Software Requirements Specification

for

Airline Tweets Sentiment Analysis using Deep Learning

Version 1.0 approved

Prepared by Shrikant Jagannath Patro

Vellore Institute of Technology, Chennai

21-05-2019

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of the document is to build an analytical software for sentiment analysis of the flight tweet by passengers. This product will assist the flight management team to know the sentiment of the people travelling through flight and serve them better. This document will cover complete requirement, Design, verification, validation and testing details.

## Document Conventions

The Document uses the following conventions.

SRS – Software requirement specification.

## Intended Audience and Reading Suggestions

This project is prototype for the sentiment analysis of the flight’s tweet and it is restricted in college premises. This document is useful for developers, project managers, marketing staff, end user, tester and document writer. This include initial requirement specification for the software developer. This will help the project manager to keep track record of the development and overall development schedule. Earlier, schedule for different development, testing and deployment is planned using SRS. The marketing staff can market product based on the specification and feature mention in SRS. Different type of Testing and necessary detail is available in document. The document can detail in following sequence, Introduction, Overall Description, External interface requirement, Software feature. Requirements include functional, Non-functional Requirement, Performance Requirement, Safety requirement, Security requirement and Business rule.

## Product Scope

The purpose of the sentiment analysis using Deep Learning is to identify the sentiment from the tweet generated by flight passenger. This will classify the tweets mostly into three category Good, Bad, Neutral. This will help the flight management staff to know better their customer. This software based on the principle of Deep learning. This will perform analyses on the tweets and customer experience. The business will perform better in near future by making appropriate suggestions analyzed by the software. Better business and marketing strategy can be developed using knowledge from this software. Different ways to perform Analysis using Machine Learning and Deep Learning along with their pros and cons is mentioned in referenced document.

## References

[1] Apporv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau “Sentiment Analysis of Twitter Data”, Department of Computer Science, Columbia University.

[2]Walaa Medhat, Ahmed Hassan, Hoda Korashy “Sentiment Analysis Algorithms and applications: A Survey”, Ain Shams Engineering Journals (2014).

[3] <https://codete.com/blog/real-time-sentiment-analysis-with-machine-learning/>

[4] <https://www.kaggle.com/bertcarremans/deep-learning-for-sentiment-analysis/notebook>

[5] <https://stanfordnlp.github.io/CoreNLP/>

[6] <http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/>

[7] <https://en.wikipedia.org/wiki/Principal_component_analysis>

[8] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

[9] <https://www.kaggle.com/bertcarremans/deep-learning-for-sentiment-analysis/data>

# Overall Description

## Product Perspective

The origin of the product is specified as additional service to the general overall functionality of the flight management. This is add on service to flight management system. The product construction mechanism broadly divided into 4 part. This are Set-Up of project, Data preparation, Deep Learning. The Set-Up phase appear initially deal with initialization of variable and data loading. Data preparation phase will make data ready for the Deep learning. Initially raw data cannot be used directly for Machine Learning and Deep Learning task. So, Data preparation appear before Deep Learning. Once data is ready to be processed then In that case Deep Learning is used to perform analysis and conclude result.

## Product Functions

**Login:** User will be able to login into system. If user have no account. Then, System will ask user for Sign In followed by Sign In.

**Enrollment:** Customer can register them with more than one account, provided that sufficient details are mentioned.

**Book Flights:**The user can use the Book Flights function to purchase seats for an airplane flight.

**Reserve Seats:** The user can use the Reserve Seat function to reserve seats for an airplane flight.

**Flight Status:** This section shall allow the user – whether enrolled or not – to view flight information that matches input criteria.

**Flight Schedules:** This section of the system shall allow a user to query flight schedules based upon simple input criteria. The user will provide departure and arrival cities, and a departure/return date.

**My Account:** This section gives the user the power to view, save, edit or delete the information stored in his/her account. The user can check his/her accumulated points, look at the status of a flight that was booked,

**Logout**: The Logout section provides a way for the user to securely log out of the system.

**Classification of Comment:** This function is major functionality of the proposed product. It will classify the comment in test phase into Good, Bad and Neutral Comment. The proportion of this comment to analyze the overall performance of the Flight Management and services.

**Appropriate technique:** Deep Learning is more suitable for large amount of Data. In case of twitter data, when data size is extremely large. So, to handle such cases Deep learning outperforms traditional machine learning algorithm.

