**SOFTWARE**

**REQUIREMENTS**

**SPECIFICATION**

**For**

**Airline Reservation System**

**Prepared by: -**

*Jayasudha T*

*Kanishkasree R*

*Srimathi Priya M*

1. **Introduction:**
   1. **Purpose:**

The main objective of this document is to illustrate the requirements of the project Airline Reservation System. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The purpose of an airline reservation system is to streamline and automate the process of booking and managing flight reservations. It enables passengers to check seat availability, select flights, and make reservations through various channels. The system helps airlines efficiently manage flight schedules, allocate seats, and update fare information. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

* 1. **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: 14
* Convention for body
* Font face: Times New Roman
* Font style: Bold
  1. **Scope of Development Project:**

The airline reservation system is a web-based flight booking engine that is used to conduct flight bookings. It involves Airline schedules, passenger reservations and Passenger Name Records. This Project is basically an interface to carry out reservations on the provided airline from any place. With the help of this system, customers can view all the different flight’s availability with different timings for a particular date and it also allows them to reserve a seat, cancel a reservation or modify it.

The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules. The language used for developing the project is Java as it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost, and development process.

* 1. **Definitions, Acronyms and Abbreviation:**

JAVA -> Platform Independence

SQL -> Structured Query Language

ER -> Entity Relationship

UML -> Unified Modeling Language

IDE -> Integrated Development Environment

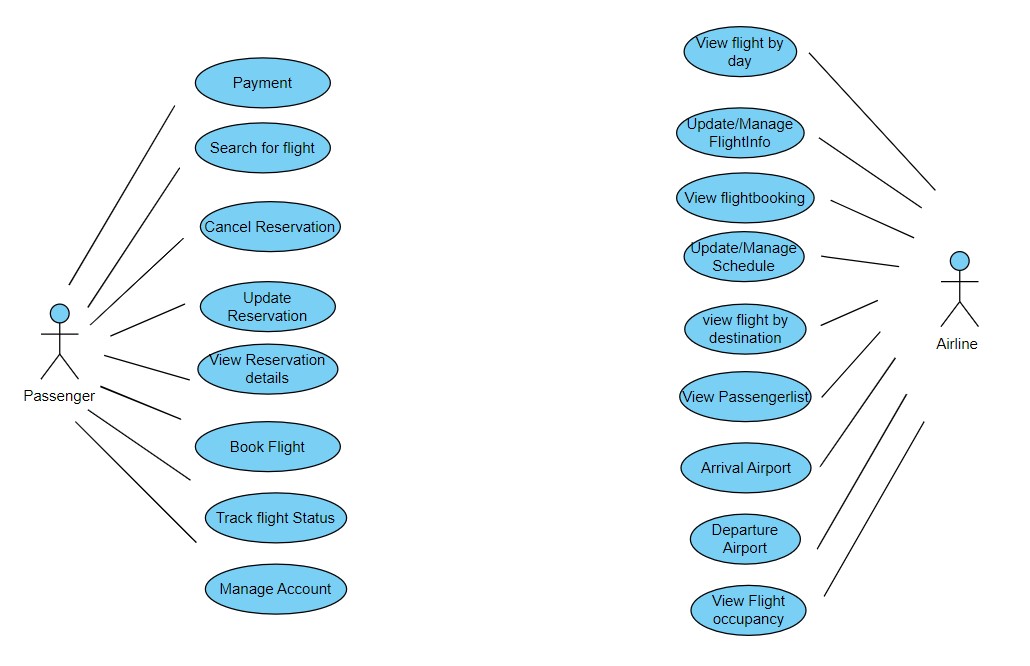
SRS -> Software Requirement Specification

**1.5 References:**

* Books
* Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson
* Software Requirements (Microsoft) Second Edition By Karl E. Wiegers
* Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman
* Websites
* [**https://www.oneclickitsolution.com/blog/airline-reservation-system/**](https://www.oneclickitsolution.com/blog/airline-reservation-system/)
* [**https://www.slideshare.net/MostafaMorsyMohamed/airline-reservation-system-project-report-1**](https://www.slideshare.net/MostafaMorsyMohamed/airline-reservation-system-project-report-1)

