**SOFTWARE REQUIREMENTS**

**SPECIFICATION**

**for**

**Airline Ticket Booking And Conformation System**

**(ATBCS)**

**Version 1.0 Prepared by**

**Harshith Chowdary Meduri(21CS10042)**

**Vikhyath Gummadi(21CS10028)**

**Nihith Maddi(21CS10040)**

**Contents**

[1 Introduction 5](#_Toc130849706)

[1.1 Purpose 5](#_Toc130849707)

[1.2 Document Conventions 5](#_Toc130849708)

[1.3 Intended Audience and Reading Suggestions 5](#_Toc130849709)

[1.4 Project Scope 6](#_Toc130849710)

[1.5 References 6](#_Toc130849711)

[2 Overall Description 7](#_Toc130849712)

[2.1 Product Perspective 7](#_Toc130849713)

[2.2 Product Functions 7](#_Toc130849714)

[2.3 User Classes and Characteristics 8](#_Toc130849715)

[2.4 Operating Environment 8](#_Toc130849716)

[2.5 Design and Implementation Constraints 9](#_Toc130849717)

[2.6 User Documentation 9](#_Toc130849718)

[2.7 Assumptions and Dependencies 9](#_Toc130849719)

[3 External Interface Requirements 10](#_Toc130849720)

[3.1 User Interfaces 10](#_Toc130849721)

[3.2 Hardware Interfaces 11](#_Toc130849722)

[3.3 Software Interfaces 11](#_Toc130849723)

[3.4 Communications Interfaces 11](#_Toc130849724)

[4 System Features 12](#_Toc130849725)

[4.1 Register a Passenger 12](#_Toc130849726)

[4.2 LOGIN 12](#_Toc130849727)

[4.3 Profile 13](#_Toc130849728)

[4.4 Update Password 13](#_Toc130849729)

[4.5 Booking Flight Ticket 13](#_Toc130849730)

[5 Other Nonfunctional Requirements 15](#_Toc130849731)

[5.1 Performance Requirements 15](#_Toc130849732)

[5.2 Safety Requirements 15](#_Toc130849733)

[5.3 Security Requirements 15](#_Toc130849734)

[5.4 Business Rules 15](#_Toc130849735)

[6 Other Requirements 16](#_Toc130849736)

[6.1 Appendix A: Glossary 16](#_Toc130849737)

[6.2 Appendix B: Analysis Models 16](#_Toc130849738)

# Introduction

## Purpose

The purpose of this project is to develop an application to provide a secure and convenient way to book airplane tickets and also confirm the vaccination status of passengers

## Document Conventions

The following standards document conventions are used in our document: IEEE STD 830 -1998, IEEE Standard for Software Requirement Specification

## Intended Audience and Reading Suggestions

This document is intended to assist the users when they use the software and for developers and the project managers to plan their project and implement the software required.

This Software Requirement Specification document is divided into Six subsections: Section 1: Introduction

Section 2: Overall Description of the Software giving information about functions, user classes, operating environment, constraints and documentation.

Section 3: External Interface Requirements giving a brief introduction to user, hardware, software and communications interfaces.

Section 4: Detailed functional requirements of different features. Section 5: List of non-functional requirements

Section 6: Other requirements

## Project Scope

The main goal of the project is to build an online Airline ticket booking system which allows the user to enter their destination and boarding airports along with the date of travel. Based on the above information the system will show different flights scheduled on that day. The passenger can book ticket as per to his choice and he must upload his vaccination certificate. The system scans the QR code in certificate and verifies it. If the verification was successful he will then proceed to payment gateway. If payment was successful he will get a confirmation from system and the travel distance is then updated.

## References

The documents and websites referred to, are as follows:

* IEEE STD 830 -1998, IEEE Standard for Software Requirement Specification.

# Overall Description

## Product Perspective

The Airline Ticket Booking and Confirmation System is itself a self-contained product, which provides a platform for the passengers to book the airline tickets from available planes and passenger can get discounts based on their past ffm(frequent flier miles).

