

National Institute of Co-operative Development

Polgolla

Diploma in Information Technology – 2019/2020

**Ticket Management System**

Final Project

Student Name : - W.G.M.M.Ajmal

ID Number :-DIT/FT/2019/14/363

Submission Date : -2020/12/14

# **DECLARATION**

I declare that the project entitled is an outcome of my own effort under the guidance of Mr. Dimuthu Kumara. The project is submitted to the National Institute of Co-operative development for the partial fulfilment of the Diploma in Information Technology course.

----------------------------------------

Signature of Course Coordinator

----------------------------------------

Date

# **ACKNOWLEDGEMENT**

My most profound gratitude goes to my, Course Coordinator – Sectional Head of IT Mrs. I. Namali Nanayakara who gave me the opportunity to do this project. I am extending my sincere and heartfelt thanks to our esteemed guide Mr. Kumara Dimuthu, who provided the overall support and encouragement and for showing us the right way. I am extending my sincere thanks to our respected Computer Instructor of the IT Division at National Institute of Co-operative Development Mr. Chanaka Dissanayake, for allowing us to use the facilities available. Finally, I would like to thank my parents and friends for the support and encouragement they have given us during the course of my work. Words cannot fully express my appreciation to all individual who have contributed to this work.

# **ABSTRACT**

Ticket Management System by Manual way is tedious process, since it involves work load and time consumption. In this system, we can easily manage the all ticket system operations.

The main feature of the Developed Ticket Management System is to allocate for the passengers and manage employees. Identification of the drawbacks of the existing system leads to the development computerized ticket management system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing ticket management system. Less human error, Strength and strain of manual labor can be reduced, High security, Data Redundancy can be avoided to some extent, Data consistency, Easy to handle, Easy to update the data, Easy record keeping Backup data can be easily generated.

This project is carried out using Visual Studio 2015 Enterprises as Front-End and MySQL sever database as Back-End. This Ticket Management System has Admin Login, Guest Login. This ticket management system is running through these dual login systems. Admin can manage all the operations in the system and a guest can log into the system and can reserve their bookings for their travel. This system will ease the work for passengers who willing to book tickets for their travel.

**Table of Contents**

[**DECLARATION** 2](#_Toc58769998)

[**ACKNOWLEDGEMENT** 3](#_Toc58769999)

[**ABSTRACT** 4](#_Toc58770000)

[**LIST OF FIGURES** 8](#_Toc58770001)

[**LIST OF TABLES** 11](#_Toc58770002)

[**CHAPTER 01** 13](#_Toc58770003)

[**INTRODUCTION** 13](#_Toc58770004)

[**Introduction** 14](#_Toc58770005)

[**1.1Background** 14](#_Toc58770006)

[**1.2 Purpose** 14](#_Toc58770007)

[**1.3 Objectives and Scope** 15](#_Toc58770008)

[**1.4 Project Schedule** 16](#_Toc58770009)

[**1.5 Approach** 17](#_Toc58770010)

[**CHAPTER 02** 18](#_Toc58770011)

[**SOFTWARE REQUIREMENT** 18](#_Toc58770012)

[**SPECIFICATION** 18](#_Toc58770013)

[**Software Requirement Specification** 19](#_Toc58770014)

[**2.1 User Requirement Specification** 19](#_Toc58770015)

[**2.2 System Requirement Specification** 19](#_Toc58770016)

[**2.2.1 Functional System Requirements** 19](#_Toc58770017)

[**2.2.3 Non-Functional Requirement Specification** 20](#_Toc58770018)

[**2.3 Hardware Requirements.** 20](#_Toc58770019)

[**2.4 Software Requirements.** 20](#_Toc58770020)

[**2.5 Software Tool Used** 21](#_Toc58770021)

[**CHAPTER 03** 22](#_Toc58770022)

[**SYSTEM ANALYSIS** 22](#_Toc58770023)

[**3.System Analysis** 23](#_Toc58770024)

[**3.1 Data Collection Methods** 23](#_Toc58770025)

[**3.2 Existing System** 23](#_Toc58770026)

[**3.3 Proposed System** 23](#_Toc58770027)

[**3.4 Literature Review** 24](#_Toc58770028)

[**CHAPTER 04** 26](#_Toc58770029)

[**SYSTEM DESIGN** 26](#_Toc58770030)

[**4.System Design** 27](#_Toc58770031)

[**4.1 System Architecture** 27](#_Toc58770032)

[**4.2 ER-Diagram** 28](#_Toc58770033)

[**4.3Use Case Diagram** 30](#_Toc58770034)

[**CHAPTER 05** 31](#_Toc58770035)

[**SYSTEM IMPLIMENTATION** 31](#_Toc58770036)

[**5. SYSTEM IMPLIMENTATION** 32](#_Toc58770037)

[**5.1 Interfaces** 32](#_Toc58770038)

[**5.1.1. Loading Page** 32](#_Toc58770039)

[**5.1.2. Admin Login** 32](#_Toc58770040)

[**5.1.3. Admin & User Home Page** 33](#_Toc58770041)

[**5.1.4. Customer Registration** 34](#_Toc58770042)

[**5.1.5. Flight** 34](#_Toc58770043)

[**5.1.6. Flight Ticket (Customer)** 35](#_Toc58770044)

[**5.1.7. Loading Page for Hotel View** 35](#_Toc58770045)

[**5.1.8. View of Hotels** 36](#_Toc58770046)

[**5.1.9. Transport Options** 36](#_Toc58770047)

[**5.1.10. Issue Salary** 37](#_Toc58770048)

[**5.1.11. Report of Flights** 37](#_Toc58770049)

[**5.1.12. Report of Hotel** 38](#_Toc58770050)

[**5.1.13. Report of Customer** 38](#_Toc58770051)

[**5.1.14. Report of Transport** 39](#_Toc58770052)

[**5.1.15. Print Preview of Employee Salary Receipt** 39](#_Toc58770053)

[**5.1.16. Manage Users** 40](#_Toc58770054)

[**5.1.17. Manage Employee** 40](#_Toc58770055)

[**5.1.20. Manage Transport** 41](#_Toc58770056)

[**5.1.21. Manage Flight** 42](#_Toc58770057)

[**5.1.22. Email view for hotel booking** 42](#_Toc58770058)

[**5.1.23. Email view for hiring vehicles** 43](#_Toc58770059)

[**5.1.24. Email message registered customers** 43](#_Toc58770060)

[**5.2 Coding (Selected)** 44](#_Toc58770061)

[**5.2.1. Coding for Customer Module** 44](#_Toc58770062)

[**5.2.2. Coding for Dashboard** 44](#_Toc58770063)

[**5.2.3. Coding for Login** 45](#_Toc58770064)

[**5.2.4. Coding for Message Box** 46](#_Toc58770065)

[**5.2.5. Coding for Login** 46](#_Toc58770066)

[**5.2.5. Coding for Display Meals (Selected)** 47](#_Toc58770067)

[**5.2.6. Coding for Flight Bookings (Selected)** 48](#_Toc58770068)

[**5.2.7. Coding for Insert Records** 49](#_Toc58770069)

[**5.2.8. Coding for Update Records** 49](#_Toc58770070)

[**5.2.9. Coding for Delete Records** 49](#_Toc58770071)

[**5.2.10. Coding for Data grid view (Cell Click and view)** 50](#_Toc58770072)

[**5.3 Database Tables** 51](#_Toc58770073)

[**5.3.1. Accounts** 51](#_Toc58770074)

[**5.3.2. Trans** 51](#_Toc58770075)

[**5.3.3. Tra\_booked** 52](#_Toc58770076)

[**5.3.4. Manage Users** 52](#_Toc58770077)

[**5.3.5. Salary** 52](#_Toc58770078)

[**5.3.6. Place** 53](#_Toc58770079)

[**5.3.7. Payment** 53](#_Toc58770080)

[**5.3.8. Issue** 53](#_Toc58770081)

[**CHAPTER 06** 58](#_Toc58770082)

[**EXPERIMENTS & RESULTS** 58](#_Toc58770083)

[**6.EXPERIMENTS & RESULTS** 59](#_Toc58770084)

[**6.1. Testing Admin Login** 59](#_Toc58770085)

[**6.2. Testing Customer Registration** 60](#_Toc58770086)

[**6.3. Testing Hotel Booking** 61](#_Toc58770087)

[**6.4. Testing Vehicle Hiring** 62](#_Toc58770088)

[**6.5. Testing Flight Options** 63](#_Toc58770089)

[**7. Conclusion** 64](#_Toc58770090)

[**8.Future Scope** 64](#_Toc58770091)

[**9.References** 65](#_Toc58770092)

# **LIST OF FIGURES**

[Figure 2 Classical Waterfall Model – Approach of Project 16](#_Toc58771520)

[Figure 3 System Architecture Graphical View 26](#_Toc58771521)

[Figure 4 ER-Diagram – Ticket Management System 27](#_Toc58771522)

[Figure 5 Use Case Diagram – Ticket Management System 29](#_Toc58771523)

[Figure 6 Loading Page 31](#_Toc58771524)

[Figure 7 Admin Login 31](#_Toc58771525)

[Figure 8 Admin Home Page 32](#_Toc58771526)

[Figure 9 User Home Page 32](#_Toc58771527)

[Figure 10 Customer Registration 33](#_Toc58771528)

[Figure 11 Fight 33](#_Toc58771529)

[Figure 12 Flight Ticket 34](#_Toc58771530)

[Figure 13 Loading Page for Hotel View 34](#_Toc58771531)

[Figure 14 View of Hotels 35](#_Toc58771532)

[Figure 15 Transport Options 35](#_Toc58771533)

[Figure 16 Issue Salary 36](#_Toc58771534)

[Figure 17 Report of Flights 36](#_Toc58771535)

[Figure 18 Report of Hotels 37](#_Toc58771536)

[Figure 19 Report of Customer 37](#_Toc58771537)

[Figure 20 Report of Transport 38](#_Toc58771538)

[Figure 21 Employee Salary Receipt 38](#_Toc58771539)

[Figure 22 Manage Users 39](#_Toc58771540)

[Figure 23 Manage Employee 39](#_Toc58771541)

[Figure 24 Manage Hotel 39](#_Toc58771542)

[Figure 25 Manage Meals 40](#_Toc58771543)

[Figure 26 Manage Transport 40](#_Toc58771544)

[Figure 27 Manage Flight 41](#_Toc58771545)

[Figure 28 Email View for hotel booking 41](#_Toc58771546)

[Figure 29 Email view for hiring vehicles 42](#_Toc58771547)

[Figure 30 Email message for registered Customers 42](#_Toc58771548)

[Figure 31 Coding for customer module 43](#_Toc58771549)

[Figure 32 Coding for dashboard 43](#_Toc58771550)

[Figure 33 Coding for Login 44](#_Toc58771551)

[Figure 34 Coding for Message Box 45](#_Toc58771552)

[Figure 35 Coding for Login 45](#_Toc58771553)

[Figure 36 Coding for Display Meals 46](#_Toc58771554)

[Figure 37 Coding for Flight Booking 47](#_Toc58771555)

[Figure 38 Coding for Insert Records 48](#_Toc58771556)

[Figure 39 Coding for Update Records 48](#_Toc58771557)

[Figure 40 Coding for Delete Records 48](#_Toc58771558)

[Figure 41 Coding for Data gird view 49](#_Toc58771559)

[Figure 42 Testing Admin Login 1- HMS 58](#_Toc58771560)

[Figure 43 Testing Admin Login 2 58](#_Toc58771561)

[Figure 44 Testing Customer Register 1 59](#_Toc58771562)

[Figure 45 Testing Customer Registration 59](#_Toc58771563)

[Figure 46 Testing Hotel Booking 1 60](#_Toc58771564)

[Figure 47 Testing Hotel Booking 2 60](#_Toc58771565)

[Figure 48 Testing Vehicle Hiring 61](#_Toc58771566)

[Figure 49 Testing Vehicle Hiring 2 61](#_Toc58771567)

[Figure 50 Testing Flight Option 62](#_Toc58771568)

# **LIST OF TABLES**

[Table 3 Accounts 51](#_Toc58765376)

[Table 4 Trans 51](#_Toc58765377)

[Table 5 Tra\_booked 52](#_Toc58765378)

[Table 6 Manage Users 52](#_Toc58765379)

[Table 7 Salary 52](#_Toc58765380)

[Table 8 Place 53](#_Toc58765381)

[Table 9 Payment 53](#_Toc58765382)

[Table 10 Issue 53](#_Toc58765383)

[Table 11 Hotel Booked 54](#_Toc58765384)

[Table 12 Hotel 54](#_Toc58765385)

[Table 13 Food 54](#_Toc58765386)

[Table 14 Flight 55](#_Toc58765387)

[Table 15 Emp 55](#_Toc58765388)

[Table 16 Customer 56](#_Toc58765389)

[Table 17 Code 56](#_Toc58765390)

[Table 18 Booking 56](#_Toc58765391)

[Table 19 Booked 57](#_Toc58765392)

[Table 20 Add Ticket 57](#_Toc58765393)

[Table 21 Add Flight 57](#_Toc58765394)

[Table 22 Testing Admin Login 59](#_Toc58765395)

[Table 23 Testing Customer Registration 60](#_Toc58765396)

[Table 24 Testing Hotel Booking 61](#_Toc58765397)

[Table 25 Vehicle Hiring 62](#_Toc58765398)

# **CHAPTER 01**

# **INTRODUCTION**

## **Introduction**

### **1.1Background**

This Ticket Management System is developed for aircore ticketing services in favor of the ticket management team which help themselves to save the records of the employees, passengers. It helps them from the manual work from which it is very difficult to find the record of the passengers, employees, payments, meals, bookings and other things.

This solution is developed on the plight of the ticket management team; through this they cannot require so eminent person to handle and manage the affairs of the passengers & employees in the services, all you need to do is to login as admin and you can see the information of all the passengers, employees, bookings and other things.

This system is fully computerized and having more flexible and safe options, and it is more securable way of store records of company therefore this TMS is much useful for the current ticketing functions.

### **1.2 Purpose**

The purpose of this project is to make an automated system to carry out the various operations of Aircore Ticketing Company. This Ticket Management System will provide the ease of use to the admin of the company by performing all the work on a computer system rather than following a manual approach. This approach helps improving the reliability of the data maintained and provides a fast and efficient interface for the users of the software.

### **1.3 Objectives and Scope**

This software product the Ticket management to improve their services for all the employees and passengers. This also reduce the manual work of the person in admin panel and bundle of registers that were search when to find the information of the previous passenger, employee, payments and bookings. Through this system we can save the records of the passengers, employees and others in database. The database of the system will help to record all the details of the passengers, employees and others at once and help to manage records at any time.

* To automate each and every activity of the manual system, which increases its throughput.
* To provide a quick response with very accurate information as and when required.
* Reduce the cost of maintenance.

### **1.4 Project Schedule**

As I scheduled my project work I have started my project work on October 1st week, it took 1 week for requirement gathering for the project, and another 2 weeks for Designing part of the project. At the starting of the 1 week of November I have started to code my project and it took 2 weeks, after that I have taken a testing part to my project and, debugged errors. As I scheduled I have completed my project on end of the November (4th Week).

### **1.5 Approach**

#### **Classical Waterfall Model**

In order fulfill the requirements as highlighted in the objectives I have chosen Classical Waterfall Model to be used in this system to be developed. It is very easy to handle and identify the user requirements and easy to develop the project according to the user requirements. Therefore, I have chosen this software model to develop the project. Reasons for choosing this methodology is, it supports high user involvement, easy to use & understand, taking reviews in the completion of every phase and appropriate model for this project.

The phases of Classical Waterfall Model are;

Figure 2 Classical Waterfall Model – Approach of Project

# **CHAPTER 02**

# **SOFTWARE REQUIREMENT**

# **SPECIFICATION**

## **Software Requirement Specification**

## **2.1 User Requirement Specification**

The user requirements for this system is to make the system fast and flexible, less prone to error, reduce expenses and save the time.

* Less human error.
* Strength and strain of manual labor can be reduced.
* High Security.
* Data redundancy.
* Data consistency.
* Easy to handle.
* Easy data updating.

## **2.2 System Requirement Specification**

### **2.2.1 Functional System Requirements**

It defines the functional requirements that applicable to the Hostel Management System. These are the sub modules of the system.

