Software Requirement Specification (SRS) for Tracking and Mining Flight Prices (TAMFLIP)

1. Introduction:

1.1. **Purpose:** Tracking and Mining Flight prices is a software to help the user find out the cheapest and most convenient flights and also track them or predict future prices for flights.

1.2. Scope:

1.2.1 Inscope

- Find cheapest flights between places.
- Send notifications about the tracked flights.
- Predict the dates of flights where the cost might be cheapest.
- Track flight rates over a period of time.

1.2.2 Out of Scope

- Book flight tickets.
- Suggest travelling places.

1.3. Definitions, Acronyms, Abbreviations:

1.3.1. Acronyms and Abbreviations:

- API: Application Programming Interface.
- TAMFLIP: Tracking and Mining Flight Prices.
- SRS: Software Requirement Specification.
- UI: User Interface

1.3.2. Definitions:

- API: A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service (freely available online flight info APIs in this case)
- Flight: Informal term for an aeroplane trip.

1.4. **Overview:** The rest of this SRS is organized as follows: Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed. Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given. Section 4 gives some possible future extensions of the system. Finally the appendices in Section 5 show examples of the user interface and details of prediction.

2. Overall Description

2.1. **Product Perspective :**TAMFLIP is geared towards people who want to find the best flights for a trip they are considering, track the prices of these flights with daily notifications, predict the prices of these flights in the future to make changes to their planned trips or just want to get insights into the fluctuation of flight prices over time. It is supposed to be clean and to the point without visual clutter and easy to use for the user.

TAMFLIP is intended to be a web application that can run across multiple devices with varying screen sizes and browsers.

2.2. Product Function

Class of use cases	**Description of Use Cases**
Use cases related to searching the flight	Select Arrival and Departure locations
	Select date from dropdown calendar
	Select mode of travel
	Select number of passengers
	Select flight class type
Use cases related to sending alerts, tracking prices and unsubscribing	Select the flight to track
	Send the alert to the user emails
	Unsubscribe tracking emails
Use case related to showing statistics	Show history of prices of the selected flight
Use cases related to price prediction	Select the flight for which prices have to be predicted
	Predict the prices of the selected flights
Use cases related to sorting and filtering flights	Sort searched flights
	Filter flights
Other use cases	Show/hide flight information

2.3. User Characteristics

- The app does not assume any knowledge on the part of the user about flights besides basic travel information like arrival and departure destinations, no.of passengers etc.

- The flight statistics might require some interpretation on the part of the user like understanding basic bar graphs.

2.4. Principal Actors

The two principal actors in TAMFLIP are user and system.

2.5. General Constraints

- The user must have a connection to the server in which the website(application) is hosted to connect to and use it.

- The user must have a valid email address to receive notifications from the system about flight prices.

2.6. Assumptions and Dependencies

- The proper functioning of TAMFLIP depends on the stability of the user's connection with the server where the app is hosted.

- If the server is down, the user will not be able to use the app.

- The flight information is acquired from various online apis and the working of the application is conditional on the accuracy and availability of this information.

3. Specific Requirements

3.1. Functional Requirements (Use Cases)

Use cases related to searching the flights

Use Case 1: Select Arrival and Departure locations.

Primary Actor: User
Precondition: None
Main scenario:

1. User starts typing the location.
2. A dropdown list suggesting the cities is shown to the user (cities matching the user input).
3. User selects a city from the list.

4. Arrival and departure locations are registered by the system.

Alternate scenario:

1. No cities with the name entered by user are present.
    a. Prompt the user to enter a valid city name.

Use Case 2: Select date from a dropdown calendar.

Primary Actor: User

Precondition: None

Main scenario:

1. User selects the box to enter the travel date.
2. A dropdown calendar showing the available dates is displayed.
3. User selects the required date.
4. User selected date is registered by the system.

Use Case 3: Select mode of travel

Primary Actor: User

Precondition: None

Main Scenario:

1. User selects the dropdown list to select the mode of travel between the destinations.
2. Mode of travel will be either one-way, round-trip or multi-city.
3. The option is set default to one-way mode of travel.

Use Case 4: Select number of passengers

Primary Actor: User

Precondition: None

Main scenario:

1. User selects the box to enter the number of passengers.
2. An interactive counter showing the number of adults, children is shown.
3. User selects the appropriate number of people in each category.

Use Case 5: Select flight class type (Economy, Business etc.)

Primary Actor: User

Precondition: None

Main scenario:

1. User selects the box to confirm the class.
2. A dropdown calendar showing the available classes is displayed.
3. User selects the required class.
4. User selected class is registered by the system.
5. Defaults to economy class.

Use cases related to sending alerts, tracking prices and unsubscribing

Use Case 6: Select the flight to track.

Primary Actor: User

Precondition: None

Main scenario:

1. User selects the flight he/she wants to track.
2. User information (i.e email) is collected.
3. User information and the flight details are registered by the system.