**Availability of High End infrastructure**: In recent Time, the availability of the high end machine specification at lower price is possible. This motivate us to use high end machine and take complete advantage of the Deep learning feature for the Classification task.

**Lack of Feature knowledg**e: When there is lack of Domain understanding for feature inspection, Deep learning Technique outshines others as less requirement for feature engineering.

## User Classes and Characteristics

User of the system will be able to retrieve the nature of the comment. There is no limitation in term of word usage. Any review is accepted by the deep learning frame work and classify the comment. The user from different class will be able to generate their comment in form of tweet. This tweeted data can be analyzed.

**Management Staff**- Naïve class of the user. They need to feed the data to Neural Network. The Neural Network is pre-trained. They need to use the data, feed it to neural network and neural network will generate result accordingly.

**User-** User has properties like Name, Address, Age and Associated with Flight Miles accumulated and Credit Card information.

## Operating Environment

**Software Platform**

Python 3.7.2

Basic Packages: Pandas, Numpy, re, collections and matplotlib.

Data Preparation and Modeling packages: sklearn, nltk, keras,

Jupyter Notebook

Anaconda3

**Hardware platform**

CPU: Intel Core i7-8750H

RAM: 16GB

Storage: 512GB SSD

Battery: 80Whr

**Operating System**

Windows / Linux

## Design and Implementation Constraints

**The limitation with respect to Software Policy**: If any strict guideline with respect to processing of the specific comment in near feature is introduced then, these will not completely explore the capabilities of the implementation.

**Hardware Limitation**: The hardware limitation for the implementation will exist if sufficient infrastructure for the hardware implementation as prescribed by operating environment is not provided and will affect the performance of the system.

**Memory Requirement**: for best performance at least 1TB to 4TB hard drive and 16GB RAM is necessary, if not provided performance will degrade.

**Programming Language Requirement**: Python Programming language are strictly required for implementation.

**Review or Comment Language**: Comment written in English can be reviewed, if any other language appear it cannot be processed.

**Programming Standard and Constraint**: This include the generic standard used for the development of the program using python programming and appropriate handling of the data and python packages.

## User Documentation

User Manual will be delivered as aid in helping customer (Management Staff) to understand the usage of the project.

## Assumptions and Dependencies

**TensorFlow Framework**: This is an open source product provide by google for high end matrix manipulation and other computation involved in Natural Language processing, Speech Recognition, Image and Video processing.

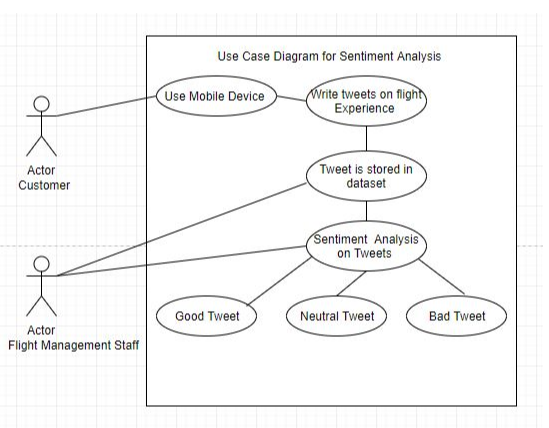
**Education Level of Management Staff**: The Management staff should be educated enough to python application and basic understanding of working of the tweeter.

**Reusability**: The internal structure of the deep learning framework provided by Keras on top the TensorFlow is used as Neural Net. It is not developed in this project is reused component.

# External Interface Requirements

## User Interfaces

[use Case Diagram]



## *C:\Users\shrikant\Desktop\content\Fast track Sem\software engineering and modeling\Digital Assignment\Diagram\use case diagram for Sentiment Analysis.JPG*

## Hardware Interfaces

System with the following Hardware Specification is used as interface to interact with the software system:

CPU: Intel Core i7-8750H

RAM: 16GB

Storage: 512GB SSD

Battery: 80Whr

## Software Interfaces

Application Developed using following product is acting as interface:

Python 3.7.2

Basic Packages: Pandas, Numpy, re, collections and matplotlib.

Data Preparation and Modeling packages: sklearn, nltk, keras,

Jupyter Notebook

Anaconda3

## Communications Interfaces

This software will work on the stored data in system disk space. Any web browser will support the Jupyter Notebook and Anaconda3. Preferably Google Chrome.