* Admin
* Guest
* Meals
* Hotel
* Transport
* Reservation
* Tally up
* Manage

### **2.2.3 Non-Functional Requirement Specification**

* Performance Requirements.
* Safety Requirements.
* Security Requirements.
* Flexibility
* Reusability

## **2.3 Hardware Requirements.**

* Processor
* RAM
* Hard Disk
* Keyboard
* Monitor or LCD

## **2.4 Software Requirements.**

* Operating System: Microsoft Windows.
* Front-end Tool: Visual Basic 2015 –Enterprise.
* Back-end Tool: MySQL Database.

## **2.5 Software Tool Used**

* **Front-End Tool – Visual Basic 2015 Enterprises**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. The integrated debugger works both as a source-level debugger and a machine-level debugger.

I have used Visual Studio tool to code this software, and also I have used this software to testing part of the system and debugged errors by testing the system. It is a very efficient IDE to develop any software.

* **Back-End Tool – MySQL**

MySQL is an Oracle-backed open source relational database management system based on Structured Query Language. MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and software development.

I have used MySQL database to save all records of the system. It eases the work of the project development.

# **CHAPTER 03**

# **SYSTEM ANALYSIS**

## **3.System Analysis**

### **3.1 Data Collection Methods**

* Documentations (Reports, PDF, Books etc.)
* Observations
* Questionnaires

### **3.2 Existing System**

The existing Ticket Management System is a manual way of handing records of ticketing services.

* More human error.
* More strength and strain of manual labor needed.
* Repetition of some procedures.
* Lack of security.
* Data redundancy.
* Record keeping is difficult.
* Time consuming.

### **3.3 Proposed System**

The proposed Ticket Management System is fully computerized system. It provides easy and quick wat to access the records. It has authorization schemes. It reduces the consuming of time. Through this system we can save the records of the passengers, employees and other module records in database easily. It provides an easy way to manage (Add, Update, Delete, View, Search, Filter) records of ticketing services.

### **3.4 Literature Review**

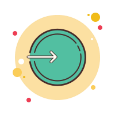
* According to the view of (Patrick J. TooleSudhir KrishnaswamyNima MoayediDavid N. LordFrancisco J. Gutierrez, 2007) embodiments of the invention include a system and set of processes for managing tickets. The system maintains a database of tickets belonging to a company and the allotment of the tickets to employees and clients. The system provides interfaces for requesting tickets and managing ticket requests. The system optionally provides automated request processing, ticket resale and purchase and electronic distribution.
* According to the view of (panelNorazahMohd , SukiaNorbayahMohd Sukib, 2017) In the context of intense market competition, airlines are enriching their business operations by offering flight ticket booking apps that can be downloaded on mobile devices. This study aims to examine the intention of individuals to use such apps, and uses Structural Equation Modelling (SEM) to analyses the data gathered from individuals in Malaysia. Perceived usefulness represents the greatest influence on individuals in respect of their intention to engage with such an app offered on a mobile device. Airline companies should consider using advances in ICT within their overall portfolio of marketing strategies, if they wish to become more competitive in the current market. They should utilize the interactive and attractive features of online channels in order to encourage more individuals to try their flight ticket booking apps on mobile devices. The proposed model could be used as a baseline model in future research.
* According to the view of (Kola Ayanlowo, O. Shoewu , Segun O. Olatinwo Omitola , Damilola D. Babalola, 2014) hostels Consequently it has increased knowledge and helped produce a population of enlightened citizens who can easily abide by the rules of civilized society and contribute meaningfully to the process of democratic governance. Most of the newly established educational institutions however, are using the old conventional techniques for managing their assets especially hostel facilities. These old techniques with its inherent limitations have impacted negatively on the overall organizational efficiency of this educational systems. In this paper, the development of an automated hostel accommodation management system is proposed.
* Finally, according to my view, I have introduced a feature in my system that a passenger can surf the travel agency details under the supervision of the receptionist in the computer system, therefore passenger can know more information about the travelling services, if the passenger wishes to allocate a ticket, she/he can log into the system by getting a username and password from the receptionist and log into the system. Passenger can fill a register form which is shown in the system to reserve/get their id. Admin can manage all the details about passengers such as view, search, update, delete and filter. As it is admin can add and manage the hotel records as well as admin can add and manage employee records who are working in the company and admin can manage the all other modules too. I hope it will work out very efficiently in future for any ticket management systems. This Ticket Management System works very flexibly and efficiently for the ticket service companies and support the user to operate the system very easily.

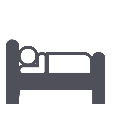
# **CHAPTER 04**

# **SYSTEM DESIGN**

## **4.System Design**

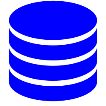
### **4.1 System Architecture**

********

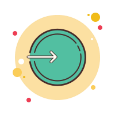
****



****

****



****

****

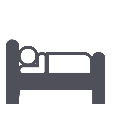
****

Figure 3 System Architecture Graphical View

### **4.2 ER-Diagram**

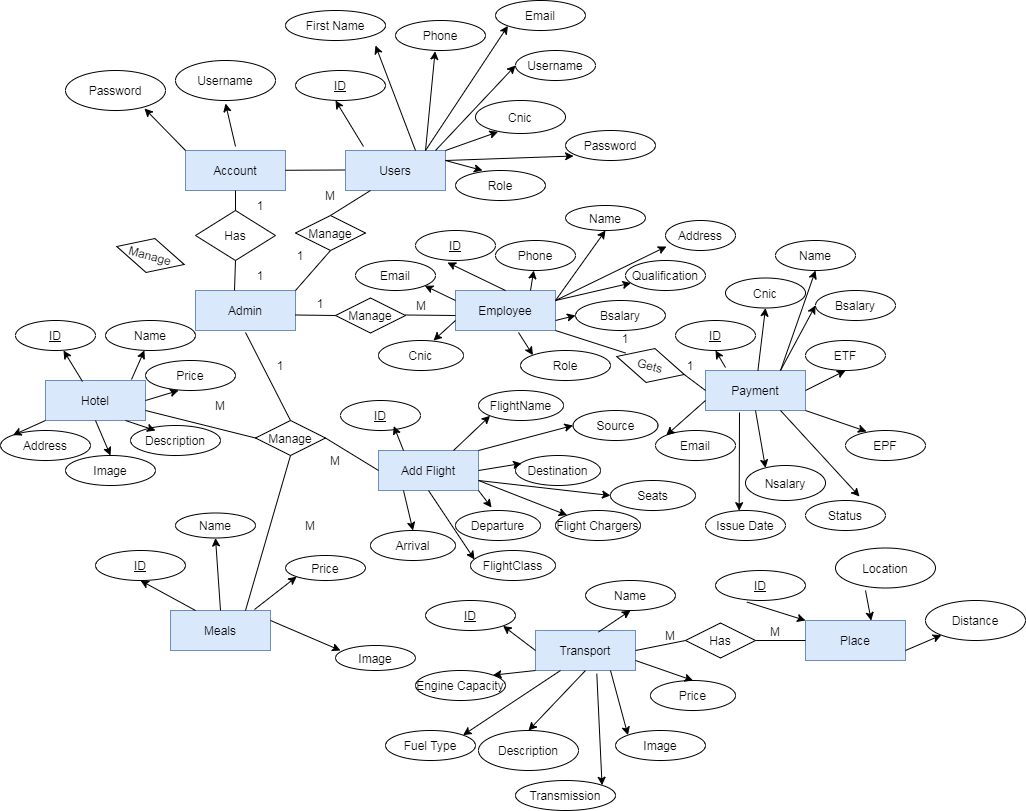
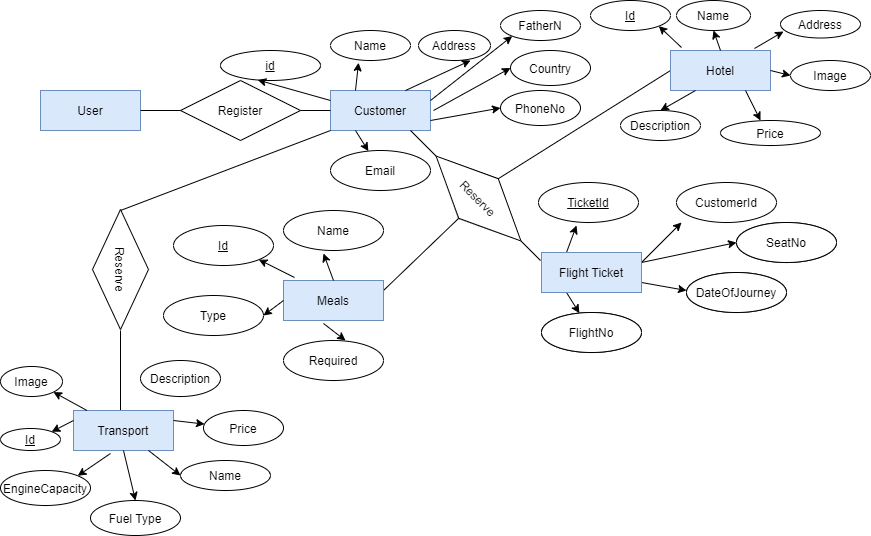
****

Figure 4 ER-Diagram – Ticket Management System



### **4.3Use Case Diagram**

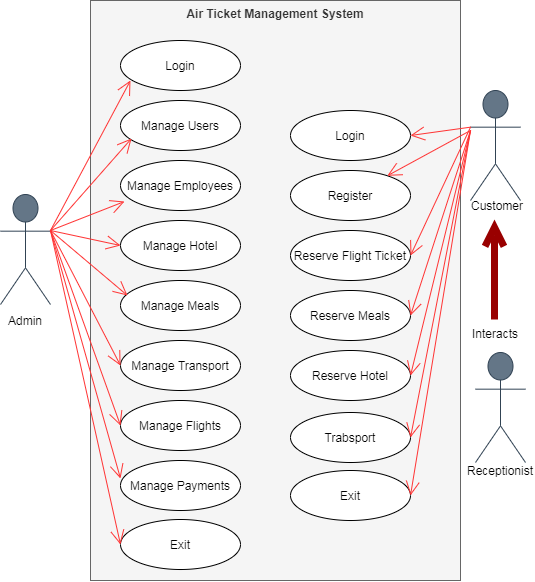
****

Figure 5 Use Case Diagram – Ticket Management System

# **CHAPTER 05**

# **SYSTEM IMPLIMENTATION**

## **5. SYSTEM IMPLIMENTATION**

## **5.1 Interfaces**

### **5.1.1. Loading Page**

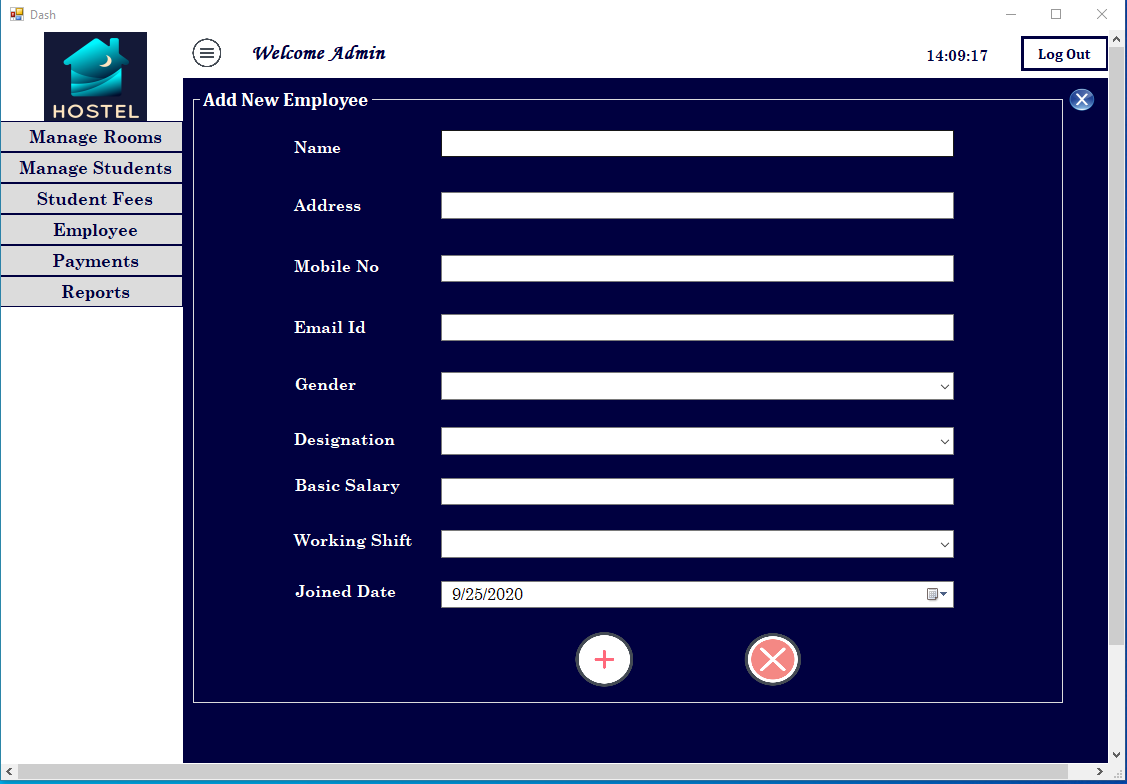
****

Figure 6 Loading Page



### **5.1.2. Admin Login**

Figure 7 Admin Login



### **5.1.3. Admin & User Home Page**

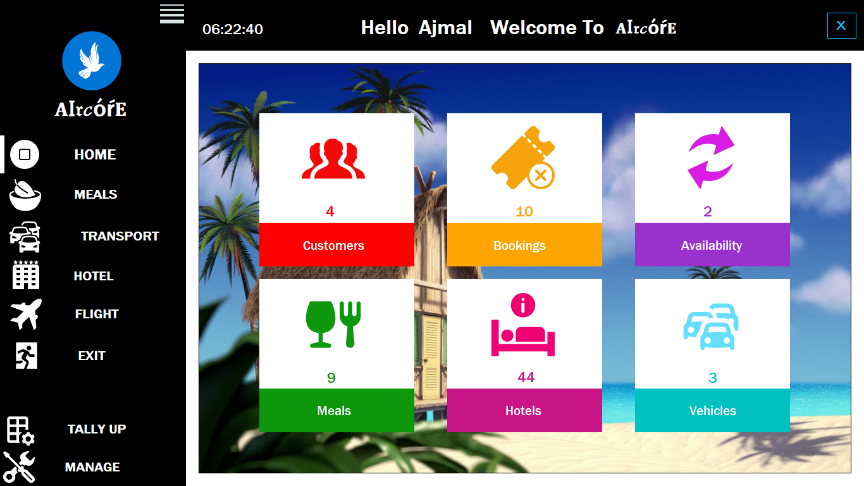
****

Figure 8 Admin Home Page

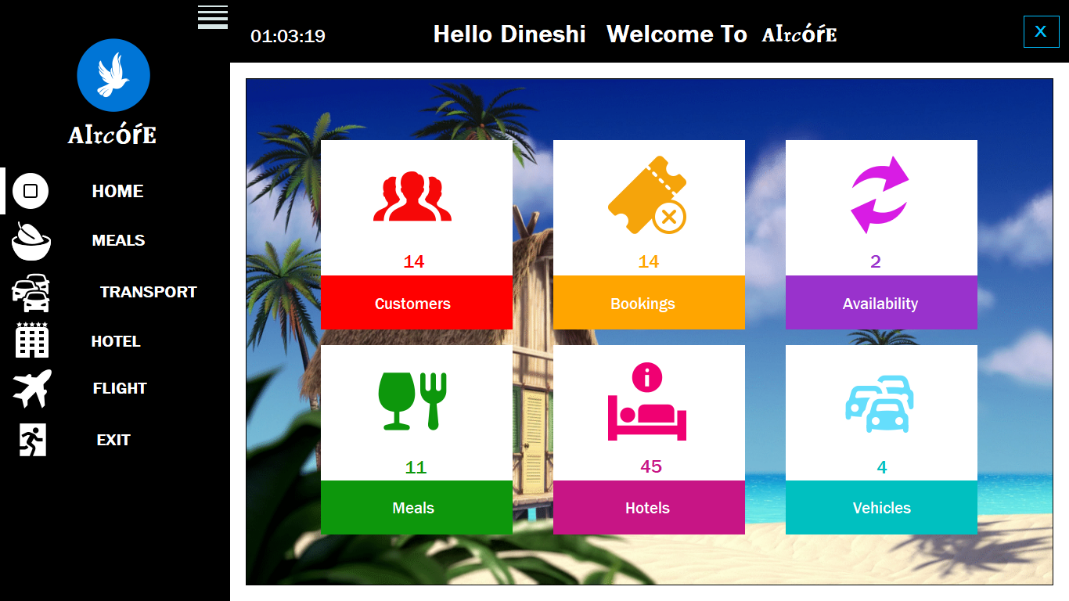
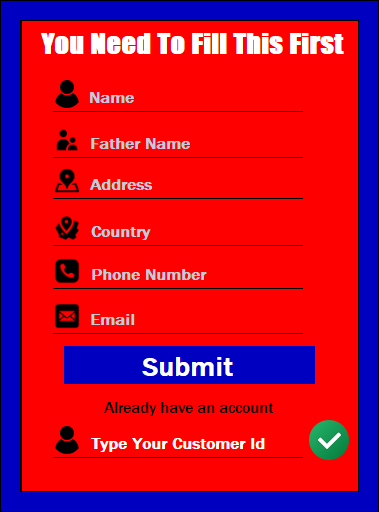
****

