Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: AeroCast

Project Analysis Report

Ahmet Alparslan Çelik, Ömer Berk Uçar, Muhammed Yusuf Satıcı, Yasin İlkağan Tepeli, Ahmet Taha Albayrak

Supervisor: Çiğdem Gündüz Demir
Jury Members: Selim Aksoy and Mustafa Özdal

Progress Report

Nov 05, 2017

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2

Table of Contents

1. Introduction

While planning a holiday or a business trip, everyone once in a while tries to find the best airfare deal according to their needs. In general, it is a tedious work due to the dynamic changes in prices. Airlines change their prices according to various criteria such as supply and demand, how far the departure date is, which day of the week the flight is, etc. Thus, finding the cheapest ticket by searching manually can be cumbersome. Furthermore, it is necessary to optimally find the best purchase date for a particular route within the traveller's budget.

With the current technologies, it is possible to track airfares, get notifications in case of a drop or increase in price and even get rough predictions if the price will increase or decrease for a particular route. Nevertheless, the existing methods do not provide user specific airplane predictions. Therefore, in order to purchase the ideal cheapest ticket, travellers will need to use separate services. Moreover, if they were to look for different travelling plans, the number of different routes they need to check will increase making things even more time consuming and complicated.

Aerocast aims to minimize the efforts put into finding the best airfare deal available. It will be a web application, which collects the available data scattered through different web services and analyzes them in a user specific way. Therefore, it will assist travellers to find the best airfare in the minimum possible time. The users can search and track their flight of interest months before the departure and can get elusive intelligence. Regarding to the personal preferences, users can provide preferences such as carrier type or cabin class, Aerocast will list the most related results.

In the report, an analysis of the system we will develop is provided. Initially, the existing systems together with their major differences of the current systems are given. Then the details of the system are given together with its unequalled features. Later, functional, non-functional and pseudo requirements are listed. Afterwards, scenarios and their common use cases are presented. The object and dynamic models are presented and explained and finally the user interface mock-ups are listed with brief descriptions.

2. Current System

There are various applications providing services regarding the airfares and flights. Two kind of applications show some similarities with the proposed system of AeroCast and two main applications are discussed as current systems in this report.

1. Applications, providing flight information to users. These applications enable the user to search, buy and track the current flights for various destinations, origins, departure dates and carrier codes. However, they do not make forecasts for airfares.

   Skyscanner[1]:

   - It enables the users to search the current flight prices based on the criteria specified by the user.
   - The user can display the prices of the flights in different sort orders based on the price or duration of the flights.
   - Skyscanner enables the users to track their flights and it alerts the users when the prices of the flights change.
   - The users need to register into the skyscanner system to get alerts on the flights.
   - The skyscanner also let the users choose multiple destinations for their flight searches.
   - Skyscanner does not provide any forecasting system and it does not show the previous flight data.

   AirfareWatchDog[2]**:**

   - It enables the users to find current cheap flight tickets based on the criteria specified by the user.
   - The users are able to track flights and get alerts by providing their email address.

- AirfareWatchDog does not provide any forecasting system and it does not show the previous flight data.
- AirfareWatchDog enables the users to compare cheap flights from different sources.

Expedia[3]**:**

- Similar to the above applications, Expedia enables the users to search, track and filter the current flight prices.
- The users can create lists of favorite flights and trips.
- The users need to log into the system to track the flights and create the lists.
- Expedia also provides the users the option of searching for flights and hotels together.
- Expedia does not provide any forecasting system and it does not show the previous flight data.

TripAdvisor[4]**:**

- TripAdvisor also enables the users to search for the current cheap flight prices.
- The users can save the searches and see their lastly searched flights.
- TripAdvisor provides the option of comparing its results with the results of Expedia, in which TripAdvisor mostly gives worse results than Expedia.
- TripAdvisor does not provide any forecasting system and it does not show the previous flight data.

2. Applications providing forecasts for the flight prices. These applications enable the user to buy the flight tickets at the time that the flight prices are going to be low. Mostly, these applications do not provide flight data and make only purchase date predictions.

Hopper[5]:

- Hopper is a mobile application that aims to predict when the users should buy the flight tickets.

- Hopper shows the current cheap flight price to the user and enables the user to track the flights.
- Hopper does not require the users to log into the system to use its functionalities.
- Hopper makes predictions as buy now or wait.
- It does not make specific purchase date predictions and it does not show expected prices for the flights.
- Hopper does not provide previous flight data to the users.
- Hopper sends notifications to the users when the prices for one of the tracked flights change.

Fairfly[6]:

- Fairfly makes airfare predictions for corporate travel.
- Fairfly only serves to the travel management companies and enterprises. It is not open to the public.

Kayak[7]**:**

- It enables the users to search the current flight prices based on the criteria specified by the user.
- Kayak enables the user to track, filter and sort the search results based on multiple criteria.
- Kayak make predictions as the prices of the flights are expected to decrease or increase in the next week. Kayak make forecasts only for seven days.
- It does not make specific purchase date or departure date predictions and it does not make any prediction for the dates later than one week.
- It does not show the expected prices.
- Kayak has an explore option that shows the previous flight price data on a map. The user can specify the origin and date interval in the explore option and Kayak displays the past prices of the flights found in the

specified date interval and having the specified origin. The past prices are shown on top of the destinations in a world map.

- The user can interactively use the map by clicking on the destinations in order to search for current cheap airfares and get predictions for the next week.

3. Proposed System

3.1. Overview

AeroCast is a web application that aims to enable the users to find the optimum time to buy the flight tickets. AeroCast makes purchase date predictions which tries to determine when the prices of the flights are predicted to be the lowest and it makes departure date forecast that tries to find which departure dates are expected to have low flight prices. In addition to the purchase and departure date forecasts, the users see the expected prices for the predicted purchase dates. Also, AeroCast shows the history of flight prices to the users and it enables the users to track the flights. The users can get notifications when the predictions change and they can give feedback to the system as well as the particular predictions. By using the AeroCast system, the users are able to get comprehensive insight about the flights based on the date forecasts and flight statistics.

AeroCast gets various specifications from the users such as origin, destination, departure date interval, cabin class and carrier code to make forecasts for the specified search of the user. Based on the criteria specified by the user, AeroCast determines the best departure dates having the lowest prices and for these departure dates, AeroCast predicts the purchase dates having the expected lowest price. Therefore, unlike the current systems, AeroCast enables the users to get a two staged prediction regarding the departure and purchase dates along with the expected price of the flight. Also, AeroCast provides the option of making predictions for multiple departure dates, which makes the system more flexible. AeroCat provides filtering, sorting and tracking options to the users and it enables the users to receive notifications for the changes in the flight

prices. Unlike most of the current systems, the users not only see the current prices and the future predicted prices of the flights but they also see the past price data of the flights. AeroCast visualizes the history of price data along with the predicted prices of the future purchase dates on an interactive graph. AeroCast also continuously collects flight data from various sources and it updates the airfare forecasts based on the newly collected data. Therefore, AeroCast provides a dynamic service to the users that offers a comprehensive amount of past and future data and makes flexible multi-staged airfare forecasts.

