AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

Dept. of Computer Science
Faculty of Science and Technology

CSC2210: OBJECT ORIENTED PROGRAMMING 2

Spring 2023-2024

Section: [L]

Group No: 24

Project Report On

TravelEase – A Simplifed and Unified Ticket Booking System

Supervised By

Zahiduddin Ahmed

Submitted By:

Name	ID
1. MEHERAJ HASAN	22-49363-3
2. H. M. SAIFULLAHIL MAZID	20-44018-2
3. SHAMS PAHLOWAN SOAD	22-49342-3
4. FOISAL AHMED CHOWDHURY	22-49332-3

CO2: Display and verify the mean of a real-life Project using the concepts of C# Graphical User Interface based environment with database integration to depict a desktop-based application.

Assessment Criteria	Not Attended/ Incorrect (0)	Inadequate (1-2)	Average (3)	Good (4)	Excellent (5)
Evaluation Criteria	**Evaluation Definition**				**Total = 15**
Requirement fulfillment	Properly demonstrate a real-life scenario-based project with proper functional requirement identification for the Object-Oriented Programming project development activities.				5
Validation	Ensuring the ability of students' proper demonstration on validation forms in their system in terms of dealing with the data.				5

Verification	Identifying if the students can verify the system data along with proper functional requirements in terms of data flow.	5

CO3: Prepare and Explain a real life desktop based application synthesizing several component of C# along with development tools to adhere the given requirements.

Assessment Criteria	Not Attended/ Incorrect (0)	Inadequate (1-2)	Average (3)	Good (4)	Excellent (5)
Evaluation Criteria	**Evaluation Definition**				
Organization of the application	Fails to identify any suitable real time application or requirements for project development activities related to OOP.	Limited understanding about the project scopes and scenarios or identification of functional requirements.	Lacks depth or relevance to OOP project development activities and may contain inaccuracies. Real-life scenarios are mentioned, but the discussion lacks depth or clarity.	Consider and integrate the idea of several core aspects of the project along with relevance to real-life scenarios. Demonstrating a solid understanding of the application presentation.	Generalize and exhibits an exceptional understanding of project preparation according to a to real-life scenarios. Also contains proper and insightful identification of the system which is comprehensive and precise.
Representation and Integration of Database	Fails to identify and present any understanding or implementation of database. Also failed to integrate the data with the project itself.	Limited understanding of the database concepts or their proper way of using in a real time project. While some attempt may be made to implement but it is incomplete or poorly executed, leading to inadequate design.	Lacks depth or relevance to database integration with the application. Shows a basic understanding but some aspects may be missing or incorrectly implemented, resulting in partial or inconsistency. May lack proper normalization.	Integrate the database with the forms properly and implements it with proper validation which is mostly accurate and comprehensive, ensuring the proper handling of data input and verification along with general normalization.	Exhibits an exceptional understanding and implementation of database ensuring attention to detail, and robust data manipulation procedures and contributing to the overall clarity.
Graphical User Interface	Fails to present or prepare GUI based application interfaces. There is no	Limited understanding of graphical user interfaces. Lack of	Shows a basic understanding of creating user interfaces. Most of them	Effectively identifies and meet the consider the simplicity. Design related works are	Exhibits an exceptional work design following a high standard of simple and elegant work.

	evidence of creating or integrating such things according to their usefulness.	design knowledge. Very poor attempt to make such things which are currently obsolete or can't be identified as coherent.	are interconnecte d but maybe some of them lack it. However, most of it can be described as user friendly.	mostly accurate and taken proper attention to ensuring a user-friendly coherent system.	Several controls and mechanism has been organized in a preferred way according to the coherent usage.

Chapter 1: Introduction

TravelEase is a comprehensive online ticketing system designed to streamline the ticket purchasing process for various modes of transportation. Whether it's a bus ride, train journey, or flight, TravelEase ensures a hassle-free experience for both travelers and service providers.

User Functionality:

- **Registration:** Users can create accounts securely by providing personal information such as first name, last name, gender, email, phone number, and date of birth. National ID (NID) upload is available.
- **Ticket Purchasing:** Registered users can browse available vehicles, select their desired mode of transportation, choose departure and arrival destinations, and select travel dates.
- **Ticket Management:** Customers can view their purchase history, including details of tickets bought, canceled, or refunded. They can also print tickets directly from the system.
- **Seat Selection:** The system provides seat availability information, allowing users to choose their preferred seats during the booking process.
- **Payment Integration:** TravelEase integrates with popular payment gateways, offering users a variety of payment options such as credit/debit cards, net banking, and digital wallets.

Employee Functionality:

- **Company Management:** Employees associated with transportation companies can manage company-specific operations within the system.
- **Fleet Management:** Modular admins oversee the addition and deletion of vehicles, as well as changes to vehicle routes and schedules.
- **Ticket Management:** Employees can cancel tickets, adjust seating arrangements, and set rules for ticket refunds.

System Admin Functionality:

- **System Oversight:** The system admin oversees the entire platform, managing user accounts, approving company registrations, and enforcing system policies.
- **Collaboration:** System admins work closely with modular admins to ensure smooth operation and resolve any issues that may arise.
- **Policy Enforcement:** System admins enforce policies related to user behavior, company conduct, and system integrity.

TravelEase revolutionizes the ticketing experience by offering a user-friendly interface, comprehensive ticket management features, and secure payment processing. It caters to the needs of both travelers and transportation companies, making travel arrangements seamless and efficient for all parties involved.

Chapter 2: User Stories

Use Case Name	Actor	User Story
Registration	Passenger, Modular Admin, Admin	Registers to the system with all the credentials.
Login	Passenger, Modular Admin, Admin	Signs in/ logs in to the system to access the features of the system.
Schedule Checking	Passenger, Modular Admin, Admin	Can check the transport schedules
Purchase History	Passenger, Modular Admin, Admin	Checks the purchase history according to the actor.
Select Transport Vehicle	Passenger	Selects transport for journey
Purchase ticket	Passenger	Purchases the desired ticket for journey
Payment	Passenger	Pays for the purchased ticket
Download ticket	Passenger	Downloads/prints the purchased ticket
Cancel purchase	Passenger	Cancels the booked/purchased ticket
Refund request	Passenger	Requests for a refund upon ticket cancellation
Account Modification	Passenger	Modifies the passenger credentials
Increase/Decrease Trip number	Modular Admin	Changes the number of trips that a vehicle may have.
Cancel trip	Modular Admin	Cancels any upcoming trips
Change route	Modular Admin	Alters the routes for a vehicle or group of vehicles
Changes seat count	Modular Admin	Changes the number of seat available in a vehicle.
Sets rule for refund	Modular Admin	Refund rules are set by the Modular Admins
Request to add company	Modular Admin	Requests to add transport service company to the system.
Approve company	Admin	Approves companies to provide transport services.
Block user	Admin	Blocks any registered passenger under any rule breaking
Block company	Admin	Blocks any registered company under rule breaking
Restrict vehicle	Admin	Emergency or forcefully restricts any vehicle

Chapter 3: Database Queries

- **Database Creation**

```
CREATE DATABSE TravelEase
```

- **BusSeatTB**

```
CREATE TABLE [dbo].[BusSeatTB](
	[vehicleID] [int] NOT NULL,
	[seatNumber] [varchar](15) NOT NULL,
	[seatStatus] [bit] NULL,
	[userID] [varchar](25) NULL
) ON [PRIMARY]

ALTER TABLE [dbo].[BusSeatTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])

ALTER TABLE [dbo].[BusSeatTB]  WITH CHECK ADD FOREIGN KEY([vehicleID])
REFERENCES [dbo].[VehicleTB] ([vehicleID])
```

- **CompanyTB**

```
CREATE TABLE [dbo].[CompanyTB](
	[companyID] [int] NOT NULL,
	[compName] [varchar](25) NOT NULL,
	[bdRegID] [varchar](25) NOT NULL,
	[companyStatus] [bit] NULL,
PRIMARY KEY CLUSTERED
(
	[companyID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
```

- **CompanyVehicleTB**

```
CREATE TABLE [dbo].[CompanyVehicleTB](
	[serial] [int] IDENTITY(1,1) NOT NULL,
	[companyID] [int] NOT NULL,
	[vehicleID] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[serial] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[CompanyVehicleTB]  WITH CHECK ADD FOREIGN KEY([companyID])
REFERENCES [dbo].[CompanyTB] ([companyID])

ALTER TABLE [dbo].[CompanyVehicleTB]  WITH CHECK ADD FOREIGN KEY([vehicleID])
REFERENCES [dbo].[VehicleTB] ([vehicleID])
```

- **DestinationTB**

```
CREATE TABLE [dbo].[DestinationTB](
	[destinationID] [int] IDENTITY(1,1) NOT NULL,
	[desFrom] [varchar](15) NOT NULL,
	[desTo] [varchar](15) NOT NULL,
	[vehicleTypeID] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[destinationID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[DestinationTB]  WITH CHECK ADD FOREIGN KEY([vehicleTypeID])
REFERENCES [dbo].[VehicleTypeTB] ([vehicleTypeID])
```

- **LaunchSeatTB**

```
CREATE TABLE [dbo].[LaunchSeatTB](
	[vehicleID] [int] NOT NULL,
	[seatNumber] [varchar](15) NOT NULL,
	[seatStatus] [bit] NULL,
	[userID] [varchar](25) NULL
) ON [PRIMARY]

ALTER TABLE [dbo].[LaunchSeatTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])

ALTER TABLE [dbo].[LaunchSeatTB]  WITH CHECK ADD FOREIGN KEY([vehicleID])
REFERENCES [dbo].[VehicleTB] ([vehicleID])
```

- **LoginCredentialsTB**

```
CREATE TABLE [dbo].[LoginCredentialsTB](
	[userID] [varchar](25) NOT NULL,
	[userName] [varchar](50) NOT NULL,
	[userPassword] [varchar](255) NOT NULL,
PRIMARY KEY CLUSTERED
(
	[userID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY],
UNIQUE NONCLUSTERED
(
	[userName] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[LoginCredentialsTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])
```

- **MAdminCompanyTB**

```
CREATE TABLE [dbo].[MAdminCompanyTB](
	[serial] [int] IDENTITY(1,1) NOT NULL,
	[companyID] [int] NOT NULL,
	[MAdminNumber] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[serial] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[MAdminCompanyTB]  WITH CHECK ADD FOREIGN KEY([companyID])
REFERENCES [dbo].[CompanyTB] ([companyID])

ALTER TABLE [dbo].[MAdminCompanyTB]  WITH CHECK ADD FOREIGN KEY([MAdminNumber])
REFERENCES [dbo].[ModularAdminTB] ([MAdminNumber])
```

- **ModularAdminTB**

```
CREATE TABLE [dbo].[ModularAdminTB](
	[MAdminNumber] [int] IDENTITY(1,1) NOT NULL,
	[userID] [varchar](25) NULL,
PRIMARY KEY CLUSTERED
(
	[MAdminNumber] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[ModularAdminTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])
```

- **PassengerPaymentTb**

```
CREATE TABLE [dbo].[PassengerPaymentTb](
	[serial] [int] IDENTITY(1,1) NOT NULL,
	[transactionID] [int] NOT NULL,
	[passengerNumber] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[serial] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[PassengerPaymentTb]  WITH CHECK ADD FOREIGN KEY([passengerNumber])
REFERENCES [dbo].[PassengerTB] ([passengerNumber])

ALTER TABLE [dbo].[PassengerPaymentTb]  WITH CHECK ADD FOREIGN KEY([transactionID])
REFERENCES [dbo].[PaymentTB] ([transactionID])
```

- **PassengerTB**

```
CREATE TABLE [dbo].[PassengerTB](
	[passengerNumber] [int] IDENTITY(1,1) NOT NULL,
	[userID] [varchar](25) NULL,
PRIMARY KEY CLUSTERED
(
	[passengerNumber] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[PassengerTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])
```

- **PaymentTB**

```
CREATE TABLE [dbo].[PaymentTB](
	[transactionID] [int] NOT NULL,
	[amount] [float] NOT NULL,
	[cardNumber] [int] NOT NULL,
	[cardType] [varchar](10) NOT NULL,
	[payDate] [date] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[transactionID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
```

- **PlaneSeatTB**

```
CREATE TABLE [dbo].[PlaneSeatTB](
	[vehicleID] [int] NOT NULL,
	[seatNumber] [varchar](15) NOT NULL,
	[seatStatus] [bit] NULL,
	[userID] [varchar](25) NULL
) ON [PRIMARY]

ALTER TABLE [dbo].[PlaneSeatTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])

ALTER TABLE [dbo].[PlaneSeatTB]  WITH CHECK ADD FOREIGN KEY([vehicleID])
REFERENCES [dbo].[VehicleTB] ([vehicleID])
```