Figure 9 User Home Page

****

### **5.1.4. Customer Registration**

Figure 10 Customer Registration

### **5.1.5. Flight**

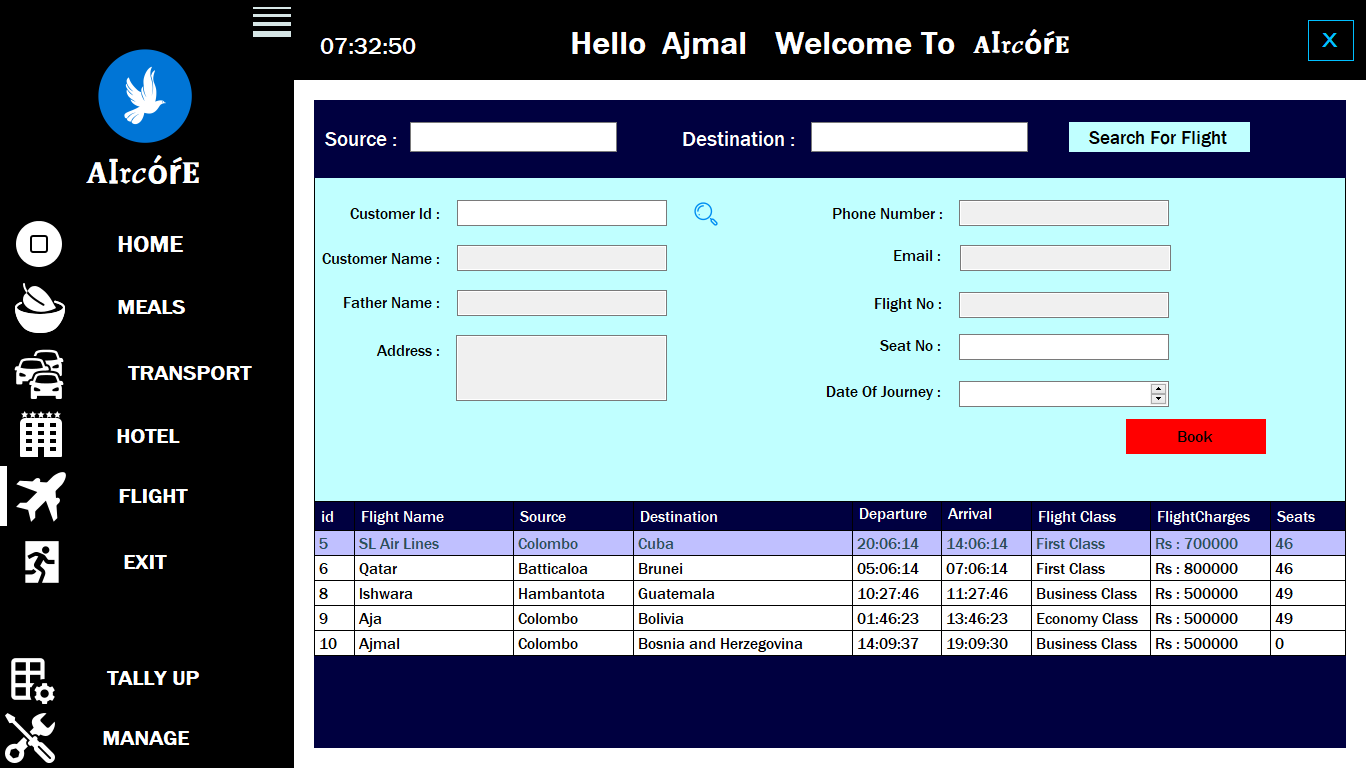


Figure 11 Fight

### **5.1.6. Flight Ticket (Customer)**

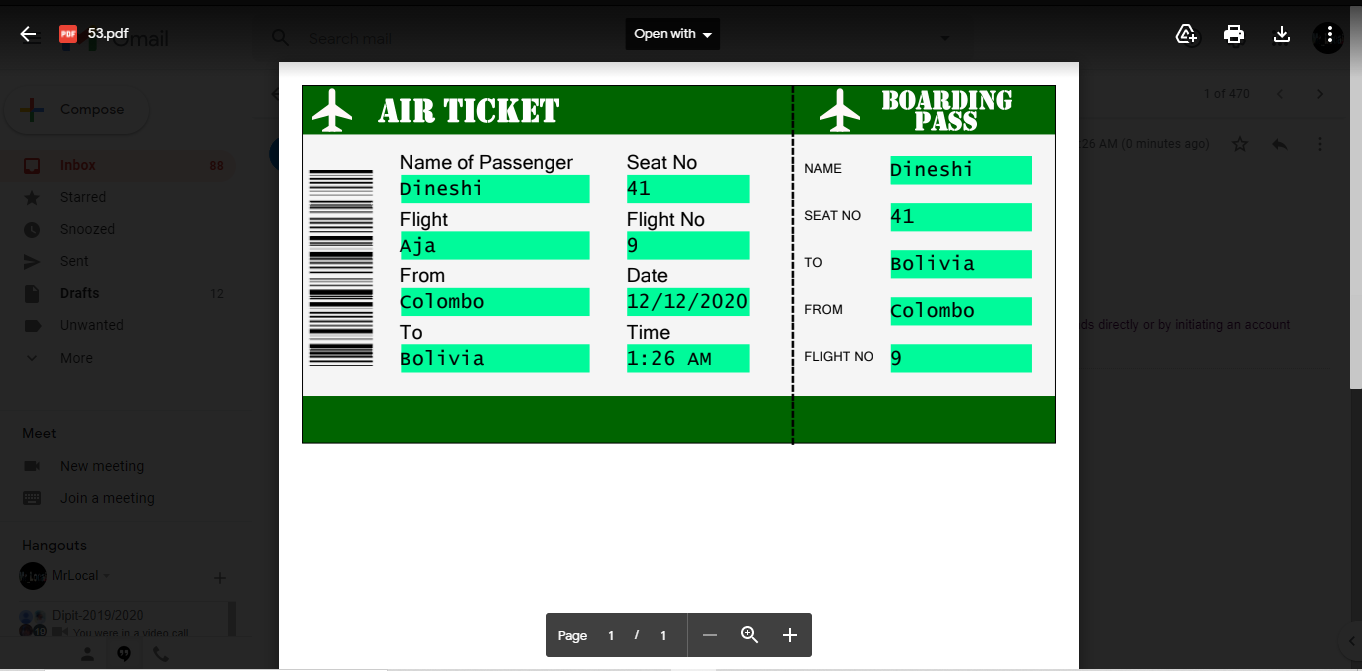


Figure 12 Flight Ticket

### **5.1.7. Loading Page for Hotel View**

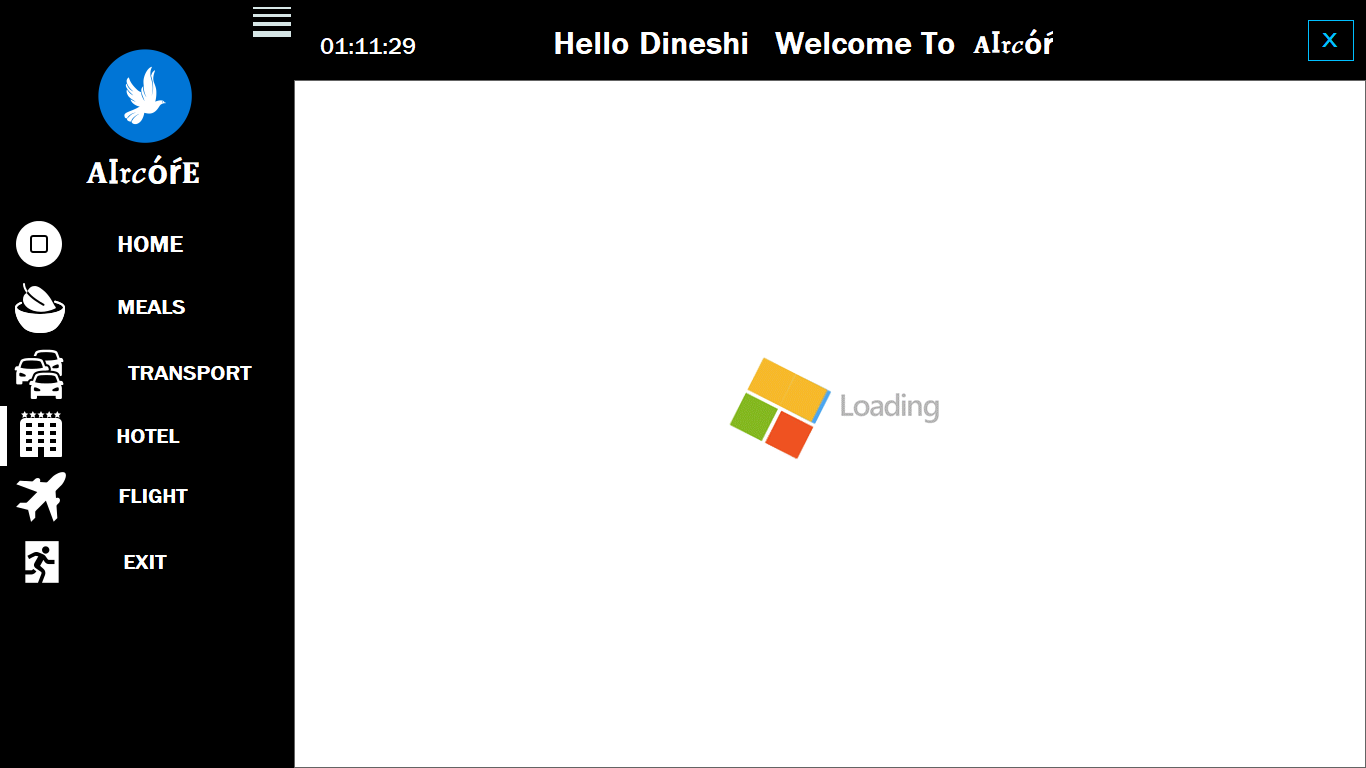


Figure 13 Loading Page for Hotel View

### **5.1.8. View of Hotels**

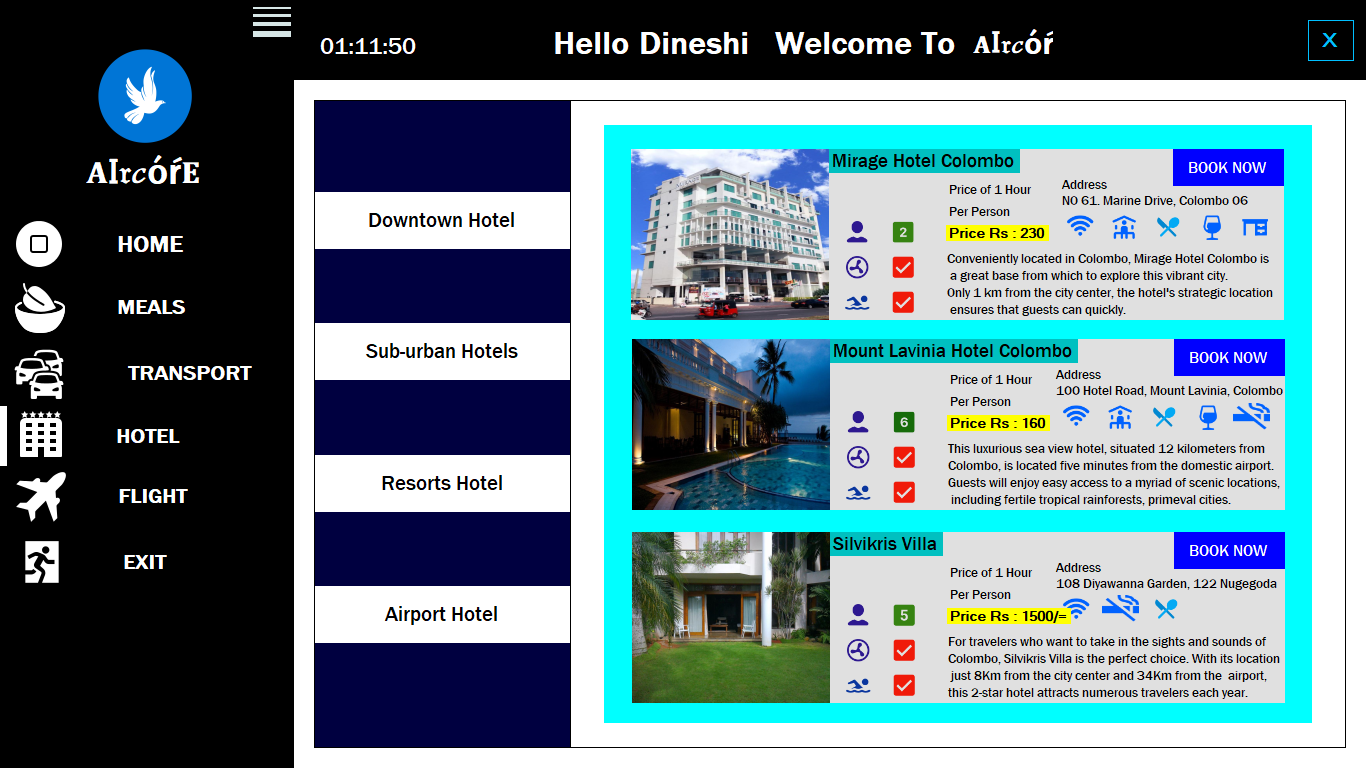


Figure 14 View of Hotels

### **5.1.9. Transport Options**

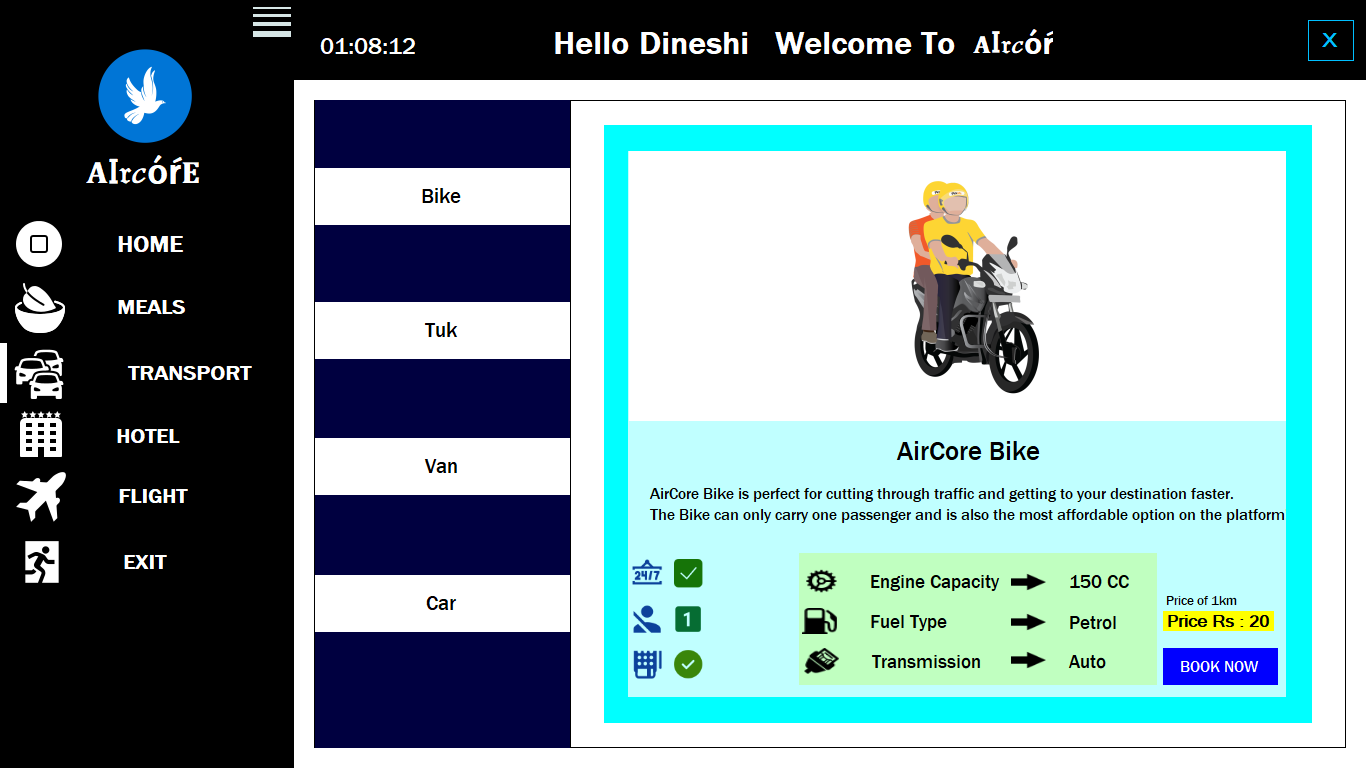


Figure 15 Transport Options

### **5.1.10. Issue Salary**

****

Figure 16 Issue Salary

### **5.1.11. Report of Flights**

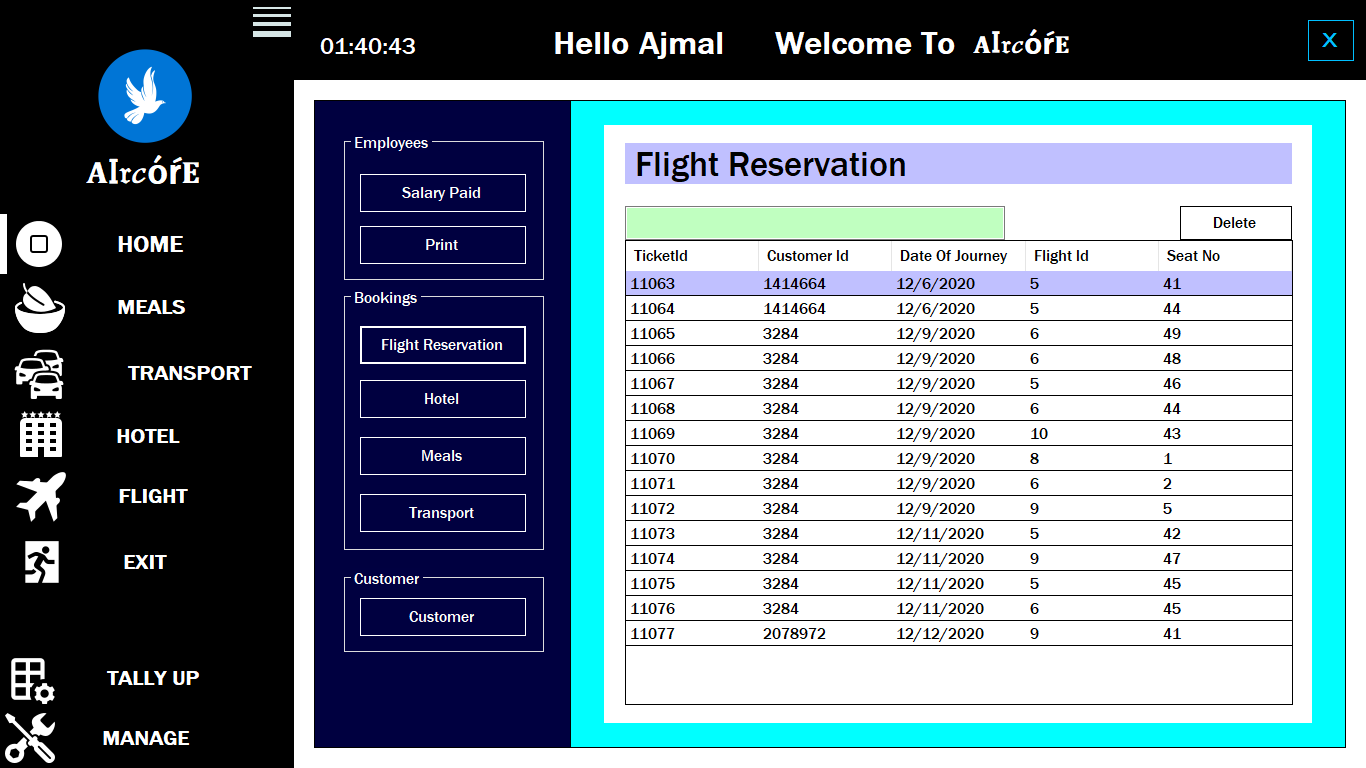
****

Figure 17 Report of Flights

### **5.1.12. Report of Hotel**

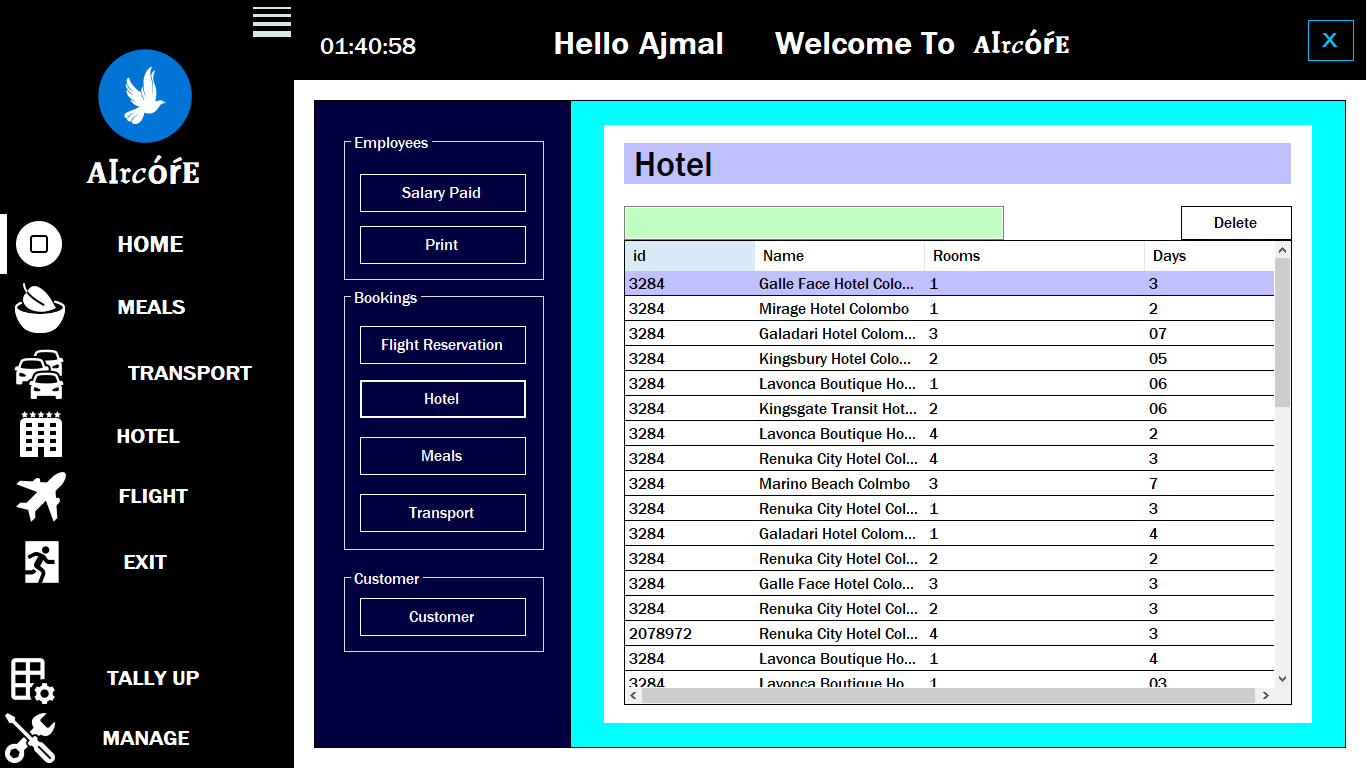
****

Figure 18 Report of Hotels

### **5.1.13. Report of Customer**

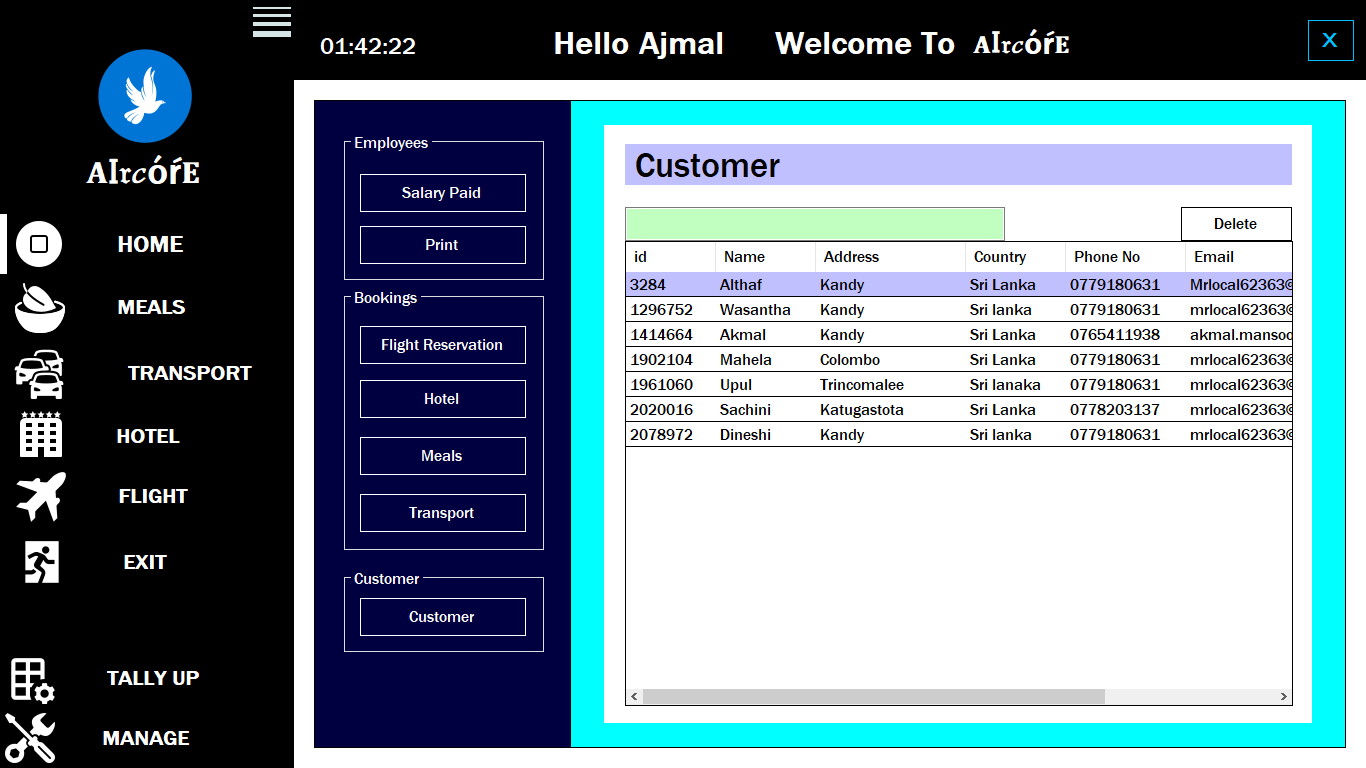
****

Figure 19 Report of Customer

### **5.1.14. Report of Transport**

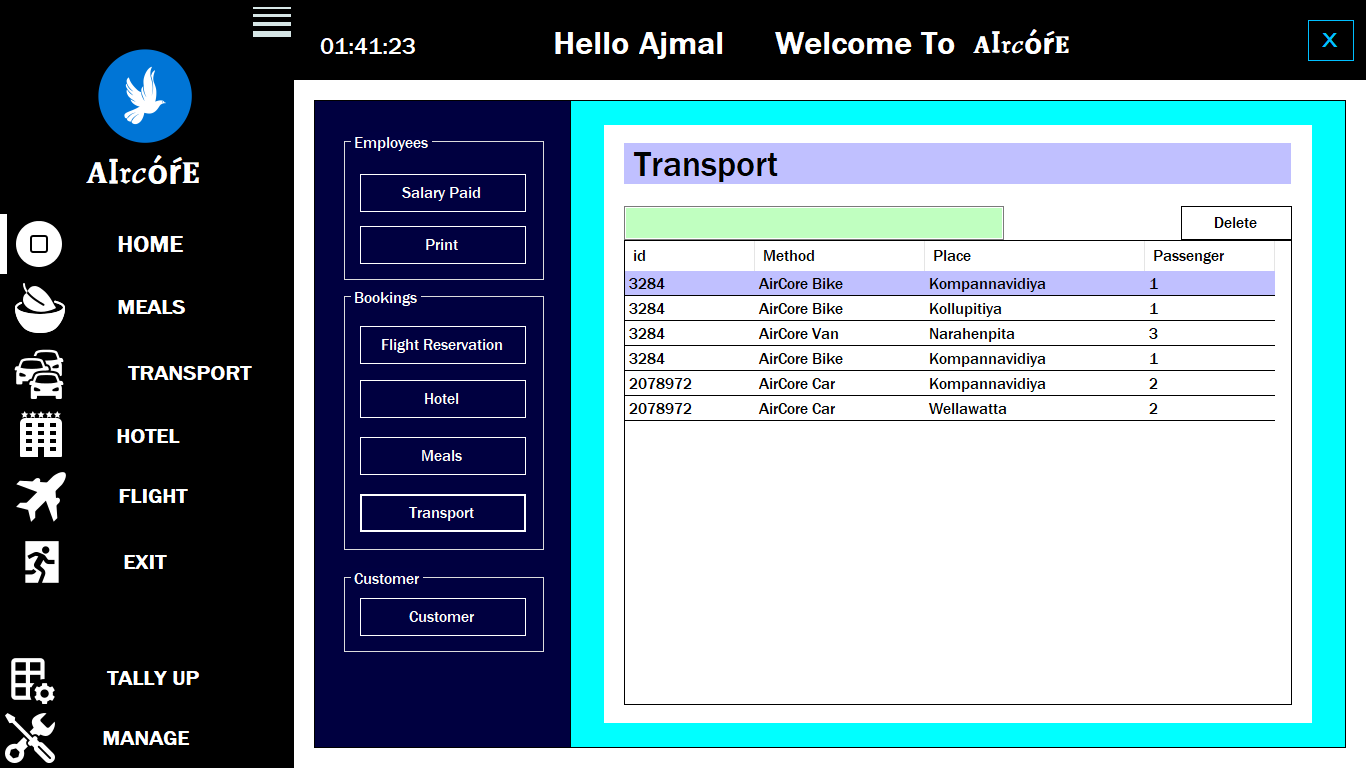
****

Figure 20 Report of Transport

### **5.1.15. Print Preview of Employee Salary Receipt**

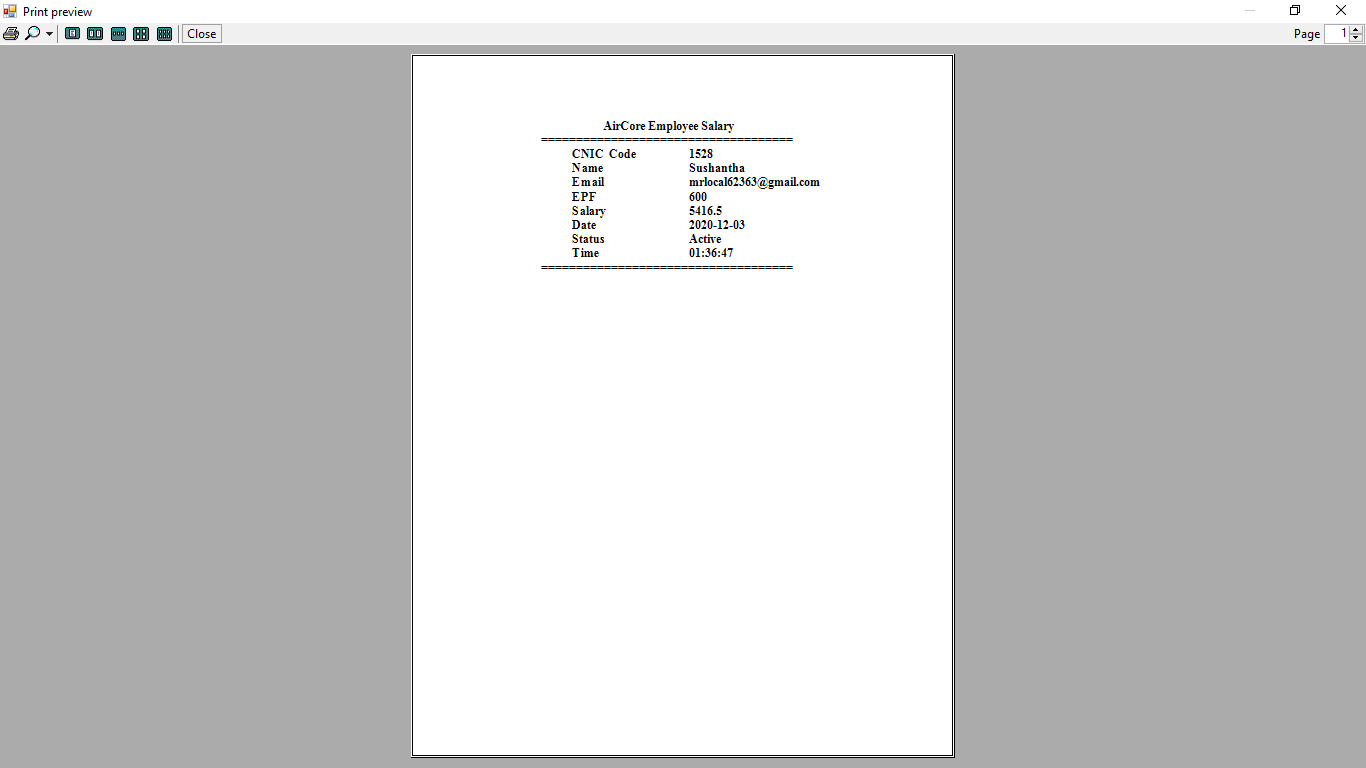


Figure 21 Employee Salary Receipt

### **5.1.16. Manage Users**

Figure 22 Manage Users

### **5.1.17. Manage Employee**

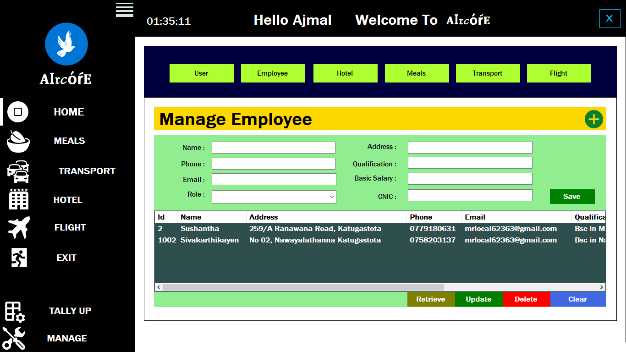


Figure 23 Manage Employee

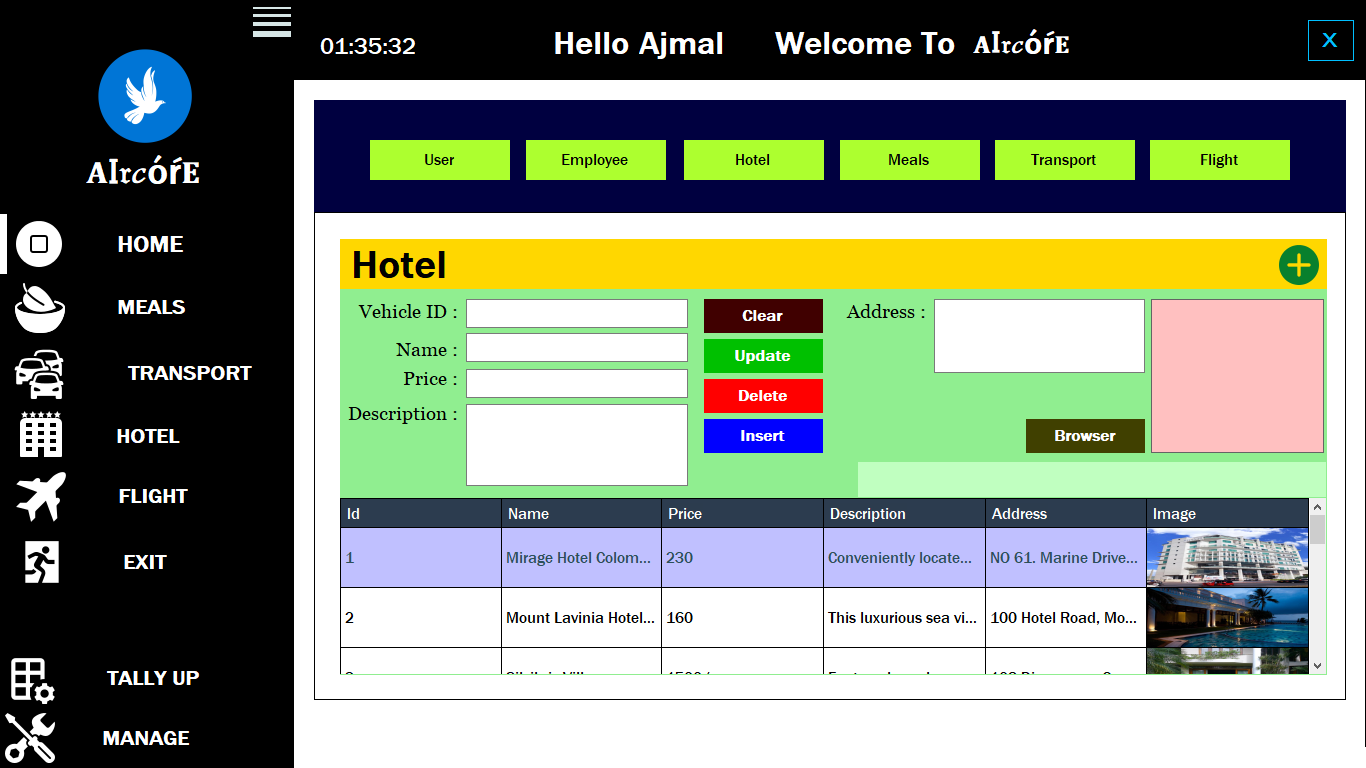
**5.1.18. Manage Hotel**

Figure 24 Manage Hotel

