CSC 340 Software Engineering Software Requirements Documentation (SRD)

Flight Finder

9.8.2020

Version 1.0

By: IE8 is the Hallowed King

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Teegan Krieger Michael Chapman Willie Holmes	Initial Version of the document, including basic layout and simple descriptions of each section	11/29/2020

Table of Contents

1. Introduction

1.1.1 Title

Flight Finder

1.1.2. Team Name

IE8 is the Hallowed King

1.1.3 Date

09/08/2020

1.1.4 Team Members

Michael Chapman, Teegan Krieger, Willie Holmes

1.1.5 Honor Code

I HAVE ABIDED BY THE UNCG *Academic Integrity Policy* ON THIS ASSIGNMENT.

1.2 Purpose

This is the Systems Requirements Document (SRD) for the semester project for class CSC 340. The goal of this project is to create the app *Flight Finder*. Flight Finder will allow users to connect to a database to access flight information from one locale to another, and allow users to save and view at a later time, specific quotes that they select.

1.3 Document conventions

Term	Description
Quote	An estimated price for a desired flight.
HTTP	Hypertext Transfer Protocol – The standard protocol used for transferring data over the internet.
API	Application Program Interface – A programming interface between a web server and a client.

1.4 Intended audience

This document is intended to provide guidance for the members of the development team to understand the full scope of the project, as well as provide Professor Quigley with an understanding of the product the group is attempting to build.

1.5 Scope

The purpose of this project is to improve each team member’s understanding of proper software engineering techniques as well as create an environment that prepares each team member for future team projects. The secondary goal of this project is to create a product that is presentable to Professor Quigley come the due date.

2. General Description

2.1 Product perspective

The product provides users with a simple and intuitive means of browsing flight quotes and saving quotes they might wish to purchase.

2.2 Product features

The product will provide users with an easy experience to plan flights. The product will provide the ability to search for and save flight quotes. The product will also provide users with a way to find the cheapest flights outbound from their local airport. Users will also be able to build trips with multiple flight quotes.

2.3 User class and characteristics

The product is intended for any person who is in need of an easy solution for browsing and planning what flight(s) they would like to purchase.

2.4 Operating environment

- Client Application
- Operating Systems: Windows 10, macOS, Linux
- Platform: Java

2.5 Constraints

The following constraints will be encountered:

- Project deadline is November 30th. Time is short.
- API calls are limited for free API keys. Product should account for this.
- New techniques that none of the developers have experience with must be applied, this will slow down production.

3. System Requirements

3.1 Functional requirements

The Application Shall be able to query a API to receive flight quote, locale, and currency data. The Application Shall require to receive text and mouse input from the user, and a display with a minimum of 1024x768 Resolution. The Operating system must be able to RUN a Java Virtual machine that supports file I/O and Class reflection.

4.External Interface Requirements

4.1 User Interfaces

The User Interface shall be built using the Java.Swing Library and NetBeans Gui Builder Plugin.

Each Screen Layout will be its own JPanel and a Single Java Swing Window will control switching between screens using a Conventional Menu Bar.

Below is a list of the Functions that each Screen Menu will have.

Main Menu Bar

File

Exit Application.

Views

Switch to Standard Search View.

Switch to Cheapest Flight to Anywhere View.

Switch to Trip Viewer View.

Edit

Settings

Standard Search View.

Set Departing Date: shall be implemented with a standard date picker.

Set Sort Mode: Shall be implemented with a standard DropDown box.

Set Departing Location: Shall be implemented with two parts, an input field where the user can start typing in the beginning of a location name, and a List of possible suggestions.

Set Destination: Shall be implemented the same as the Departing Location.

QueryAPI (Search): Shall be implemented as a standard button.

Add Flight to Saved Trip: Shall be implemented as a standard DropDown box.

Cheapest Flight to Anywhere View.

Set Departing Date: shall be implemented with a standard date picker.

Set Departing Location: Shall be implemented with two parts, an input field where the user can start typing in the beginning of a location name, and a List of possible suggestions.

QueryAPI (Search): Shall be implemented as a standard button.

Add Flight to Saved Trip: Shall be implemented as a standard DropDown box.

Trip Viewer View.

Select Trip: Shall be implemented with a standard DropDown box.

Delete selected Trip : Shall be implemented as a standard button that Triggers a confirmation dialog.

Create New Trip: Shall be implemented as a text input Field with a standard Button.

Settings

Set Country: Shall be implemented as a standard DropDown box.

Set Currency Code: Shall be implemented as a standard DropDown box.

4.2 Hardware Interfaces

- Windows 10
- macOS
- Linux
- A system with an internet connection.
- System must be able to make HTTP requests
- System must allow access to the Filesystem

4.3 Communications Interfaces

The product will communicate with an API using HTTP.

4.4 Software Interfaces

The following Software interfaces are used within the product.

- Operating System – The product will support three widely used Operating Systems, those being Windows 10, macOS and Linux.
- Java – The language of choice for its simplicity.
- Java Swing – The framework chosen for creating Simple UI elements within a Java environment.
- SkyScanner API – Used for looking up flight quotes and obtaining currency and country information
- OkHTTP Library – Used for making HTTP requests within the Java environment
- LGoodDatePicker Library – Provides a simple implementation of a calendar-styled date and time picker.
- Org.Json Library – Used for creating and parsing JSON objects used throughout the application

5. Non-Functional Requirements

5.1 Performance requirements

The performance requirements need to be specified for every functional requirement. The rationale behind it also needs to be elaborated upon.

5.2 Safety requirements

The product must be released in stable condition.
Safeguards should be put in place to prevent the program from reading or writing files outside of the program's data folder.

5.3 Security requirements

No user details are ever prompted for or collected by the product; thus, no special security practices need be considered.

5.4 Software quality attributes

Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.

5.5 Other requirements

These may include the legal requirements, resource utilizations, future updates etc.