# System Features

[Class Diagram]

## System Feature 1

This will provide complete details about Requirement:

**4.1.1 Sentiment Analysis**

|  |  |  |
| --- | --- | --- |
| **Description & Priority:** | This section gives the power to the management staff to analyze every tweet of customer and find the problem, nature of the tweet. This is classified with respect to airline. | |
|  | . | |
| **Inputs:** | Tweets provided by user on flight experience | |
|  |  |
|  |  |
| **Source:** | All data are inputs from user. |
| **Output:** | Sentiment of the Tweets. |
| **Destination:** | The analysis result will be reflected in near future. | |
|  |  |
| **Pre-Conditions:** | The user must have an account with the website and must be | |
|  | logged in prior to access his/her account. |
| **Post-Conditions:** | All changes submitted by the user are applied to the user | |
|  | account on completion of the function. |

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

**3.1 General Requirements**

**4.1.2 Login**

**Description and Priority:**

This function allows a registered user to login his account using his frequent flyer number with the airline and password. If a user is not registered, the website should allow the user to enroll first. The system will check both the frequent flight number and password, when a user attempts to login.

In most case, the frequent flight number is convenient for both the user and system performance. The user easily memorizes his or her flight numbers but not a dull string. For the system, when provided the flight number, flight information will be delivered at the same time. Therefore, such operation reduces the second query chance.

Theoretically, more than one record can retrieve by user’s frequent flight number and password. Two or more users may have chosen the same password and same flight number. The way to break a tie is that system will go further to ask user’s email confirmation to identify.

**Inputs:**

Frequent flyer number and password

**Source:**

All inputs are provided by user.

**Outputs:**

Indication that user is logged in to the system.

**Destination:**

**Requires:**

The outputs are displayed on the screen as well as stored in the

system.

The user provides login information including frequent flyer

number and password.

**Pre-Conditions:**

User is not logged in to system. User has previously enrolled in system.

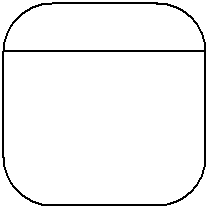
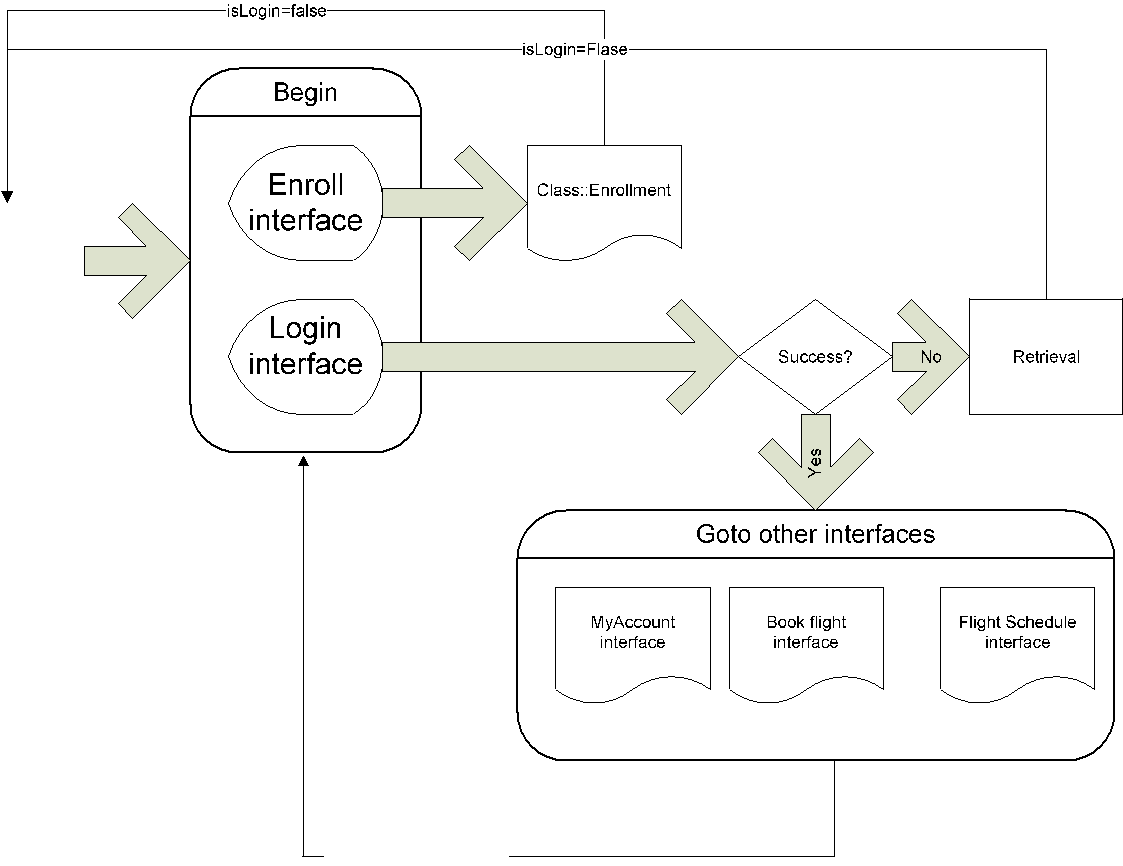
**Post-Conditions:**