**5.1.19. Manage Meals**

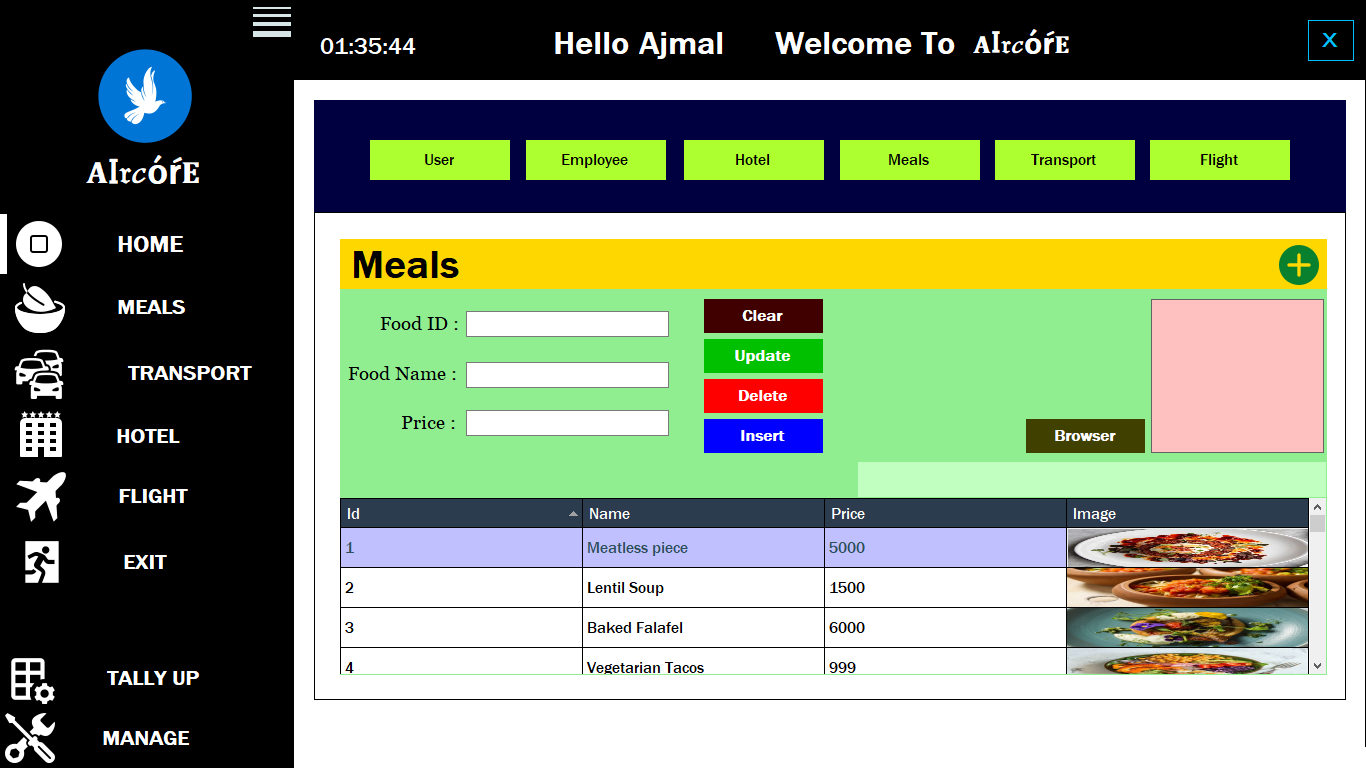
****

Figure 25 Manage Meals

### **5.1.20. Manage Transport**

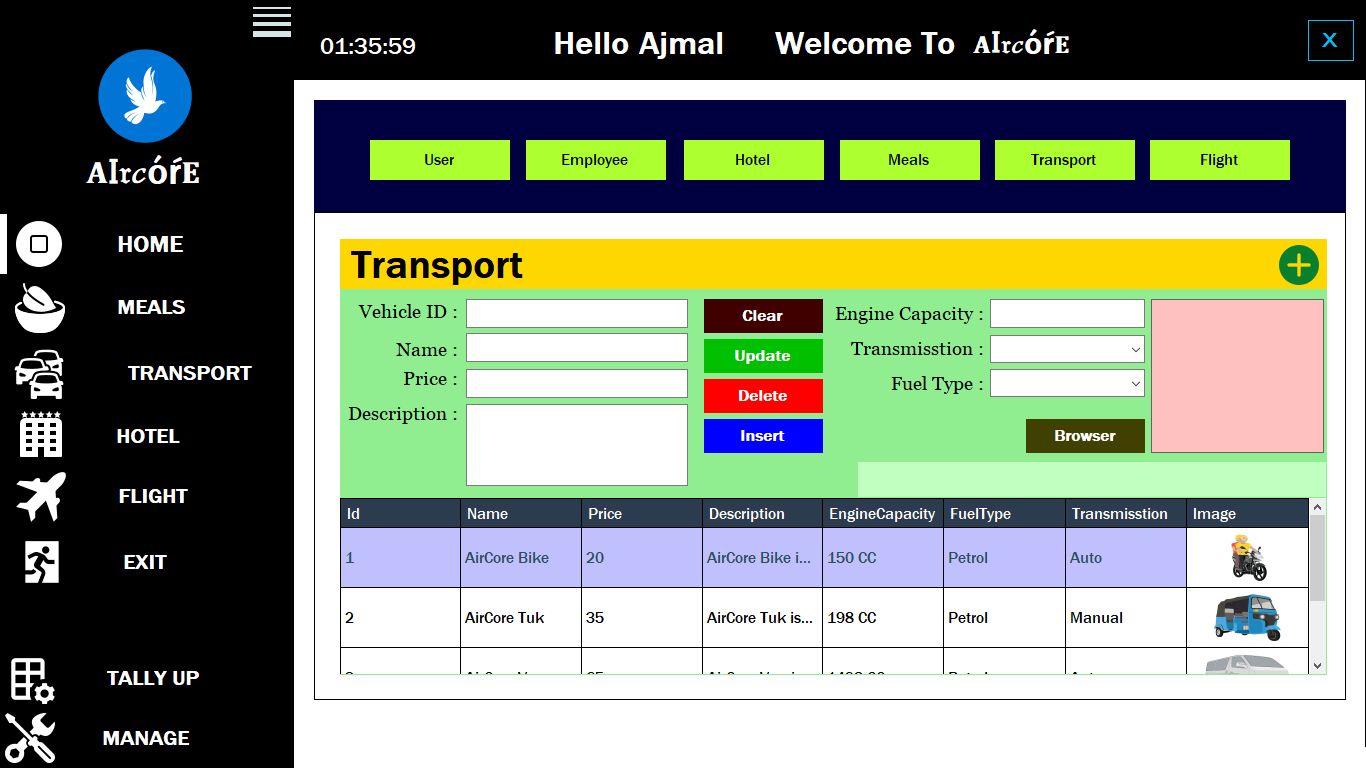
****

Figure 26 Manage Transport

### **5.1.21. Manage Flight**

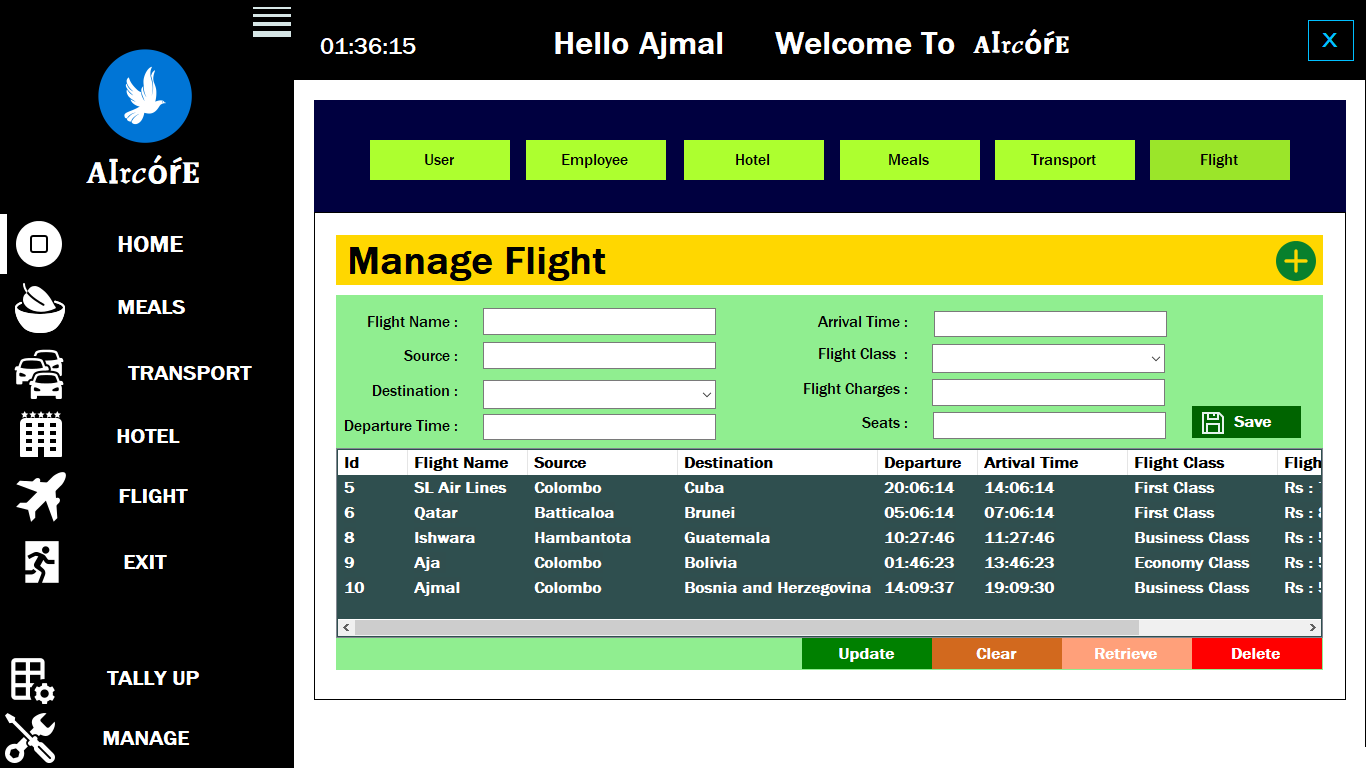


Figure 27 Manage Flight

### **5.1.22. Email view for hotel booking**

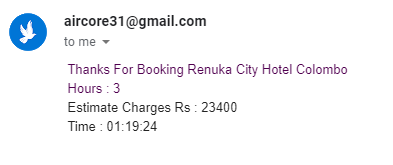


Figure 28 Email View for hotel booking

### **5.1.23. Email view for hiring vehicles**



Figure 29 Email view for hiring vehicles

### **5.1.24. Email message registered customers**

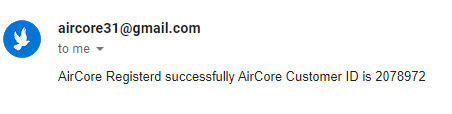
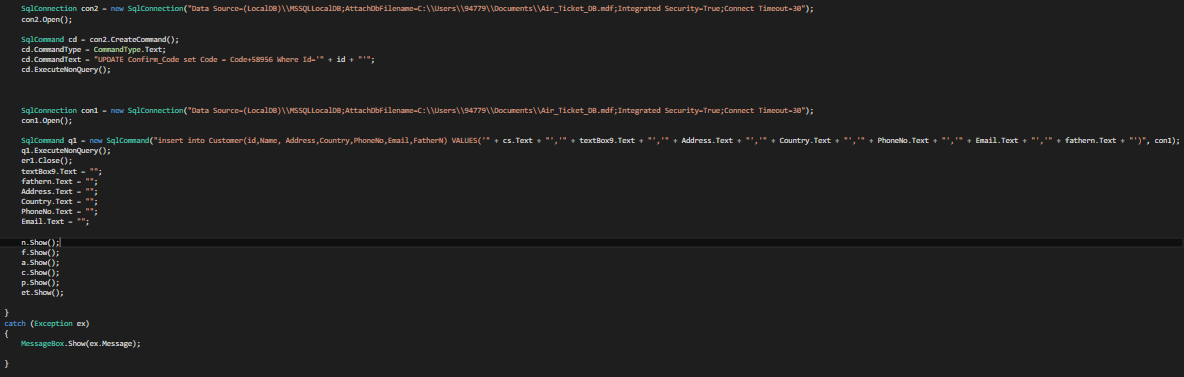


Figure 30 Email message for registered Customers

## **5.2 Coding (Selected)**

### **5.2.1. Coding for Customer Module**



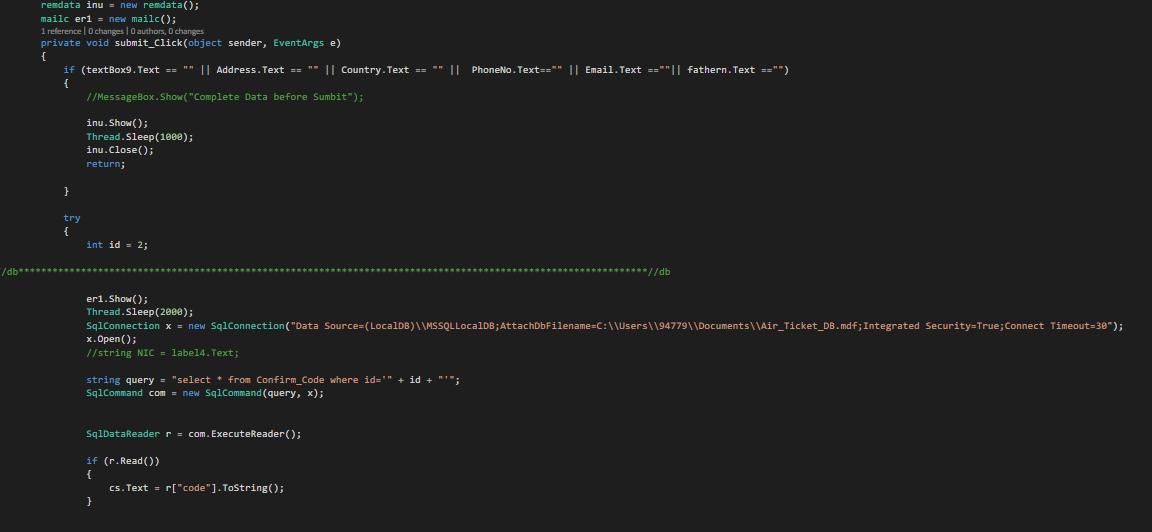


Figure 31 Coding for customer module

### **5.2.2. Coding for Dashboard**

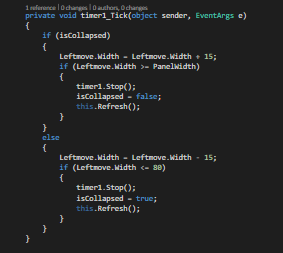
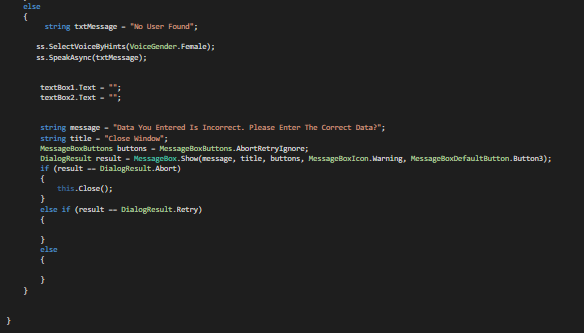


Figure 32 Coding for dashboard

### **5.2.3. Coding for Login**



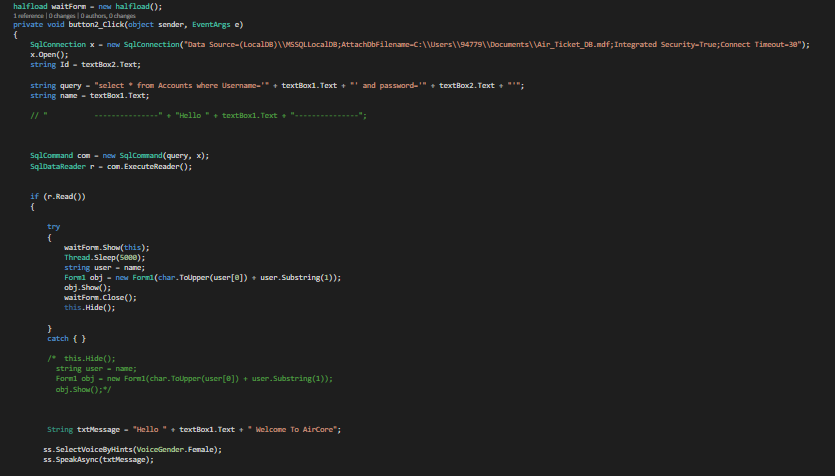


Figure 33 Coding for Login

### **5.2.4. Coding for Message Box**

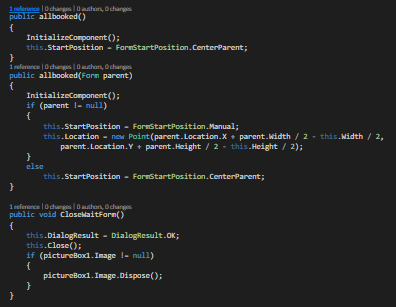


Figure 34 Coding for Message Box

### **5.2.5. Coding for Login**

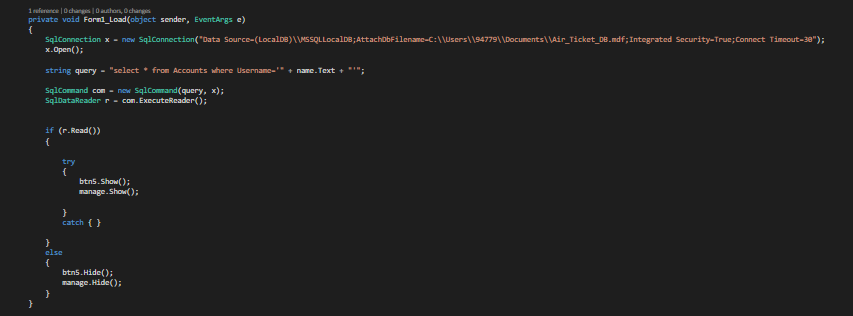
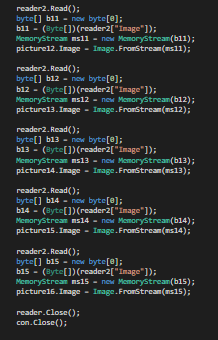
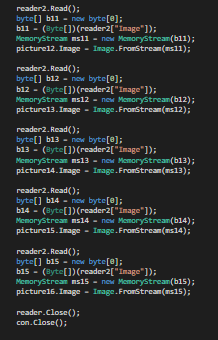


Figure 35 Coding for Login

### **5.2.5. Coding for Display Meals (Selected)**



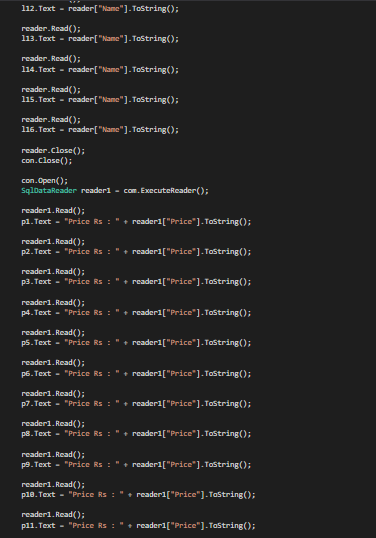
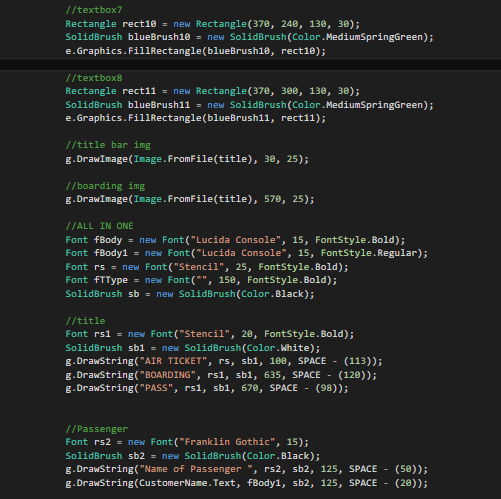


