Application Development I (Desktop)

Marie Andrews & Vishnu Narayanan

Final Project Deliverable 3: Airline Ticketing System



Contents

[Project Scope 3](#_Toc173679880)

[Users 3](#_Toc173679881)

[Clientele 3](#_Toc173679882)

[Project Functionalities 3](#_Toc173679883)

[Landing Page 3](#_Toc173679884)

[Show All Button 4](#_Toc173679885)

[Search with Filters 4](#_Toc173679886)

# Project Scope

## Users

The users of the Airline Ticketing System application will be both regular users seeking flights at the major airlines utilizing the platform as well as travel agents seeking flights for third party clients. Travel agents could particularly find the application useful as it could assist in streamlining aspects of their job duties, but all users should find the application suited to searching flights across a number of airlines based on personally selected filter parameters.

## Clientele

The primary clientele for the Airline Ticketing System application is the independent travel agent industry. Independent travel agents are not affiliated with major travel brands and often work from home. Our desktop application is ideal for this use case and would enable a small independent agent to compete with larger agencies volume wise by providing an ad-free, reliable, safe environment for acquiring the best rate on airline tickets.

# Project Functionalities

## Landing Page

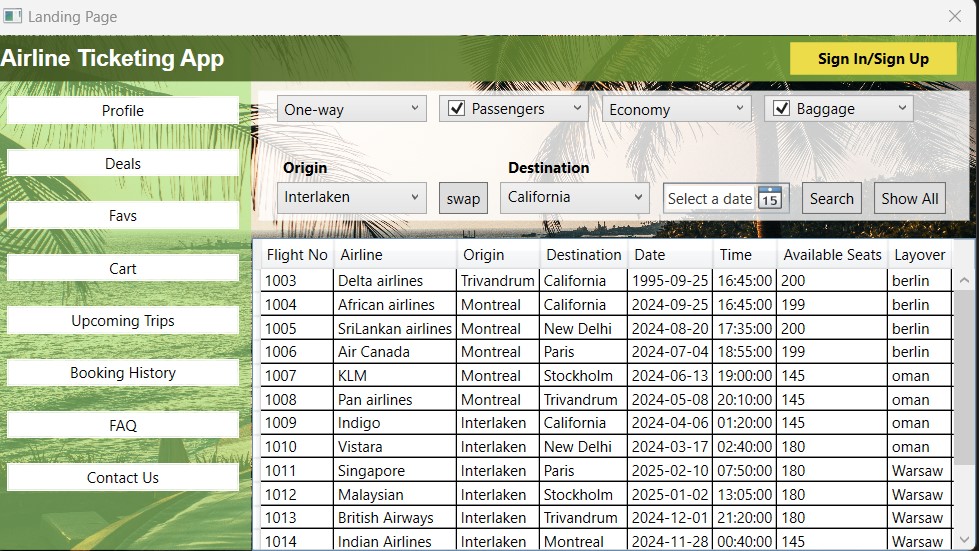
The application opens on the Landing Page:

A screenshot of a computer

Description automatically generated

### Show All Button

When a user clicks the button labeled Show All, the display grid is populated with all available flights in the database. The button click event calls method *refreshDisplayGrid().* This method is an asynchronous method that allocates the result of the REST Api method *getAllFlights()* into a List of Flight objects called allFlights. List allFlights populates the display grid.



### Search with Filters

When a user selects filters from the available options and clicks the button labeled Search, the display grid is populated with only flights matching the selected filters. The button click event retrieves the data from the selected filters and creates a new FlightsFilter object. The new FlightsFilter object is passed to the REST Api method *GetFlightsByFilter().* The result of this asynchronous method is stored in a List of Flight objects called availableFlights. List availableFlights populates the data grid.

A screenshot of a computer

Description automatically generated