3.2. Functional Requirements

3.2.1. User Profile

- The user should be able to register into the system by providing name, name, password and email address information.
- The email address should be unique for each user.
- The user should be able to sign in into the system by using email and password information of his previously created user account.
- The user should be able to sign in by using his facebook or gmail accounts.
- After signing in, the user should be able to update his profile information.
- The user should be able to receive notifications from the system and should be able to view or remove the notifications from his profile.
- The user should be able to change the email notification settings of the system from his profile.
- The system may support SMS notifications for the future development of the project.

3.2.2. Time Prediction

- The user must be able to send requests to the system by specifying the origin, destination and departure dates as well as the number of adult and child passengers information.
- The user must be able to choose one way or return flight tickets in his request.
- The user should be able to specify his flight type as direct or indirect as default.

- The user should be able to choose cabin class and carrier code of the flight.
- The carrier code, cabin class and flight type information is optional.
- The system should be able to return responses for the departure and purchase date forecasts based on the criteria specified in the request.
- The user should be able to display the responses in three different orders based on the prices of the flights and the duration of the flights.

3.2.3. Purchase Time Estimation

- The system should estimate the purchase dates with minimum prices and the expected values of the prices.
- The system should return purchase date responses for each departure date which the flight ticket prices are predicted to be the lowest.
- The system should return the expected price and other available flight information as a part of the response.
- The user should be able to track the purchase date responses of the system.

3.2.4. Departure Time Estimation

- The system should be able to rank the departure dates as having low, medium or high prices based on their price intervals and stability of their prices.
- The system should display the departure dates of the flights on a colored graph in which the blue lines represent the departure dates with low and stable prices whereas the red lines represent the departure dates with high and unstable prices.

3.2.5. Route and Flight Tracking

- The user should be able to track the routes and flights only if they log in.
- The user should be able to display his saved routes from his profile.
- After selecting the saved route, the user should be able to display his favorite tracked flights on that particular route.
- The system should keep updating the tracked searches and flights as the new data arrives.

- When the departure date of a flight expires, the system should remove the route from tracked routes.
- When the estimated dates for a saved request expire, the system should update the response with a new estimate.
- The system should notify the user about the changes in the tracked responses.

3.2.6. Flight Statistics

- The users should be able to view the flight statistics and graphs for their routes and flights.
- By selecting the flight, the user should be able to display the past changes in the prices of the flights as a solid line on a graph along with the predicted prices of the flight as a dashed line on the same graph.
- The user should be able to display the exact price and date values on the graph by moving the cursor on top of a particular point on the line.

3.2.7. Data

- The system should store the notifications, profile information and the saved routes and flights of the user.
- The system should keep the data of the previous flights such as price, departure date, origin, destination, flight type(direct or transit), carrier code, cabin class.
- The system should store all airport names, codes and cities in the world.
- The system should search and suggest the destination and origin cities as well as the airport names as the user types.

3.2.8. Automatic Data Update

- The system should collect 600,000 flight data from the most frequently used 10,000 flight routes worldwide daily.
- The system should start collecting flight data from routes that user searched if there is not enough data.
- The system should update its price analysis based on the collected data.
- The system should regularly check the results of the saved routes and flights and update these responses based on the changes in the price analysis.

3.2.9. Feedback

- The user should be able to send feedbacks to the system in order to evaluate the application and price predictions.

3.3. Nonfunctional Requirements

3.3.1. Availability

- The system should provide service continuously other than the system maintenance hours.
- The system maintenance should be scheduled between 20.00 - 22.00 on the first Monday of each month.

3.3.2. Maintainability

- Since the modules are working independently, the addition of new functionalities and the changes in the existing functionalities should not affect the system.
- A crashed component should not affect other components.

3.3.3. Response Time

- The system should predict the purchase and departure dates of the new request and list them at most 5 seconds.
- Pages should be loaded under two second.

3.3.4. Scalability

- The system should fulfill 10,000 users requests daily.
- The system should be able to store at least 10 GB of NoSQL data for the first year which continuously increase 100 MB in size for each day.

3.3.5. Security

- The system should store passwords after hashing with SHA-256.
- The number of requests will be limited to 200 per IP address in order to protect the flight data from malicious actions.
- The web service should be protected from DDOS attacks using the Cloudflare service.

3.3.6. Reliability

- The system should accurately collect data of the flights on a daily basis and accurately predict the departure and purchase dates of the flights based on the collected data.

3.3.7. Usability

- The interface should be user-friendly and the data is neatly shown.
- The system should display graphs smoothly and continuously.

3.3.8. Portability

- The system should be run on desktop web platforms and mobile web platforms.

3.3.9. Extensibility

- The system should allow developers to add new components and services on it.

3.4. Pseudo Requirements

3.4.1. Implementation Constraints

- The software will have server-client architecture.
- Git version control system together with GitHub will be used to manage project by the development team.
- OOP paradigm with the basics of SOLID [8] principles of will be followed.
- The web platform is required to be responsive and the targeted web platforms are Google Chrome, Opera, Firefox, Safari and Edge.
- For the front-end, ReactJS will be used and Swagger will be used to determine the specification of REST API.
- Back-end will be written in JAVA. For search-as-you-type feature, ElasticSearch will be used. Spring will be the Java framework. Deployment-Dependency tool will be Maven. MongoDB will be the NOSQL database for storage.
- For the machine learning Python will be used as programming language. As a web framework, Flask will be used. MongoDB will be the NOSQL Database. For the Pyhton Programming Language package management, Anaconda will be used. For scheduling and periodic tasks, CeleryTaskQueue will be used.

3.4.2. Economical Constraints

- Hosting service is needed to deploy this project and the payment for hosting is 20 USD per month.
- Domain name payment for the website is 5 USD per year.
- All framework and libraries to be used will be available and free for commercial-use.
- Travelpayouts, MakeMyTrip and QPX Express API will be used to fetch flight data. Using QPX Express API will cost 0,035 USD per query above 50 query.

3.4.3. Time Constraints

- The first version of this project will be launched in 8 months.
- The system needs to start collecting the data for each flight twelve month before the departure date of that particular flight.
- The system needs to gather at least 10 days of flight data for each flight before displaying the statistics graphically regarding that particular flight.

3.4.4. Ethical Constraints

- The Software will not share users' data with third-parties in order to protect the user privacy.
- The data we collected from web services will not be sold to any other company for any kind of profit.
- We will abide by the IEEE Code of Ethics in the development of the project [9].
- Users are considered to accept the Software as a Service agreement when they use the service.