- **RefundRuleTB**

```
CREATE TABLE [dbo].[RefundRuleTB](
	[serial] [int] IDENTITY(1,1) NOT NULL,
	[refund_80_percent] [int] NULL,
	[refund_60_percent] [int] NULL,
	[refund_40_percent] [int] NULL,
	[no_refund] [int] NULL,
	[MAdminNumber] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[serial] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[RefundRuleTB]  WITH CHECK ADD  CONSTRAINT
[FK_RefundTB_MAdminNumber] FOREIGN KEY([MAdminNumber])
REFERENCES [dbo].[ModularAdminTB] ([MAdminNumber])

ALTER TABLE [dbo].[RefundRuleTB] CHECK CONSTRAINT [FK_RefundTB_MAdminNumber]
```

- **RefundTB**

```
CREATE TABLE [dbo].[RefundTB](
	[refundID] [int] IDENTITY(1,1) NOT NULL,
	[transactionID] [int] NOT NULL,
	[refundDate] [date] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[refundID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[RefundTB]  WITH CHECK ADD FOREIGN KEY([transactionID])
REFERENCES [dbo].[PaymentTB] ([transactionID])
```

- **SeatNumbersTB**

```
CREATE TABLE [dbo].[SeatNumbersTB](
	[ticketID] [int] NOT NULL,
	[seats] [varchar](20) NOT NULL,
	[vehicleID] [int] NOT NULL
) ON [PRIMARY]

ALTER TABLE [dbo].[SeatNumbersTB]  WITH CHECK ADD FOREIGN KEY([ticketID])
REFERENCES [dbo].[TicketTB] ([ticketID])

ALTER TABLE [dbo].[SeatNumbersTB]  WITH CHECK ADD FOREIGN KEY([vehicleID])
REFERENCES [dbo].[VehicleTB] ([vehicleID])
```

- **TicketDestinationTB**

```
CREATE TABLE [dbo].[TicketDestinationTB](
	[serial] [int] IDENTITY(1,1) NOT NULL,
	[ticketID] [int] NULL,
	[destinationID] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[serial] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[TicketDestinationTB]  WITH CHECK ADD FOREIGN KEY([destinationID])
REFERENCES [dbo].[DestinationTB] ([destinationID])

ALTER TABLE [dbo].[TicketDestinationTB]  WITH CHECK ADD FOREIGN KEY([ticketID])
REFERENCES [dbo].[TicketTB] ([ticketID])
```

- **TicketTB**

```
CREATE TABLE [dbo].[TicketTB](
	[ticketID] [int] NOT NULL,
	[buyDate] [date] NOT NULL,
	[fare] [float] NOT NULL,
	[seatAmount] [int] NOT NULL,
	[vehicleID] [int] NOT NULL,
	[userID] [varchar](25) NOT NULL,
	[journeyDate] [date] NULL,
PRIMARY KEY CLUSTERED
(
	[ticketID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[TicketTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])

ALTER TABLE [dbo].[TicketTB]  WITH CHECK ADD FOREIGN KEY([vehicleID])
REFERENCES [dbo].[VehicleTB] ([vehicleID])
```

- **TrainSeatTB**

```
CREATE TABLE [dbo].[TrainSeatTB](
	[vehicleID] [int] NOT NULL,
	[seatNumber] [varchar](15) NOT NULL,
	[seatStatus] [bit] NULL,
	[userID] [varchar](25) NULL
) ON [PRIMARY]

ALTER TABLE [dbo].[TrainSeatTB]  WITH CHECK ADD FOREIGN KEY([userID])
REFERENCES [dbo].[UserTB] ([userID])

ALTER TABLE [dbo].[TrainSeatTB]  WITH CHECK ADD FOREIGN KEY([vehicleID])
REFERENCES [dbo].[VehicleTB] ([vehicleID]
```

- **UserTB**

```
CREATE TABLE [dbo].[UserTB](
	[userID] [varchar](25) NOT NULL,
	[fName] [varchar](50) NOT NULL,
	[lName] [varchar](50) NOT NULL,
	[nid] [varchar](17) NOT NULL,
	[gender] [varchar](7) NOT NULL,
	[dob] [date] NOT NULL,
	[phone] [varchar](14) NOT NULL,
	[email] [varchar](25) NOT NULL,
	[residence] [varchar](100) NOT NULL,
	[userStatus] [smallint] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[userID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[UserTB]  WITH CHECK ADD CHECK  (([dob]<getdate()))
```

- **VehicleTB**

```
CREATE TABLE [dbo].[VehicleTB](
	[vehicleID] [int] IDENTITY(1,1) NOT NULL,
	[vehicleName] [varchar](50) NOT NULL,
	[vehicleTypeID] [int] NULL,
	[BDRegistrationNumber] [varchar](50) NOT NULL,
	[MAdminID] [int] NULL,
	[availableSeats] [int] NULL,
	[vehicleStatus] [int] NULL,
	[destinationID] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[vehicleID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]

ALTER TABLE [dbo].[VehicleTB]  WITH CHECK ADD FOREIGN KEY([destinationID])
REFERENCES [dbo].[DestinationTB] ([destinationID])

ALTER TABLE [dbo].[VehicleTB]  WITH CHECK ADD FOREIGN KEY([MAdminID])
REFERENCES [dbo].[ModularAdminTB] ([MAdminNumber])

ALTER TABLE [dbo].[VehicleTB]  WITH CHECK ADD FOREIGN KEY([vehicleTypeID])
REFERENCES [dbo].[VehicleTypeTB] ([vehicleTypeID])
```

- **VehicleTypeTB**

```
CREATE TABLE [dbo].[VehicleTypeTB](
	[vehicleTypeID] [int] IDENTITY(1,1) NOT NULL,
	[vehicleTypeName] [varchar](50) NOT NULL,
	[totalSeats] [int] NULL,
PRIMARY KEY CLUSTERED
(
	[vehicleTypeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY],
UNIQUE NONCLUSTERED
(
	[vehicleTypeName] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
```

- **Use Case Diagram**

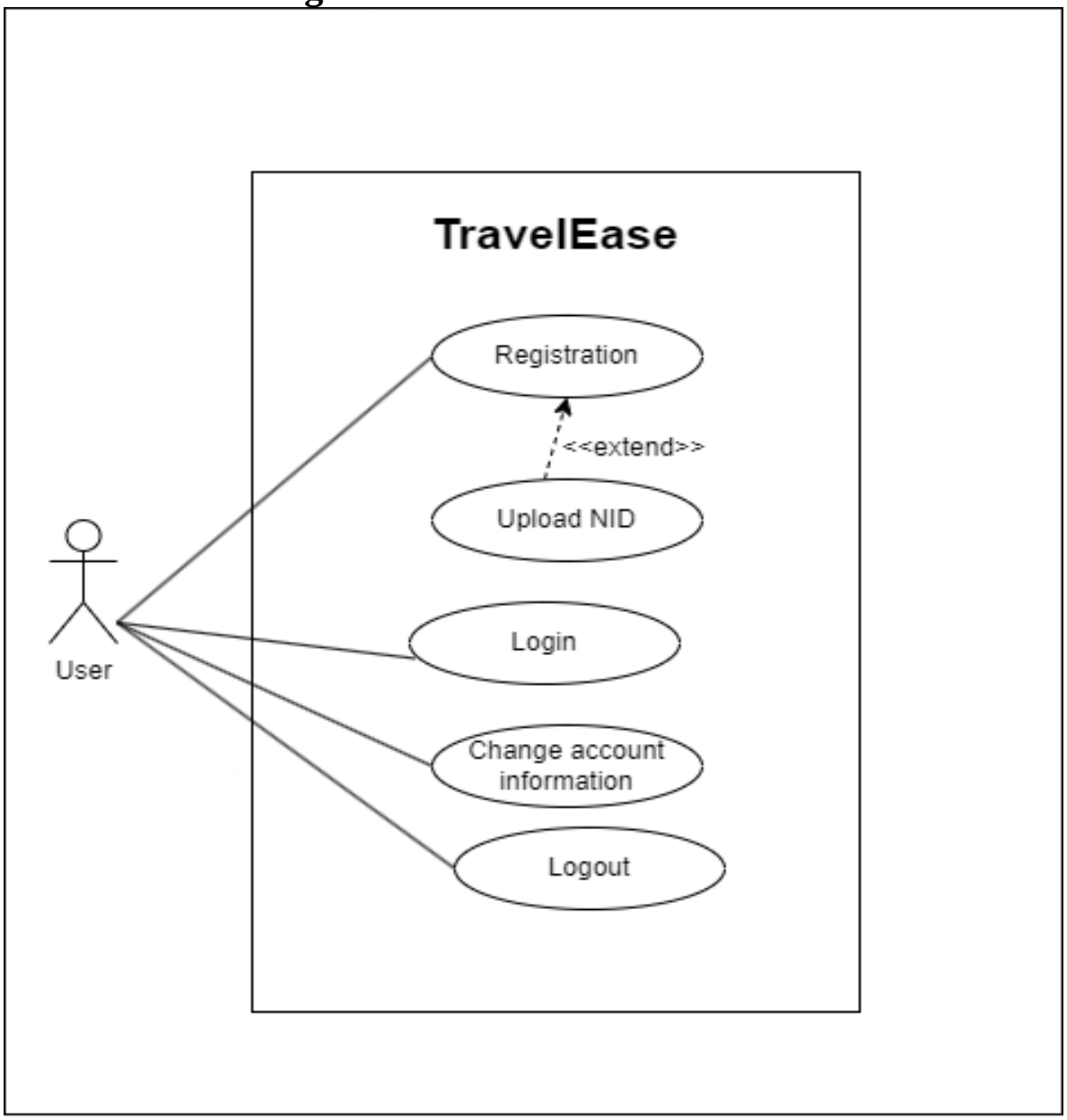


Figure: Use Case Diagram (User)

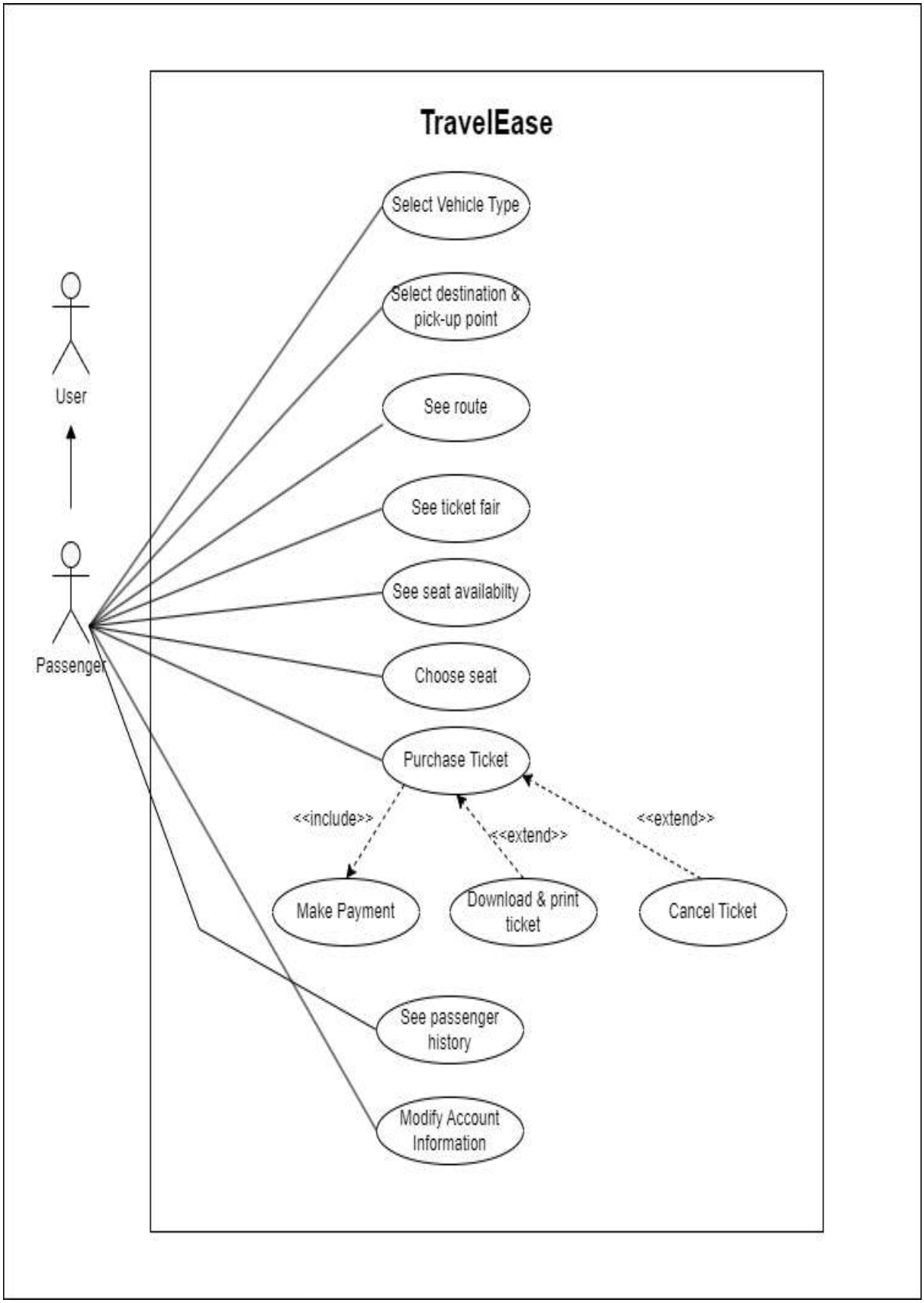


Figure: Use Case Diagram (Passenger)

