

---

# **SOFTWARE REQUIREMENTS SPECIFICATION**

**For**

**Airline Reservation System**

**Prepared by:-**

A. Sam Meshach-22AD078

Kk .Ravi Shankar-22AD076

R.Vishal-22AD123

# **1. Introduction**

## **1.1 Purpose**

The purpose of the Airline Reservation System is to provide a comprehensive and efficient solution for managing flight bookings, passenger information, and ticketing processes within an airline. The system aims to streamline the reservation management workflow for airline staff, offering a user-friendly interface for creating, modifying, and canceling flight reservations.

## **1.2 Document Conventions**

- Entire document should be justified.
- Convention for Main title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 14
- Convention for Sub title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 12
- Convention for body
  - Font face: Times New Roman
  - Font Size: 12

## **1.3 Scope of Development Project**

The Airline Reservation System development project entails the creation of a sophisticated and user-centric platform designed to optimize the workflow of airline staff and enhance the overall passenger experience. The system's scope encompasses a myriad of features, starting with robust user authentication and authorization mechanisms to ensure secure access for different roles within the airline, such as administrators and booking agents. The core functionalities include modules for efficient flight booking and ticketing, seat booking, and cancellation rules that consider timing, penalties, and rebooking options.

A key aspect is the meticulous management of passenger information, with a centralized database capturing and updating details while adhering to data privacy regulations. Real-time seat availability checks are integrated to provide instant information to airline staff, with a visually intuitive interface displaying seat layouts. The system also boasts an information-rich module for detailed flight information, offering insights into departure and arrival times, duration, and airline specifics. Integration with a reliable Database Management System (DBMS) ensures efficient data storage and retrieval, with careful consideration given to scalability and performance. The development project places a high emphasis on adherence to industry standards, necessitating ongoing compliance with regulations and proactive adjustments to the system. Scalability and future enhancements are integral components of the project, ensuring the system's adaptability to evolving industry requirements.

User documentation and training materials are developed comprehensively to equip users with the necessary knowledge. Rigorous testing and quality assurance procedures, coupled with robust security measures, round out the project's scope, aiming to deliver a secure, reliable, and user-friendly Airline Reservation System that meets the diverse needs of airline stakeholders and passengers alike

## 1.4 Definitions, Acronyms and Abbreviations

JAVA -> platform independence  
SQL-> Structured query Language  
ER-> Entity Relationship  
UML -> Unified Modeling Language  
IDE-> Integrated Development Environment  
SRS-> Software Requirement Specification