3.4.5. Sustainability Constraints

- Accuracy of randomly selected predictions will be checked on a daily basis and in case of a drop in accuracy, the admins of the system will be notified.
- There will be advertisements to compensate the spendings.
- The users may send feedback to the program.

3.5. System Models

3.5.1. Scenarios

Scenario 1

Scenario Name: RegisterAsKamil

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil, who flies frequently and wants to learn how he can find cheap flight tickets, learns that there is a website called AeroCast, which makes airfare forecasts.
2. Kamil opens the web page of the AeroCast and decides to register into the AeroCast system.
3. Kamil presses the "Register" button on the homepage and sees the registration page on the screen.
4. Kamil enters his email address.
5. Kamil enters his name.
6. Kamil enters his password and confirms the password.
7. Kamil presses the "Register" button on the registration page.
8. Kamil sees the account activation page on the screen saying that an email with the activation code has been sent to your email account.
9. Kamil opens the email he received from AeroCast and types the activation code from the email into the activation page.
10. Kamil clicks to the "Activate the account" button on the screen.
11. Kamil sees the homepage on the screen with his email information on the top right corner.

Scenario 2

Scenario Name: LogInAsKamilWithUserAccount

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil opens the web page of the AeroCast and presses the "Log In" button on the screen.
2. Kamil sees three options as "Log in with user account", "Log in with Facebook account" and "Log in with Google account".
3. Kamil chooses "Log in with user account" option.
4. Kamil sees the log in page on the screen.
5. Kamil types his email address and password.
6. Kamil sees the homepage on the screen with his name as "profile" button on the top right corner.

Scenario 3

Scenario Name: LogInAsKamilWithGoogleAccount

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil opens the web page of the AeroCast and presses the "Log In" button on the screen.
2. Kamil sees three options as "Log in with user account", "Log in with Facebook account" and "Log in with Google account".
3. Kamil chooses "Log in with Google account" option.
4. Kamil is redirected to Google Login page to login to his Google account.
5. After log in, Kamil is directed to Home Page of AeroCast.

Scenario 4

Scenario Name: LoginAsKamilWithFacebookAccount

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil opens the web page of the AeroCast and presses the "Log In" button on the screen.

2. Kamil sees three options as "Log in with user account", "Log in with Facebook account" and "Log in with Google account".
3. Kamil chooses "Log in with Facebook account" option.
4. Kamil is redirected to Facebook Login page to login to his Facebook account.
5. After log in, Kamil is directed to the home Page of the AeroCast.

Scenario 5

Scenario Name: ViewProfileOfKamil

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil clicks his name on the top right corner of the screen.
2. Kamil sees the small pop-up menu, which includes links to the "Profile", "Tracked Routes and Flights" and "Log Out" pages.
3. Kamil chooses the "Profile" link on the menu.
4. Kamil sees the profile page with his name, email address, "Change password" division, current currency as USD and "Notifications" toggle button as enabled.
5. Kamil views the content of the profile page and presses the homepage logo.
6. Kamil sees the home page on the screen.

Scenario 6

Scenario Name: UpdateProfileOfKamil

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil clicks his name on the top right corner of the screen.
2. Kamil sees the small pop-up menu, which includes links to the "Profile", "Tracked Routes and Flights" and "Log Out" pages.
3. Kamil chooses the "Profile" link on the screen.
4. Kamil sees the profile page on the screen and he clicks onto the email address field in the profile page.

5. Kamil notices that "save" button has appeared at the bottom of the page.
6. Kamil types his new email address into the email address field.
7. Kamil clicks onto the currency field and chooses the new currency from the drop down menu as TL. He also presses the "Notifications" button to disable the notifications.
8. Kamil presses onto the save button.
9. Kamil sees that the "save" button has became unclickable and a messsage saying that your changes are successfully completed has appeared on the page.
10. Kamil views the new content of the profile page and clicks the return button to open the home page.
11. Kamil sees the home page on the screen.

Scenario 7

Scenario Name: ChangePasswordOfKamil

Participating Actor Instances: User Kamil

Flow Of Events:

1. In the profile page, Kamil presses the "Change password" text.
2. Change password panel appears under the "Change password" text.
3. Kamil types the new password and confirms the password.
4. Kamil clicks to the "Save" button on the screen.
5. Kamil sees the profile page with a message saying that your changes are successfully completed and the password has been changed.

Scenario 8

Scenario Name: ForgotPassword

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil opens the home page screen and presses the log in button.

2. Kamil sees three options as "Log in with user account", "Log in with Facebook account" and "Log in with Google account".
3. Kamil chooses "Log in with user account" option.
4. Kamil cannot remember his password and presses the "forgot password?" button on the log in page.
5. On the opened pop-up screen, Kamil enters his email address and clicks on "send" button.
6. Kamil notices that a message has appeared on the page saying that a message has been sent to his email account.
7. Kamil closes the browser window.
8. Kamil opens his email account and clicks to the link to his profile page.
9. In the new browser window, which displays Kamil's profile page, Kamil types his new password and confirms the password.
10. Kamil presses the "Save" button.
11. Kamil sees the profile page with his profile information and a message saying that your changes are successfully completed.

Scenario 9

Scenario Name: searchOnFlightsWithSpecificDepartureDate

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil sees the flight type, number of passengers, origin and destination fields as well as a departure and return dates at the center of the home page.
2. Kamil presses the "Extras" text and views the additional fields as carrier code and cabin class and direct flight option.
3. Kamil enters Ankara as the origin and Istanbul as the destination airports.
4. Kamil chooses 05-05-2017 as the departure date and 15-05-2017 as the return date from the calendar which opens when he clicks on departure and return date fields.

5. Kamil selects round trip as the flight type and economy as the cabin class. He selects one adult and no child.
6. Kamil presses the "Forecast Airfares" button.
7. Kamil sees the results of the search on the screen.

Scenario 10

Scenario Name: searchOnFlightWithARangeOfDepartureDates

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil sees the origin, destination, departure date, return date and travelers fields on the screen.
2. Kamil presses the "Extras" button and views the additional fields as carrier code and cabin class and direct flight option.
3. Kamil enters Istanbul as the origin and London as the destination.
4. Kamil chooses from 08-05-2017 to 12-05-2017 as the departure dates and from 15-05-2017 to 18-05-2017 as the return dates from the calendar which opens when he clicks on the departure and return date fields.
5. Kamil selects return trip as the flight type, TK as the carrier code and business as the cabin class. He selects one adult and no child.
6. Kamil presses the "Forecast Airfares" button.
7. Kamil sees the results of the search on the screen.