Alternate scenario:

1. User information is already available (from the same session).
    a. We don't collect the user information (i.e email) again.

Use Case 7: Notification system.

Primary Actor: System

Precondition: None

Main scenario:

1. System sends requests to Flight APIs to get the details regarding the tracked flights.
2. System sends the alerts regarding the flight details to the users who opted to get the alerts.

Use Case 8: Unsubscribe from email

Primary Actor: User

Precondition: Users must have received the alert from TAMFLIP.

Main Scenario:

1. User selects the unsubscribe button.
2. User is redirected to the main website.
3. A prompt containing all the tracking flights is opened.
4. User selects the flight to unsubscribe.
5. System registers the information to unsubscribe.

Use case related to showing statistics

Use Case 9: Show history of prices of the selected flight

Primary Actor: User

Precondition: Must have searched for flights first

Main Scenario:

1. User selects the unsubscribe button.
2. User clicks the show statistics button on the search result.
3. Flight prices over multiple days are displayed.
4. Current day and lowest price are highlighted in the graph.

Use cases related to price prediction

Use Case 10: Subscribing to flight price predictions

Primary Actor: User

Precondition: None

Main scenario:

1. User selects the flight he/she wants to get predictions for.
2. User information (i.e email) is collected.
3. User information and the flight details are registered by the system.

Alternate scenario:

1. User information is already available (from the same session).
    a. We don't collect the user information (i.e email) again.

Use Case 11: Predict the prices.

Primary Actor: System

Precondition: None

Main scenario:

1. System gets the past prices of the flights which are requested by the users (to predict).
2. Time series analysis or machine learning methods are used on the gathered data to appropriately predict the future prices.
3. A notification regarding the prediction is sent over email to the concerned user.

Use cases related to sorting and filtering flights

Use Case 12: Sort searched flights

Primary Actor: User

Precondition: Must have entered From and destination points.

Main Scenario:

1. Use selects a sort option from the sort by drop down (like price, no.of stops, travel time etc.)
2. System sorts the already shown results based on the selected criteria.

Use Case 13: Filter flights

Primary Actor: User

Precondition: Must have entered From and destination points.

Main Scenario:

1. Use selects a filter option from the sort by drop down (like price, no.of stops, departure, arrival, travel time etc.)
2. System filters the information based on selected criteria.

Other use cases

Use Case 14: Show/hide flight information

Primary Actor: User

Precondition: Must have searched for flights first.

Main Scenario:

1. User selects the down button for every flight, to show more details of the journey.
2. If the details are already displayed, users can select the up button to hide the details.

3.2. **Performance Requirements**

- Should run on any modern browser and devices that support these browsers. Provided the framework is available and responsive the results are displayed in under 5 seconds.

3.3. **Design Constraints**

- Security: The mail addresses of users should be stored in a secure server.
- Fault Tolerance: The user info should not be lost if the server crashes or is down.

3.4. **External Interface Requirements**

The user screen contains a text field and drop down menus to select the parameters for the search and and a search button. Once the user makes a search the results (if available) are displayed below the search options. Each flight result then has the option to track, predict and show statistics. Clicking on this shows a graph depicting prices.

4. **Future Extensions**

Allow users to search with voice. Use NLP to get required info from these voice commands. Suggest trips based on the user's activity. Show flight jokes on screen. Display hotel booking also. Suggest departure destination based on IP of user.

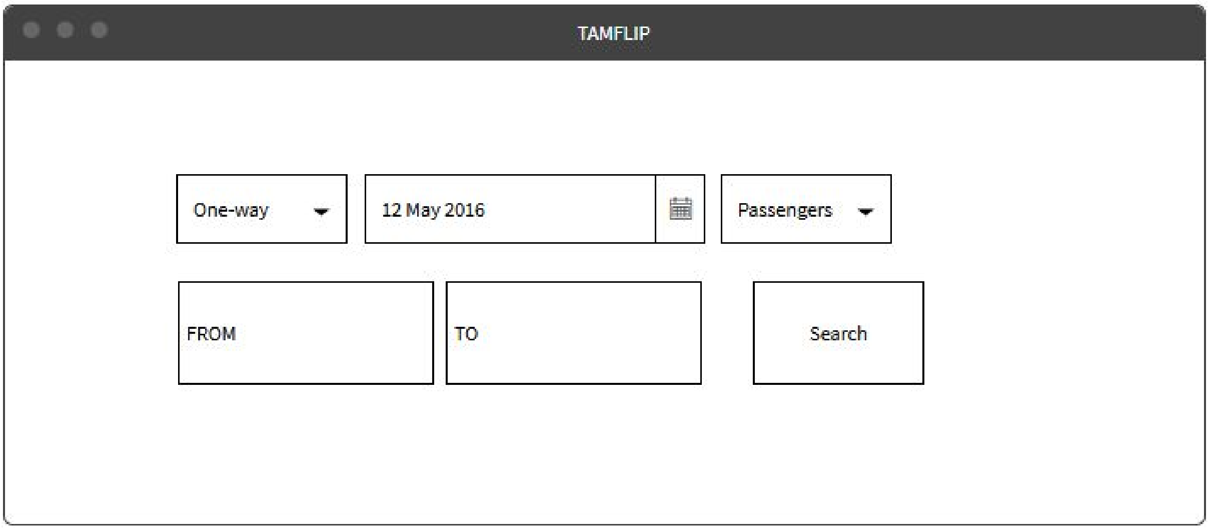
5. **Appendix**

5.1. **Appendix A:**

Predicting flight prices: The flight prices are predicted using linear regression with day of flight, number of days from departure, airline, destination, origin, flight model etc.