Figure 36 Coding for Display Meals

### **5.2.6. Coding for Flight Bookings (Selected)**



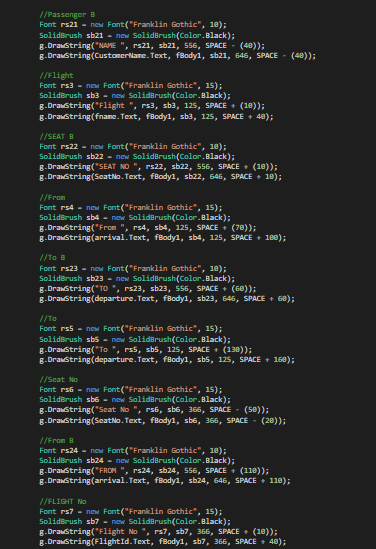


Figure 37 Coding for Flight Booking

### **5.2.7. Coding for Insert Records**



Figure 38 Coding for Insert Records

### **5.2.8. Coding for Update Records**

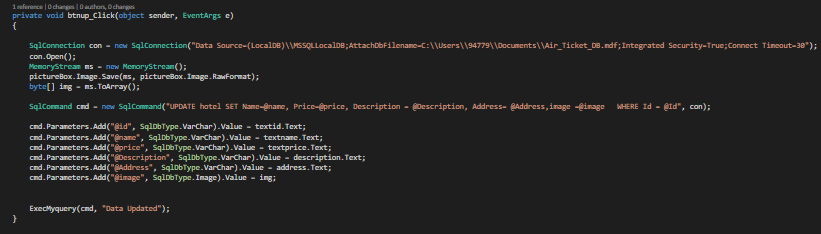


Figure 39 Coding for Update Records

### **5.2.9. Coding for Delete Records**

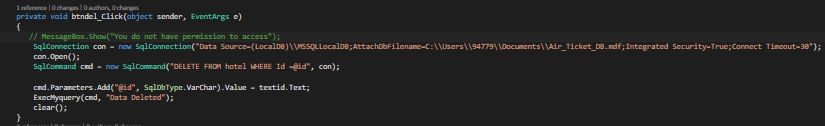
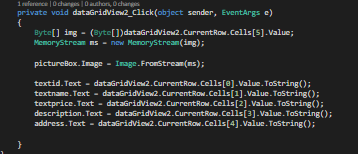


Figure 40 Coding for Delete Records

### **5.2.10. Coding for Data grid view (Cell Click and view)**

****

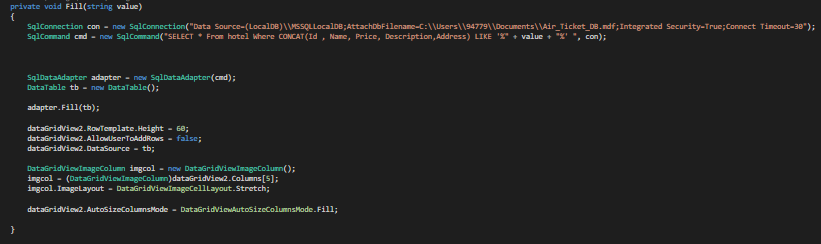
****

Figure 41 Coding for Data gird view

## **5.3 Database Tables**

### **5.3.1. Accounts**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Username | nchar(10) |
| Password | nchar(10) |

Table 1 Accounts

### **5.3.2. Trans**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Name | nvarchar(50) |
| Price | nvarchar(50) |
| Description | nvarchar(MAX) |
| EngineCapacity | nvarchar(50) |
| FuelType | nvarchar(50) |
| Transmission | nvarchar(50) |
| Image | image |

Table 2 Trans

### **5.3.3. Tra\_booked**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Method | nvarchar(50) |
| Place | nvarchar(50) |
| Passenger | nvarchar(50) |

Table 3 Tra\_booked

### **5.3.4. Manage Users**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| First\_Name | text |
| Phone | text |
| Cnic | text |
| Username | text |
| Password | ntext |
| Role | text |