Scenario 11

Scenario Name: TrackingRoutesFromIstanbulToLondon

Participating Actor Instances: User Abbas

Flow Of Events:

1. Abbas sees the results of his search as two separate lists. Abbas notices that the first list contains the results for departure dates from 24.08.2017 to 28.08.2017 whereas the second list contains the results for return dates 15-05-2017 to 18-

05-2017. Abbas notices that each result includes the purchase date, expected price and carrier code of the flight.

2. For each departure date, Abbas examines the purchase date that the price of the flight is expected to be the lowest.
3. For each return date, Abbas examines the purchase date that the price of the flight is expected to be the lowest.
4. Abbas chooses the Cheapest Tab to sort the results.
5. Abbas sees the sorted results, which start from the departure date with the lowest expected price and ends at the departure date with the highest expected price.
6. Abbas clicks "Track Route" to track all departure dates.
7. Abbas sees the Log In pop-up since he needs to log in to access the tracking service.
8. Abbas logs in with his user account and he is redirected to the search results page after logging in.
9. Abbas sees that his search result is automatically added to the tracked routes.

Scenario 12

Scenario Name: TrackingFlightsFromIstanbulToLondon

Participating Actor Instances: User Abbas

Flow Of Events:

1. Abbas sees the result of his search which has departure dates from 24.08.2017 to 28.08.2017 as a list.
2. Abbas chooses the "Fastest" tab to sort the results.
3. Abbas sees the sorted results which start from the departure date with the shortest flight duration and ends at the departure date with the longest flight duration.
4. Abbas presses "Track" button near the first departure date in the list to track the flight which departs at 25.08.2017.

5. Abbas also presses "Track" button near the third departure date in the list to track the flight which departs at 26.08.2017.
6. Abbas notices that the two buttons has changed from "Track" to "Tracked".
7. Abbas returns to the home page by clicking the return button.

Scenario 13

Scenario Name: TrackingFlightsFromAnkaraToIstanbul

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil sees the result of his search which has departure date of 24.08.2017.
2. Kamil sees one flight with the departure date of 24.08.2017 as a result.
3. Kamil sees the price change graph below the flight information.
4. Kamil clicks "Track" buttom to track the departure date.
5. Kamil notices that the "+" button has been changed to "Tracked".
6. Kamil returns to the home page by clicking the "Return" button.

Scenario 14

Scenario Name: TrackingForIstanbulParisFlight

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil sees the result of his search which has departure date of 24.08.2017.
2. Kamil notices that there is a message on top of the page saying that due to the inadequate amount of data for the particular flight specified in the search, the predictions are made based on the data from other flights on the same route and as the new data arrives for this flight, the predictions will be updated.
3. Kamil clicks "Track Flight" button.
4. Kamil notices that the "Track Flight" button has been changed to "Tracked".

Scenario 15

Scenario Name: ViewTrackedResults

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil clicks his name on the top right of the screen.
2. Kamil sees the small pop-up which includes links to the "Profile", "Tracked Routes and Flights" and "Sign Out" pages.
3. Kamil chooses "Tracked Routes and Flights".
4. Kamil sees the list of the tracked routes. Kamil notices that above the result list there are two links that are for tracked routes list and tracked flights list.
5. Kamil clicks on a route which directs to the result page which is same as the search result of the route except "Track" button.
6. Kamil returns to the previous page by clicking on return button of the browser.
7. Kamil clicks on the flight text on the top of the page. Kamil sees the list of tracked flights in the screen.
8. Kamil clicks on a flight to look at the detailed information.
9. Kamil sees the page for this specific flight.

Scenario 16

Scenario Name: ViewGraphForTrackedFlight

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil is on the "Tracked Routes and Flights" from the menu which opens when he clicks on his name at the top right of the screen.
2. Kamil clicks on the flight text at the top of the page.
3. Kamil clicks on the first flight in the screen which departs at 25.08.2017.
4. Kamil sees the page for this specific flight.
5. Kamil sees the Price-Date graph on the screen.
6. Kamil moves the cursor to the line and sees the price for a specific date on the line.

7. Kamil does not like the price change as it varies too much and clicks the "Untrack Flight" button at the left of the web page..
8. Kamil clicks on the aerocast logo to return the homepage.

Scenario 17

Scenario Name: UntrackFlight

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil wants to untrack one of his flights so he clicks on the "Tracked Routes Flights" link on the profile pop-up.
2. Kamil clicks on the flight text on the top of the page.
3. Kamil sees the list of the flights he has tracked.
4. Kamil finds the Ankara-İstanbul flight with the departure date 22.08.2017 and clicks "Untrack" button.
5. Kamil sees that the flight is removed from the list of tracked flights.

Scenario 18

Scenario Name: NotificationForPredictionUpdate

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil sees an email in his email account regarding an update on a tracked flight.
2. He wonders what might have happened and goes to the "aerocastapp.com".
3. He logs into his account.
4. He sees the "notifications" text with the number "1".
5. He clicks the "notifications" text and sees a mini pop-up menu regarding the notifications.
6. Kamil sees the notification "Your best deal for Ankara-Istanbul flight has been updated".

7. Kamil clicks on the notification and views the new purchase date for the tracked flight.

Scenario 19

Scenario Name: FeedbackFromKamil

Participating Actor Instances: User Kamil

Flow Of Events:

1. Kamil clicks on the button "Give Feedback" at the bottom of the page.
2. Kamil views the feedback page.
3. Since Kamil is logged into the AeroCast, his email address is written in the email textbox by the system.
4. Kamil wants to be informed on his other email as well
5. so he writes "kamil.abbas@bilkent.edu.tr" on email textbox.
6. Kamil writes "Page is too Slow" on the title textbox.
7. Kamil writes "Your estimation speed is okay but pages are opening too slow. Please get rid of this problem".
8. Kamil clicks on the "Submit" button.
9. Kamil sees the information popup which says "Thank you for your feedback, we will inform you about this subject as soon as possible".