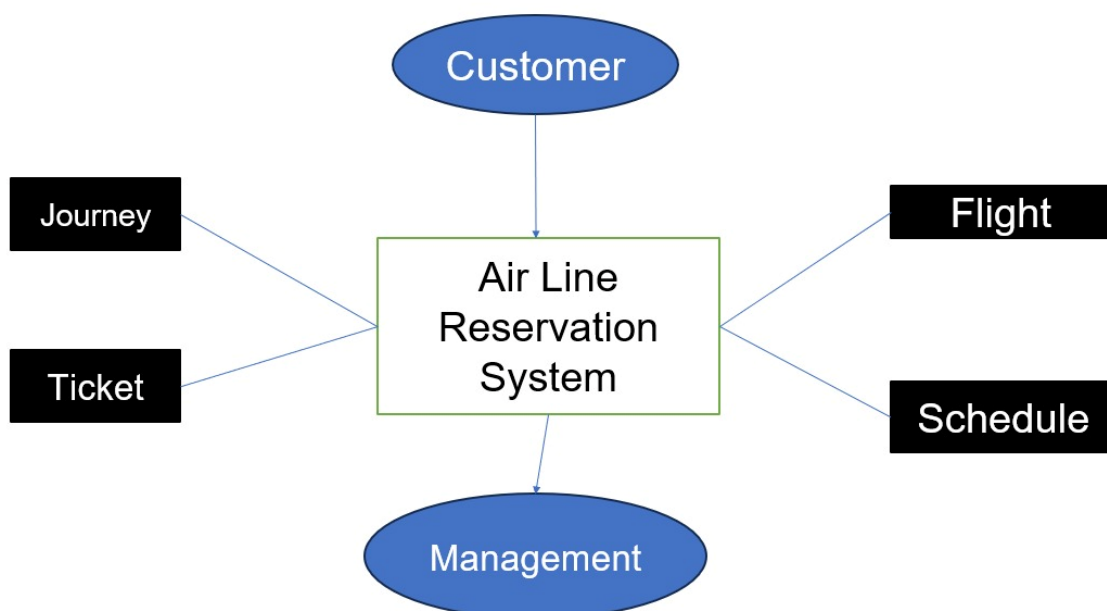
## 1.5 References

- Amadeus. (2010). *Amadeus / About*. Retrieved from <https://amadeus.com/en/about>
- American Airlines "Magnetronic Reservisor". American Airlines C.R. Smith Museum. Retrieved 3 August 2014.
- Beal, V (2018). Database Retrieved March 4, 2019, from <https://www.webopedia.com/TERM/D/database.html>
- Bourgeois, D. (2014). Information Systems Development. In *Information Systems for Business and Beyond* [PDF] (2nd ed., p. 104). Retrieved from <https://bus206.pressbooks.com/chapter/chapter-10-information-systems-development/>
- Businessdictionary (2012) <http://www.businessdictionary.com/definition/Global-DistributionSystem-GDS.html> accessed on May 26, 2012
- BusinessDictionary.com. Retrieved May 30, 2019, from BusinessDictionary.com website: <http://www.businessdictionary.com/definition/guaranteed-reservation.html>

