

FLIGHT BOOKING

CONTENTS

1. [Problem Statement 2](#_bookmark0)
2. [WIREFRAMES 3](#_bookmark1)
3. [Business Requirements: 4](#_bookmark2)
4. [Proposed Rest Endpoints to be exposed 5](#_bookmark3)
   1. [Rest APIs: 5](#_bookmark4)
5. [Key Rubrics/Expected Deliverables 5](#_bookmark5)
6. [Platform 5](#_bookmark6)

# PROBLEM STATEMENT

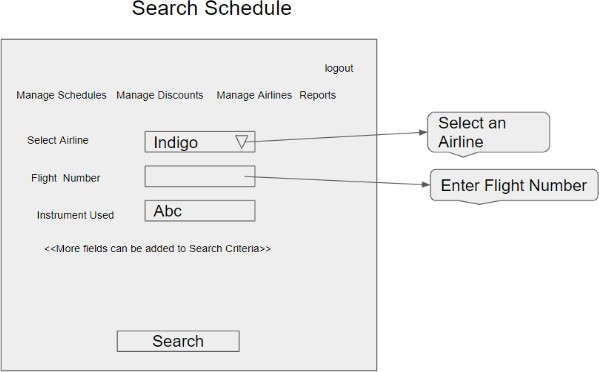
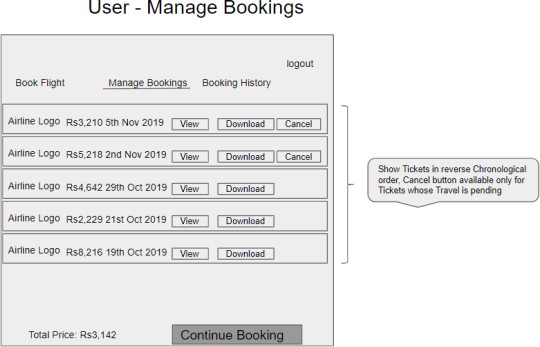
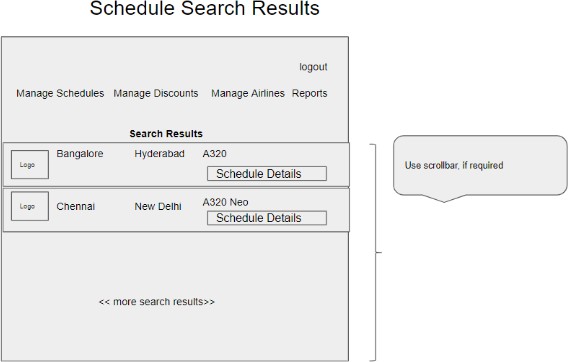
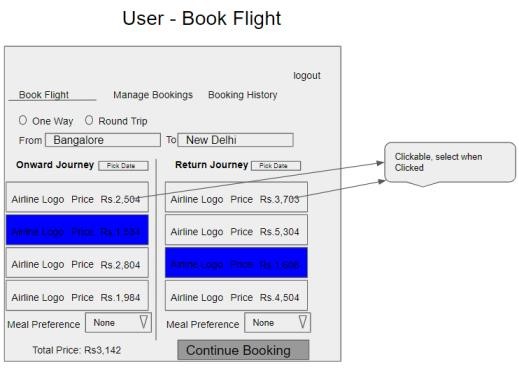
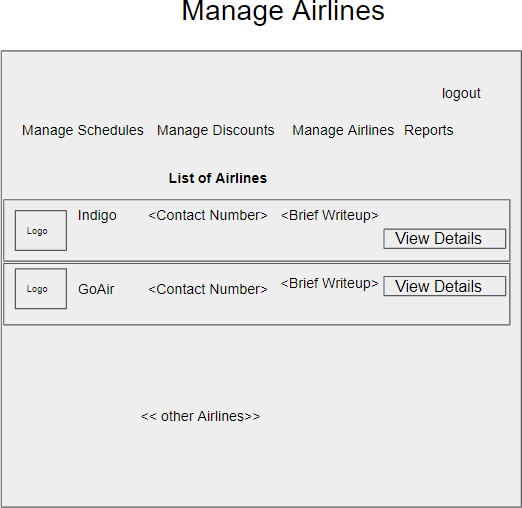
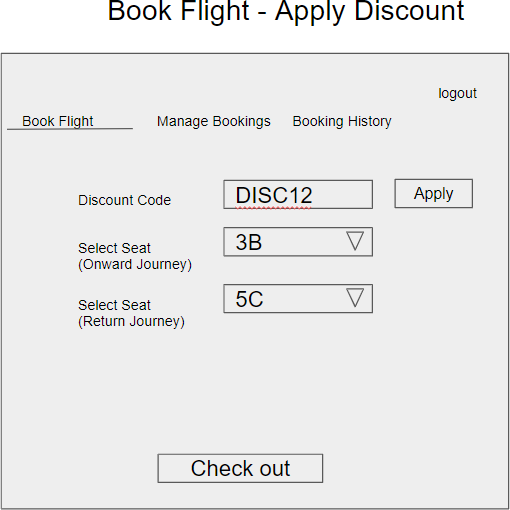
Build a software system which lets user search for a Flight Ticket and book it & includes Admin related activities. User can also cancel or update the Ticket.

Below are the different roles, which need to be supported by above Software System.