3.5.2. Use Case Model

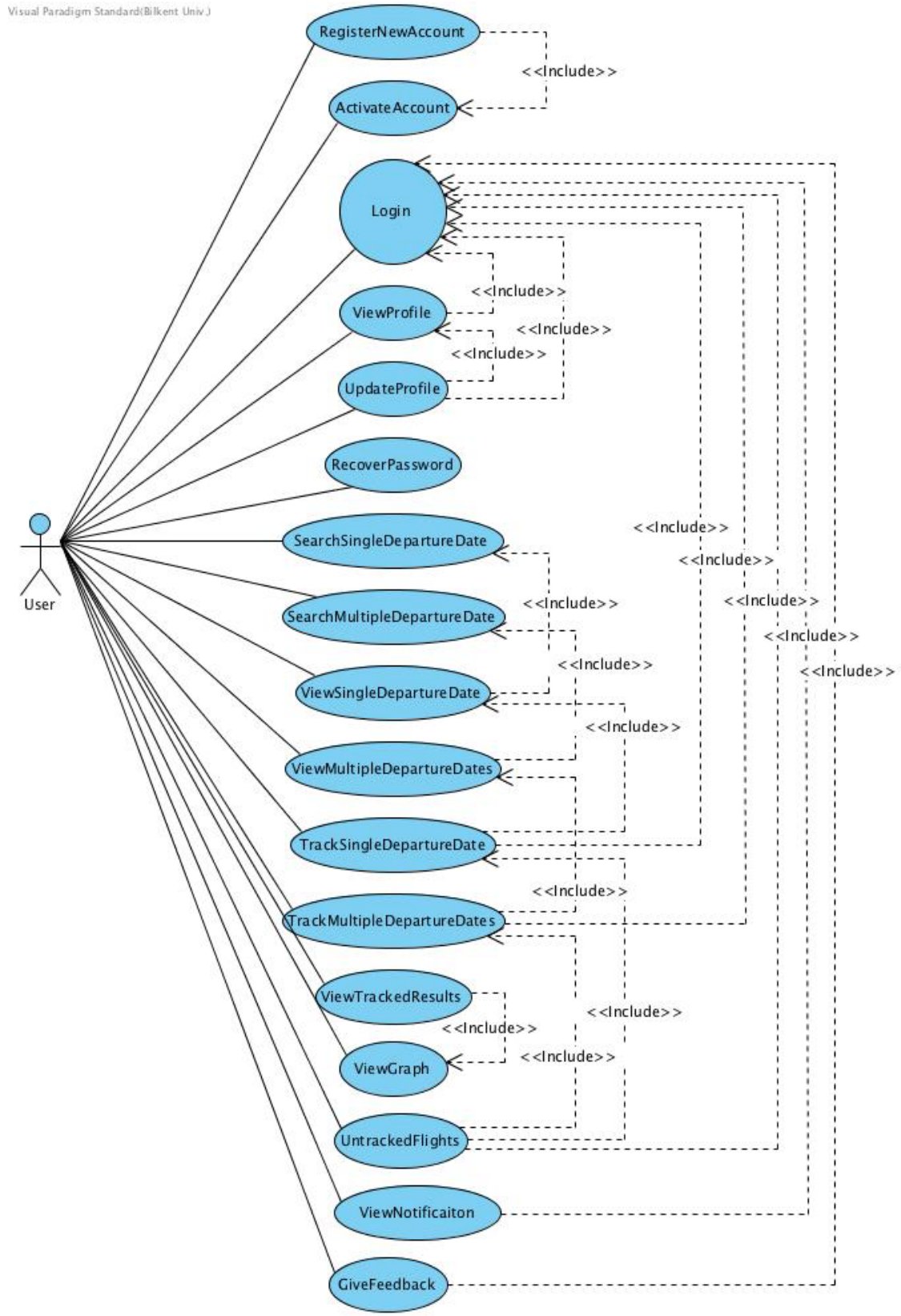

Figure 1 - Use Case Diagram

3.5.2.1. Register

Use Case Name: Register

Participating Actors: User

Main Flow of Events:

1. The user clicks on the "Sign Up" button.
2. "Sign Up" pop-up appears.
3. The user enters his/her emai address.
4. The user enters his/her name.
5. System checks if the email has been used.
6. System confirms that email is available to use.
7. The user enters his/her password.
8. The user enters his/her password again.
9. System checks if passwords are within the limitations and two passwords match.
10. System confirms that passwords are usable and they match
11. The user agrees to the terms and conditions of use.
12. The user presses 'Sign Up' button.
13. System creates the user as unactivated and open the activation popup.
14. System sends an email to the user which includes the activation code.

Alternative Flow of Events 1:

A. The user enters an e-mail that is not available.

4A. System checks whether e-mail is available or not.

5A. System detects that email is not available and warns user to enter an e-mail that is available.

Alternative Flow of Events 2:

B. The user enters passwords which are not within limitations and matching with each other.

4B. System checks if passwords are within limitations and two passwords match.

5B. System detects that passwords are not within limitations and two passwords are not matching.

6B. Warns user to enter proper passwords which are matching with each other.

Entry Conditions:

- User is on HomePage OR
- Login Pop-up is opened.

Exit Conditions:

- User is on Homepage OR
- Register pop-up has opened with a warning

3.5.2.2. Activate Account

Use Case Name: Activate Account

Participating Actors: User

Main Flow of Events:

1. The user receives the mail with an activation code.
2. The user sees the activation code and types it into the textbox on Activation Popup.
3. System checks whether the activation code is true or not.
4. System confirms the code and activates the account.
5. The website "aerocastapp.com" is reloaded with an successful activation info.

Entry Conditions:

- Activation popup is opened..

Exit Conditions:

- User is on Homepage

3.5.2.3. Log In

Use Case Name: Log In

Participating Actors: User

Main Flow of Events:

1. The user enters his/her e-mail.
2. The user enters his/her password.
3. The user presses 'Log In' button.

4. System checks whether inputs are correct and matching or not.
5. System confirms that credentials are correct and they match, it closes the Login popup.
6. System refreshes the page with the member functionalities.

Alternative Flow of Events:

1. The user enters wrong e-mail and/or password or they do not match
2. System checks whether inputs are correct and matching or not
3. System detects wrong credentials and does not allow Logging in
4. Login popup appears again with a red warning text

Entry Conditions:

- Login Popup is opened

Exit Conditions:

- Login Popup is opened OR
- Login popup is on and the authentication failed

3.5.2.4. View Profile

Use Case Name: View Profile

Participating Actors: User

Main Flow of Events:

1. The user clicks the "Profile" button on the top right corner of the screen.
2. A small pop-up menu opens, which includes brief information about the profile and links to the "Profile", "Tracked Flights And Routes" and "Log Out" pages.
3. The user presses "Profile" link on the screen.
4. Profile page, which includes user's email address, name, "Change password" button, current currency and a toggle button to enable or disable notifications, opens.
5. The user examines the currency and notification settings.
6. The user return to the home page.

Entry Conditions:

- Any page opened at the website and user is logged in.

Exit Conditions:

- User is on homepage.

3.5.2.5. Update Profile

Use Case Name: Update Profile

Participating Actors: User

Main Flow of Events:

1. The user edits e-mail, name, current currency and change status of notifications.
2. The user clicks "Save" button.
3. System checks whether inputs are correct or missing.
4. System updates the user's profile.

Alternative Flow of Events 1:

A. The user enters an e-mail that is not available.

3A. System checks whether e-mail is available or not.

4A. System detects that email is not available and warns user to enter an e-mail that is available.

Alternative Flow of Events 2:

B. The user leaves an input blank.

3B. System checks every input whether inputs are missing.

4B. System detects that some inputs are missing and warns the user to fill those missing inputs.

Entry Conditions:

- Profile page is opened and the user is logged in.