Figure: Use Case Diagram (Modular Admin)

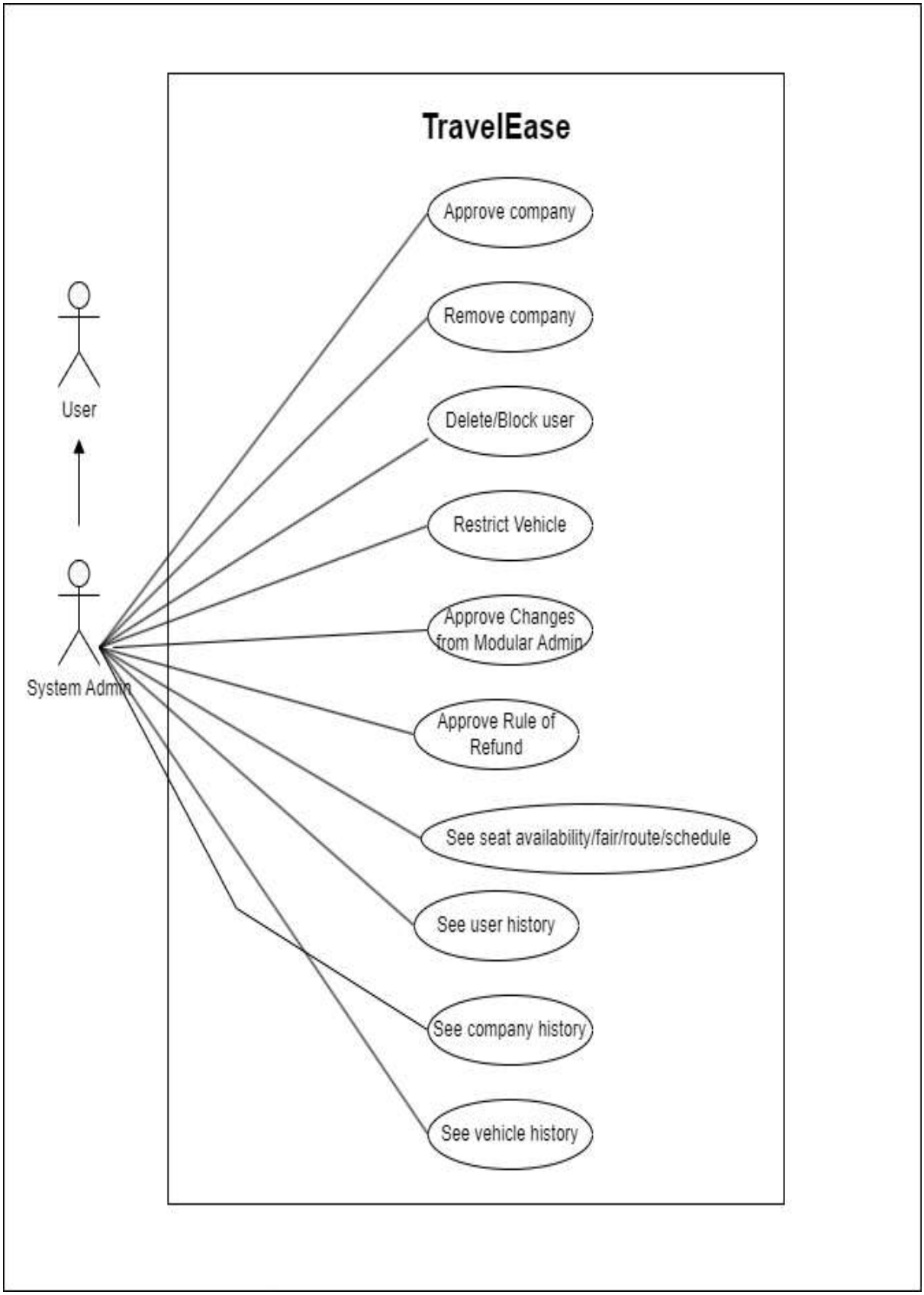
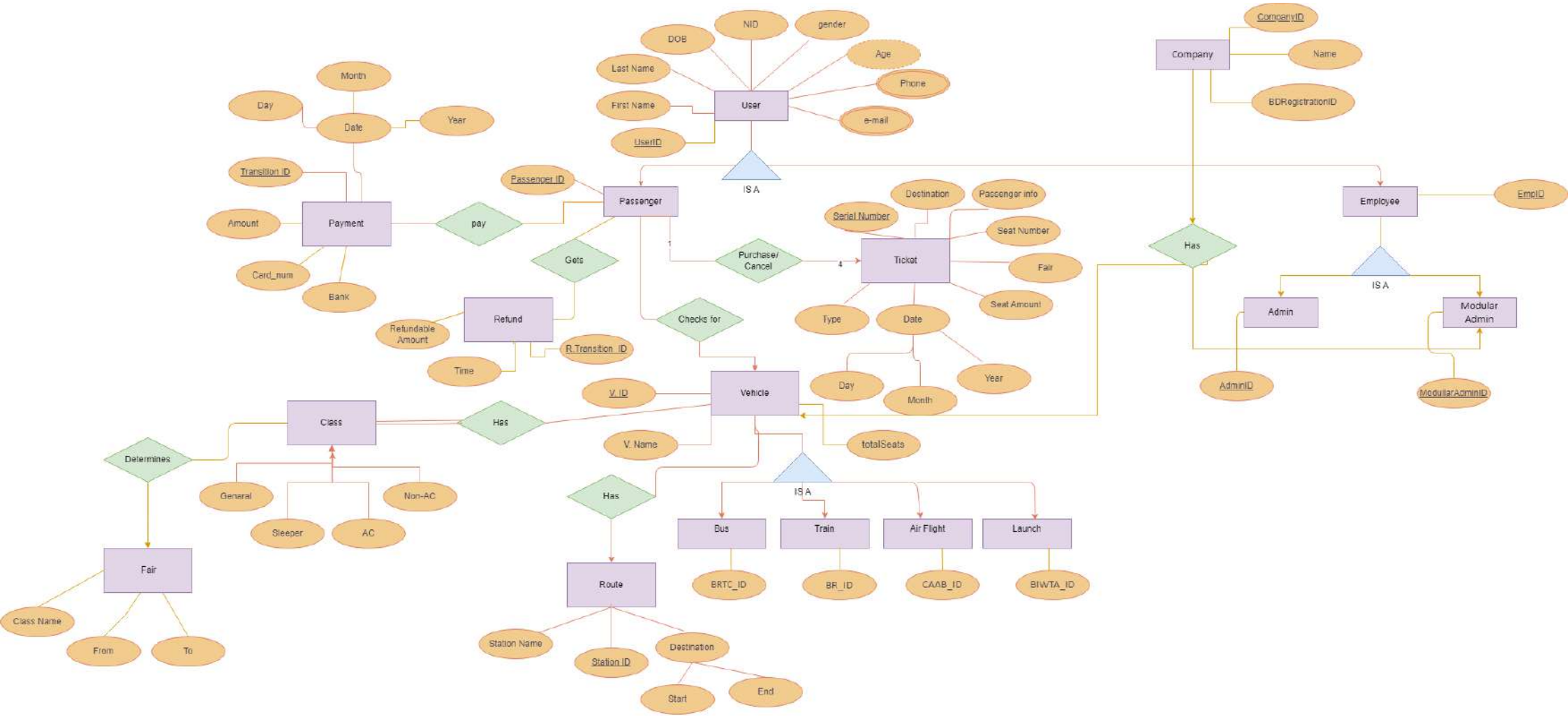
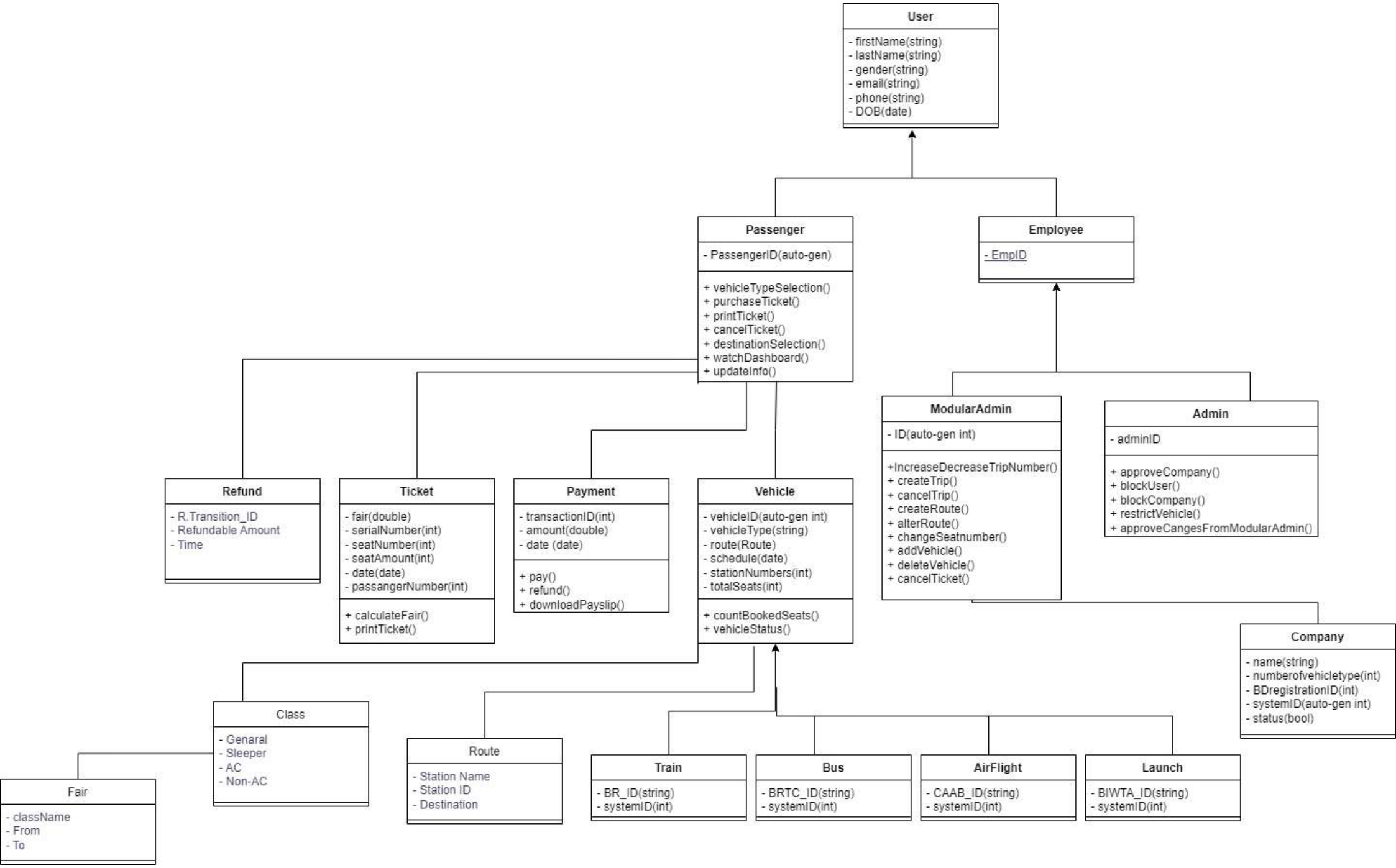


Figure: Use Case Diagram (System Admin)


Month
Day
Date
Year
Transition ID
Amount
Payment
Card_num
Bank
pay
Passenger ID
Passenger
Gets
Refund
Refundable Amount
Time
R Transition ID
Checks for
NID
gender
DOB
Age
Last Name
Phone
First Name
User
e-mail
UserID
ISA
1
Purchase/ Cancel
4
Destination
Passenger info
Serial Number
Seat Number
Ticket
Fair
Seat Amount
Type
Date
Day
Month
Year
Vehicle
V_ID
V. Name
totalSeats
ISA
Bus
Train
Air Flight
Launch
BRTC_ID
BR_ID
CAAB_ID
BIWTA_ID
Has
Route
Station Name
Station ID
Destination
Start
End
Class
Has
General
Sleeper
AC
Non-AC
Determines
Fair
Class Name
From
To
CompanyID
Company
Name
BDRegistrationID
Has
Employee
EmpID
ISA
Admin
Modular Admin
AdminID
ModularAdminID