Table 4 Manage Users

### **5.3.5. Salary**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Month | ntext |
| Date | nchar(10) |

Table 5 Salary

### **5.3.6. Place**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Location | varchar(MAX) |
| Dis | varchar(50) |

Table 6 Place

### **5.3.7. Payment**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| CNIC | varchar(50) |
| Name | varchar(50) |
| Bsalary | varchar(50) |
| EPF | varchar(50) |
| ETF | varchar(50) |
| Nsalary | varchar(50) |
| Issue\_Date | varchar(50) |
| Status | varchar(50) |
| Email | varchar(50) |

Table 7 Payment

### **5.3.8. Issue**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Cnic | varchar(50) |
| Doc | image |

Table 8 Issue

**5.3.9 Hotel\_booked**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Name | varchar(50) |
| Room | varchar(50) |
| Days | varchar(50) |

Table 9 Hotel Booked

**5.3.10 Hotel**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Name | varchar(MAX) |
| Price | varchar(MAX) |
| Description | varchar(MAX) |
| Address | varchar(MAX) |
| Image | image |

Table 10 Hotel

**5.3.11 Food**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Name | varchar(MAX) |
| Price | varchar(MAX) |
| Image | image |

Table 11 Food

**5.3.12 Flight**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| FlightId | int (Auto Increment) |
| FlightName | text |
| DepaurtureTime | time |
| Destination | text |
| FlightClass | nchar(10) |
| FlightChargers | int |
| ArrivalTime | time(7) |
| Date | date |
| Seats | int |