3.5.2.6. Recover Password

Use Case Name: Recover Password

Participating Actors: User

Main Flow of Events:

1. The user clicks the "Forgot Your Password?" button on Login Popup.

2. The user enters his/her e-mail adress.
3. System checks whether e-mail adress is registered or not.
4. System sends an e-mail which contains password recovery link.
5. The user sees the recovery e-mail.
6. The user clicks the password recovery link.
7. The user is redirected to profile page which the user can change the password.
8. The user enters his/her new password.
9. The user re-enters his/her password.
10. System checks if passwords are within the limitations and two passwords match.
11. System confirms that passwords are usable and they match.
12. The user clicks "Save" button.
13. System changes his/her account's password.

Alternative flow of Event:

A. The user enters the passwords which are not within the limitations and matching with each other.

10A. System checks if passwords are within the limitations and two passwords match.

11A. System detects that passwords are not within limitations and two passwords are not matching.

12A. System warns user to enter proper passwords which are matching with each other.

Entry Conditions:

- Login Popup is opened

Exit Conditions:

- The user is on Homepage

3.5.2.7. Search

Use Case Name: Search

Participating Actors: User

Main Flow of Events:

1. The user chooses the location where the flight departures.
2. The user chooses the location where the flight arrives.
3. The user selects the number of passengers and their types.
4. The user clicks the "extras" button.
5. System shows "cabin class" and "airlines" options about the flight.
6. The user selects his/her preferred cabin class.
7. The user selects his/her preferred airlines.
8. The user chooses departure date(s).
9. The user clicks the "Search" button.
10. System checks if the departure date(s) is valid or not.
11. System displays the results.

Alternative flow of Event:

A. User did not choose valid departure date(s)

10A. System warns the user to select a valid departure date

Entry Conditions:

- User is on Homepage

Exit Conditions:

- Result page is opened.

3.5.2.8. View Single Departure Date

Use Case Name: ViewSingleDepartureDate

Participating Actors: User

Main Flow of events:

1. The user chooses a single departure date and searches for result.
2. AeroCast renders the result page which includes a two-way flight having a purchase date estimation, departure date, carrier code and a price graph for each flight.
3. AeroCast draws a price graph which points out minimum price and current price. Above the graph carrier code and other flight information are written. At the right of the graph, AeroCast displays the "Track" button.

4. The user examines the flight information at the left of the page.
5. The user finds the price of the flight on any day by hovering the mouse pointer on the graph.
6. The user examines the return flight informaition at right of page.
7. The user returns to the home page.

Alternative Flow of Events:

2A. The chosen date has no data on the system so AeroCast inform the user as required.

3A. AeroCast uses the data of another date to make prediction on the chosen flight.

4A. AeroCast shows its prediction on the graph. The past records of the flight are left empty.

Entry Conditions:

- The user searches a flight with single departure date.
- The user is on result page.

Exit Conditions:

- The user is on the home page.

Quality Requirements:

- The result of the search is displayed on the page less than five seconds.

3.5.2.9. View Multiple Departure Dates

Use Case Name: ViewMultipleDepartureDates

Participating Actors: User

Main Flow of events:

1. The user chooses multiple departures date and searches for the results.
2. AeroCast displays the results as a list of the best five flights chosen by the sorting criteria.
3. AeroCast displays each flight as a panel which includes flight date, minimum price and track sign.
4. The user chooses one of the results by clicking on the flight panel.

5. The panel expands vertically and show the carrier code and price graph of the flight.

6. If the user expands the panel to view the graph, the ViewGraph use case is used.

7. The user closes the flight details by clicking on the panel again.

8. The user examines the return flight panel.

9. The user clicks on "Sort By" button.

10. The user selects "Duration" in dropdown menu.

11. The user notice the changes in the return flight list.

12. The user returns to home page.

Alternative Flow of Events 1:

1A. The user specifies less than five departure dates in the search.

2A. AeroCast lists all of the chosen dates according to the chosen sorting criteria.

Alternative Flow of Events 2:

2B. When one of the chosen dates has no data, AeroCast shows prediction based on another dates' data.

3B. AeroCast displays each flight as a panel which includes flight date and track sign.

4B. The user clicks on the flight panel.

5B. Panel expands and AeroCast give information that prediction is done based on another dates' data. AeroCast shows the graph such that it has no past records but only prediction.

Entry Conditions:

- The user searches multiple dates for a route.
- The user is on result page.

Exit Conditions:

- The user is on the home page.

Quality Requirements:

- The result of the search is displayed on the page in less than five seconds.

3.5.2.10. Track Single Departure Date

Use Case Name: TrackSingleDepartureDate

Participating Actors: User

Main Flow of events:

1. The user sees the results of the search.
2. AeroCast displays the flight information on top of the page and the graph in center of the page and "Track" button at the bottom of the page.
3. The user click on "Track" button.
4. AeroCast saves the flight into the list of tracked flights of the user and it changes the button from "Track" to "Tracked".
5. The user returns to Home Page

Entry Conditions:

- The user searched a flight with single departure date.
- The user is on result page.

Exit Conditions:

- The user is on the home page.

3.5.2.11. Track Multiple Departure Dates

Use Case Name: TrackMultipleDepartureDates

Participating Actors: User

Main Flow of events:

1. The user sees the results of the search.
2. AeroCast lists the results of the searched departure dates.
3. The user chooses one of the results by clicking on the flight panel.
4. AeroCast displays details of the flight such as information, price graph and "Track" button.
5. The user clicks on "Track" button.
6. The flight is tracked and "Track" button turned "Untrack".
7. The user clicks the flight again to close the flight panel.
8. The list appears again.
9. The user clicks on the "Track" button at the bottom of the page to track all flights within departure dates.

10. Search is tracked and "Track" button turned "Untrack".

11. The user returns to home page.

Alternative Flow of Events 1:

2A. The user sees the results list on the screen.

3A. The user clicks on the "+" sign right-most of the flight panel.

4A. The chosen flight is tracked and "+" sign disappears.

5A. The user returns to home page.

Entry Conditions:

- The user searches multiple departure dates.
- The user is on the result page.

Exit Conditions:

- The user is on the home page.

3.5.2.12. View Tracked Results

Use Case Name: ViewTrackedResults

Participating Actors: User

Main Flow of events:

1. The user chooses the "Tracked Flights " link on the home page.
2. AeroCast displays the Tracked Flights and Routes page on the screen.
3. The user views the tracked searches on the screen.
4. The user examines the tracked searches.
    a. The user examines all of the departure dates, and specification of the tracked search.
    b. If the user presses on a specific departure date, ViewMultipleDepartureDates use case is used.
5. The user clicks "Flights" button.
6. The user views the specific departure date
    a. The user examines the specific departure date.
    b. The user examines the purchase date and departure date predictions for the specific departure date.

c. If the user presses on a specific departure date, the ViewGraph use case is used.