User
- firstName(string)
- lastName(string)
- gender(string)
- email(string)
- phone(string)
- DOB(date)
Passenger
- PassengerID(auto-gen)
+ vehicleTypeSelection()
+ purchaseTicket()
+ printTicket()
+ cancelTicket()
+ destinationSelection()
+ watchDashboard()
+ updateInfo()
Employee
- EmpID
Refund
- R.Transition_ID
- Refundable Amount
- Time
Ticket
- fair(double)
- serialNumber(int)
- seatNumber(int)
- seatAmount(int)
- date(date)
- passangerNumber(int)
+ calculateFair()
+ printTicket()
Payment
- transactionID(int)
- amount(double)
- date (date)
+ pay()
+ refund()
+ downloadPayslip()
Vehicle
- vehicleID(auto-gen int)
- vehicleType(string)
- route(Route)
- schedule(date)
- stationNumbers(int)
- totalSeats(int)
+ countBookedSeats()
+ vehicleStatus()
ModularAdmin
- ID(auto-gen int)
+IncreaseDecreaseTripNumber()
+ createTrip()
+ cancelTrip()
+ createRoute()
+ alterRoute()
+ changeSeatnumber()
+ addVehicle()
+ deleteVehicle()
+ cancelTicket()
Admin
- adminID
+ approveCompany()
+ blockUser()
+ blockCompany()
+ restrictVehicle()
+ approveCangesFromModularAdmin()
Company
- name(string)
- numberofvehicletype(int)
- BDregistrationID(int)
- systemID(auto-gen int)
- status(bool)
Class
- Genaral
- Sleeper
- AC
- Non-AC
Fair
- className
- From
- To
Route
- Station Name
- Station ID
- Destination
Train
- BR_ID(string)
- systemID(int)
Bus
- BRTC_ID(string)
- systemID(int)
AirFlight
- CAAB_ID(string)
- systemID(int)
Launch
- BIWTA_ID(string)
- systemID(int)

- **Data Dictionary**

1. **User:**

Key	Name	Data Type	Length	Nullable
Primary	ID	Integer	10	No
	First name	varchar	10	No
	Last name	varchar	50	No
	gender	varchar	50	No
	email	varchar	50	No
	Age	integer	3	No
	Phone	Integer	11	No
	DOB	datetime	yyyy-mm-dd	No
	NID	integer	17	No

2. **Passenger:**

Key	Name	Data Type	Length	Nullable
Primary	Passenger_ID	INTEGER	10	No

3. **Employee:**

Key	Name	Data Type	Length	Nullable
Primary	Employee_ID	INTEGER	10	No

4. **Admin:**

Key	Name	Data Type	Length	Nullable
Primary	Admin_ID	INTEGER	10	No

5. **Modular Admin:**

Key	Name	Data Type	Length	Nullable
Primary	ModularAdmin_ID	INTEGER	10	No

6. **Company:**

Key	Name	Data Type	Length	Nullable
Primary	Company_ID	INTEGER	10	No
	BDregistrationID	INTEGER	10	No
	Name	Varchar	25	No

7. **Ticket:**

Key	Name	Data Type	Length	Nullable
Primary	Serial number	INTEGER	10	No
	Seat number	Integer	2	No
	Fair	Double	10	No
	Seat amount	Integer	5	No
	Date	Datetime	yyyy-mm-dd	No

8. **Vehicle:**

Key	Name	Data Type	Length	Nullable
Primary	Vehicle_ID	INTEGER	10	No
	Total seats	INTEGER	10	No
	Name	Varchar	25	No

9. **Refund:**

Key	Name	Data Type	Length	Nullable
Primary	RefundTransactionID	INTEGER	10	No
	RefundableAmount	INTEGER	10	No
	Time	Datetime		No

10. **Payment:**

Key	Name	Data Type	Length	Nullable
Primary	Transaction_ID	INTEGER	10	No
	Amount	INTEGER	10	No
	Card number	Integer	16	No
	Bank	Varchar	10	No
	Date	Datetime	yyyy-mm-dd	No

11. **Fair:**

Key	Name	Data Type	Length	Nullable
Primary	Class name	Varchar	50	No
	From	Varchar	10	No
	To	Varchar	25	No

12. **Route:**

Key	Name	Data Type	Length	Nullable
Primary	StationID	INTEGER	10	No
	StationName	Varchar	50	No
	Start	Varchar	25	No
	End	Varchar	25	No