Table 12 Flight

**5.3.13 Emp**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Name | varchar(50) |
| Address | varchar(50) |
| Phone | varchar(50) |
| Email | varchar(50) |
| Qualification | varchar(50) |
| Bsalary | varchar(50) |
| Role | varchar(50) |
| CNIC | varchar(50) |

Table 13 Emp

**5.3.14 Customer**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Name | nvarchar(50) |
| Address | nvarchar(50) |
| PhoneNo | nvarchar(50) |
| Email | nvarchar(50) |
| Country | nchar(10) |
| FatherN | nvarchar(50) |

Table 14 Customer

**5.3.15 Code**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Code | int |

Table 15 Code

**5.3.16 Booking**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| TicketId | int (Auto Increment) |
| CustermerId | int |
| Dateofjourney | date |
| FlightId | int |
| SeatNo | int |
| Country | int |
| FatherN | int |

Table 16 Booking

**5.3.17 Booked**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| Name | varchar(50) |
| Required | varchar(50) |
| Type | varchar(50) |

Table 17 Booked

**5.3.18 AddTicket**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| TicketName | nchar(10) |
| TicketNumber | nchar(10) |
| Tickets | nchar(10) |

Table 18 Add Ticket

**5.3.19 AddFlight**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Id | int (Auto Increment) |
| FlightName | nvarchar(50) |
| Source | nvarchar(50) |
| Destination | nvarchar(50) |
| Departure | time(7) |
| Arrival | time(7) |
| FlightClass | nvarchar(50) |
| FlightCharges | nvarchar(50) |
| Seats | int |

Table 19 Add Flight

# **CHAPTER 06**

# **EXPERIMENTS & RESULTS**

## **6.EXPERIMENTS & RESULTS**

### **6.1. Testing Admin Login**

|  |  |  |
| --- | --- | --- |
| **Test** | **Check in Success** | **Check in Failed** |
| **Giving correct password** | **“Login Successful !”** | **-** |
| **Giving incorrect password** | **-** | **-** |

Table 20 Testing Admin Login

****

Figure 42 Testing Admin Login 1- HMS

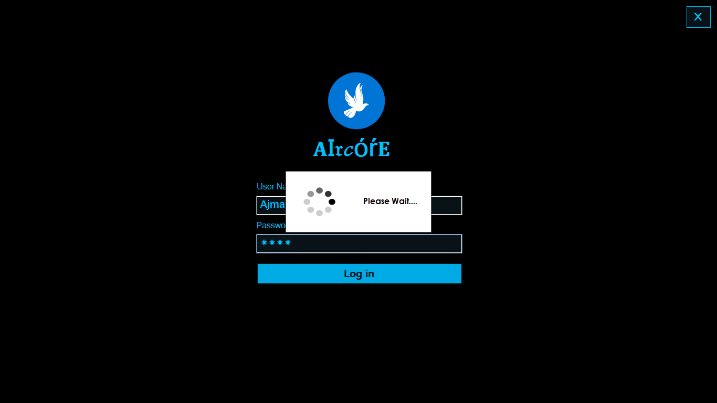
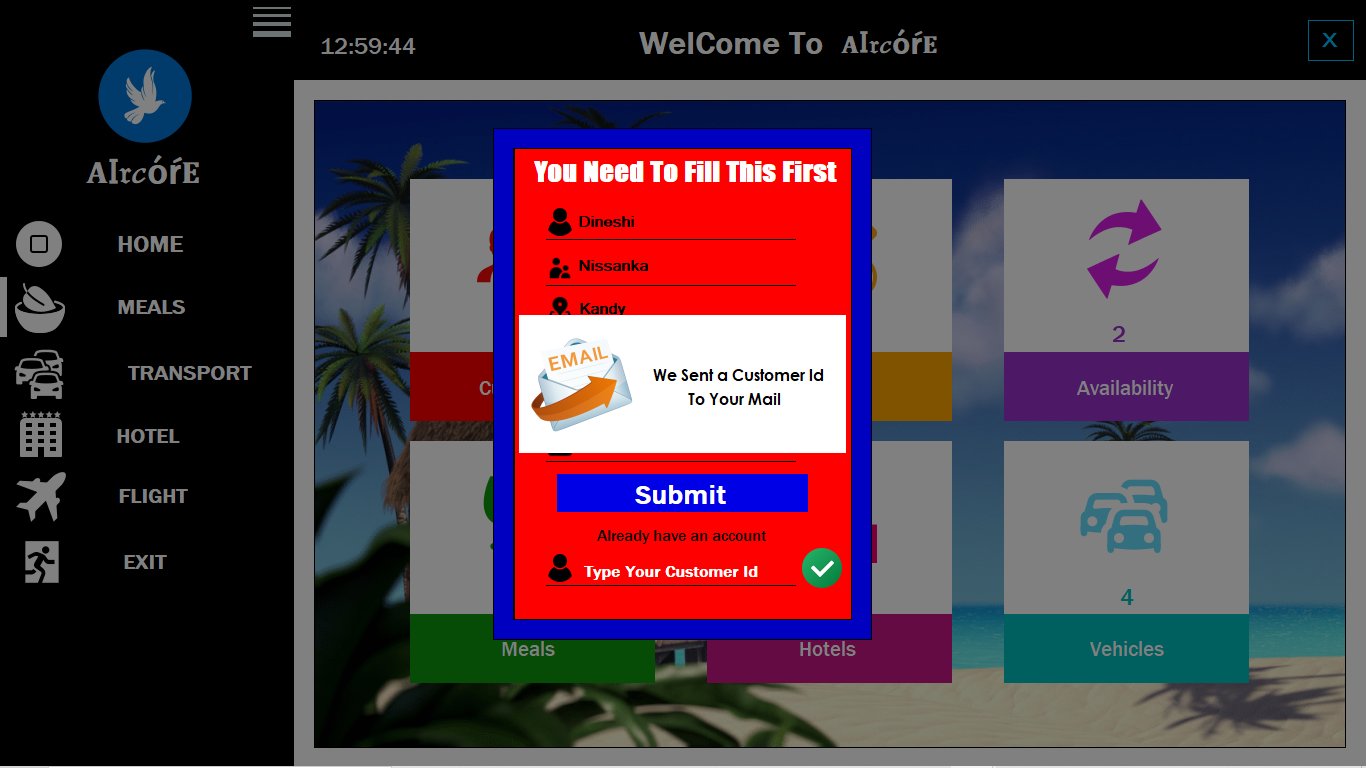
****

Figure 43 Testing Admin Login 2

### **6.2. Testing Customer Registration**

|  |  |  |
| --- | --- | --- |
| **Test** | **Check in Success** | **Check in Failed** |
| **Giving correct data entries** | **“We sent a customer id to your mail”** | **-** |
| **Giving incorrect data entries** | **-** | **“Incomplete Data Entry”** |

**** Table 21 Testing Customer Registration

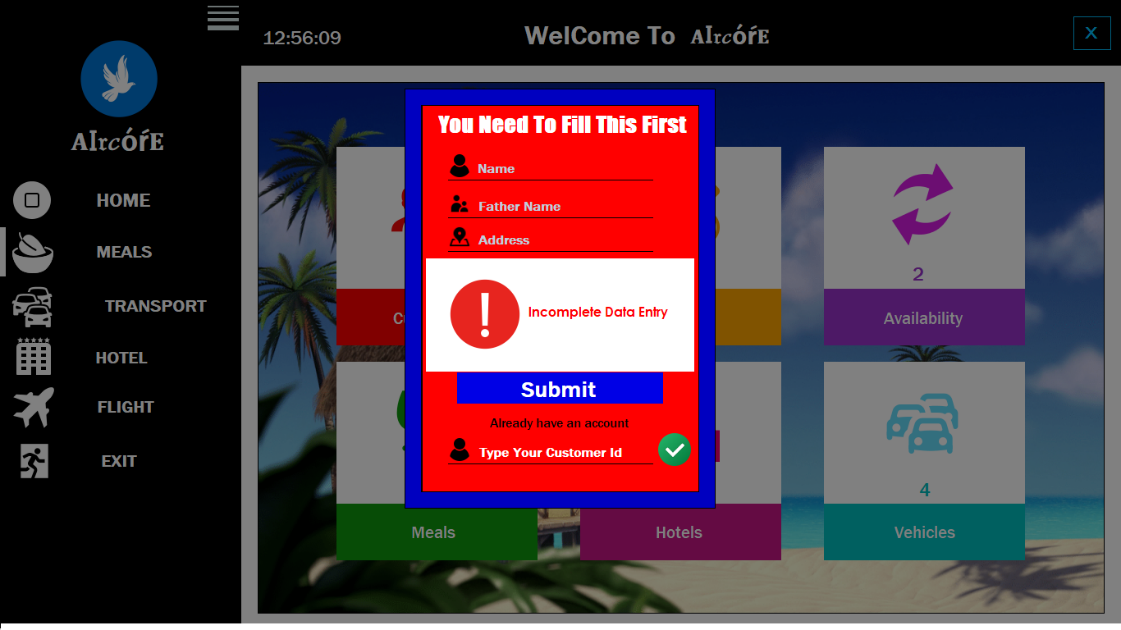
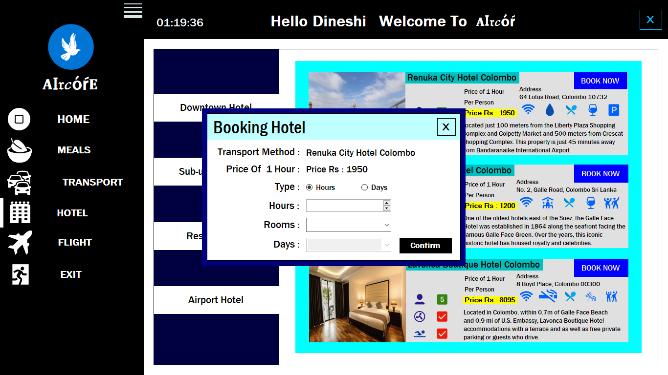
 Figure 44 Testing Customer Register 1

Figure 45 Testing Customer Registration

### **6.3. Testing Hotel Booking**

|  |  |  |
| --- | --- | --- |
| **Test** | **Check in Success** | **Check in Failed** |
| **Input correct records** | **“Booked Successfully !”** | **-** |
| **Input incorrect records** | **-** | **“Incorrect Data Entry “** |

**** Table 22 Testing Hotel Booking

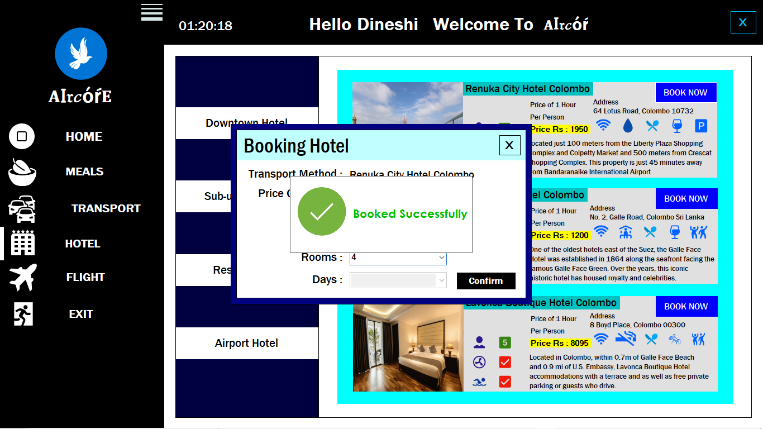
****

Figure 46 Testing Hotel Booking 1

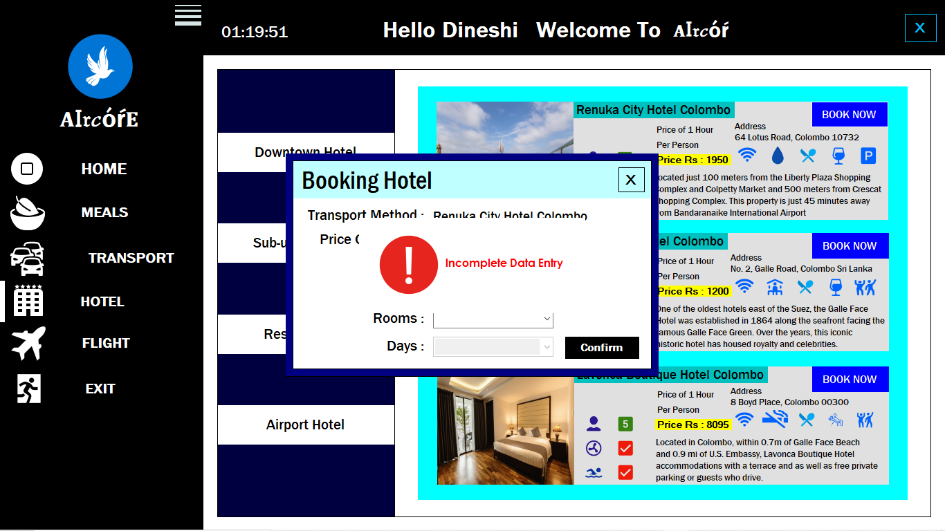
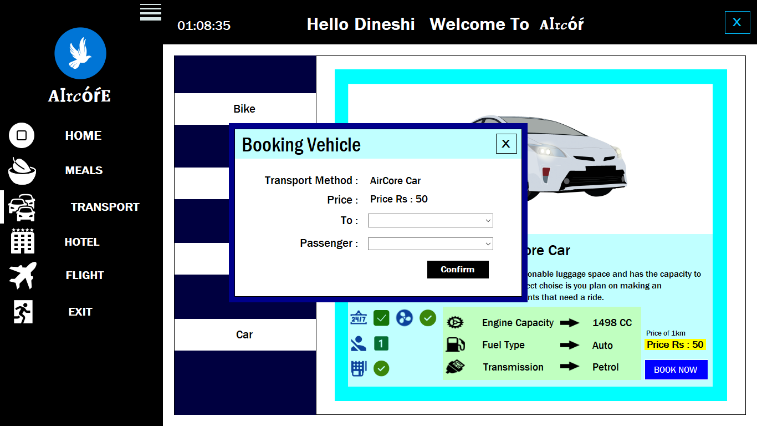
****

Figure 47 Testing Hotel Booking 2

### **6.4. Testing Vehicle Hiring**

|  |  |  |
| --- | --- | --- |
| **Test** | **Check in Success** | **Check in Failed** |
| **Input appropriate records** | **“Booked Successfully !”** | **-** |
| **Input inappropriate records** | **-** | **“Incorrect Data Entry “** |

