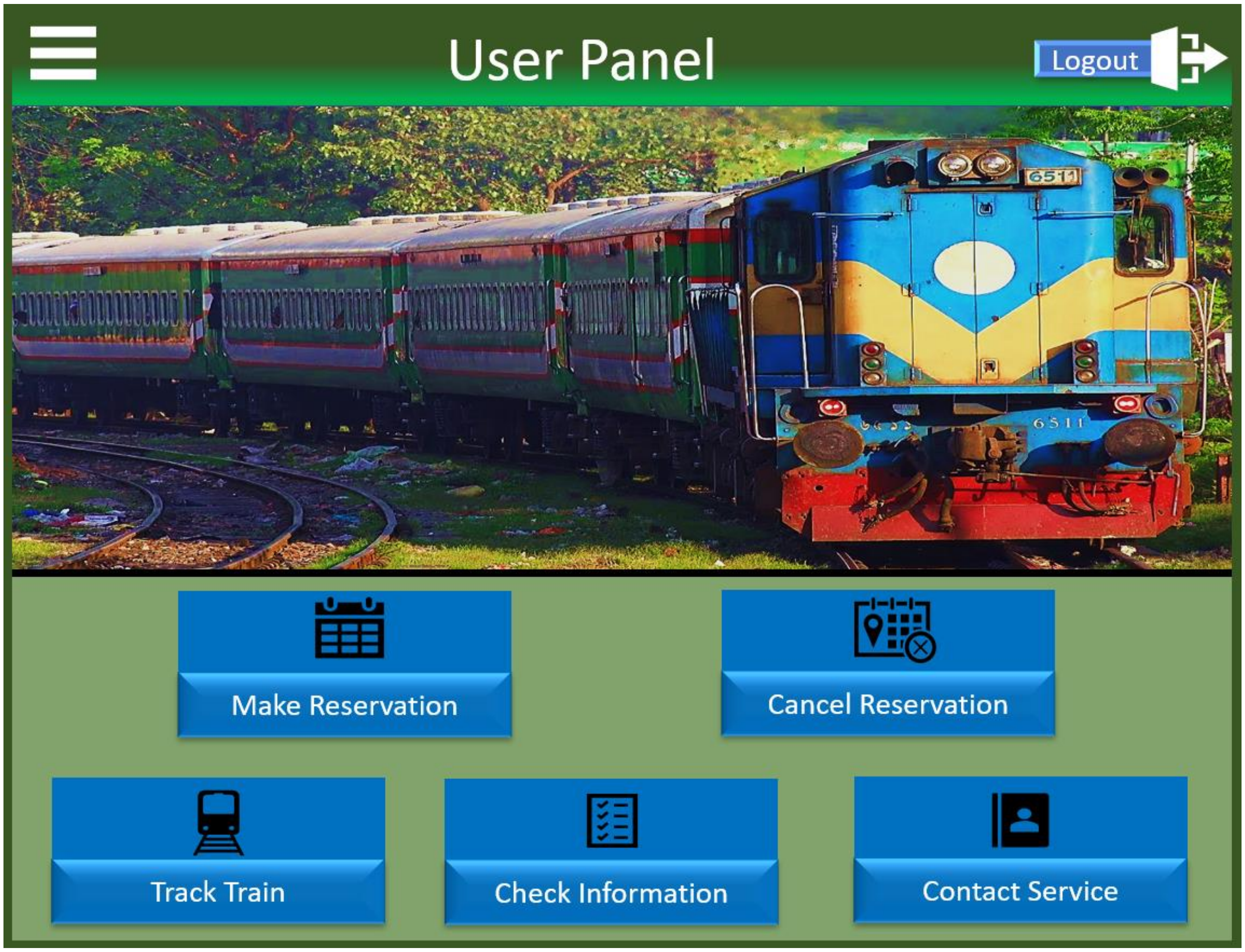
Income:



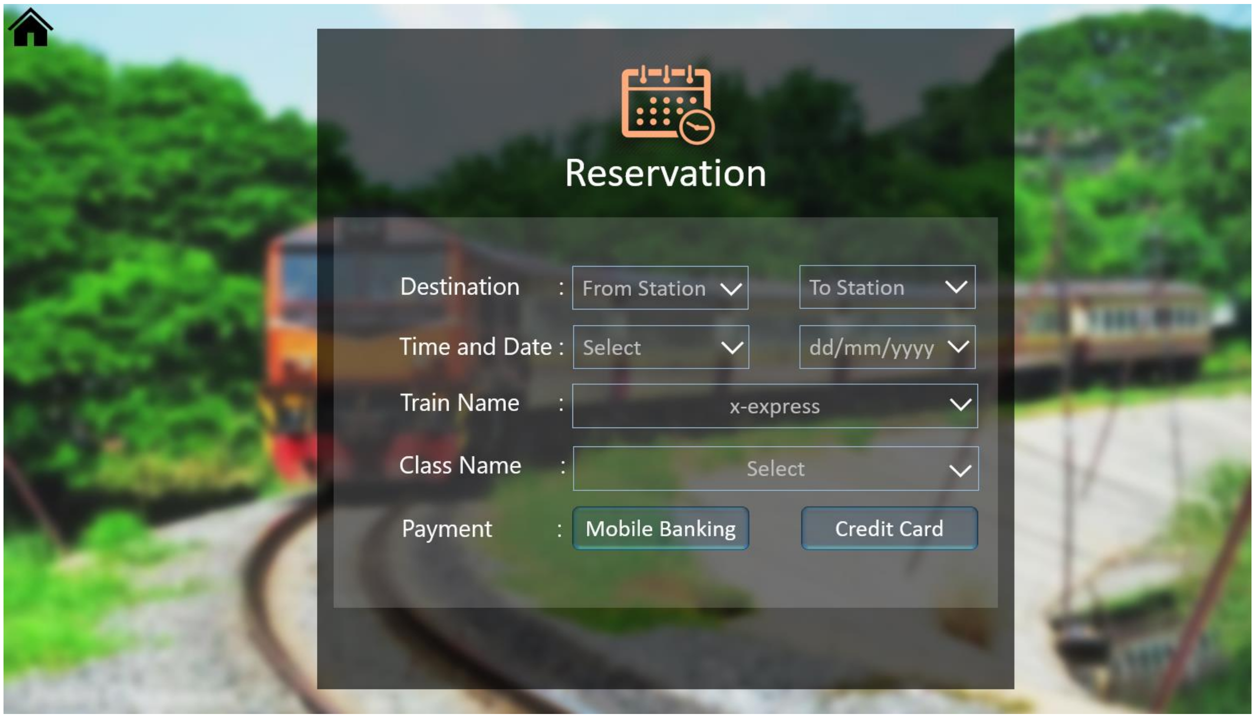
Income (Per-Day)

Total Income: 15,34,652

User Panel:


User Panel
Logout
Make Reservation
Cancel Reservation
Track Train
Check Information
Contact Service

Reservation:



Destination : From Station To Station

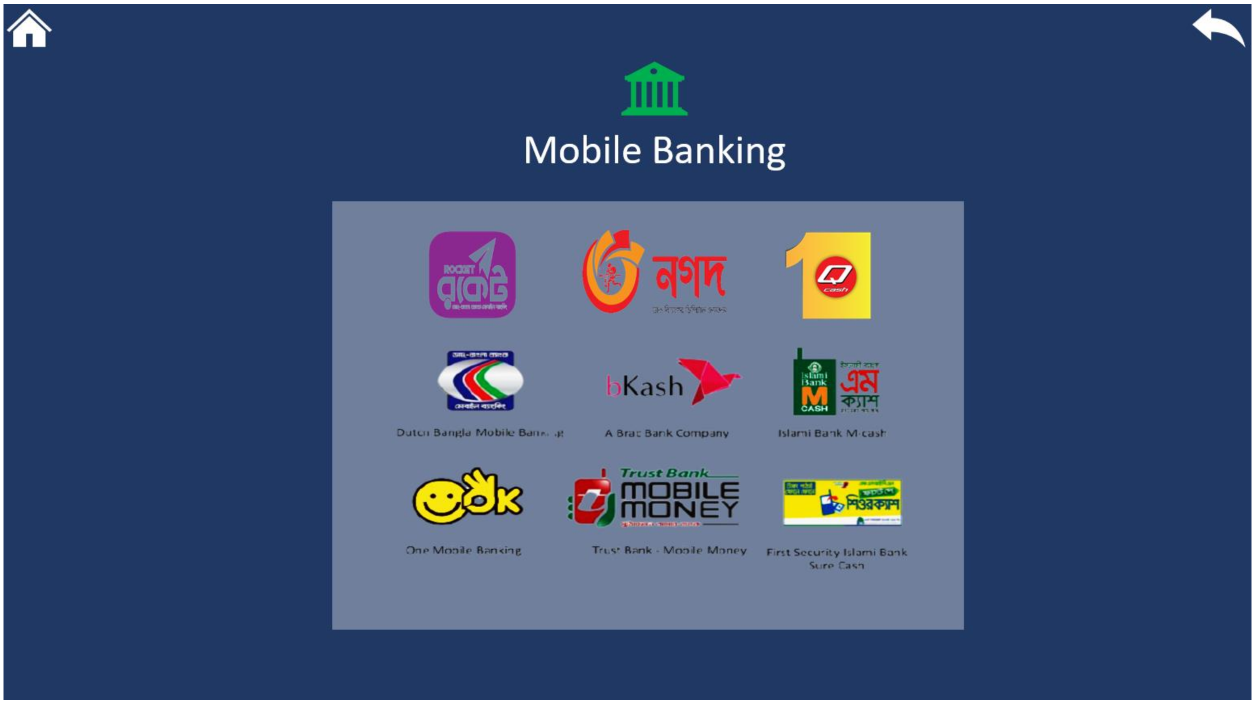
Time and Date : Select dd/mm/yyyy

Train Name : x-express

Class Name : Select

Payment : Mobile Banking Credit Card

Mobile Banking:



Mobile Banking

Dutch Bangla Mobile Banking

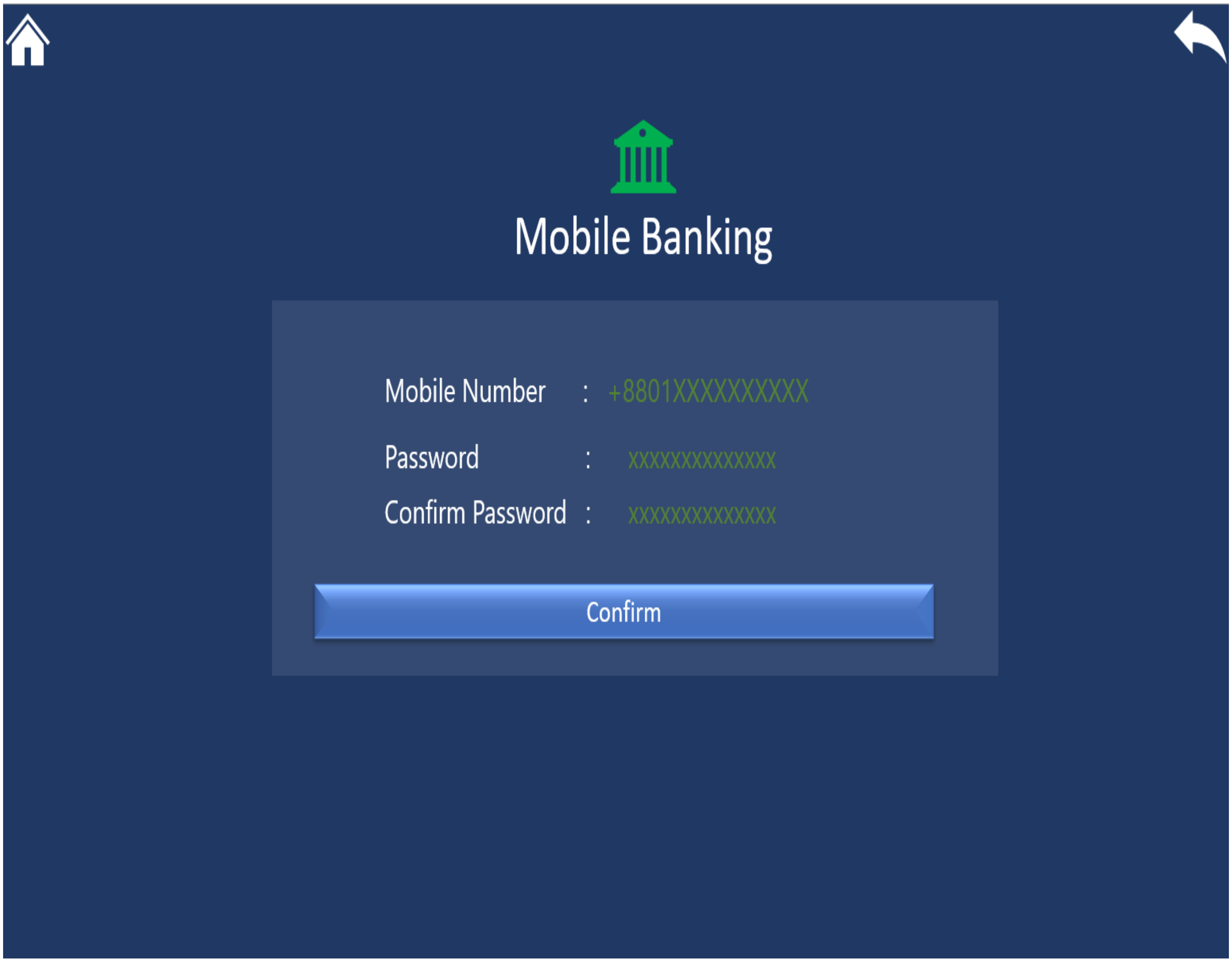
A Brac Bank Company

Islami Bank M-cash

One Mobile Banking

Trust Bank - Mobile Money

First Security Islami Bank Sure Cash



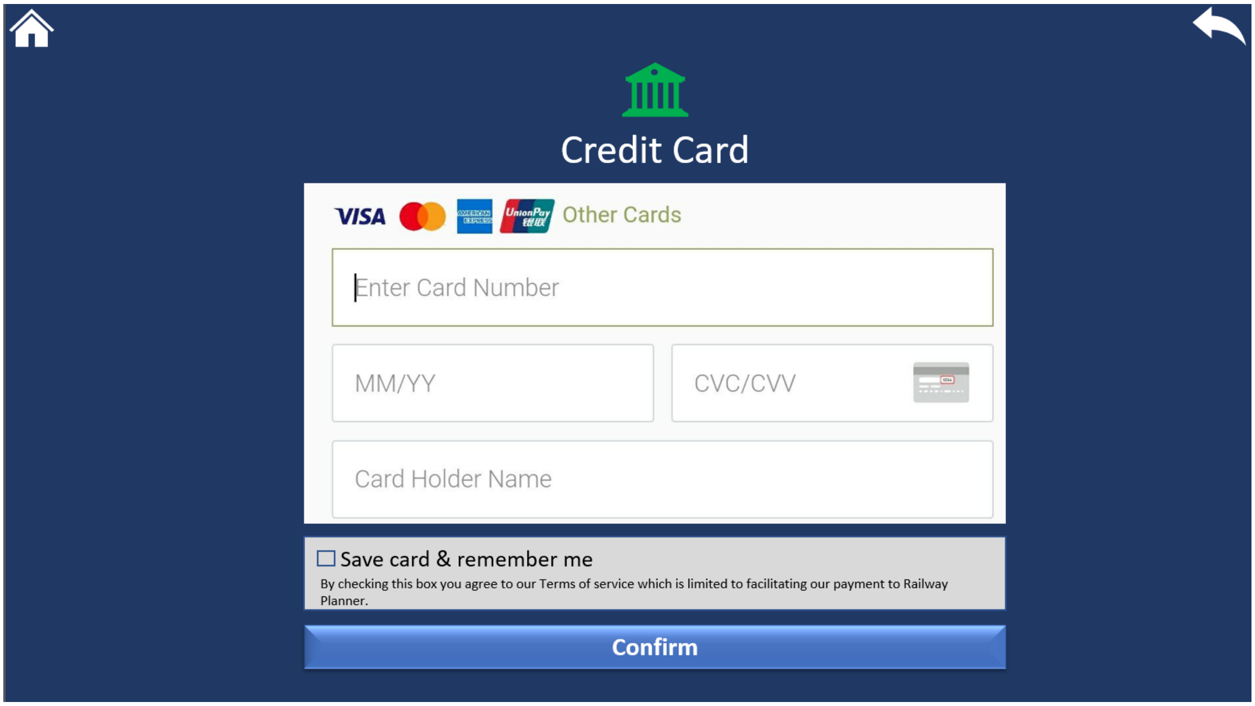
Mobile Banking

Mobile Number : +8801XXXXXXXXXX

Password : xxxxxxxxxxxxxx

Confirm Password : xxxxxxxxxxxxxx

Credit Card:



Credit Card

Enter Card Number

MM/YY

CVC/CVV

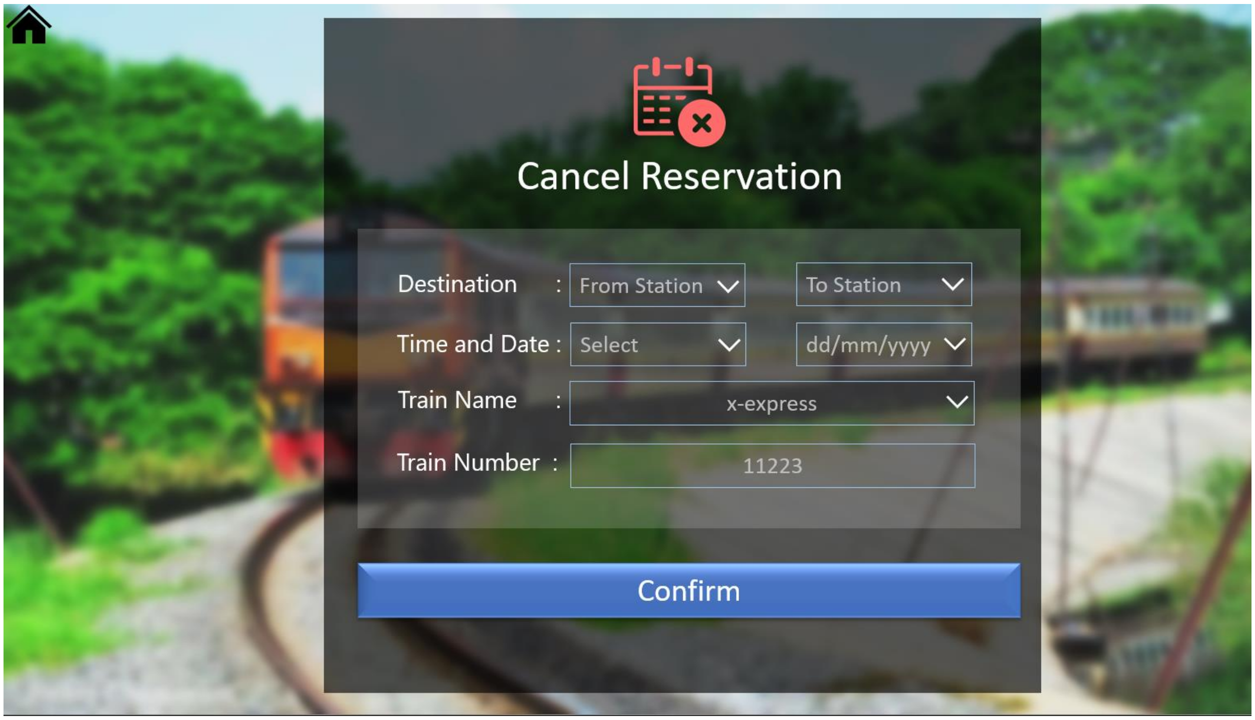
Card Holder Name

□ Save card & remember me

By checking this box you agree to our Terms of service which is limited to facilitating our payment to Railway Planner.

Confirm

Cancel Reservation:



Cancel Reservation

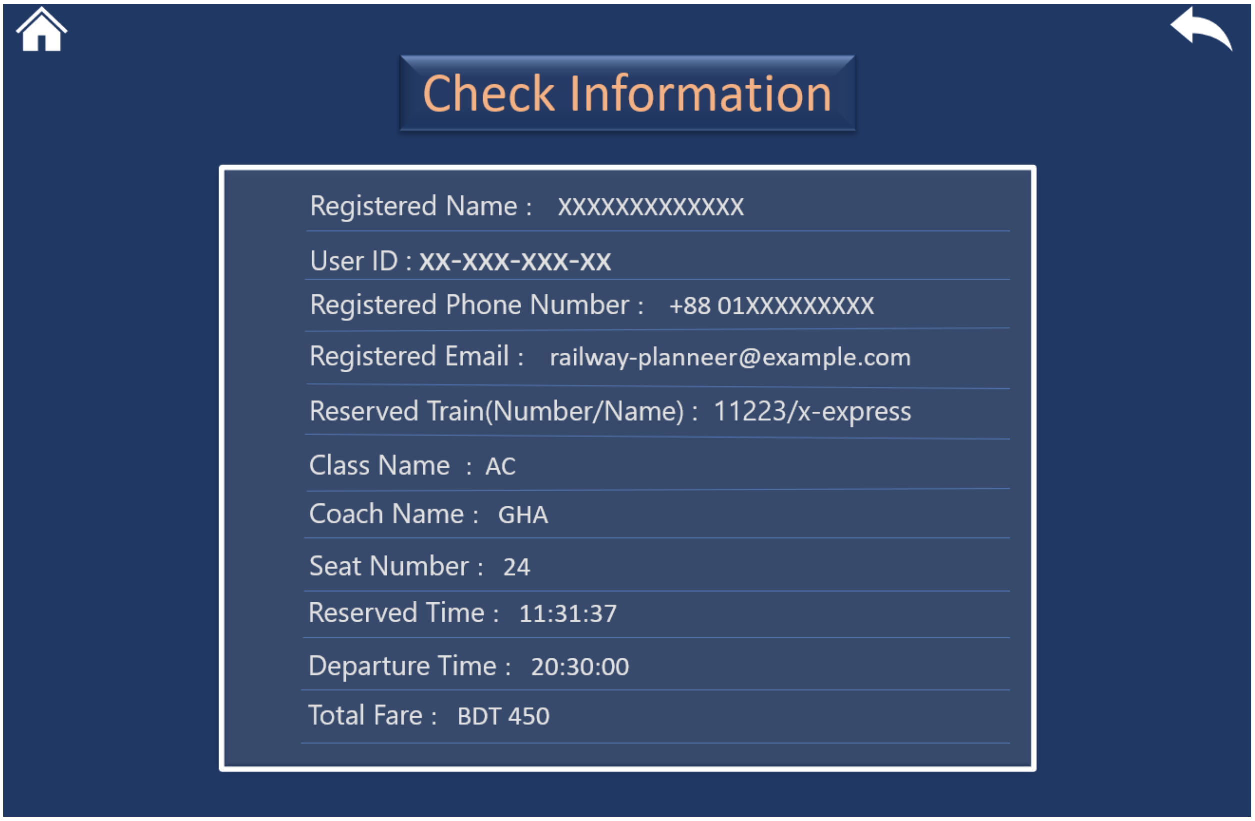
Destination : From Station To Station

Time and Date : Select dd/mm/yyyy

Train Name : x-express

Train Number : 11223

Check Information:



Check Information

Registered Name : XXXXXXXXXXXXX

User ID : xx-xxx-xxx-xx

Registered Phone Number : +88 01XXXXXXXXX

Registered Email : railway-planneer@example.com

Reserved Train(Number/Name) : 11223/x-express

Class Name : AC

Coach Name : GHA

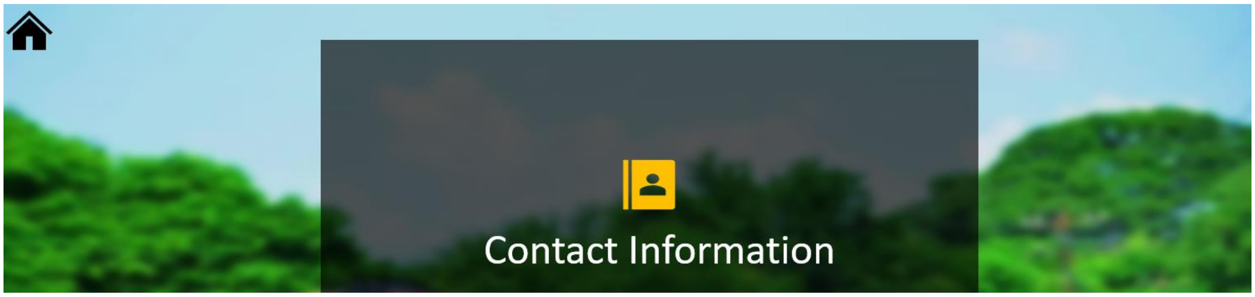
Seat Number : 24

Reserved Time : 11:31:37

Departure Time : 20:30:00

Total Fare : BDT 450

Contact Information:



Project Schedule & Work Allocation:

Week	Weekly Activity Log	Completed By
Week 1	Recruit Project Members to Form a Team	Group Work
Week 2	Project Proposal	Group Work
Week 3	Project's Use Case Diagram and Specification Submission	Group Work
Week 4	Activity Diagram	Group Work
Week 5	Class Diagram	Group Work
Week 6	Sequence Diagram	Group Work
Week 7	Principles	Group Work
Week 8	System Prototype	Group Work

References:

[1] https://easychair.org/publications/preprint/qBV6 [2]