13. **Bus:**

Key	Name	Data Type	Length	Nullable
Primary	BRTC_ID	INTEGER	10	No

14. **Train:**

Key	Name	Data Type	Length	Nullable
Primary	Train_ID	INTEGER	10	No

15. **AirFlight:**

Key	Name	Data Type	Length	Nullable
Primary	CAAB_ID	INTEGER	10	No

16. Launch:

Key	Name	Data Type	Length	Nullable
Primary	BITWTA_ID	INTEGER	10	No

Chapter 4: Screenshot of Forms**:**

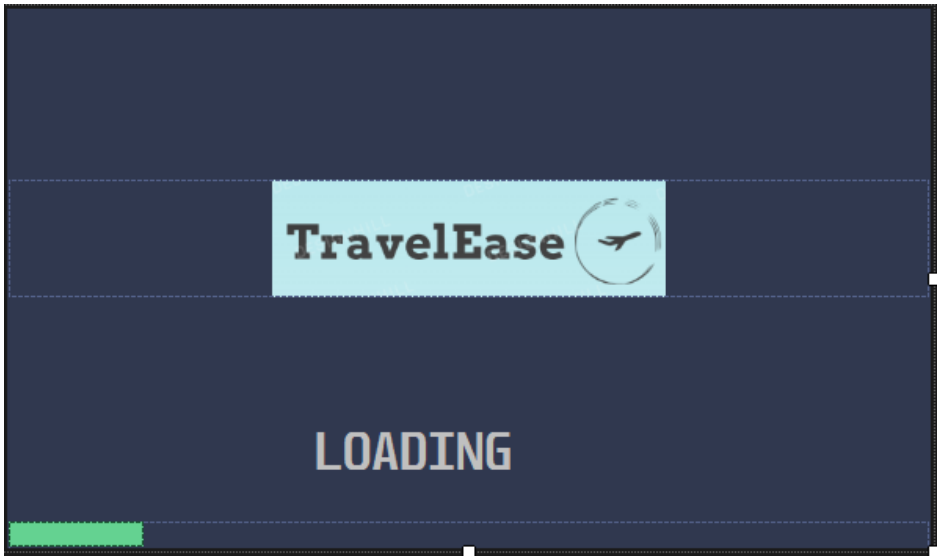


Figure: Loading Page

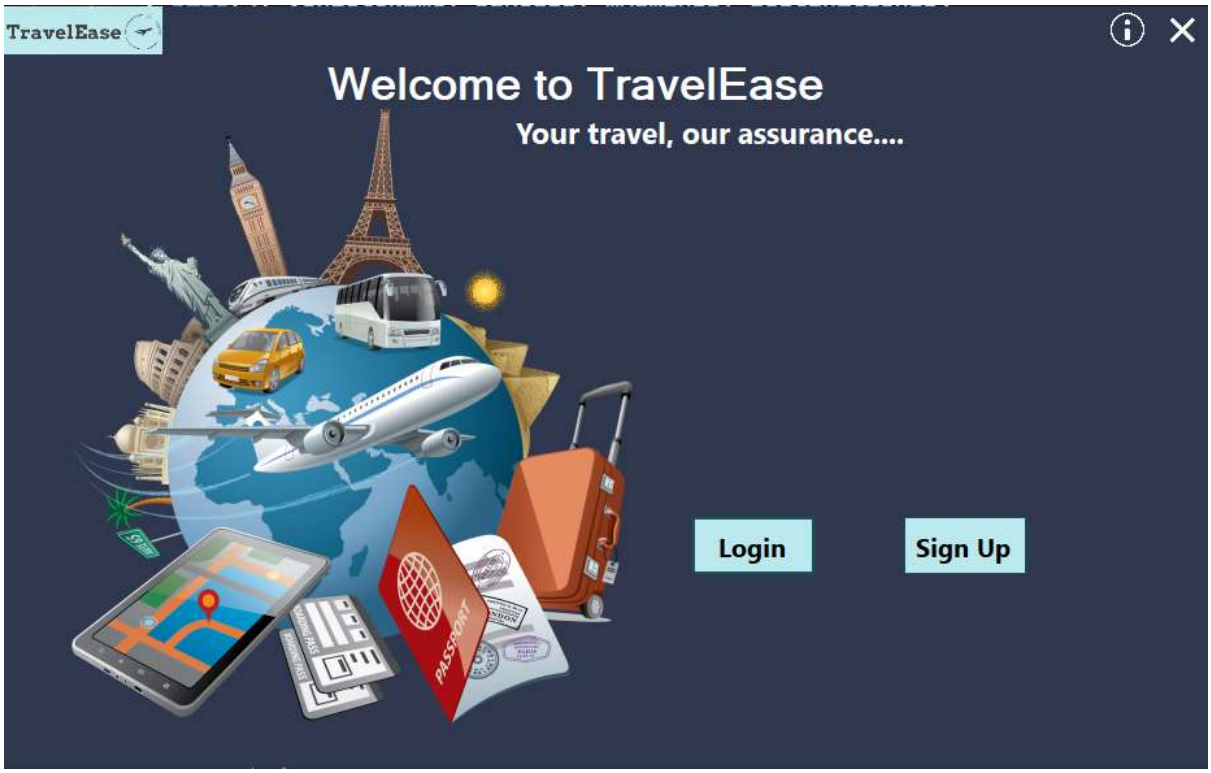


Figure: Homepage

Figure: About Page

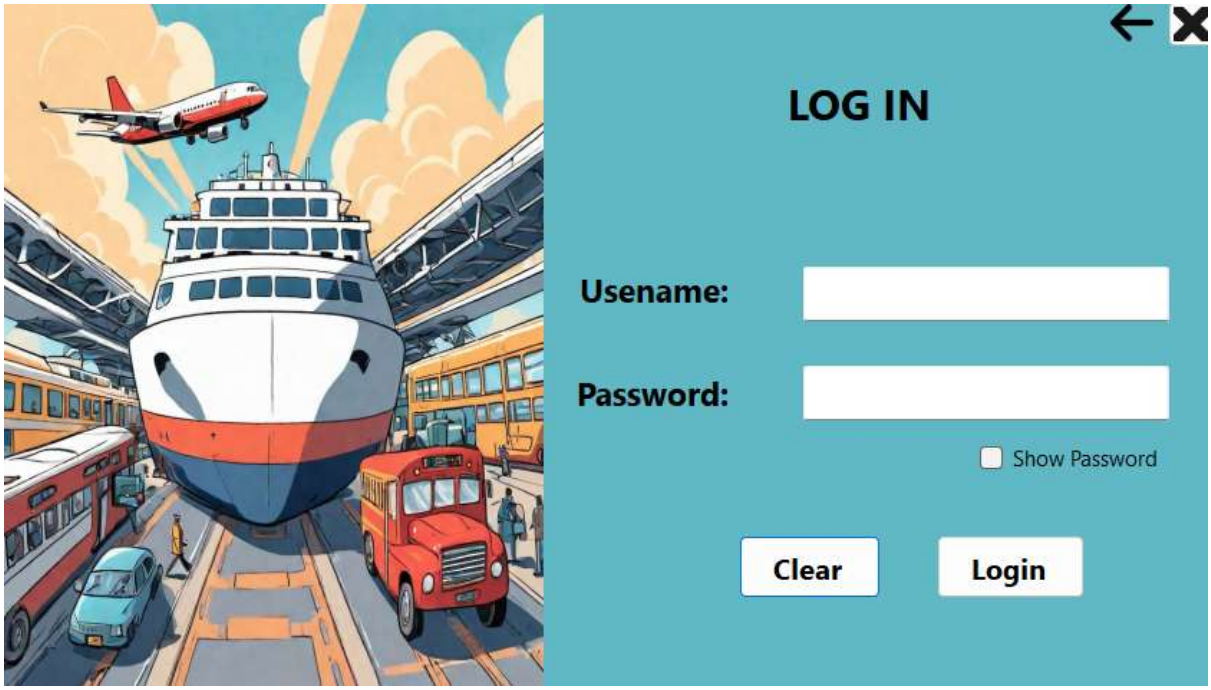


Figure: Login Page

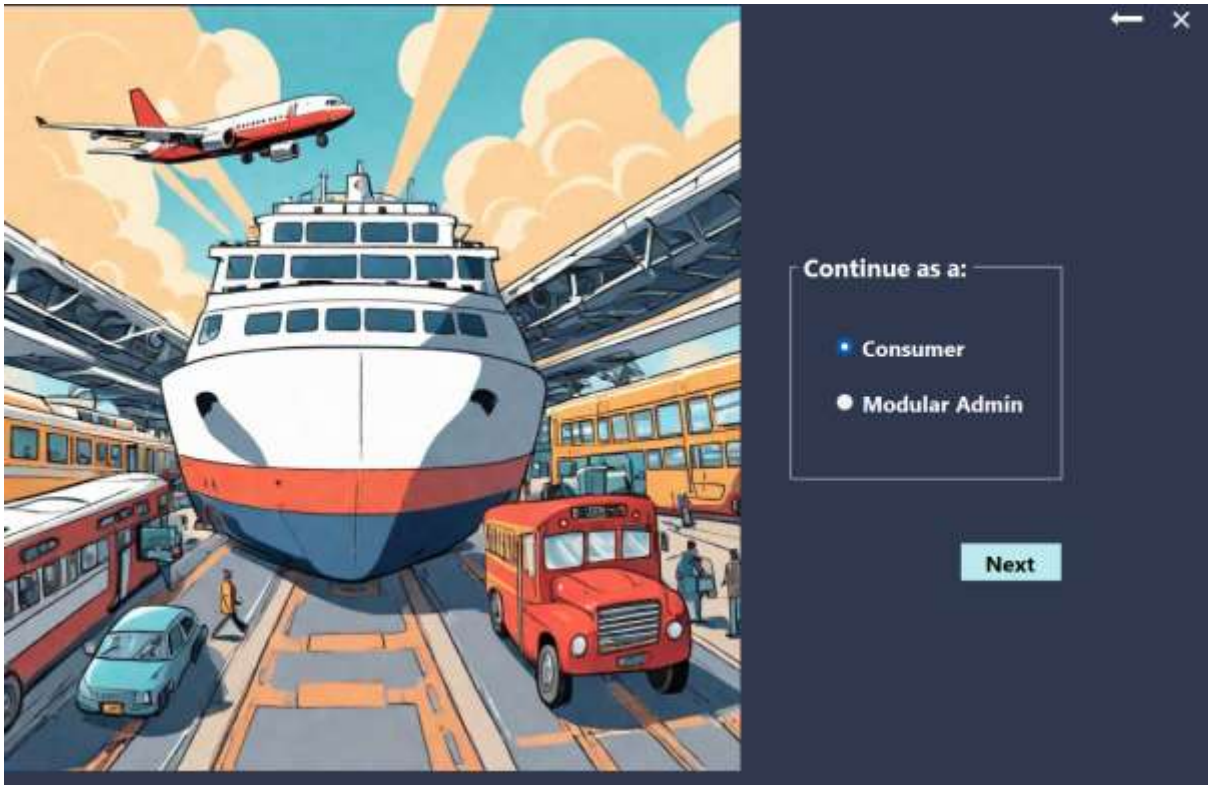


Figure: Registration Page

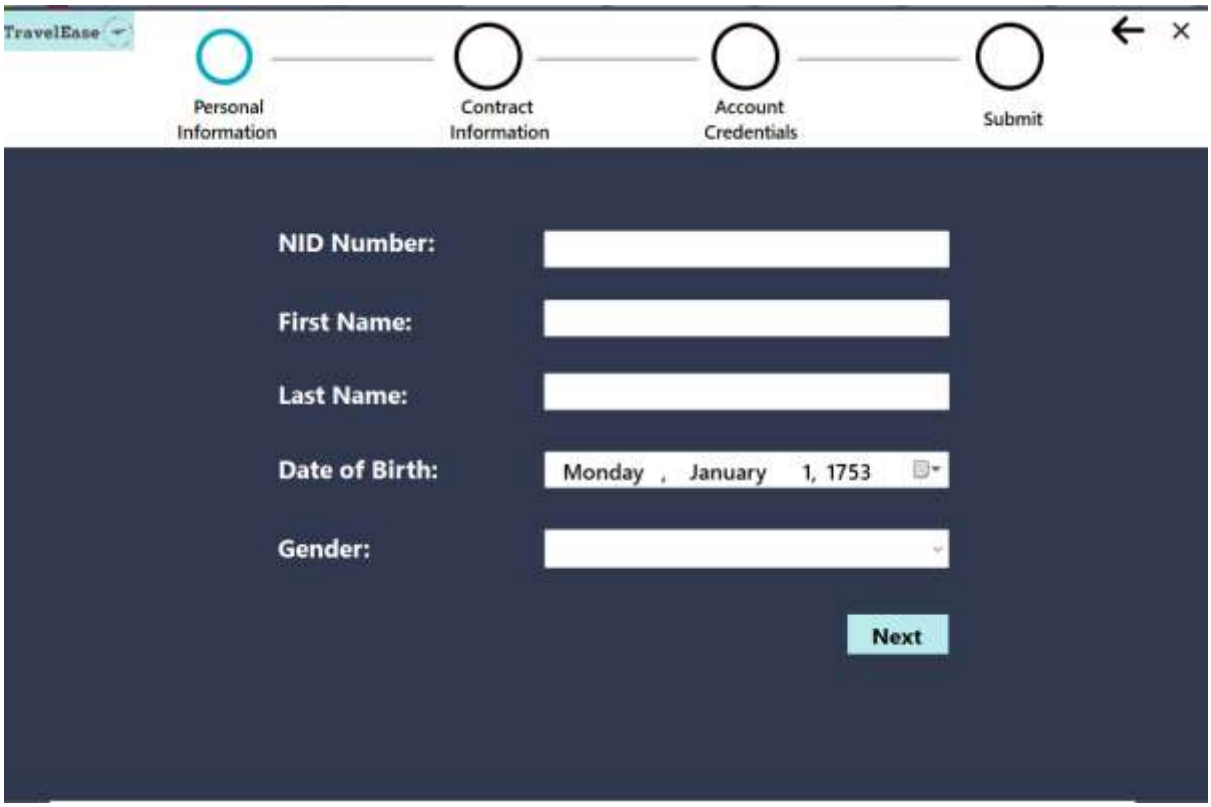


Figure: Consumer Registration Page (Panel 1)

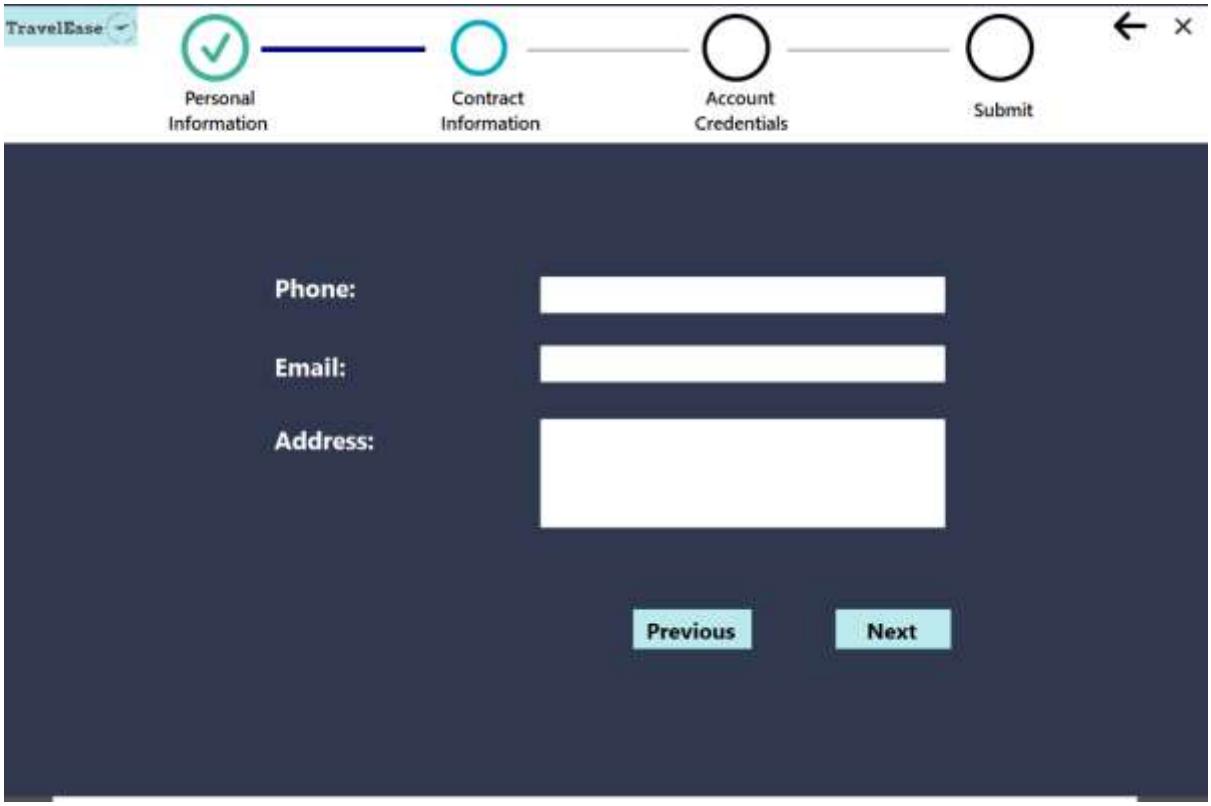


Figure: Consumer Registration Page (Panel 2)

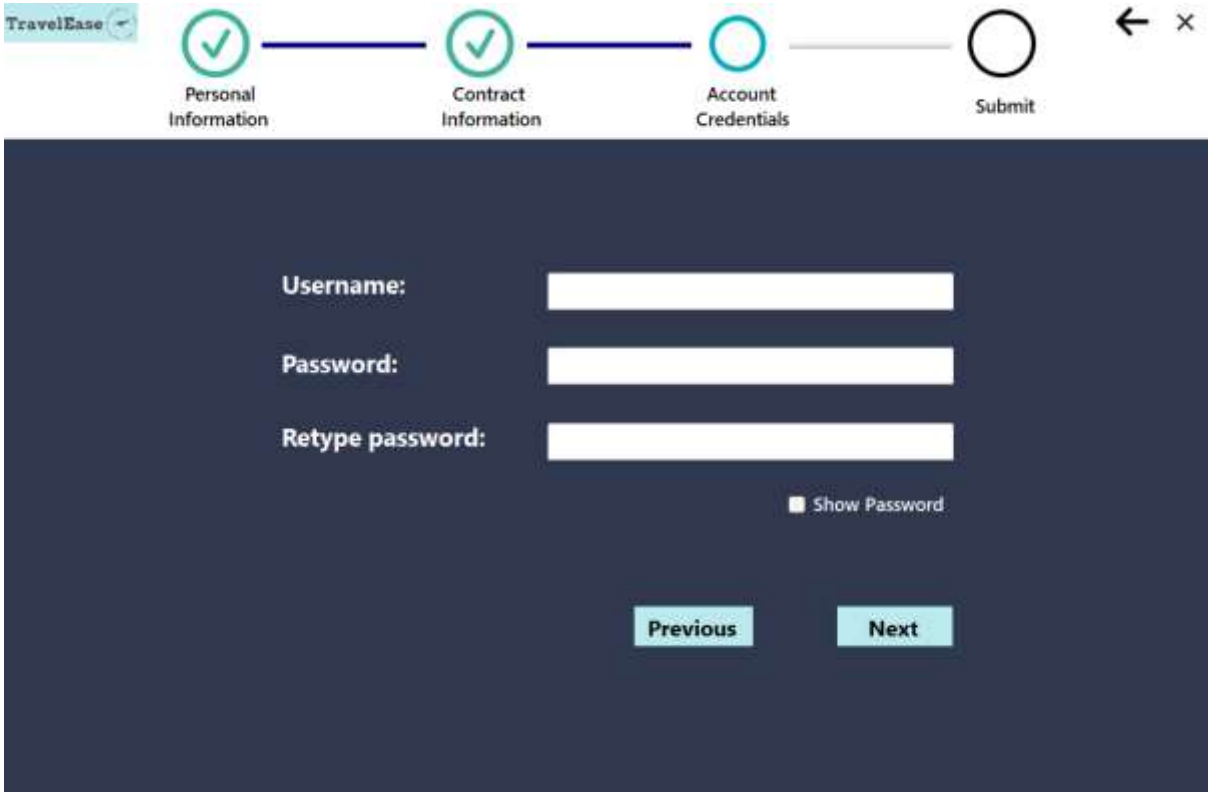


Figure: Consumer Registration Page (Panel 3)