1. **Overall Descriptions:**
   1. **Product Perspective:**

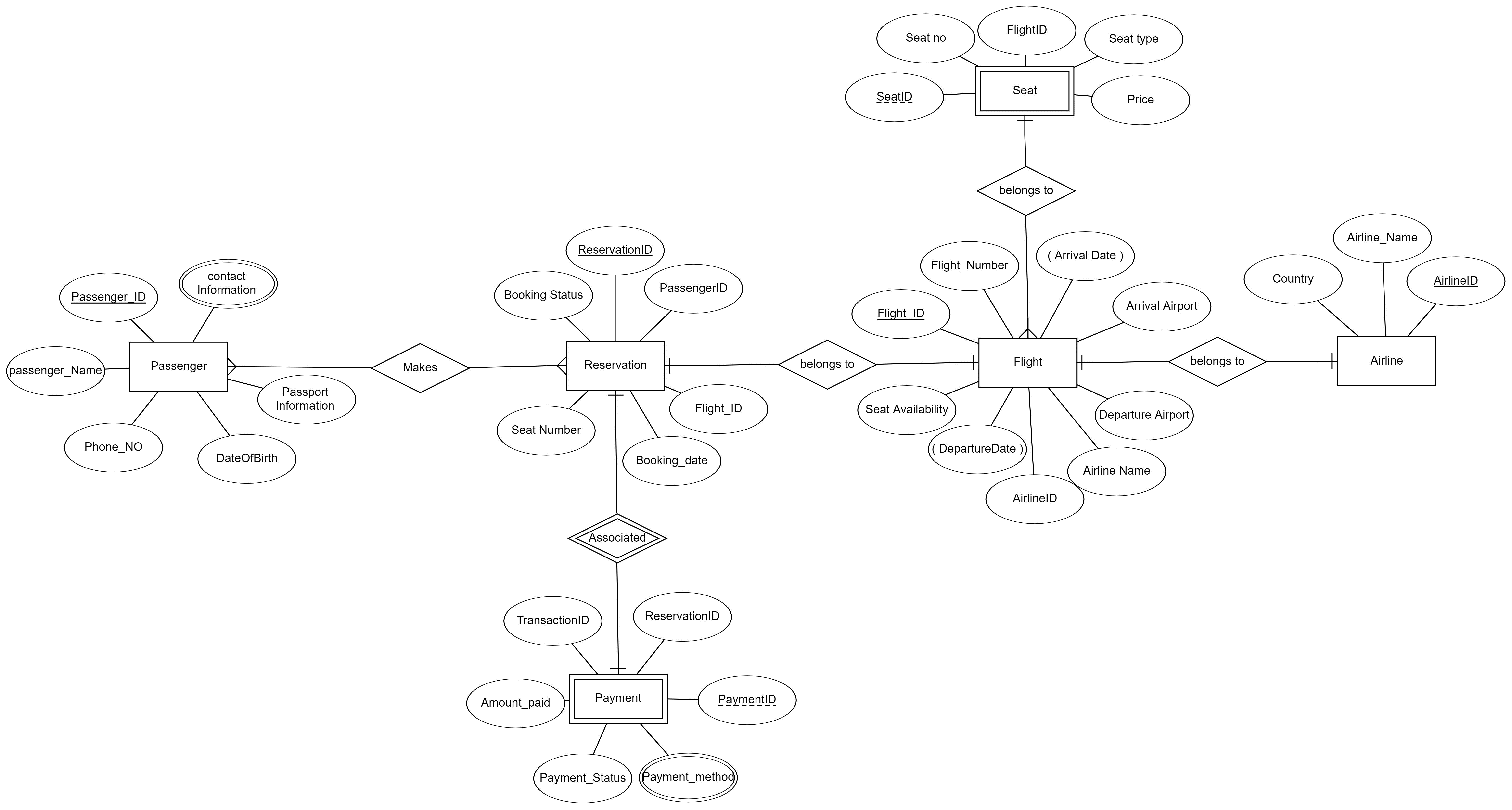
Use Case Diagram of Airline Reservation System



A use case diagram for an airline reservation system illustrates the interactions between various actors and the system's functionalities. It provides a visual representation of the system's scope and the key roles involved in its operation. Airline Reservation System's use case diagram highlights the essential functionalities of an airline reservation system, enabling users to manage their travel arrangements efficiently and administrators to maintain system operations effectively.

* 1. **Product Function:**

Entity Relationship Diagram of Airline Reservation System



The Entity Relationship diagram for an Airline Reservation System visually depicts the connections between entities like Passenger, Reservation, Payment and Airline Booking. These entities interact through relationships defined by cardinality and participation constraints. The Passenger entity represents individuals seeking air travel, while the Payment entity represents confirmed reservations. The Airline Booking entity centralizes reservation management. Cardinality and participation constraints further define these interactions. A passenger can have zero or more tickets, while a ticket must belong to one passenger. Every ticket must be associated with a valid passenger. The diagram provides a blueprint for designing a structured database for an airline reservation system.

* 1. **User Classes and Characteristics:**
* Administrator
* Create and manage user accounts:Administrators can create new user accounts for travel agents and customers. They can also edit or delete existing user accounts.
* Add and edit flights: Administrators can add new flights to the system and edit existing flights. They can also set the fares for each flight.
* Manage bookings: Administrators can view, modify, or cancel bookings. They can also refund tickets.
* View system reports: Administrators can view reports on system activity, such as the number of bookings made and the revenue generated.
* Travel Agent
* Book flights: Travel agents can book flights for customers. They can select the flight, the number of passengers, and the seating arrangements.
* View customer information: Travel agents can view customer information, such as their name, contact information, and passport details.
* Manage customer bookings: Travel agents can view, modify, or cancel customer bookings. They can also issue tickets.
* Customer
* Book flights: Customers can book flights for themselves or for others. They can select the flight, the number of passengers, and the seating arrangements.
* View booking history: Customers can view their booking history, including the flight details, the fare paid, and the booking status.
* Manage personal information: Customers can view and update their personal information, such as their name, contact information, and passport details.
  1. **Operating Environment:**