User is logged in to system, OR user is not logged in because he/she entered unrecognized information.

**Side-Effects:**

None

**Use Case:**



**4.1.3 Enrollment**

**Description and Priority:** This function allows unregistered user to enroll and to create a

new account with the website. In order to create a new account, the user has to provide required information such as first name, last name, email address and password. Other optional information, such as phone number, credit card information and mailing address, can be provided during the registration process.

The system checks if all required data are provided and then will prompt the user to enter additional information, if required. After all required information is provided, the system auto-generates a unique frequent flyer number that the user must use as username for future authentications. The system shall auto-generate this number in less than five seconds.

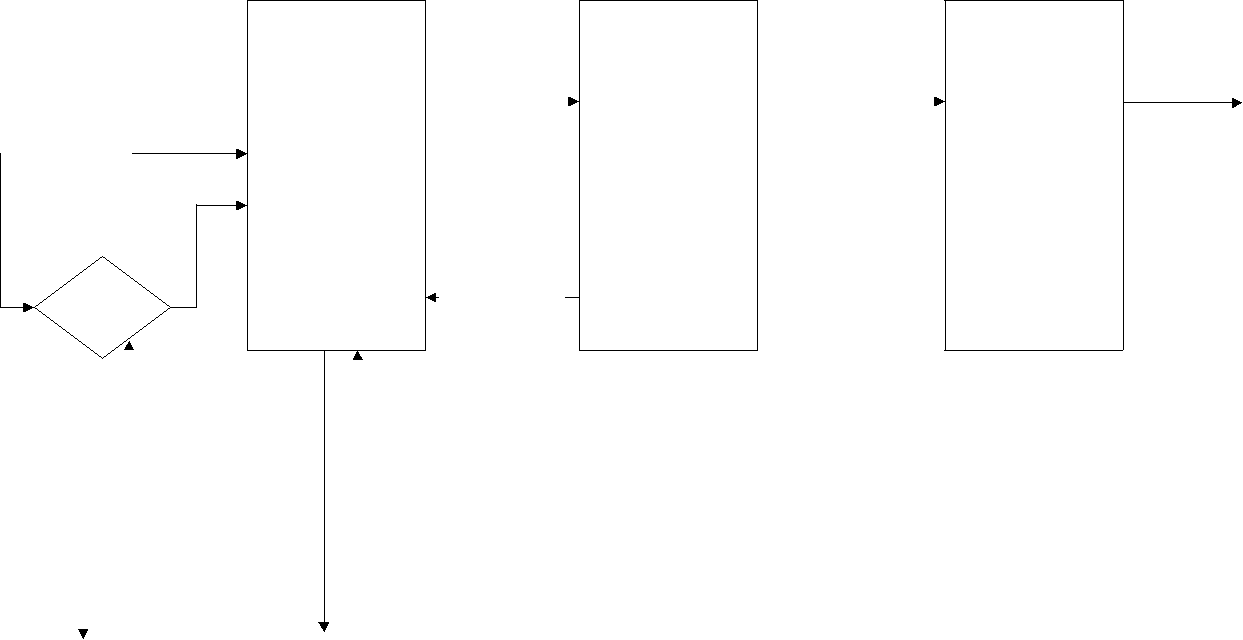
**Inputs:**

Required: First name, last name, email address and frequent flyer number

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Optional: phone number, credit card information and mailing | |
|  |  | address. |  |
| **Source:** | | All data except the frequent flyer number are inputs from the | |
|  |  | user. The frequent flyer number is from the system store. | |
| **Outputs:** | | Frequent Flyer Number |  |
| **Destination:** | | The changes are committed on the completion of the | |
|  |  | “enrollment” function to account information. All information | |
|  |  | also displays to the user via the screen. | |
| **Pre-Conditions:** | | The user must not have an existing account with the website. | |
| **Post-Conditions:** | | A user account is created and the user is able to access all | |
|  |  | functionalities provided by the function “My Account” (refer | |
|  |  | to 3.1.7) |  |
| **Side-Effects:** | | None |  |
| **Stimulus-Response:** | |  |  |
|  |  |  |  |
|  |  | **User Action** | **System Application** |
|  |  |  | 1. The system shows a list of |
|  |  |  | required data that must be |
|  |  |  | provided by the user in |
|  |  |  | order to create a new |
|  |  |  | account |
|  | 2. | The user provides | 3. The system checks if all |
|  |  | information such as first | required information have |
|  |  | name, last name, address, | been provided by the user. |
|  |  | email address, credit card | If a required data is |
|  |  | information and password | missing, the system |
|  |  | by filling fields. Some data | prompts the user to provide |
|  |  | are optional, while others | the specific data in order to |
|  |  | are required (refer to | process the account |
|  |  | section 2.2, REQ-02). | creation. |
|  |  |  | 4. When every required data |
|  |  |  | is provided, the system |
|  |  |  | creates an account and |
|  |  |  | auto-generates a unique |
|  |  |  | flyer number that must be |
|  |  |  | used as a username for |
|  |  |  | future authentication. The |
|  |  |  | system also sends the user |
|  |  |  | account information, |
|  |  |  | including the flyer number, |
|  |  |  | to the email address |
|  |  |  | provided by the user during |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | the account creation |  |
|  |  |  |  |  |  | process. |  |
| **4.1.4 Book Flights** | | |  |  |  |  |  |
| **Description and Priority:** | | | The user can use the *Book Flights* function to purchase seats | | | |  |
|  |  |  | for an airplane flight. The system shall present the user with | | | |  |