## 2. Overall Descriptions

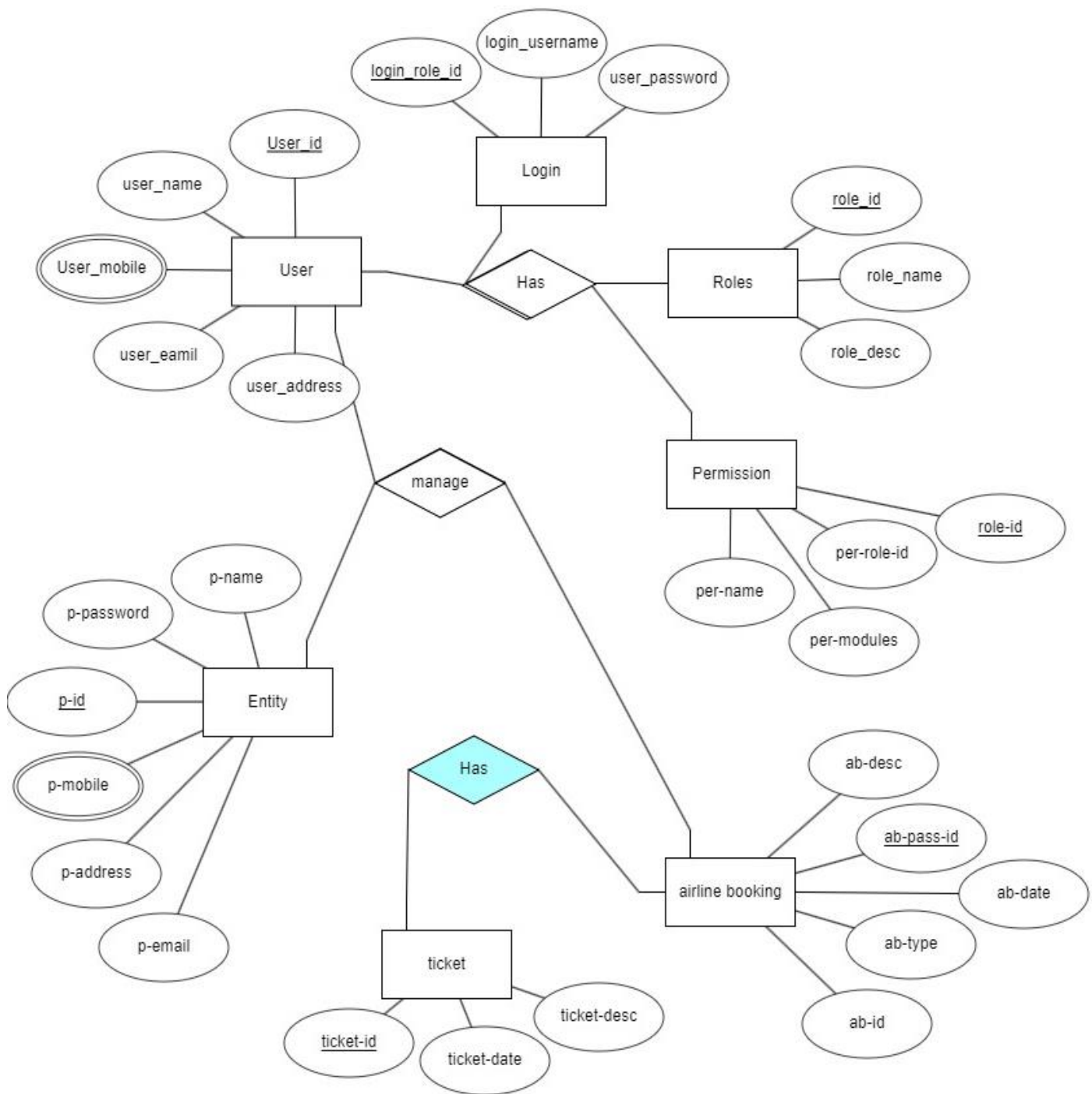
### 2.1 Product Perspective

Use Case Diagram of Library Management System



## 2.2 Product Function

Entity Relationship Diagram of Airline Reservation System



## **2.3 User Classes and Characteristics**

The Airline Reservation System caters to two distinct user classes: airline staff and passengers. Admin can issue and manage flight reservations, view available flights, maintain passenger records, and access comprehensive reports. Admin possess the authority to add new flights, modify existing flight information, and monitor the overall status of reservations and ticketing.

Additionally, they can interact with the database to ensure accurate and up-to-date information. On the other hand, Members, who are primarily passengers, engage with the system to explore available flights, make reservations, and manage their travel details. Members have the capability to view flight options, check seat availability, and confirm bookings. They can also access their reservation history, update personal information, and receive real-time notifications about their flights. The system empowers Members to navigate seamlessly through the booking process, enhancing their overall travel experience.

## **2.4 Operating Environment**

The product will be operating in windows environment. The Airline Reservation System is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox .Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

## **2.5 Assumptions and Dependencies**

Some possible assumptions and dependencies for an airline reservation system are:

- Assumptions are the conditions that are expected to be true for the system to function properly. Some examples of assumptions are:
  - The users of the system have a computer with internet access.
  - The system can communicate with the databases of different airlines and follow their business rules.
  - The system can access credit and debit card authorization services.
  - The system is compatible with the computer systems on which it has been loaded for customer use.

- Dependencies are the relationships between the system and other systems or components that affect its functionality. Some examples of dependencies are:
  - The system depends on the availability and reliability of the internet connection.
  - The system depends on the accuracy and timeliness of the flight information from the airlines.
  - The system depends on the security and privacy of the user data and transactions.
  - The system depends on the hardware and software interfaces with other systems, such as flight inventory, user query, and administrator interface.

## **2.6 Requirement**

### **Software Configuration:-**

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP

Language: Java Runtime Environment, Net beans 7.0.1 (front end)

Database: MS SQL Server (back end)

### **Hardware Configuration:-**

Processor: Pentium(R)Dual-core CPU

Hard Disk: 40GB

RAM: 256 MB or more

## **2.7 Data Requirement**

The data requirements for an airline reservation system are the inputs and outputs that the system needs to function properly. The data requirements can be divided into two categories: functional and non-functional.

Functional data requirements are the data that the system needs to perform its main functions, such as searching, booking, and canceling flights. Some examples of functional data requirements are:

- User information: The system needs to collect and store the user's personal and contact details, such as name, email, phone number, and address. The system also needs to verify the user's identity and payment method, such as credit card or debit card.

- Flight information: The system needs to access and update the flight information from different airlines, such as flight number, departure and arrival time, origin and destination, ticket price, seat availability, and baggage allowance

- Reservation information: The system needs to generate and manage the reservation information for each user, such as reservation number, flight details, passenger details, payment details, and confirmation status.

Non-functional data requirements are the data that the system needs to meet the quality and performance standards, such as security, reliability, and usability. Some examples of non-functional data requirements are:

- Security: The system needs to protect the user data and transactions from unauthorized access, modification, or deletion. The system also needs to encrypt the data and use secure protocols for communication.
- Reliability: The system needs to ensure the accuracy and timeliness of the data and functions. The system also needs to handle errors and exceptions gracefully and provide backup and recovery mechanisms.
- Usability: The system needs to provide a user-friendly and intuitive interface for the users. The system also needs to support multiple languages and currencies and provide help and feedback features.