The airline reservation system's operating environment encompasses the hardware, software, networking, and performance considerations necessary for its seamless operation. Hardware requirements for the server include a multi-core processor with at least 4 cores and 8 GB RAM, 16 GB or more RAM, 250 GB SSD or more storage, and a 1 Gigabit Ethernet or faster network interface. Clients require the latest versions of Chrome, Firefox, Safari, or Edge web browsers and Windows 10, macOS 10.15 or later, or Linux Ubuntu 18.04 LTS or later operating systems. Software requirements for the server include Windows Server 2019 or later, Linux Ubuntu 18.04 LTS or later, Apache Tomcat or equivalent web application server, MySQL database and Java programming Language. Clients require the latest versions of Chrome, Firefox, Safari, or Edge web browsers.

* 1. **Assumptions and Dependencies:**

The assumptions are: -

* The coding should be error free
* The system should be user-friendly so that it is easy to use for the users
* The users should have access to the internet and a compatible device
* The Payment processing is secure and reliable
* External systems are functional and accessible
* Communication channels are open and monitored
* The system should have more storage capacity and provide fast access to the database
* Users must have their correct username and passwords to enter into their online accounts and do actions

The dependencies are: -

* + - The specific hardware and software due to which the product will be run
    - On the basis of listing requirements and specification the project will be developed and run
    - Data security measures
    - System administration and maintenance
    - Regulations and compliance
  1. **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

* 1. **Data Requirement:**

Data requirements for an airline reservation system encompass flight, passenger, booking, and additional information. Flight data includes details like flight number, airports, times, aircraft type, seat availability, and fares. Passenger data comprises name, contact information, passport details, and frequent flyer number. Booking data involves booking number, passenger name, flight number, seat number, fare, and payment information. Additional data may include airport, airline, travel agent, and historical booking information. Data storage and retrieval typically utilize a relational database with efficient data retrieval mechanisms. Data security employs encryption and access control, while data integrity involves data validation and backups. Specific data requirements for common use cases include flight searching, booking, viewing booking history, and managing personal information. Administrator, travel agent, and customer roles have distinct data access privileges.

1. **External Interface Requirement:**
   1. **GUI:**

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as Create, Modify, viewing the details of the flight.

User Interface: -

The user interface should be designed to be simple, intuitive, and easy to navigate. It should be consistent across all screens and follow a standard template. The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module.

Login Interface: -

The login interface should be prominently displayed on the homepage. Users should be able to enter their username and password to log in. If the user enters their username or password incorrectly, an error message should appear. Users should be able to reset their password if they have forgotten it.

Search: -

The search function should be easily accessible from the homepage. Users should be able to search for flights. The search results should display a list of available flights. Users should be able to select a flight from the search results and click on the "Book" button. Users should be able to enter their passenger information, select their seats, and enter their payment information. Once the user has completed all of the required information, they should click on the "Book" button to confirm their booking.

My Account: -

Users should be able to access their My Account page from the homepage. The My Account page should display the following information User's personal information, Booking history, Past searches and Administrator Interface. The administrator interface should be separate from the user interface and require a login

1. **System Features:**

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

* User data should be treated with utmost confidentiality and protected from unauthorized access, misuse, or disclosure.
* Users should be informed about data collection practices and have control over their personal information.
* The system should be transparent in its operations and accountable for any data breaches or security incidents.
* Proper accountability which includes not allowing a member to see other member’s account. Only administrator will see and manage all member accounts

1. **Other Non-Functional Requirements:**
   1. **Performance Requirement:**

The proposed system that we are going to develop will be used as the Chief performance system. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the client.

* The system should be able to handle a large number of concurrent users without performance degradation
* The system should be able to respond to user requests quickly and efficiently
* The system should be highly available and resilient to failure
* The system should be secure and protect user data from unauthorized access
* The system should have a comprehensive performance monitoring system in place
  1. **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
* The system should be able to maintain transaction processing throughput under peak load conditions
  1. **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
  1. **Business Rule:**

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.

* 1. **User Requirement:**

The users of the airline reservation system are passengers, travel agents, and administrators. Passengers use the system to search for flights, book flights, and manage their bookings. Travel agents use the system to book flights on behalf of their clients. Administrators use the system to manage user accounts, add and edit flights, manage bookings, and view system reports. The system should be designed to be easy to use for all users, including those with limited technical knowledge. The system should provide a user-friendly interface, a comprehensive user manual, online help, and a guide to install and maintain the system.

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

1. **Other Requirements:**
   1. **Data and Category Requirement:**

There are different categories of users namely Administrator, Booking Agent, Customer 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 Administrator only have the rights to retrieve the information about database. Additionally, fare data is integral, specifying ticket prices for different classes, like economy, business, and first class, along with baggage allowances.

* 1. **Appendix:**

A: Admin, Administrator, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Customer, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features: Travel Agent; U: User, User class and characteristics, User requirement;

* 1. **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
  1. **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. In this project there are certain main classes which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here ‘Passenger’, ‘Reservation’, ‘Booking’ and ‘Payment’ are the most important classes which are related to other classes.