|  |  |  | information on all current flights. The user may then select a | | | |  |
|  |  |  | pair (departure and return) of flights on which to purchase | | | |  |
|  |  |  | seats. The user can indicate the number of seats and placement | | | |  |
|  |  |  | of such. Finally, the system shall guide the user completely | | | |  |
|  |  |  | through the checkout process. | | | |  |
| **Inputs:** | | | User information – the user must already be logged in. | | | |  |
| **Source:** | | | Inputs are from the user except flight information, which is | | | |  |
|  |  |  | retrieved from the system. | | | |  |
| **Output:** | | | The purchased seats are tied to the user’s account, so he/she | | | |  |
|  |  |  | can reserve seats later. | |  |  |  |
| **Destination:** | | | The booked flights will be stored in the user’s account | | | |  |
|  |  |  | information when the user finishes payment. The flight | | | |  |
|  |  |  | information shall also display on the screen. | | | |  |
| **Pre-Conditions:** | | | The user must have an account with the website and must be | | | |  |
|  |  |  | logged in. | |  |  |  |
| **Post-Conditions:** | | | Completion of this function guarantees that the user has seats | | | |  |
|  |  |  | on a specific flight. However, if the user wants particular | | | |  |
|  |  |  | seats, he/she must also complete the reserve seats function. | | | |  |
|  |  |  | Any successfully-booked flight from this function is assumed | | | |  |
|  |  |  | to have completed payment already. | | | |  |
| **Side-Effects:** | | | User’s account is charged. Flight is associated with user’s | | | |  |
|  |  |  | account. | |  |  |  |
| **Stimulus/Response:** | | |  |  |  |  |  |
|  |  |  | |  |  |  |  |
|  |  | **User Action** | |  |  | **System Application** |  |
|  |  |  |  |  | 1. The systems checks to see if the | |  |
|  |  |  |  |  |  | user is logged in, if not then the |  |
|  |  |  |  |  |  | systems require the user to login. |  |
|  |  |  |  |  |  | The system shows the user a list of |  |
|  |  |  |  |  |  | their already booked flights and the |  |
|  |  |  |  |  |  | available seats for those flights. |  |
|  | 2. The user selects individual seats | | |  | 3. The system checks if the selected | |  |
|  | from a list of pre-booked flights. The | | |  |  | seats are still available. If they are |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | seats are chosen in pairs so that the | available then the seats are |  |
|  | users selects individual seats for the | temporarily unavailable to other |  |
|  | departing and returning flights. | users to allow the user to confirm |  |
|  |  | their selection. The seats and |  |
|  |  | confirmation selection screen are |  |
|  |  | displayed |  |
|  | 4. The user confirms the seat selections | 5. Once confirmed, the seats are |  |
|  | on the screen. | removed from available seats and |  |
|  |  |  |
|  |  | are applied to the user account. The |  |
|  |  | user is shown a final confirmation |  |
|  |  | screen which displays selected seats |  |
|  |  | and account information. |  |
| **Activity Diagram:** | |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| START |  | Logged In |  |
|  |  |
|  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Selection |  |  |  | Seats Confirmed |  |  |  |
|  |  |  |  |  |  |  |
|  | | |  |  |  |  |  |  |  |  |
| Viewing | | | |  | Confirm |  |  |  |  |  |
| Unreserved | | | |  | Selected |  |  |  | Confirmation |  |
| Seats | | | |  | Seats |  |  |  | (Step3) |  |
| (Step1) | | | |  | (Step2) |  |  |  |  |  |