## Product Functions

The following functions are provided by the software

* Login/register by - Passenger.
* If the Passenger has lost his credentials then he can recover his login details.
* Passenger will get a OTP to his registered email id and on entering the OTP he can reset his password .
* Passenger can select date, place of arrival, place of departure, number of children, adults who will be travelling. A list of available airplanes will be shown.
* Passenger can select the airplane as per his choice.
* The system will show him the total fare along with GST.
* If the Passenger has accumulated ffm then he will avail discount.
* Passenger has to upload his vaccination certificate for verification
* After successful verification passenger can make transaction in provided gateway.
* After successful payment he will get a confirmation e-mail, that his flight ticket has been confirmed

## User Classes and Characteristics

* Passenger
* Create account with login credentials.
* Update Details.
* Book flight tickets
* Upload the vaccination certificate.
* Make payment for ticket
* Flight
* Price(in Rs)
* Carrier
* Time
* Info
* Airport
* Name
* City
* Country
* Code
* Booking
* Booking id
* User
* Flight
* Date
* Departure and Arrival
* Time of booking

## Operating Environment

The software is an online based portal and is thus platform independent and can be used in all well-known browsers.

## Design and Implementation Constraints

* Limited Storage provided in the Mongo Cloud Database.
* Server issues by Stripe API.

## User Documentation

The Webpage is user-friendly and is easy to use.

## Assumptions and Dependencies

* The user must be familiar with the internet and capable to connect to the platform with a web browser.
* No accessibility features are provided on the platform assuming that the disabled users’ device provide those if any such users use this platform.
* Connecting and using the platform requires:
  + working internet connection.
  + compatible browser.

# External Interface Requirements

## User Interfaces

The product will consist of a web-application with which the user will interact. First the user opens the homepage where he should login first and then proceed to Ticket booking. After entering journey details successfully he will be taken to verification page where he should upload his cowin certificate. After successful verification he will be taken to payment gateway and after payment he will get a confirmation mail

* Home Screen
  + Login

∗ Username

∗ Password

* + Register

∗ Username

∗ Email

∗ Password

* + Profile

∗ First name and last name

∗ Mobile

∗ Address

∗ display picture

* + Recovery

∗ OTP

* + Reset

∗ Create new password

* + Ticket Booking

∗ Selection of Arrival and Departure airports

∗ Selection of Travel Date

∗ Entering the number of children, adults, elders

∗ Selection of different classes like economy, business, first class

∗ Submitting cowin certificate.

## Hardware Interfaces

The hardware interfaces for the online portal are listed below:

* The portal can be used on any platform or PC which has a proper internet connection.
* Compatible web browser is required to avoid any problems.

## Software Interfaces

The Software interfaces for the project are listed below:

* The Project will connect to ReactJS for Front-End Responsive Pages and MongoDB Cloud Database.
* The project is a MERN Stack web app and thus requires a working internet connection and a web browser only and is operating system independent.
* The Project uses ExpressJS Framework in the server side and NodeJS in the client side. All the data coming from client side will be stored in an online MongoDB Cloud Database.

## 3.4 Communications Interfaces

All the communications will be done via the web-browser with the standard HTTP protocol.

# System Features

## Register a Passenger

* Description
  + Getting on the platform requires the user to logged in to the specific profile. This task is very important to access the all the features of a class
* Stimulus/Response Sequence
  + For Registering, User will be asked to provide specific information to make the profile.
  + As soon as the user enters they will see a general home page and there will be a link to go to the register page. There the user will enter the details in the respective boxes. After the details are authenticated/stored the user will be directed again to the home page. The user will also get a email regarding the registration into platform
* Functional Requirements
* This requires MERN framework.

## LOGIN

* Description
  + Users will enter the credentials, created during the register process and then after authentication, they will be redirected to the respective user page.
* Stimulus/Response Sequence
  + He will be taken to the home page again.
  + Now he can proceed to book tickets and can use all the features.
* Functional Requirements
  + This requires MERN framework.

## Profile

* Description
  + User can see their profile.
* Stimulus/Response Sequence
  + Content of the profile will be shown like name, email etc.
  + Logout button is also provided to log user out
* Functional Requirements
  + This requires MERN framework.

## Update Password

* Description
  + If the user forgets his/her password then he can update his password by this button.
* Stimulus/Response Sequence
  + First the user will be asked to enter his new password and once again enter the same password in confirm password box.
  + Then he should click on the button generate OTP. This will send an OTP to his registered email.
  + On entering correct OTP the user has successfully updated his password.
* Functional Requirements
  + This requires MERN framework.

## Booking Flight Ticket

* Description
  + On successful login into the website you can see Book Flight Tickets button. On clicking it you will be directed into a new page.
* Stimulus/Response Sequence
* The User will be shown 2 boxes to select arrival and departure airports.
* The User will also be shown a box to select date of travel.
* The User can see three boxes showing 3 classes of passengers namely child, adult, elder.
* The user can increase the number from default to zero accordingly
* Finally he has to choose the class of seat in which he want to travel which are business, economy, first class.
* After entering the details he will be shown different flights which are available and can choose according to his will
* After choosing the flight he will be shown subtotal, GST and total amount to pay.
* Functional Requirements
  + This requires MERN framework.