Table 23 Vehicle Hiring

****

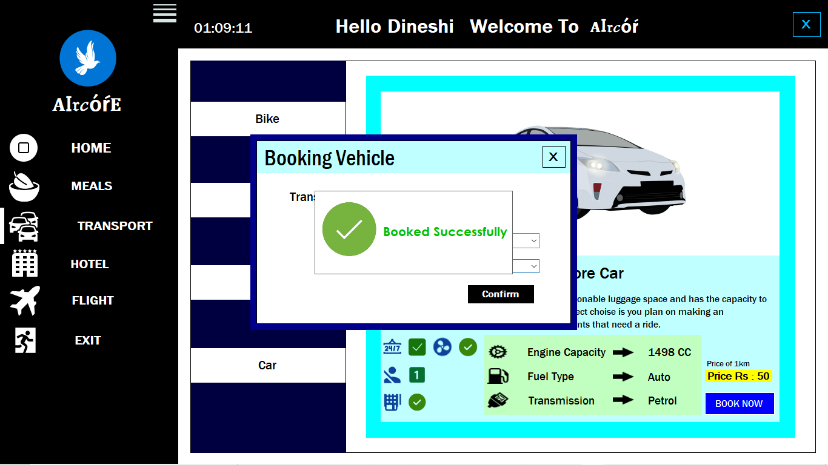


Figure 48 Testing Vehicle Hiring

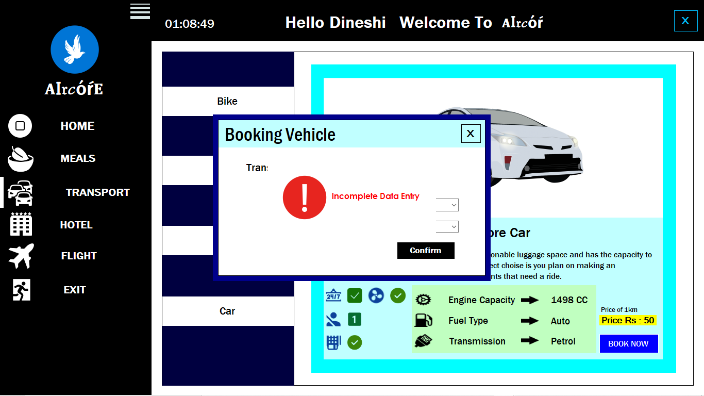
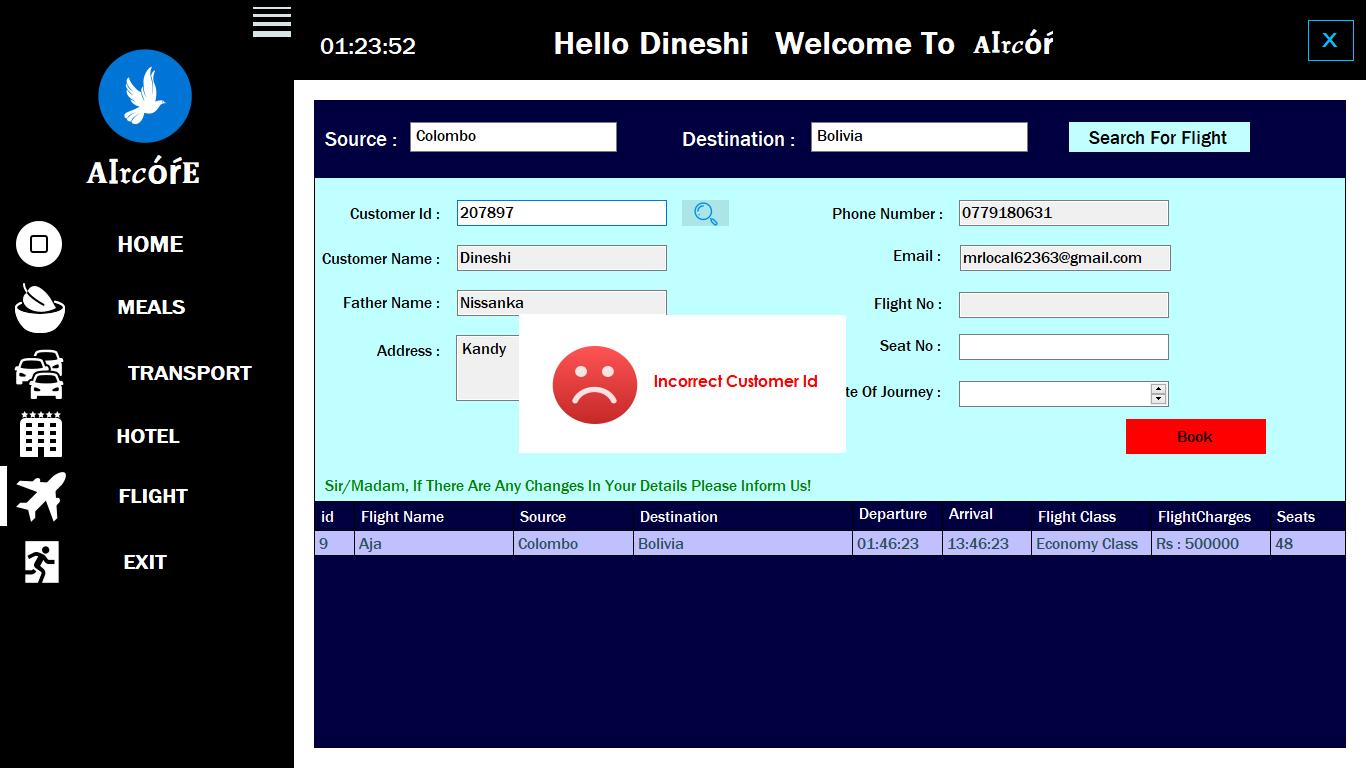
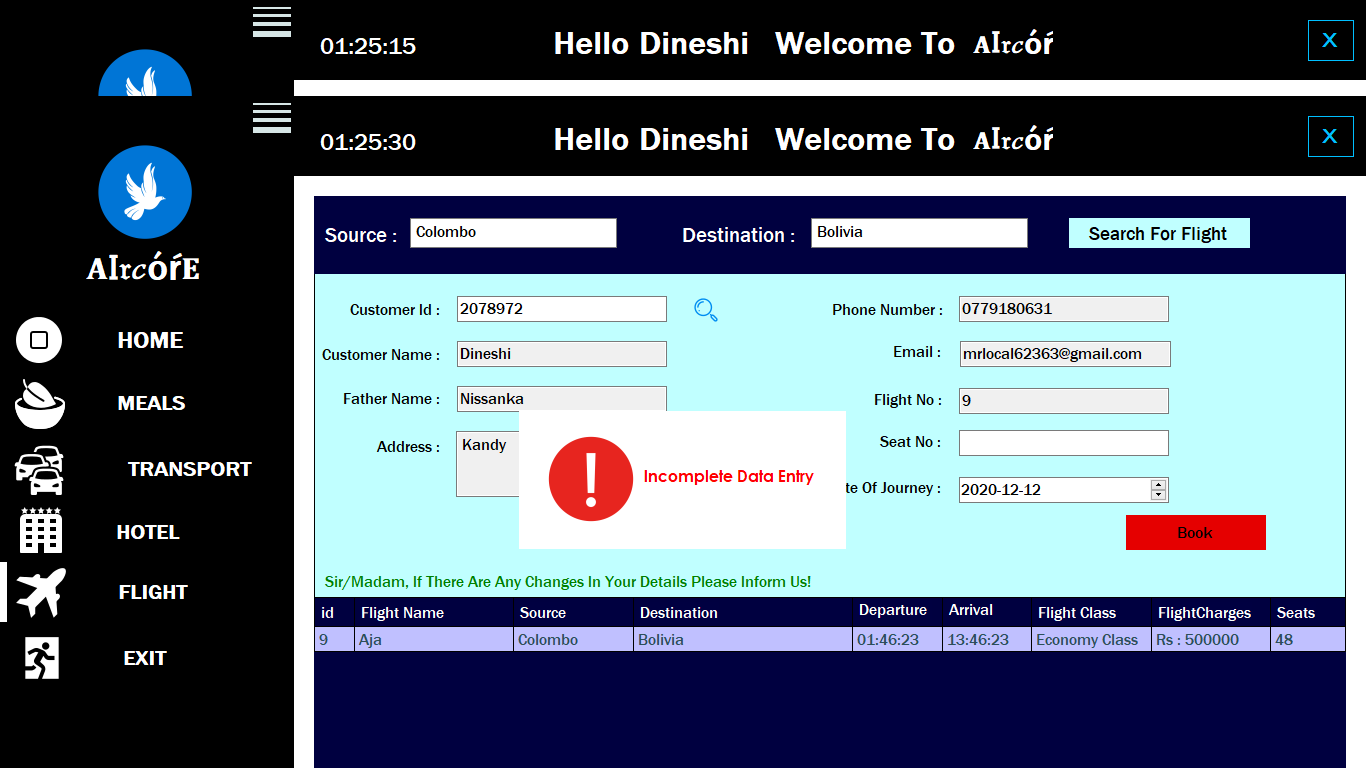
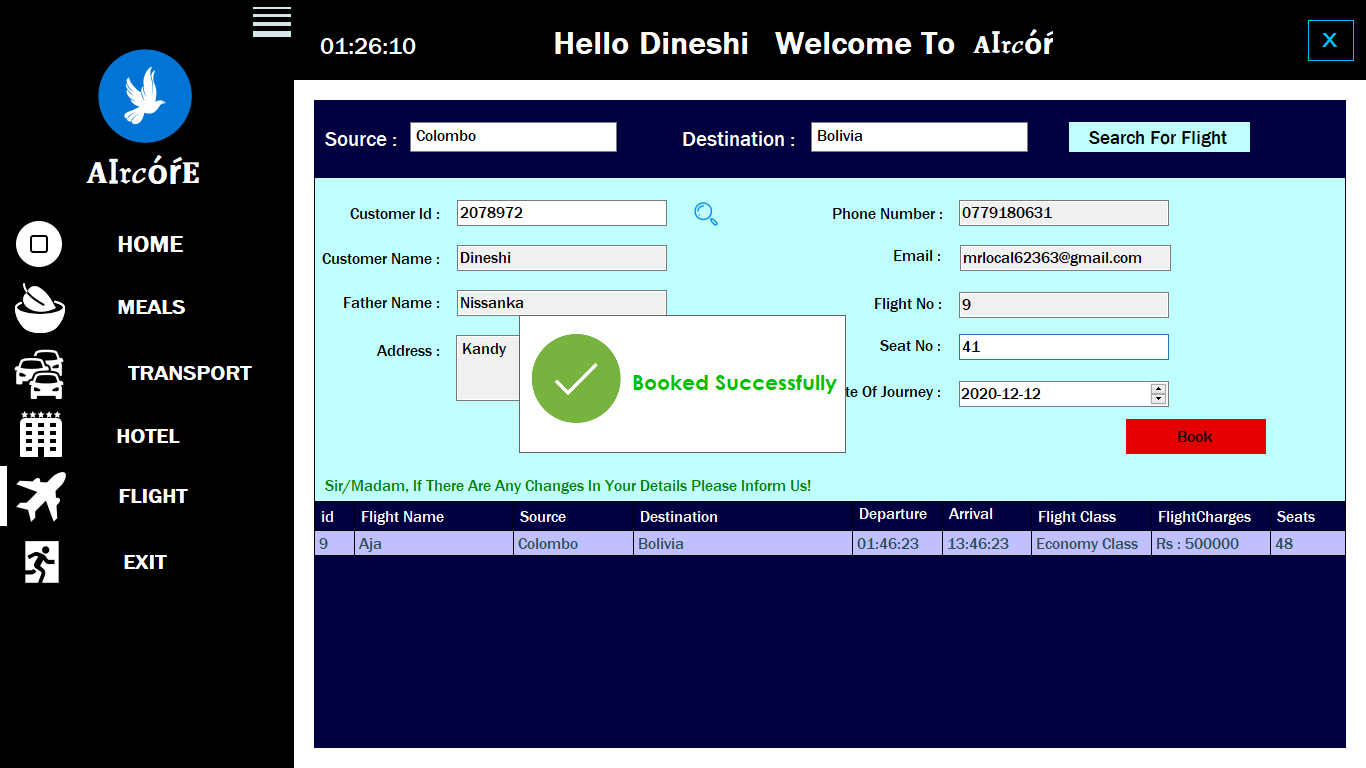
****

Figure 49 Testing Vehicle Hiring 2

### **6.5. Testing Flight Options**

****

****

****

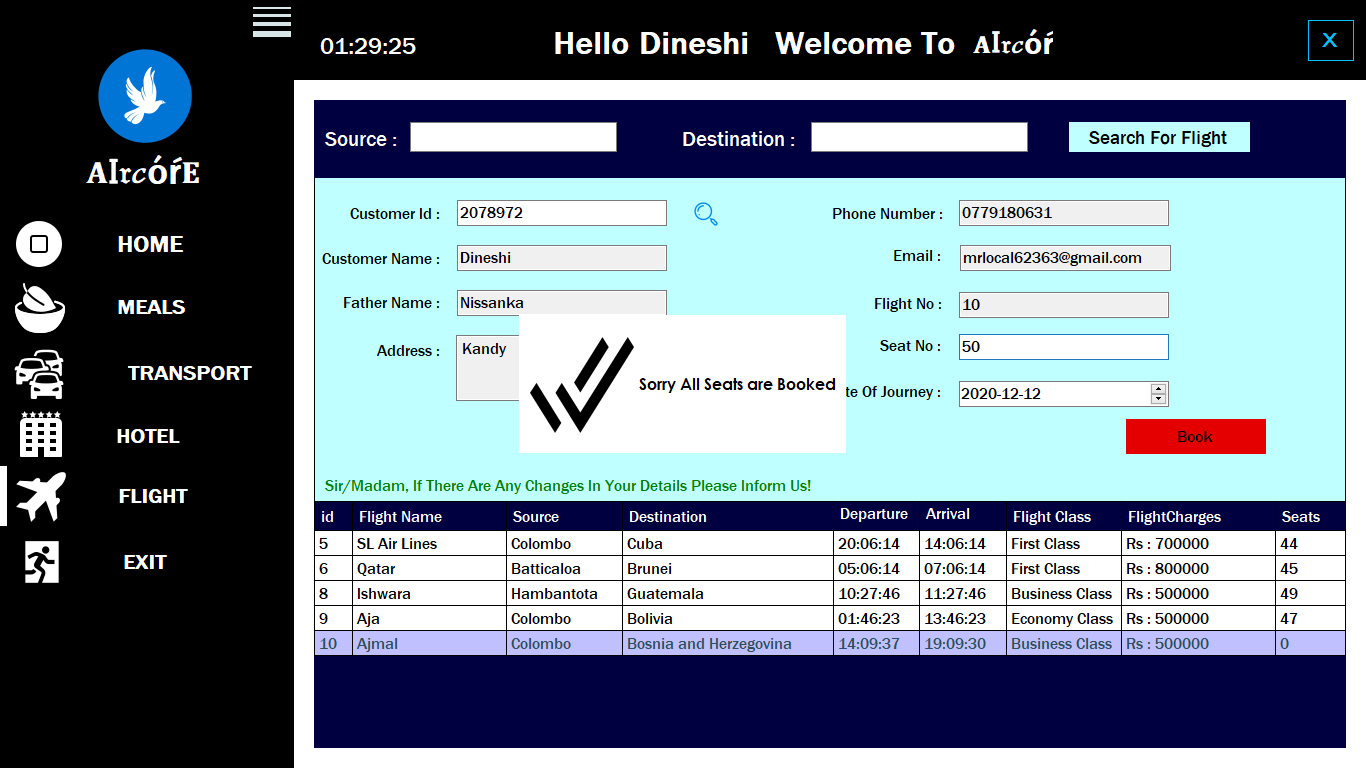
****

Figure 50 Testing Flight Option

# **7. Conclusion**

To conclude the description about the project. The project developed using C#.Net and MySQL is based on the requirement specification of the user and the analysis of the existing system with flexibility and adaptability for the future enhancement. The expanded functionality of the present software requires an appropriate approach towards software development.

Identification of the drawbacks of the existing system leads to the design of automated computerized system that will be compatible to the existing system with the system which is more user-friendly and GUI oriented.

Finally, this Ticket Management System can perform flexibly and efficiently while operate the software and more adaptability with future improvements and enhancements.

# **8.Future Scope**

There are many additional features, which can be planned to be incorporated during the future enhancement of this software. This is a Stand-Alone Ticket Management System; the future version of this software can be a web-based system for ticketing services. At the future version a passenger can surf the flight details through internet and allocate a flight ticket for themselves by register themselves to the ticketing agency. By registering themselves, they will get a username and password for their login. Therefore, passenger can visit to the system whenever and view their flight ticket allocation records.

The proposed software can perform very efficiently, as well as in the future described features can be added to this software.

# **9.References**

G. Rajkumar , T. Sivagama Sundari. (2019). Ticket Management System Based on Finger Print Authentication. *computerscijournal*.

Gutierrez, P. J. (2007, July 12). *https://patents.google.com/.* Retrieved from https://patents.google.com/: https://patents.google.com/

MUYESHI, P. (2009, April 9). *https://www.academia.edu/31425862/TICKET\_MANAGEMENT\_SYSTEM\_Submitted\_by*. Retrieved from https://www.academia.edu/: https://www.academia.edu/

panelNorazahMohd , SukiaNorbayahMohd Sukib. (2017). Flight ticket booking app on mobile devices: Examining the determinants of individual intention to use. *Elsevier*.

Patrick J. TooleSudhir KrishnaswamyNima MoayediDavid N. LordFrancisco J. Gutierrez. (2007). *https://patents.google.com/.* Retrieved from https://patents.google.com/

Sujana, A. (2013). *r Trincomalee Campus*. Retrieved from http://hdl.handle.net: http://hdl.handle.net/123456789/696

Yatera, J. (2014). *Ticket Processes management draft report.* Computer Sciencee.