5.2. Appendix B:

Mockup of the user screens and UI are attached below.



TAMFLIP

One-way

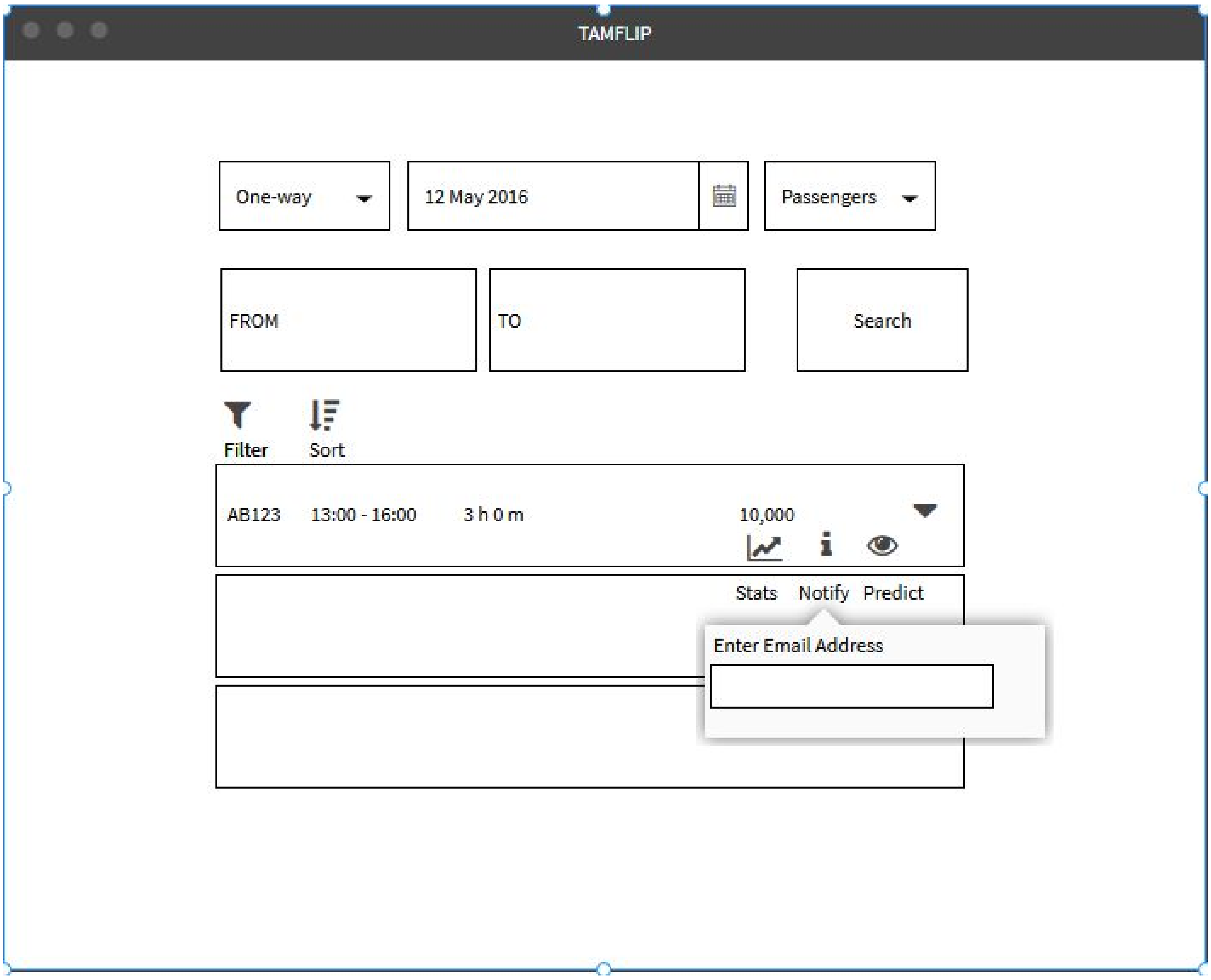
12 May 2016

Passengers

FROM

TO

Search



TAMFLIP

One-way

12 May 2016

Passengers

FROM

TO

Search

Filter

Sort

AB123 13:00 - 16:00 3 h 0 m 10,000

Stats Notify Predict

Enter Email Address