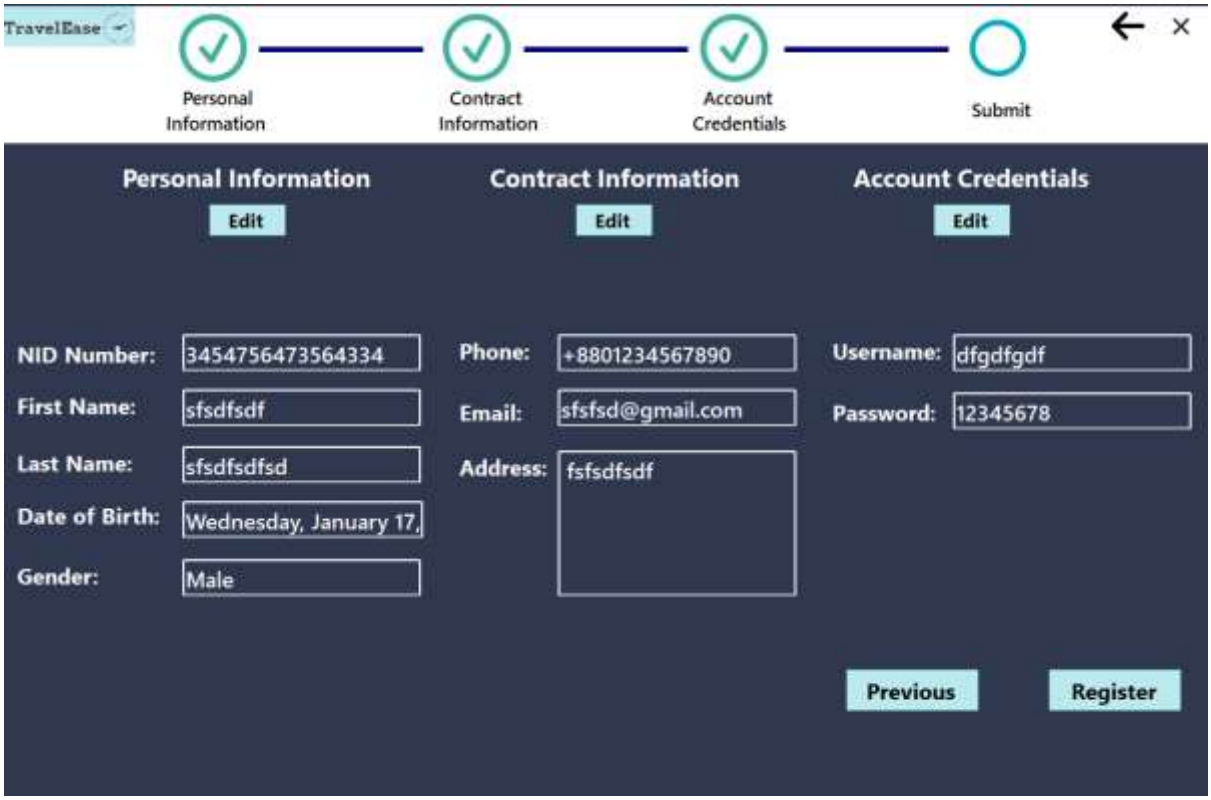


Figure: Consumer Registration Page (Panel 4)

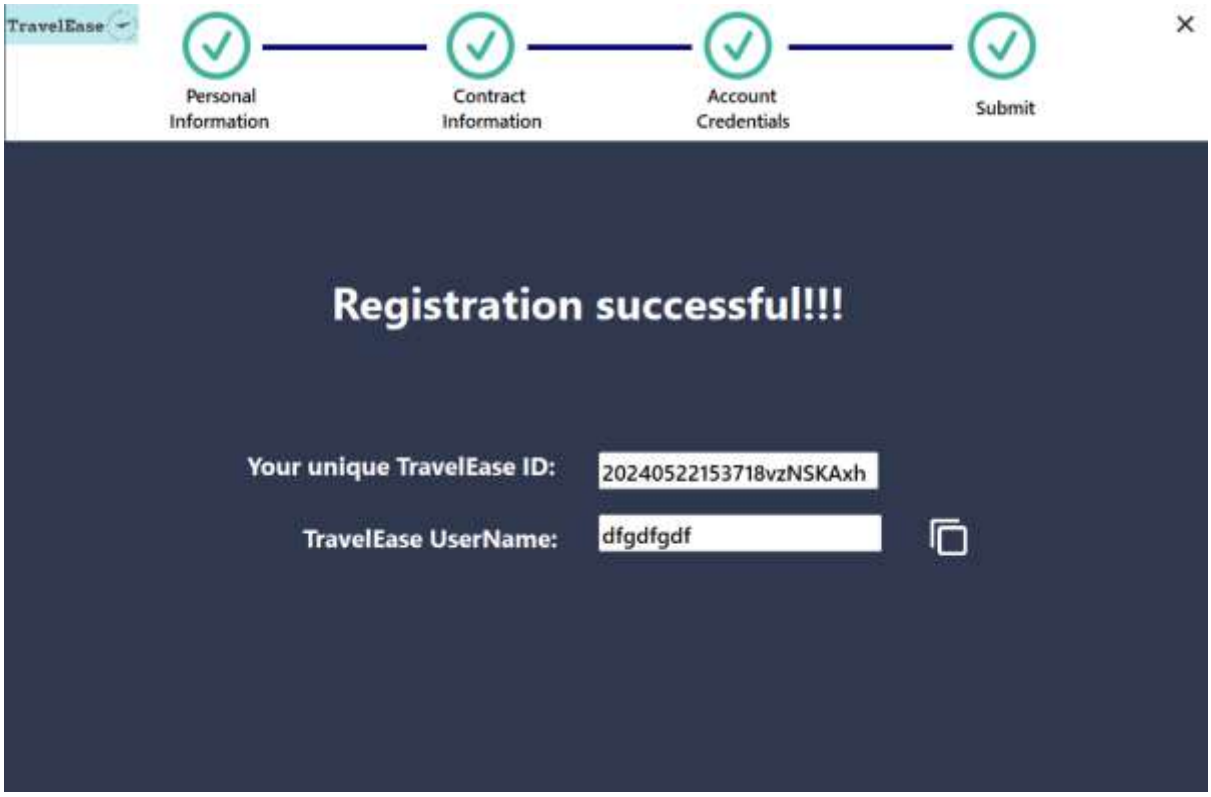


Figure: Consumer Registration Page (Panel 5)

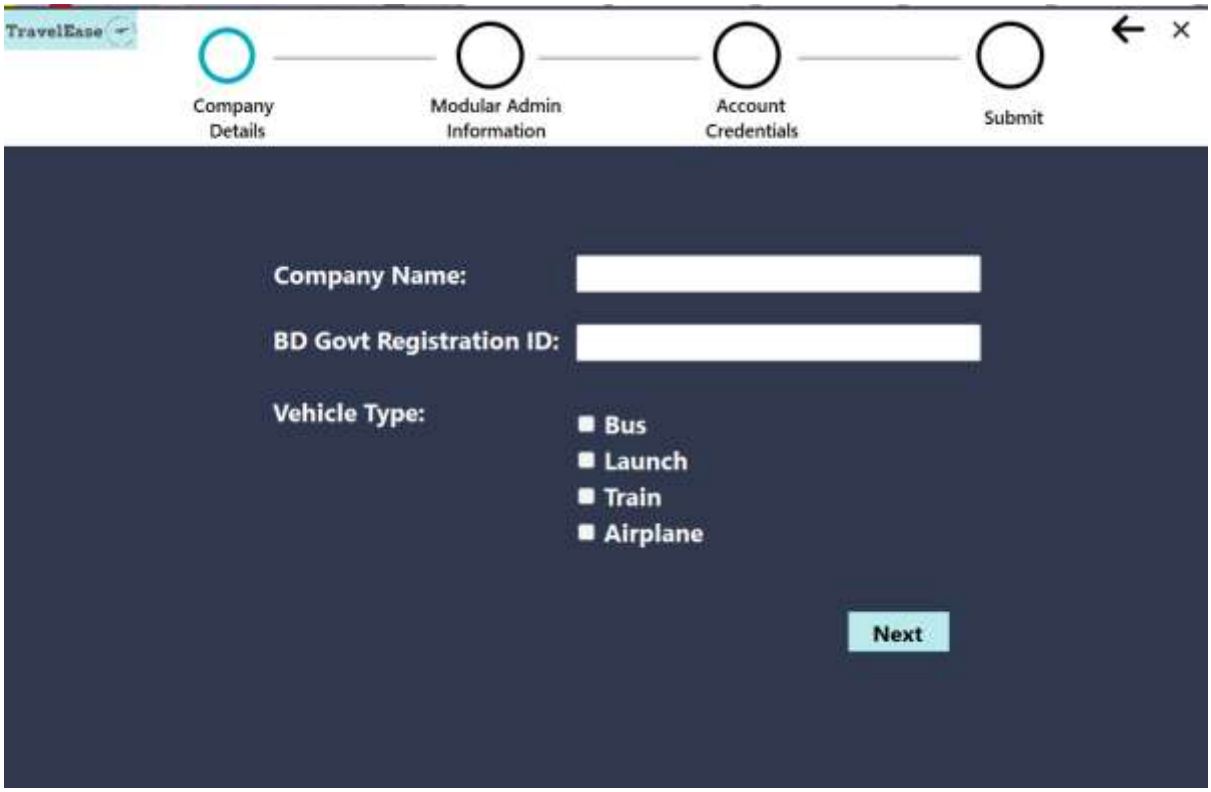


Figure: Modular Admin Registration Page (Panel 1)

Figure: Modular Admin Registration Page (Panel 2)

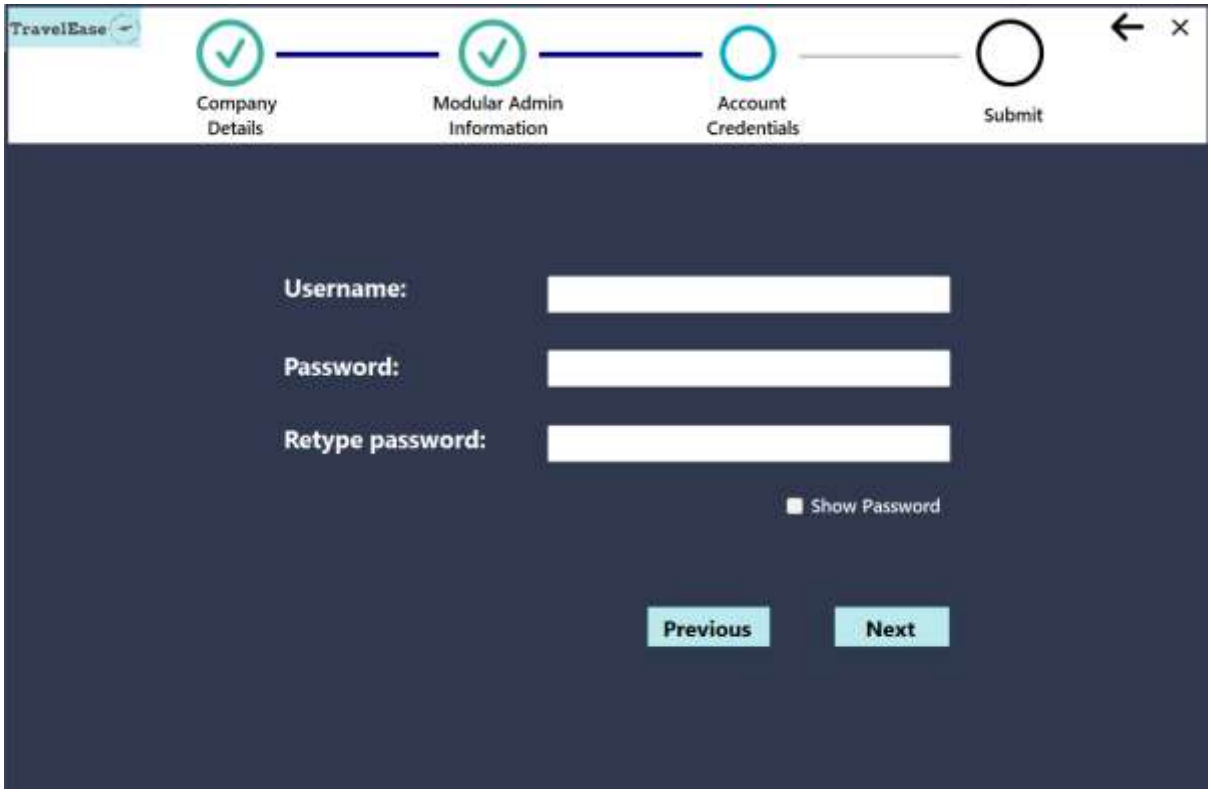


Figure: Modular Admin Registration Page (Panel 3)