Selection Rejected

Login

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Incorrect | | |  | Incorrect | |  | Incorrect | | |
|  | Login | |  | Selection | |  | Selection | | |
|  |  |  |  |  |  |  |  |  |  |



**4.1.5 Reserve Seats**

**Description and Priority:** The user can use the *Reserve Seat* function to reserve seats for an airplane flight. The seats to be reserved are initially found through the user’s previous bookings. These bookings were previously completed through the *Book Flight* function (SEE 2.2.3).

The system shall display available seats for the departing and returning flights booked by the user. The user selects seats

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | ***Page 11*** |  |
|  |  | from each flight, where the number of selected seats from each | | |  |
|  |  | flight is the number that the user booked on that particular | | |  |
|  |  | flight. Once the flight seats are selected, the user confirms the | | |  |
|  |  | seat selection. |  |  |  |
| **Inputs:** | | User information used to determine previously booked flights. | | |  |
|  |  | Selected seats will be reserved by user. | | |  |
| **Source:** | | Inputs are from the user except flight information, which is | | |  |
|  |  | retrieved by the system. | |  |  |
| **Outputs:** | | Selected Seats |  |  |  |
| **Destination:** | | The changes are committed on completion of the *Reserve* | | |  |
|  |  | *Seats* function to the user’s account information. The selected | | |  |
|  |  | seats are also displayed to the screen. | | |  |
| **Pre-Conditions:** | | The user must have an account with the website and must be | | |  |
|  |  | logged in. The user must also have previously booked flights | | |  |
|  |  | without seats reserved, either by user or system. | | |  |
| **Post-Conditions:** | | All selections of seats must be applied to the user’s account. | | |  |
| **Side-Effects:** | | The selected seats are no longer available to any other | | |  |
|  |  | customer. |  |  |  |
| **Stimulus/Response:** | |  |  |  |  |
|  |  |  |  |  |  |
|  | **User Action** |  |  | **System Application** |  |
|  |  |  | 1.The systems checks to see if the user | |  |
|  |  |  |  | is logged in, if not then the system |  |
|  |  |  |  | requires the user to login. The |  |
|  |  |  |  | system shows the user a list of all |  |
|  |  |  |  | flights with open seats that match |  |
|  |  |  |  | their requirements. |  |
|  | 2.The user selects a flight or a pair of | | 3.The system checks if the selected | |  |
|  | flights (round-trip) that they wish | |  | flight(s) has their desired number of |  |
|  | to book. They choose a number of | |  | seats still available. If they are, the |  |
|  | seats they wish to book. | |  | system indicates to the user that their |  |
|  |  |  |  | flight is in the process of being |  |
|  |  |  |  | booked and they must now complete |  |
|  |  |  |  | payment. If there aren't enough |  |
|  |  |  |  | seats, the system notifies the user |  |
|  |  |  |  | which flights were full, and returns |  |
|  |  |  |  | the user to the beginning of the |  |
|  |  |  |  | function. |  |
|  | 4.The user fills out payment | | 5.If payment is successfully confirmed, | |  |
|  | information (either credit card or | |  | the flight is updated with the new |  |
|  | reward number) |  |  |  |
|  |  |  |  |  |

number of open seats, and the user's

account is updated with the reserved

flight information. The user is

shown a final confirmation screen,

which displays flight and account

information.