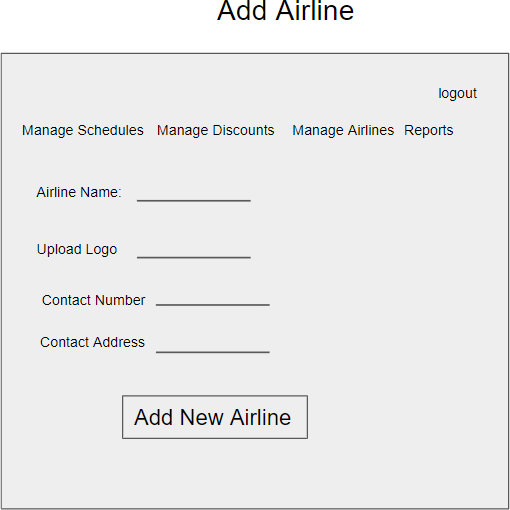
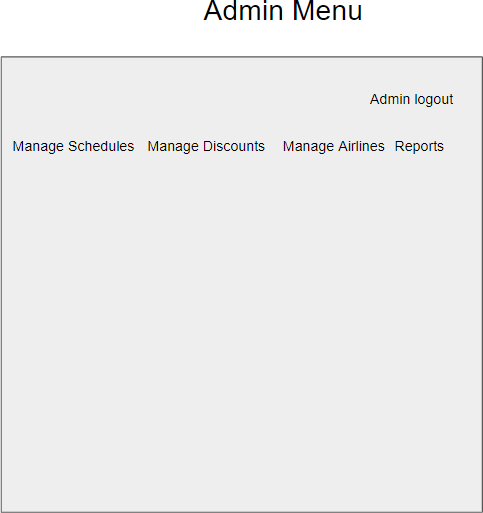
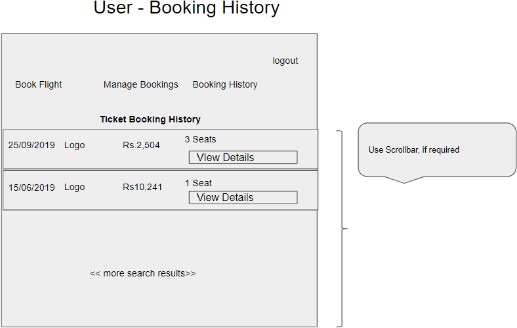
* 1. User
  2. Admin

The scope includes developing the application using dot net core as the backend and for end use react.

## For mobile application use flutter



1. WIREFRAMES

UI needs improvisation and modification as per given use case.

# BUSINESS REQUIREMENTS:

As an application developer, develop microservices with below guidelines:

|  |  |  |
| --- | --- | --- |
| User Story  # | User Story Name | User Story |
| US\_01 | User Mode | 1. User can search for a Flight based on date/time, from place/to place, one way or round trip 2. Each Search result need to display Flight Date/time, Airline Name/Logo, Price(to & round trip – TBD) 3. From Search results, User should be able to select a specific Flight and go ahead and complete Ticket Booking by providing below details    * Name and Email ID    * Number of seats to book.    * Details of each passenger (NAME:GENDER:AGE)    * Opt for Meal(Veg/Non veg)    * Select Seat Number(s) 4. On successful Ticket Booking, PNR number need to be generated, it should be possible to download TicketBooking can be done a Logged in User only 5. With email id user should be able to    * view History of Ticket Bookings,    * Cancel a Ticket only prior to a day(24 hrs) before journey date. 6. With PNR number view the booked ticket details |
| US\_02 | Admin Mode | 1. Admin shall be able to login/logout. 2. There can be pre-defined username/password for Admin. 3. Admin shall be able to add/block an Airline. When Airline is Blocked, Flights belonging to that Airline will not be shown in Ticket Search results. 4. Admin shall be able to add Inventory/Schedule of an existing Airline by specifying below details:    * flight number    * Airline    * From Place    * To Place    * Start date time,    * End date time,    * Scheduled Days(Daily, Week Days, Week Ends, For specific days specify the list of Days like Mon, Wed)    * Instrument used(A320, A320 neo, etc…)    * Total number of business class Seats |

|  |  |  |
| --- | --- | --- |
|  |  | * Total number of non-business class Seats * Ticket cost (consider taxes and other charges), * number of rows, * meal(none, veg, non veg) |

# PROPOSED REST ENDPOINTS TO BE EXPOSED

4.1 REST APIS:

|  |  |  |
| --- | --- | --- |
| **POST** | **/api/v1.0/flight/airline/register** | **New airline**  **booking** |
| **POST** | **/api/v1.0/flight/admin/login** | **Admin login** |
| **POST** | **/api/v1.0/flight/airline/inventory/add** | **Add Inventory/Schedule of an existing**  **Airline** |
| **POST** | **/api/v1.0/flight/search** | **Searches for flight** |
| **POST** | **/api/v1.0/flight/booking/{flightid}** | **Book ticket** |
| **GET** | **/api/v1.0/flight/ticket/{pnr}** | **Get Booked ticket details based on**  **PNR** |
| **GET** | **/api/v1.0/flight/booking/history/{emailId}** | **Get Booked tickets history based on**  **Email ID** |
| **DELETE** | **/api/v1.0/flight/booking/cancel/{pnr}** | **Cancel a booked**  **ticket** |

# KEY RUBRICS/EXPECTED DELIVERABLES

As an application developer:

1. Ensure layer project Structure with proper naming conventions and model classes.
2. Use application.json file to maintain all configuration like connection string
3. Implemented the layered structure - Controller, Interface, Service, DAO, Testing,

Validation, Security etc

1. Secure all Rest End Points by configuring SSL Certificate for Cloud

# PLATFORM

Use above Business Requirements to implement the below.

* 1. Use Azure to deploy application on cloud.
  2. Use Azure SQL Server as a database for the Application.
  3. Use Azure Functions and DB to build a backend process for handling requests for Flight booking App.
  4. Use Azure Service to send email after booking. Note: Minimum 2APIs (UI+Backend) to be hosted in cloud