### **3. External Interface Requirement**

#### **3.1 GUI**

The GUI (Graphical User Interface) is the external interface that allows the users to interact with the airline reservation system. The GUI should be user-friendly, intuitive, and consistent. The GUI should also provide feedback and help features for the users. Some of the GUI requirements are:

- The GUI should have a home page that displays the name and logo of the system, a menu bar with different options, and a search engine for finding flights.
- The GUI should have a login page that allows the users to enter their username and password, or create a new account if they are not registered.
- The GUI should have a registration page that allows the users to enter their personal and contact details, such as name, email, phone number, and address. The GUI should also validate the user input and generate a unique passenger ID for each user.
- The GUI should have a flight details page that displays the flight information from different airlines, such as flight number, departure and arrival time, origin and destination, ticket price, seat availability, and baggage allowance. The GUI should also allow the users to filter and sort the flights based on their preferences.
- The GUI should have a booking page that allows the users to select a flight, enter their passenger details, and confirm their reservation. The GUI should also generate a reservation number and a confirmation status for each booking.
- The GUI should have a cancellation page that allows the users to cancel their reservation by entering their reservation number and passenger ID. The GUI should also update the seat availability and the user account accordingly.
- The GUI should have a user account page that allows the users to view and edit their profile, change their password, and see their booking history. The GUI should also allow the users to print or download their tickets and invoices.
- The GUI should have an about page that displays the information about the system, such as its purpose, features, developers, and contact details. The GUI should also provide a feedback form for the users to rate and comment on the system.

## 4. System Features

System features are the high-level capabilities and functions that the system provides to the users and stakeholders. System features describe what the system can do and how it can benefit the users. Some of the possible system features for an airline reservation system are:

- Flight search: The system allows the users to search for flights based on their origin, destination, date, time, and preferences. The system displays the flight information from different airlines, such as flight number, departure and arrival time, ticket price, seat availability, and baggage allowance. The system also allows the users to filter and sort the flights based on their criteria, such as price, duration, stops, etc.
- Flight booking: The system allows the users to book a flight by selecting a flight, entering their passenger details, and confirming their reservation. The system generates a reservation number and a confirmation status for each booking. The system also verifies the user's identity and payment method, such as credit card or debit card, and processes the payment securely.
- Flight cancellation: The system allows the users to cancel their reservation by entering their reservation number and passenger ID. The system updates the seat availability and the user account accordingly. The system also refunds the payment based on the cancellation policy of the airline.
- User account: The system allows the users to create and manage their user account. The system stores the user's personal and contact details, such as name, email, phone number, and address. The system also allows the users to view and edit their profile, change their password, and see their booking history. The system also allows the users to print or download their tickets and invoices.
- Customer service: The system provides customer service features for the users and the stakeholders. The system allows the users to contact the system administrators or the airline representatives for any queries or issues. The system also provides a feedback form for the users to rate and comment on the system. The system also provides an about page that displays the information about the system, such as its purpose, features, developers, and contact details.

## 5. Other Non-functional Requirements

### 5.1 Performance Requirement

The performance requirements for an airline reservation system are the non-functional requirements that specify the quality and efficiency of the system's functions. Some examples of performance requirements are:

- The system should respond to user queries and requests within a reasonable time, such as a few seconds or minutes, depending on the complexity and volume of the data.
- The system should handle multiple concurrent users and requests without compromising the accuracy and reliability of the data and functions.
- The system should have a high availability and uptime, such as 99.9%, and minimize the downtime and maintenance periods.
- The system should have a high scalability and adaptability, such as being able to handle increasing or decreasing demand and load, and being able to integrate with new or existing systems and technologies.



## **5.2 Safety Requirement**

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

## **5.3 Security Requirement**

- System will use secured database
- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints
- Proper user authentication should be provided
- No one should be able to hack users' password
- There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

## **5.4 Requirement attributes**

- There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
- The project should be open source
- The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
- The user be able to easily download and install the system

## **5.5 Business Rules**

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## **5.6 User Requirement**

The admin provides certain facilities to the users in the form of:-

- Backup and Recovery
- Forgot Password
- Data migration i.e. whenever user registers for the first time then the data is stored in the server
- Data replication i.e. if the data is lost in one branch, it is still stored with the server
- Auto Recovery i.e. frequently auto saving the information
- Maintaining files i.e. File Organization
- The server must be maintained regularly and it has to be updated from time to time

## 6. Other Requirements

### 6.1 Data and Category Requirement

There are different categories of users namely teaching staff, Librarian, Admin, students etc. Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can be able to modify the data ,delete, append etc. All other users except the Librarian only have the rights to retrieve the information about database. Similarly there will be different categories of books available. According to the categories of books their relevant data should be displayed. The categories and the data related to each category should be coded in the particular format.