**4.1.6 Flight Status**

**Description & Priority:** This section shall allow the user – whether enrolled or not – to view flight information that matches input criteria. The user will provide:

a. A flight number and Date

OR

b. Departing/Arriving Cities and Date.

The system will display matching flight information including the following fields: o Flight Number

o Departure City

o Arrival City

o

Status (one of the following)

 In Flight

 At the Gate

 Delayed

 On Time

**Inputs:**

Departing city, Destination city, Departure date/time

**Source:**

All inputs are provided by the user.

**Outputs:**

Flight information including Flight Number, Departure City, Arrival City, and Flight Status.

**Destination:**

All outputs should display on the screen.

**Pre-Conditions:**

None.

**Post-Conditions:**

User has flight status for any matching flight.

**Side-Effects:**

None

**4.1.7 Flight Schedule**

**Description & Priority:**

This section of the system shall allow a user to query flight schedules based upon simple input criteria. The user will provide departure and arrival cities, and a departure/return date. If any flights match the criteria, the system will display the following information:

o Flight Number

o Departing City & Date/Time

o Arriving City & Date/Time

o Number of Available Seats

The system shall define a “matching” flight as one that uses the departure/arrival cities at a flight time greater or equal to the time provided by the user. Otherwise, the system shall alert the user that no matching flights can be found.

**Inputs:**

Departing City, Destination City, Departure Date/Time

**Source:**

All inputs provided by user.

**Outputs:**

Flight Information including Flight Number, Departing City & Date/Time, Arriving City and Date/Time, Number of Available Seats.

**Destination**:

All output should display to the screen.

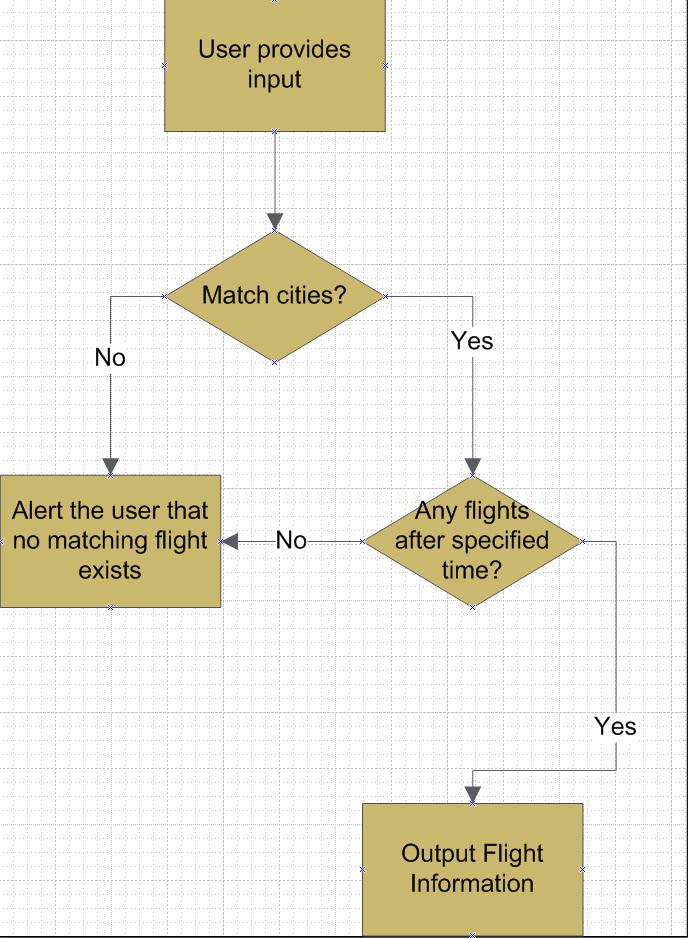
**Pre-Conditions**:

None

**Post-Conditions**:

User has flight information for any matching flight.