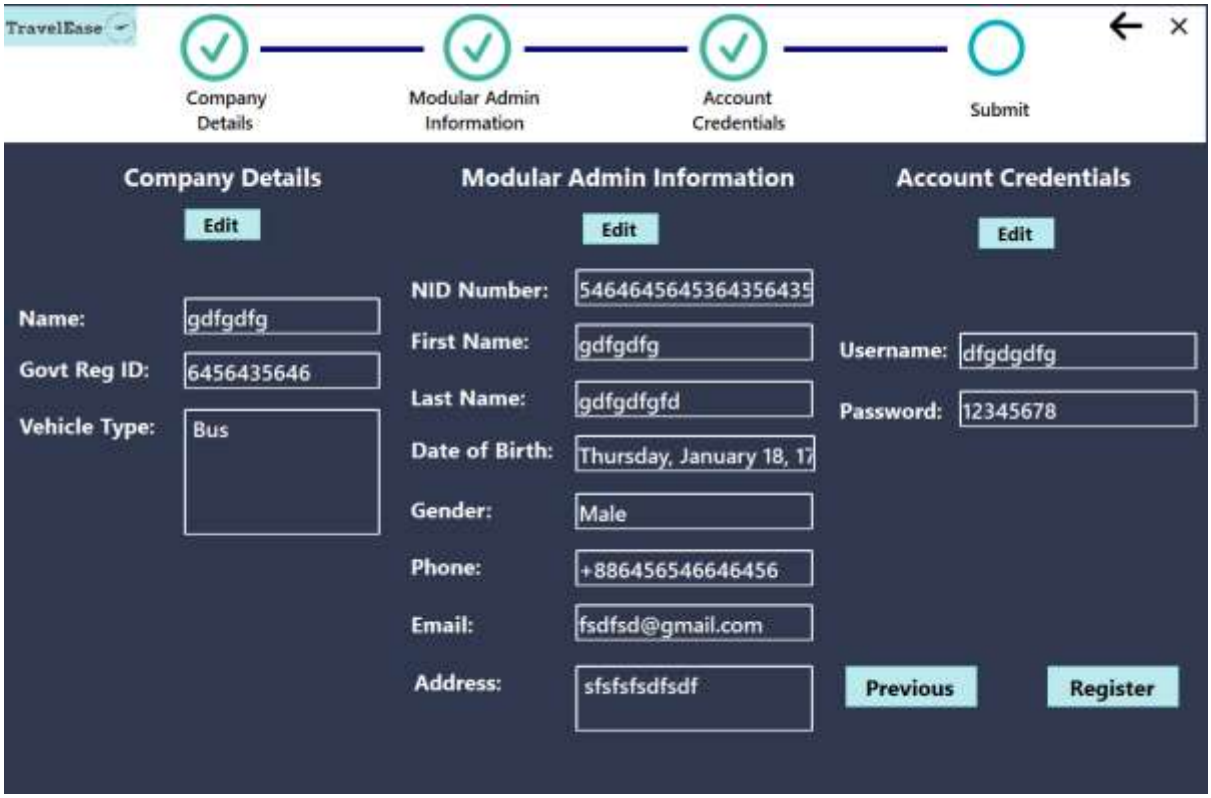


Figure: Modular Admin Registration Page (Panel 4)

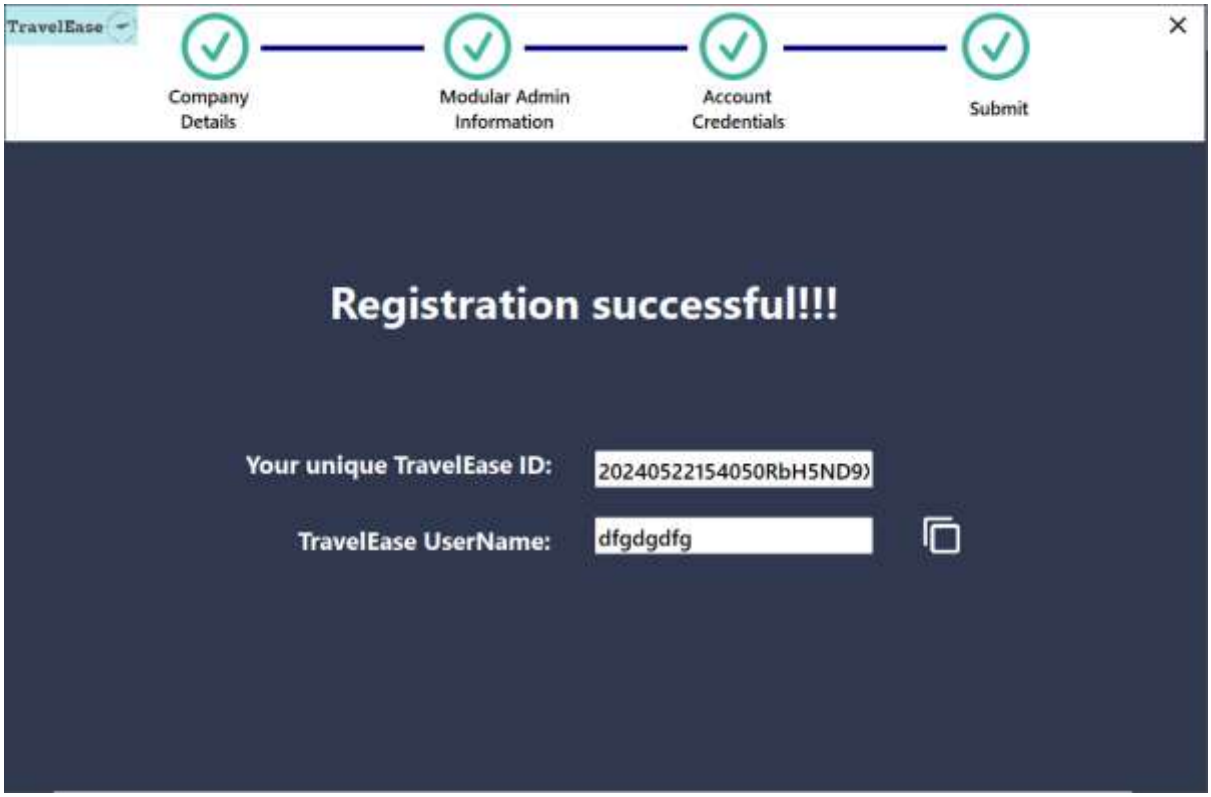


Figure: Modular Admin Registration Page (Panel 5)

Figure: Consumer Dashboard (Search Ticket)

Figure: Consumer Dashboard (Search Ticket)

Figure: Consumer Dashboard (Search Ticket)

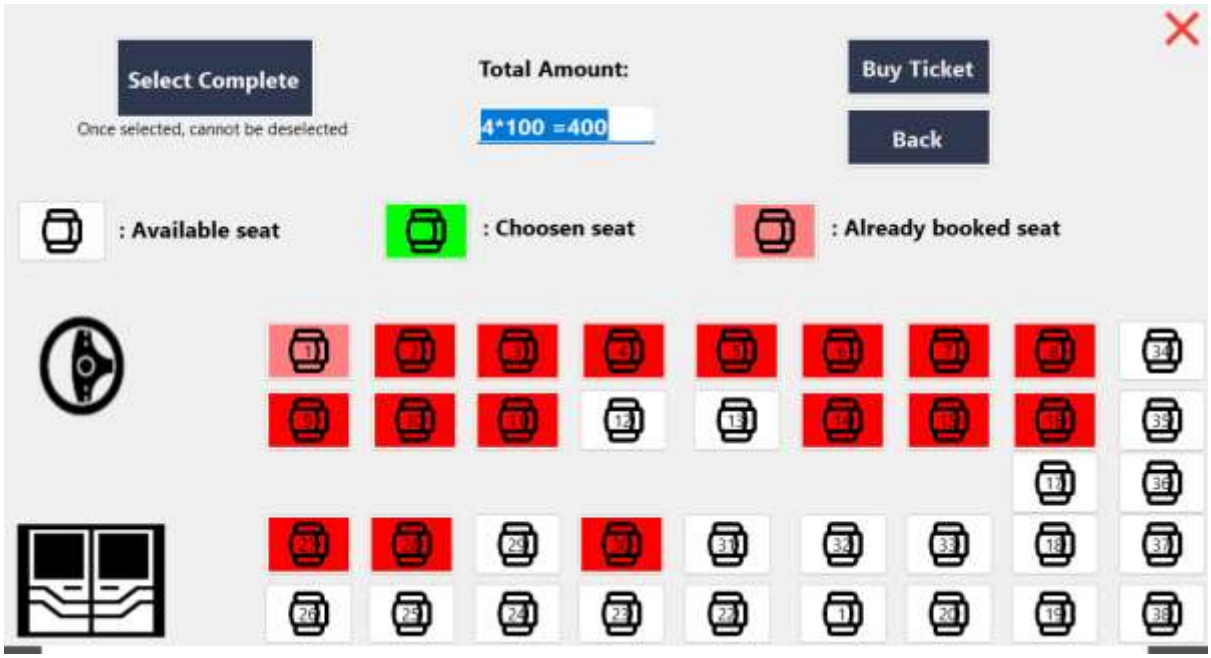


Figure: Consumer Dashboard (Book Ticket)

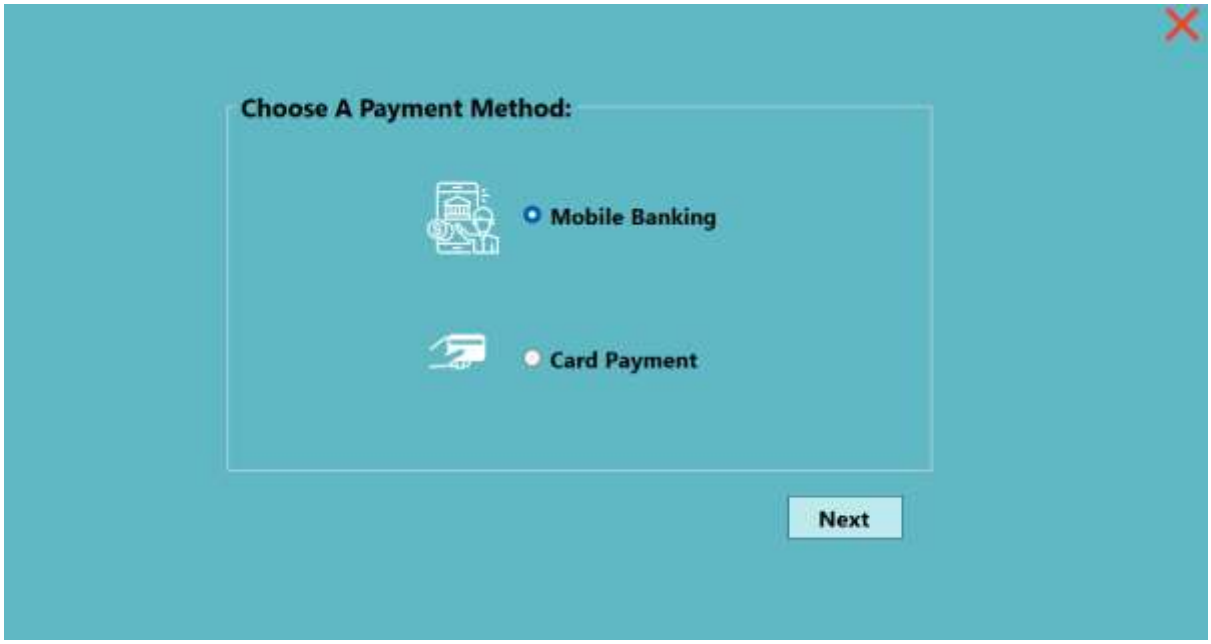


Figure: Consumer Dashboard (Payment Options)

Figure: Consumer Dashboard (Mobile Banking)

Figure: Consumer Dashboard (Card)

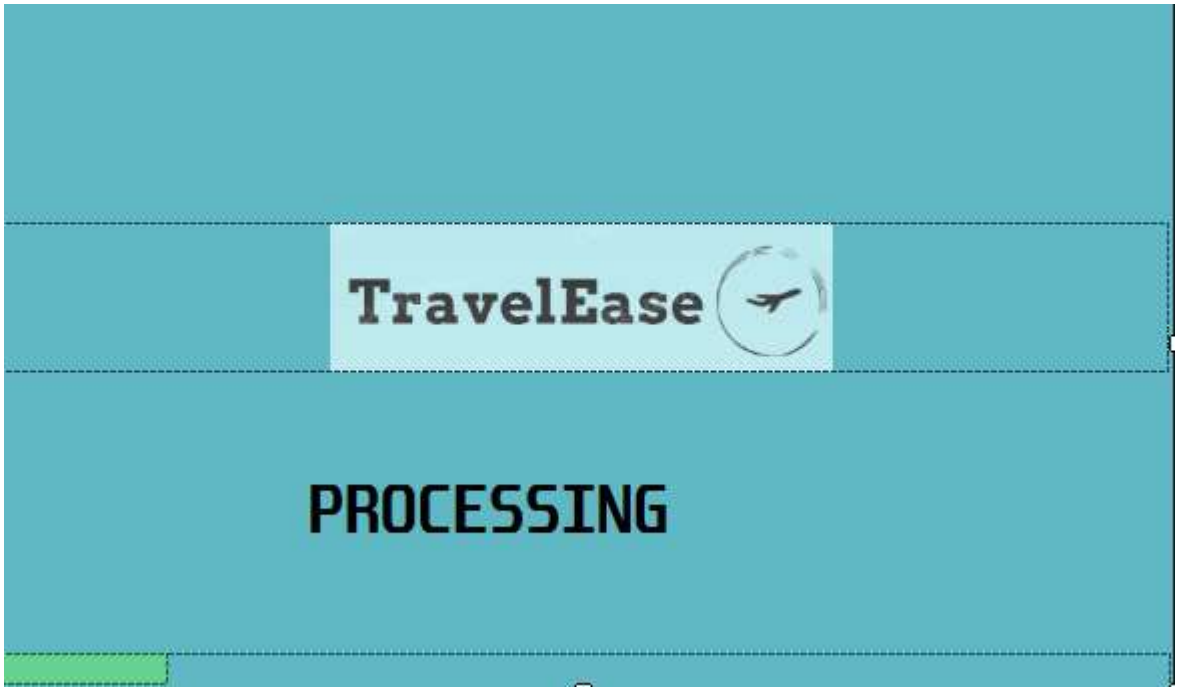


Figure: Consumer Dashboard (Payment Processing)