# Other Nonfunctional Requirements

## Performance Requirements

The user should be able to query databases quickly and the results fetched must be appropriate. This can be done by finding the right balance between performance and accuracy by using MongoDB.

## Safety Requirements

The application runs on the web browser and hence harm to the user device is minimal, whereas a lot of data is to be read and written and hence the data storage and the server damage is possible but that too during heavy usage.

## Security Requirements

The portal exchanges many critical documents between the users hence there is a mul titude of safety requirements:

* A secure Login and Password for the users.
* A secured database system so that private information cannot be accessed by other people. Different storage spaces for the users so there is no information exchange between the users unwillingly.

## Business Rules

The software should not be outsourced to any third party without prior permission. Maintaining more than one account or fake account is illegal.

# Other Requirements

* Securely store the data and backup server to prevent any server crashes.
* A server capable of handling heavy workload and queries.
* Licensing requirements: Applicable.

## Appendix A: Glossary

* Javascript : A Modern Multipurpose programming language.
* Express.js : A Javascript back-end framework for web development.
* React.js: A Javascript front-end library for building user-interfaces.
* Node.js: A backend Javascript runtime environment and executes Javascript code.
* MongoDB: Widely used NoSQL Online Cloud Database.

## Appendix B: Analysis Models

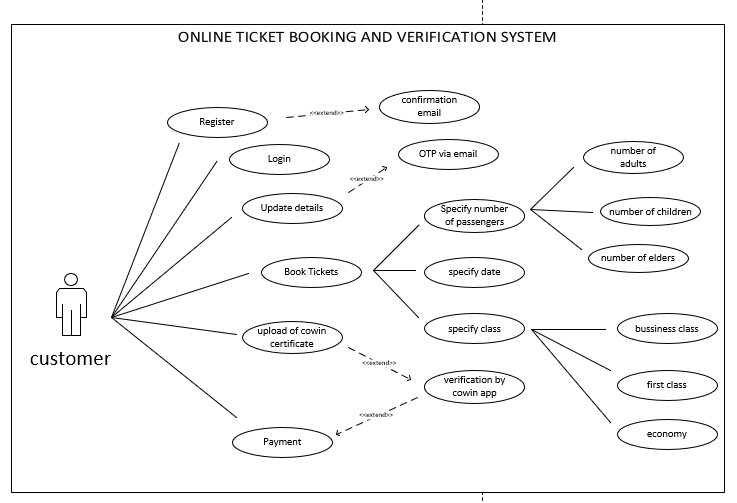
****

Figure 6.1: Use Case Diagram

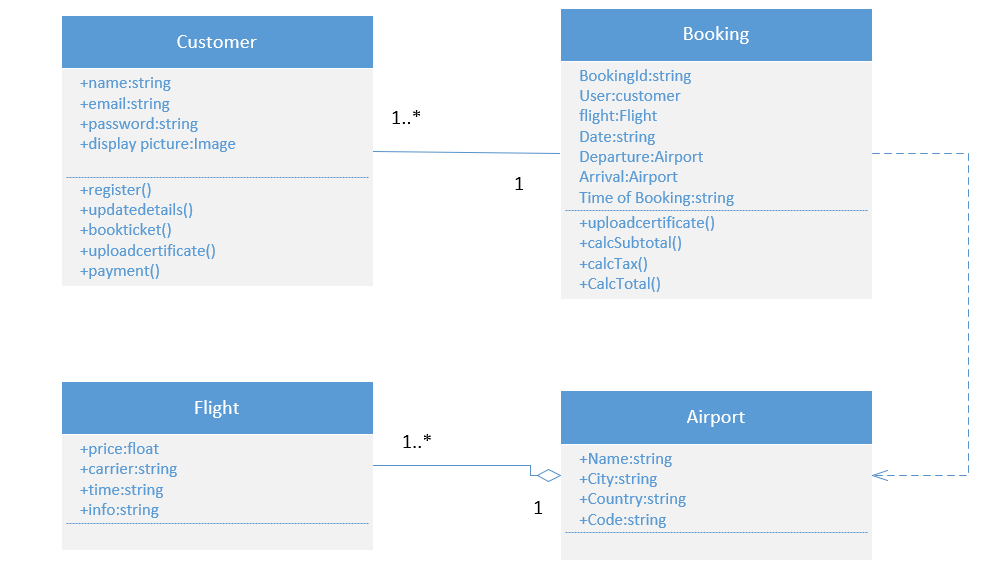


Figure 6.2: Class Diagram