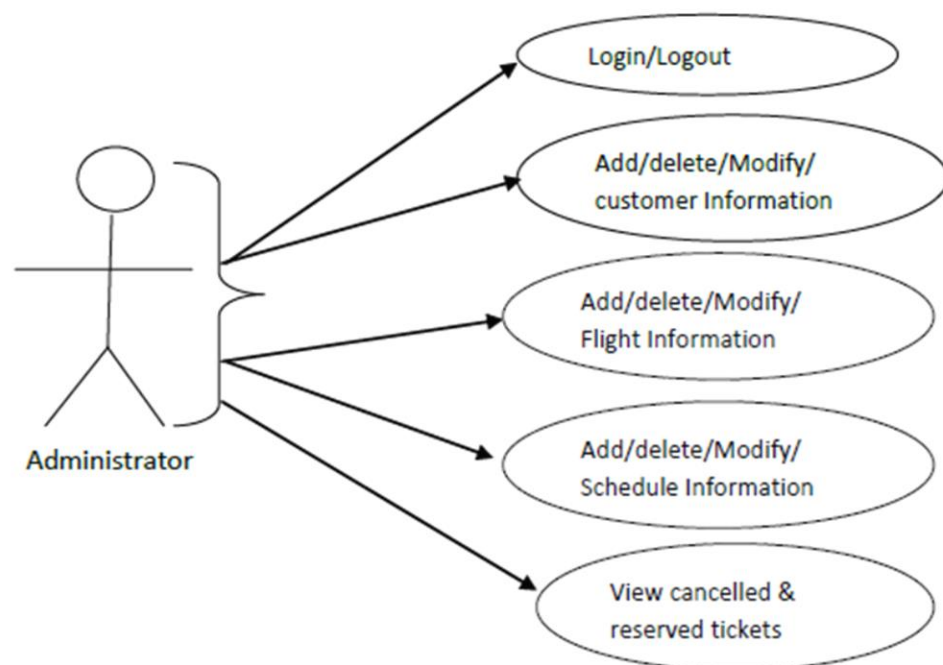
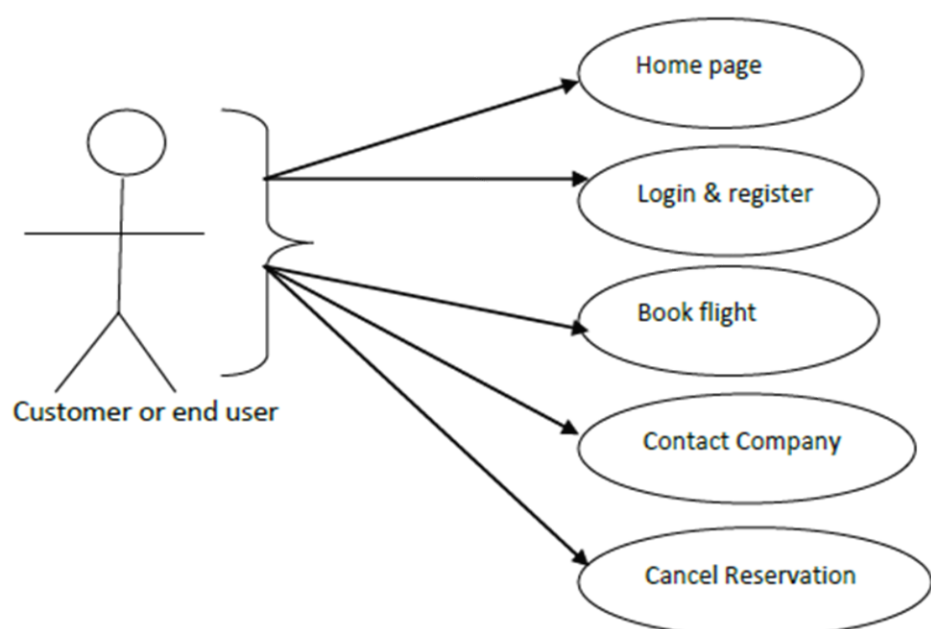
### 6.2 Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

- Administrator: A login id representing a user with user administration privileges to the software
- User: A general login id assigned to most users
- Client: Intended users for the software
- SQL: Structured Query Language; used to retrieve information from a database
- SQL Server: A server used to store data in an organized format
- Layer: Represents a section of the project
- User Interface Layer: The section of the assignment referring to what the user interacts with directly
- Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
- Data Storage Layer: The section of the assignment referring to where all data is recorded
- Use Case: A broad level diagram of the project showing a basic overview
- Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system's cases, their attributes, and the relationships between the classes
- Interface: Something used to communicate across different mediums
- Unique Key: Used to differentiate entries in a database

### 6.3 Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes' structure and their relationships to each other frozen in time represent the static model.



**Figure 3.5: Class Diagram**