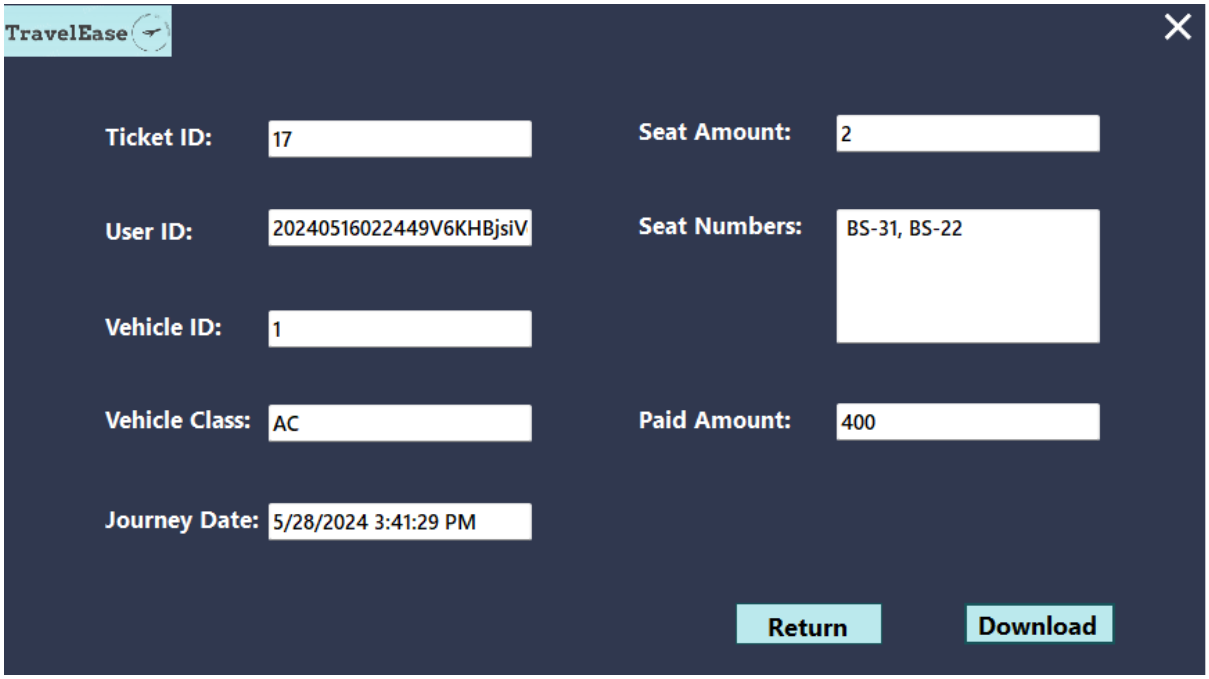


Figure: Consumer Dashboard (Download Ticket)

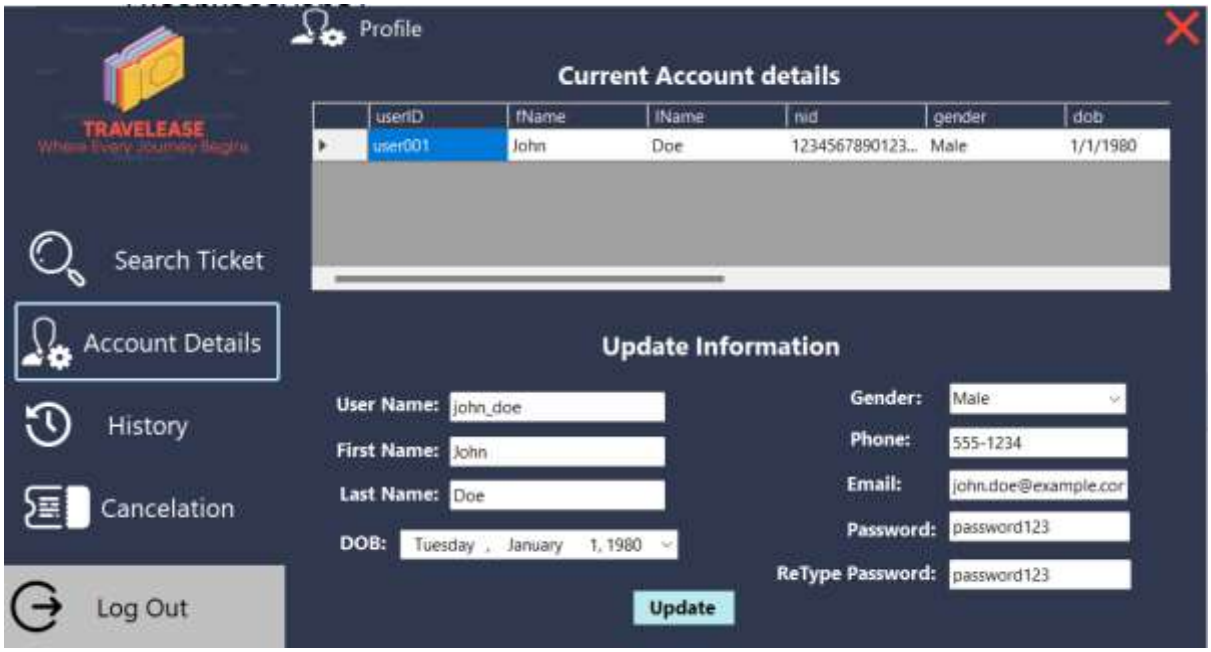


Figure: Consumer Dashboard (Account Details and Information Update)

Figure: Consumer Dashboard (Purchase History)

Figure: Consumer Dashboard (Ticket Cancelation)

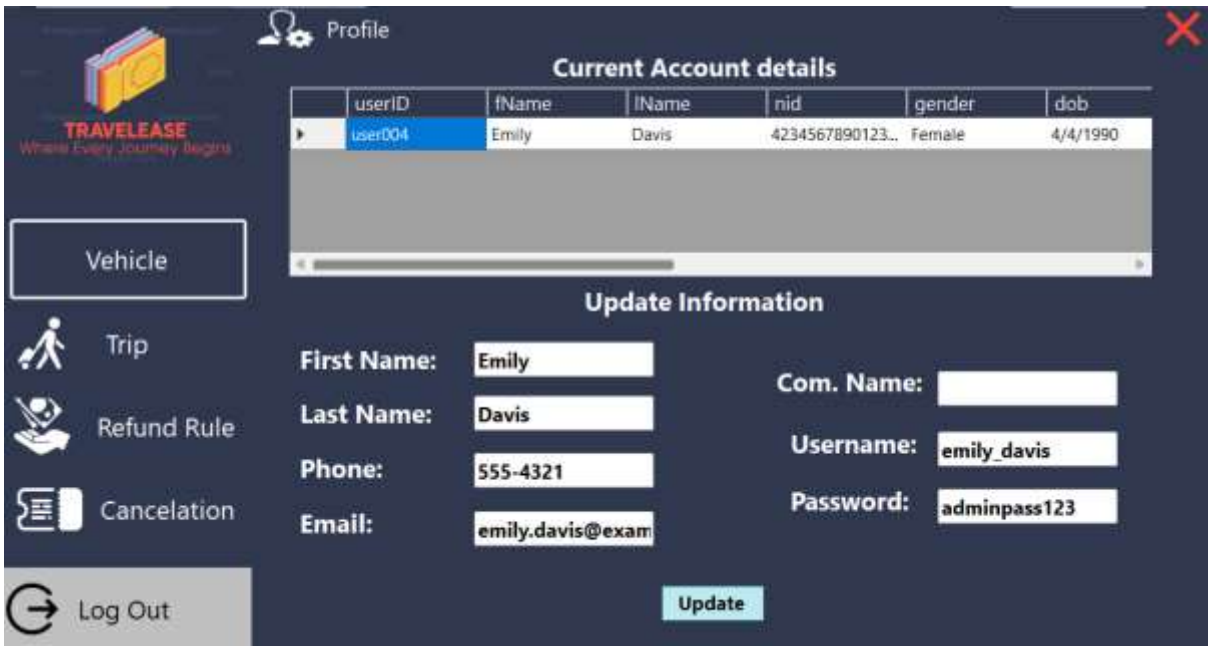


Figure: Modular Admin Dashboard (Account Details and Information Update)

Figure: Modular Admin Dashboard (Add Vehicle)

Figure: Modular Admin Dashboard (Modify Trip)

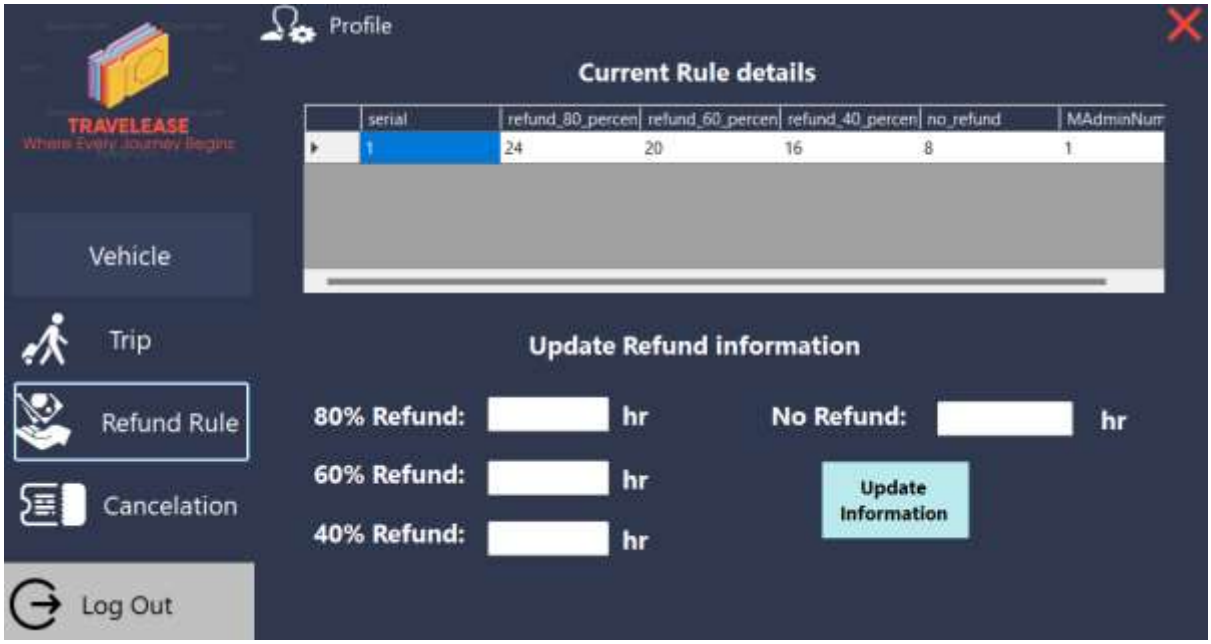


Figure: Modular Admin Dashboard (Set Refund Rule)

Figure: Modular Admin Dashboard (Cancel Tickets)

Chapter 5: Conclusion

TravelEase is an innovative solution that transforms the ticketing process for transportation services, providing a seamless and efficient platform for users and service providers alike. By integrating user registration, secure payment gateways, and comprehensive ticket management functionalities, TravelEase ensures a smooth and hassle-free experience for travelers. The platform's robust features, such as seat selection and detailed ticket history, enhance user convenience, while the administrative tools empower transportation companies and system admins to manage operations effectively.

Employees benefit from streamlined company and fleet management capabilities, ensuring operational efficiency and optimal service delivery. The collaborative approach between system admins and modular admins ensures that the platform runs smoothly, addressing any issues promptly and maintaining high standards of service integrity.

TravelEase stands out by catering to the diverse needs of the travel industry, offering a user-friendly interface, secure transactions, and an all-encompassing management system. This comprehensive approach not only simplifies the travel booking process but also strengthens the operational backbone of transportation services, setting a new standard in the online ticketing industry. Through TravelEase, the complexities of travel arrangements are minimized, ensuring that both travelers and service providers enjoy a seamless and enjoyable experience.