**Software Requirements Specification**

**for**

**<Efficient Flight Itinerary>**

**Version 1.0 draft 1**

**Prepared by <Lance D., Bradley T., Lawrence S.>**

**<CS 441 - Software Engineering>**

**<10/21/2022>**

1. **OVERALL DESCRIPTION**
   1. **Product Overview**

Our Efficient Flight Itinerary system’s primary purpose is to provide a service that simplifies and manages flight choices most applicable to a passengers predetermined itinerary & preferences.

* 1. **Product Functionality**

1. Algorithm that calculates an efficient itinerary for a customer

* Weighted by user inputs

1. User-friendly operation and display
2. Integrate within our website
3. Modifiable after calculation

* Requests if user would like to modify original inputs
* Offers recommendations or secondary options

Above requirements are ordered by priority

* 1. **Design and Implementation Constraints**

Personnel:

* Small scale Project requires small scale team for development.

Data:

* Only consider West-Central Airlines flights, manageable overhead of data.

System:

* Limited integrability due to being constrained by company and geographical area (by design to keep overhead manageable).
  1. **Assumptions and Dependencies**

With the vast number of flights that occur from a multitude of locations, it is assumed that our data for testing purposes will be sourced from a precompiled list of flights between Dallas, TX and Los Angeles, CA. It is also assumed that these flights will only be within a one week timeframe, no more and no less. The week of flights to compile a data bank from is still tbd.

This will be a suitable volume of manageable data to handle for a small scale team of three developers. Additional assumptions will be that the passenger/user is familiar with handling a keyboard and mouse to utilize our service.

1. **SPECIFIC REQUIREMENTS**
   1. **External Interface Requirements**

The only requirement of this type is the program being interoperable with the existing website of the client.

* 1. **Functional Requirements**
* program must take user inputs (preferences)
* user inputs will be exported as a txt file
* user inputs must be plugged into an algorithm to generate an itinerary
* itinerary must be displayed to the user
* user must be able to edit preferences after seeing itinerary until satisfied

* 1. **Use Case Model**
* User inputs Flight Destination/Origin port Input, date of departure/Arrival from Origin port and date of departure/arrival from destination back to origin
  + Query user for one way trip or round trip
* User inputs filter requests/requirements (Time/Cost/Layovers)
* EFI performs calculations based on weighted values for each filter category
* Once weighted options are calculated, generate & display top candidates suitable for customer
* Options that remain modifiable for user: departure/arrival dates & weighted preferences
  + Query user if they need to modify their departure/arrival dates or preferences, otherwise continue
  + Flight Destination/Origin remains static
* Once user confirms preferred flight and payment, simulate generating a confirmed flight ticket for user

**3    OTHER NON-FUNCTIONAL REQUIREMENTS**

**3.1       Performance Requirements**

* The software is expected to be able to perform its functions while the host website is available
  + Looking at near 24/7 operation
* Software should be able to run smoothly and quickly across all major operating systems and their browsers
  + You should be able to easily access and operate our application from your most common devices

**3.2       Safety and Security Requirements**

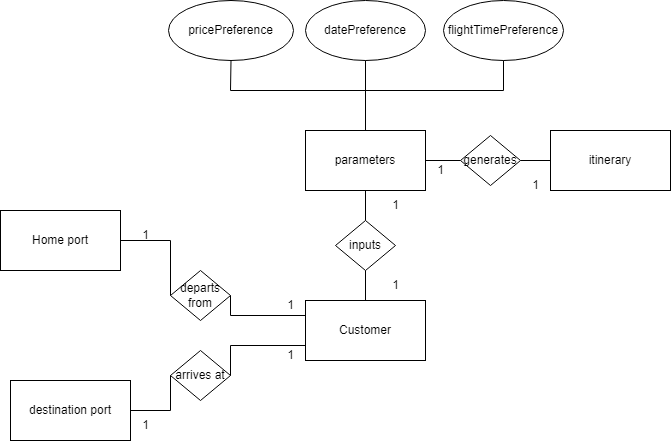
* The software will hosted on a web page so it must be secure enough to not allow users to get any data other than what the application is supposed to serve
* The software must not divulge any user data to other users

**3.3       Software Quality Attributes [non-functional]**

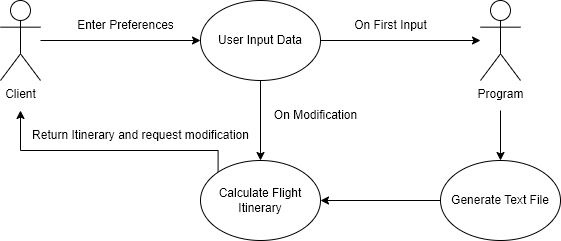
* Software will not frequently crash under operation
* Software will have a user friendly interface
* Software will be easy to operate and make it hard for there to be user error