**Activity Diagram:**



**4.1.8 My Account**

|  |  |  |
| --- | --- | --- |
|  | |  |
| **Description & Priority:** | This section gives the user the power to view, save, edit or | |
|  | delete the information stored in his/her account. The user can | |
|  | check his/her accumulated points, look at the status of a flight | |
|  | that was booked, cancel a flight that was already booked | |
|  | (optional) and change his/her address, phone number, email or | |
|  | password. This feature is not available for non-registered user. | |
| **Inputs:** | Account changes, if any, made by the user. Account changes | |
|  | include updates on first name, last name, email address, |  |
|  | mailing address, password or phone numbers. |  |
| **Source:** | All data are inputs from user. |  |
| **Output:** | None. |  |
| **Destination:** | The changes are committed on completion of the *My Account* | |
|  | function to account information. |  |
| **Pre-Conditions:** | The user must have an account with the website and must be | |
|  | logged in prior to access his/her account. |  |
| **Post-Conditions:** | All changes submitted by the user are applied to the user | |
|  | account on completion of the function. |  |
| **4.1.9 Account Log out** |  |  |
| **Description & Priority:** | The *Logout* section provides a way for the user to securely log | |
|  | out of the system. This process will save all user operations | |
|  | when he/she exits the system. If a user wishes to continue | |
|  | accessing the website, he/she must log-in again to access user | |
|  | features. |  |
| **Inputs:** | None. |  |
| **Source:** | N/A |  |
| **Outputs:** | Notification that the user is logged out. |  |
| **Destination:** | User is notified by display to screen. |  |
| **Pre-Conditions:** | User is logged in to the system. |  |
| **Post-Conditions:** | User is logged out of the system. |  |
| **Side-Effects:** | The system clears the session state for the user once logout is | |
|  | complete. |  |

**4.External Interface Requirements**

**4.1 User Interfaces**

A *Help* link will appear on every screen that describes the function of each page to the user. The implementation should be written so that blind users can still interact with the system (using a screen reader.)

**4.2 Communications Interfaces**

The system must utilize the standard Hyper Text Transfer Protocol (HTTP) to ensure maximum inter-browser compatibility. The client accesses the system through a web browser.

**5.Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The Airline Website shall have capabilities to accept 500 connections. For each session, system shall guarantee the connection time 5 minutes from last input, after which the connection will be deemed expired. A close operation will be performed when expired. This design is to satisfy each user’s usability and connection quality.
* The system shall send out verification request immediately (within 100ms) after the it receives a user submitted form.
* The system shall update all flight status information every 5 minutes.

**5.2 Security Requirements**

* Passwords must be a minimum of eight characters and must contain one to seven digits.
* Email addresses should be verified before the system grants user access. This verification shall be exercised by sending the prospective user a confirmation email after enrollment. This email must contain information specific to completing the enrollment process.
* All exchanges from client to server involving private data shall occur using the highest available level of secure connection (e.g., https).

**5.3 Software Quality Attributes**

**5.3.1** **Usability:**

The airline website design shall allow deployment on both Windows and UNIX (Linux) servers. The design should support Windows Server 2003, Linux 2.6.x, V10 UNIX and later.

**5.3.2 Robustness:**

The system design shall include recovery scenarios allowing the ability to restore a state no older than one business day old.

**6. Appendix A: Glossary**

*Authentication*

The process of identifying an individual, usually based on username and password

*Cached*

A form of storing information/data, usually this data is repeatedly accessed.

*CSS*

Cascading Style Sheets is a feature to give users and developers more control on how web site pages are displayed.

*Database*

Is a structured collection of records or data that is stored in a computer system. In our system, this may pertain to flight records or user information.

*Dynamic Links*

A pointer to a particular scope called during runtime.

*Encryption Algorithm*

A mathematical procedure for performing encryption on data, which is translating data into secret code.

*HTTP*

Hyper Text Transfer Protocol is the underlying protocol used by the World Wide Web. It defines how messages are formatted and transmitted and what actions should be taken in response to various commands.

*Hyper Links*

Also called link, is a directly followable reference within a hypertext document.

*Input Criteria*

A defined particular group of criteria, which defines inputs.

*Query or Queries*

A form of questioning. In this document, query pertains to a search entered by a user into a search engine to return results.