**4   MODEL DRIVEN APPROACH**

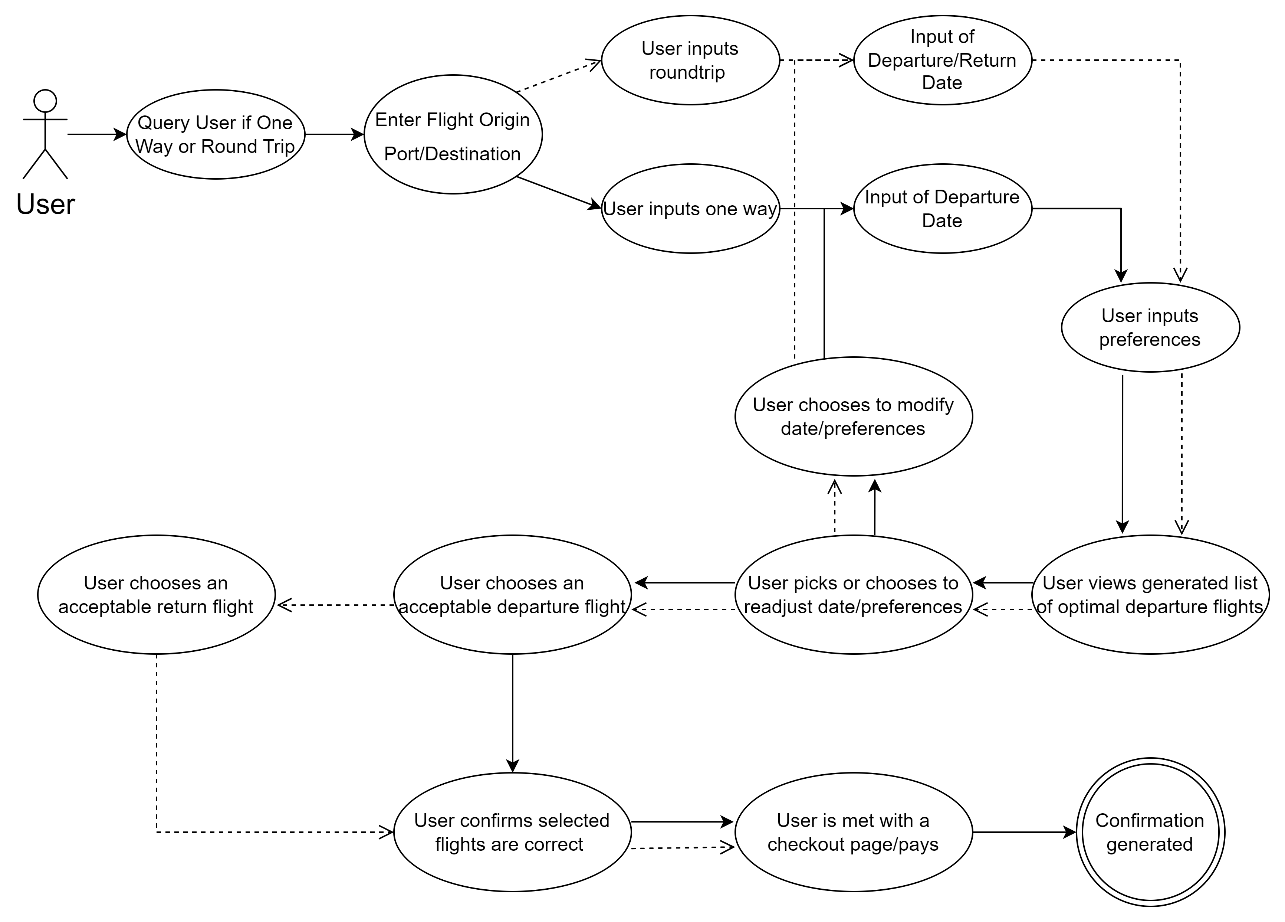
**4.1 Activity modeling**

****

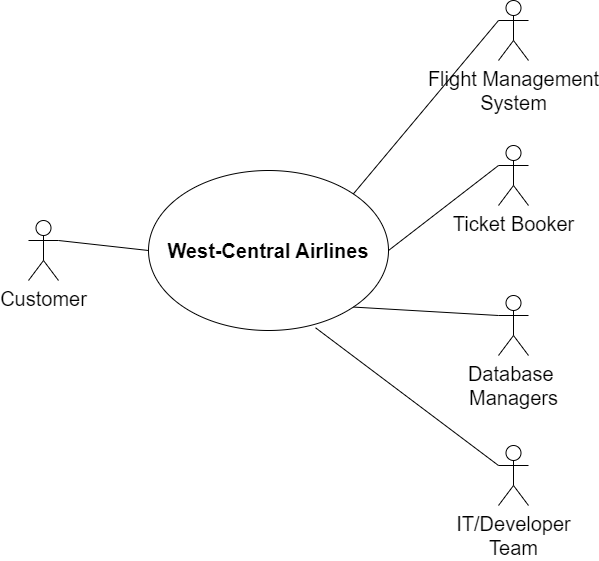
**4.2   Use case modeling**

****

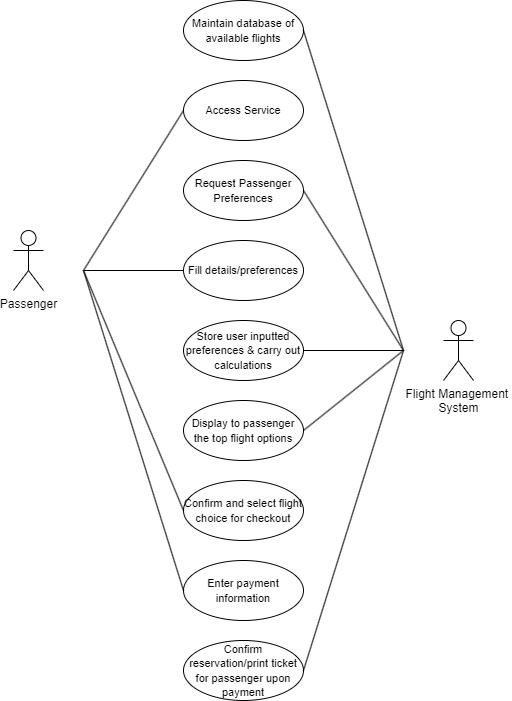
**4.3 Sequence modeling**

****

**4.4 Level 0 use case modeling**

****

**4.5 Level 1 use case modeling**

****