Alternative Flow of Events 1:

2A. The list of searches is empty.

3A. The user returns to home page.

Alternative Flow of Events 2:

5B. The list of flight dates is empty.

6B. The user returns to home page.

Entry Conditions:

- The user is on the home page.
- The user is logged into AeroCast.

Exit Conditions:

- The user is on the home page.

Quality Requirements:

- The list of tracked results is loaded into the page no later than two seconds.

3.5.2.13. View Graph

Use Case Name: ViewGraph

Participating Actors: User

Main Flow of events:

1. The user presses on a flight result.
2. AeroCast responds by representing a graph to the user. The graph includes the history of price changes as a solid line and the expected future prices as a dashed line.
3. The user views the graph on the page.
    a. The user moves the cursor on top of the points on the graph.
    b. AeroCast displays the price and date values of the specific points on the graph.
    c. The user examines the price and date values.
4. The user closes the graph by clicking the return button.

Entry Conditions:

- The user is on tracked departure dates page. OR
- The user is on the search results page.

Exit Conditions:

- The user is on the tracked departure dates page. OR
- The user is on the search results page.

Quality Requirements:

- The graphs are loaded into the page no later than two seconds.

3.5.2.14. Untrack Flights

Use Case Name: UntrackDepartureDates

Participating Actors: User

Main Flow of events:

1. The user presses the "Untrack" button next to a tracked flight.
2. AeroCast removes the chosen flight from the list of tracked flights of the user.
3. The user presses "Untrack" button at the bottom..
4. AeroCast removes the chosen tracked search and all departure dates of that search from the list of tracked departure dates of the user.

Alternative Flow of Events:

1A. The user presses the untrack button located in the upper right corner of the graph page in order to untrack a specific departure date.

   a. AeroCast removes the chosen departure date from the list of tracked specific departure dates of the user.
   b. AeroCast disables the untrack button in the graph page.
   c. After the user leaves the graph page, the untracked departure date is no longer available in the Tracked Flights page of the user.

Entry Conditions:

- The user is on the Tracked Flights page.
- The user is on the Graph page of a tracked flight.

- The user is logged into AeroCast.

Exit Conditions:

- The chosen departure date is untracked.

3.5.2.15. View Notifications

Use Case Name: ViewNotifications

Participating Actors: User

Main Flow of events:

1. The user receives an email from AeroCast regarding a new notification.
2. The user opens the homepage screen.
3. AeroCast displays the number of new notifications on the notifications symbol located in the right top corner of the page.
4. The user presses on the notifications symbol.
5. The user views the new notifications in a pop-up menu.
6. The user selects a notification from the pop-up menu.
7. AeroCast displays the Tracked Flights page which includes the departure date with the notification represented as a star shaped indicator.
8. The user views the departure date having the notification.
9. AeroCast removes the star shaped notification indicator from the tracked departure date.
10. AeroCast removes the notification from the pop-up menu opened from the notifications symbol.

Alternative Flow of Events 1:

1A. The notifications pop-up menu is empty since there are no tracked results of the user or there are no new notifications for the tracked results of the user.

2A. The user closes the notifications pop-up menu.

Alternative Flow of Events 2:

6B. The user removes the notification from the pop-up menu without openning the tracked departure dates.

9B. AeroCast removes the star shaped notification indicator from the tracked departure dates.

10B. AeroCast removes the notification from the pop-up menu.

Alternative Flow of Events 3:

1C. The user directly opens the Tracked Departure Dates page by clicking on the Track Departure Dates link in the home page. The user skips the events from 4 to 6 in the main flow of events.

Entry Conditions:

- The user is logged into AeroCast.

Exit Conditions:

- The user is on the home page.

Quality Requirements:

- The lists of tracked results is loaded into the page no later than two seconds.

3.5.2.16. Give Feedback

Use Case Name: Give Feedback

Actors: User

Main Flow of Events:

1. The user clicks the "Give Feedback" button at the bottom of the page.
2. Feedback page appears on the screen.
3. The user enters his/her e-mail address, title of the feedback and the feedback message to the textbox.
4. The user clicks the "Submit" button.
5. The user returns to home page.

Alternative Flow of Events:

3A. Since user is logged in and his email is already appears in email textbox, user just enters title and message of feedback.

4A. The user clicks the "Submit" button.

5A. The user returns to home page.

Entry Conditions:

- User is on the Home Page

Exit Conditions:

- User is on the Home Page

3.5.2.17. Give Flight Feedback

Use Case Name: Give Flight Feedback

Actors: User

Main Flow of Events:

1. The user sees the "Feedback" popup on the screen.
2. AuroCast displays the flights which are expired.
3. The user clicks "I liked" button.
4. AuroCast redirects the user to the Home Page

Entry Conditions:

- User is on the Home Page
- The user is logged into AeroCast.
- The tracked flights and/or routes expire.

Exit Conditions:

- User is on the Home Page

3.5.3. Object And Class Model

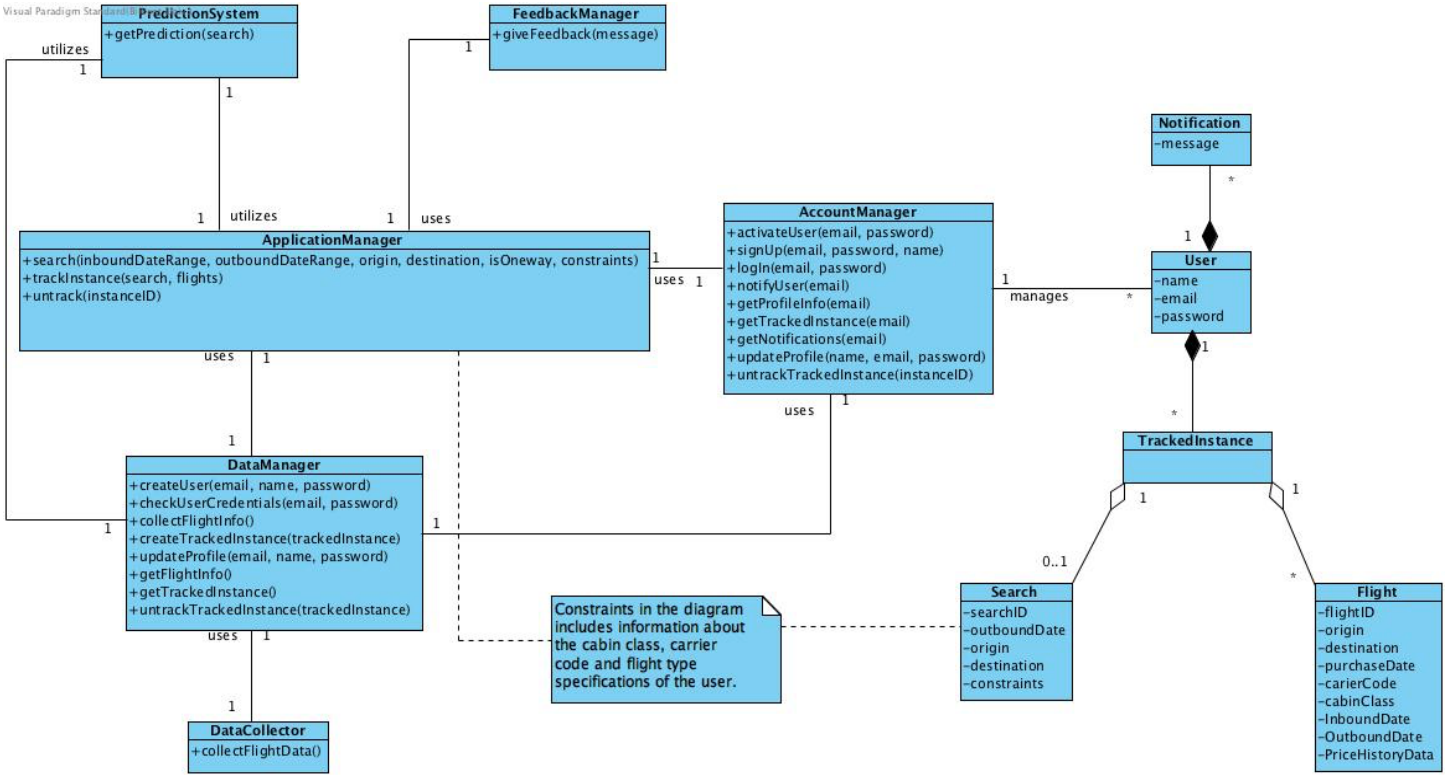

Figure 2 - Object and Class Diagram

User: This class will represent a user of the application. A user will have identifying information such as name, email and password. User class also will contain the current tracked flights and routes.

TrackedInstance : This class will represent tracked flights or tracked routes of the application. Tracked Instance will contain search and flight.

Search: This class will represent a search of the application. A search will have specific search id, outbound date, origin airport , destination airport and constraints which is specified by user.

Flight: This class represent a flight of the application. It will have flight information. This information will be origin and destination airport, carrier code, cabin class , inbound and outbound date. .Flight class will also contain the prediction result of a specific date and price history of the cheapest flight in this specific date.

Notification: This class will represent notifications. It will contain the notification message.

AccountManager: This class will control and maintain the accounts of a user. It will be responsible for creating a new account, singing the user in, sending activation email and password recovery email and notification email. AccountManager class will also responsible for accesing the user's tracked flight and route information.

DataCollector: This class will be responsible for collecting data from 3rd party applications such as TravelPayout, QPX Express. This data will contain data the cheapest ticket for every route.

FeedbackManager: This class will be resposible for collect feedback from user.

PredictionSystem: This class will be used to generate price prediction of the flight based on the user's selected constraints by using data provided by the DataManager.

DataManager: This class will be responsible for managing the users of application, users' tracked flights and routes and data provided by DataCollector.

ApplicationManager: This class will be used to serve user for searching , tracking flight and route, tracking flight and/or route. This class will also be responsible for generating search instance.

3.5.4. Dynamic Models

3.5.4.1. Sequence Diagrams

- **Register Sequence Diagram**

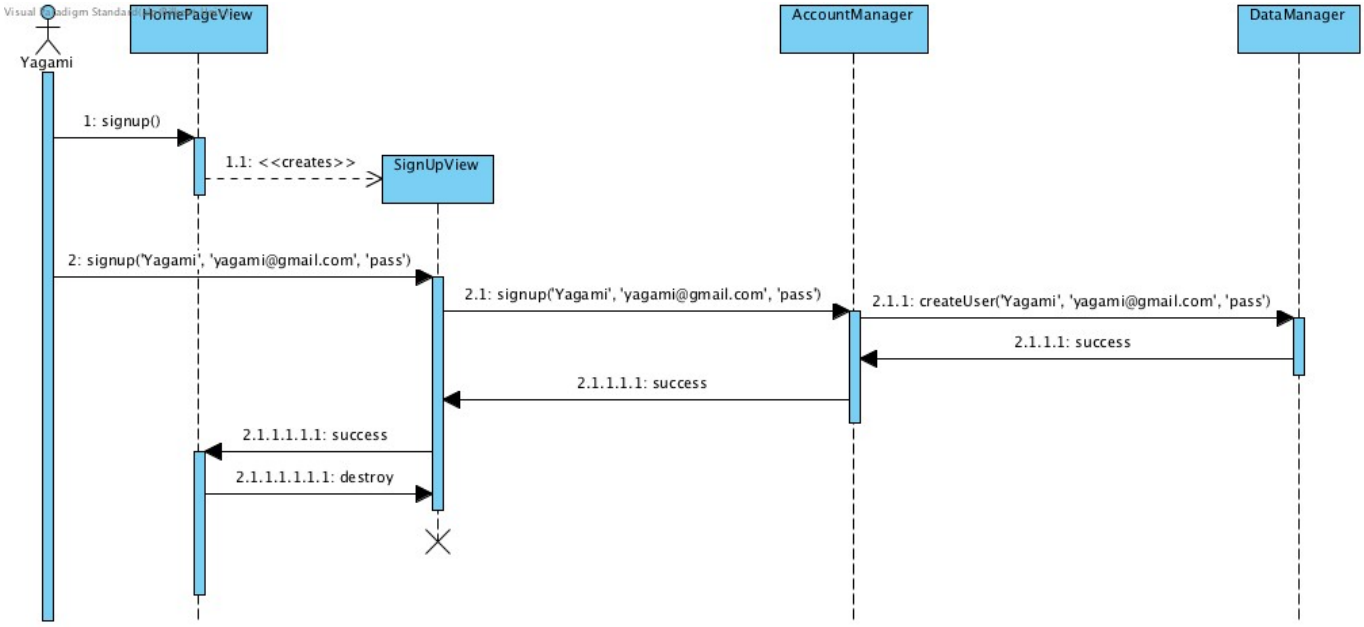

Figure 3 - Register Sequence Diagram

The sequence diagram represents the sequence of events for registering into AeroCast system. Yagami starts to register in the *HomeView* by choosing the sign-up option. The user fills and submits the form in the *SignUpView* to continue to the registration. The necessary information is passed to the *DataManager* to create the new user account. After the successful creation of the user in the system, the *DataManager* and *AccountManager* signals the success and *SignUpView* notifies the *HomeView* about the successful creation of the new user account. The *SignUpView* is removed after the successful creation of the account and the user is directed to the *HomeView*. *HomeView* shows the information related to the newly created user account.

- **Successful Login Sequence Diagram**

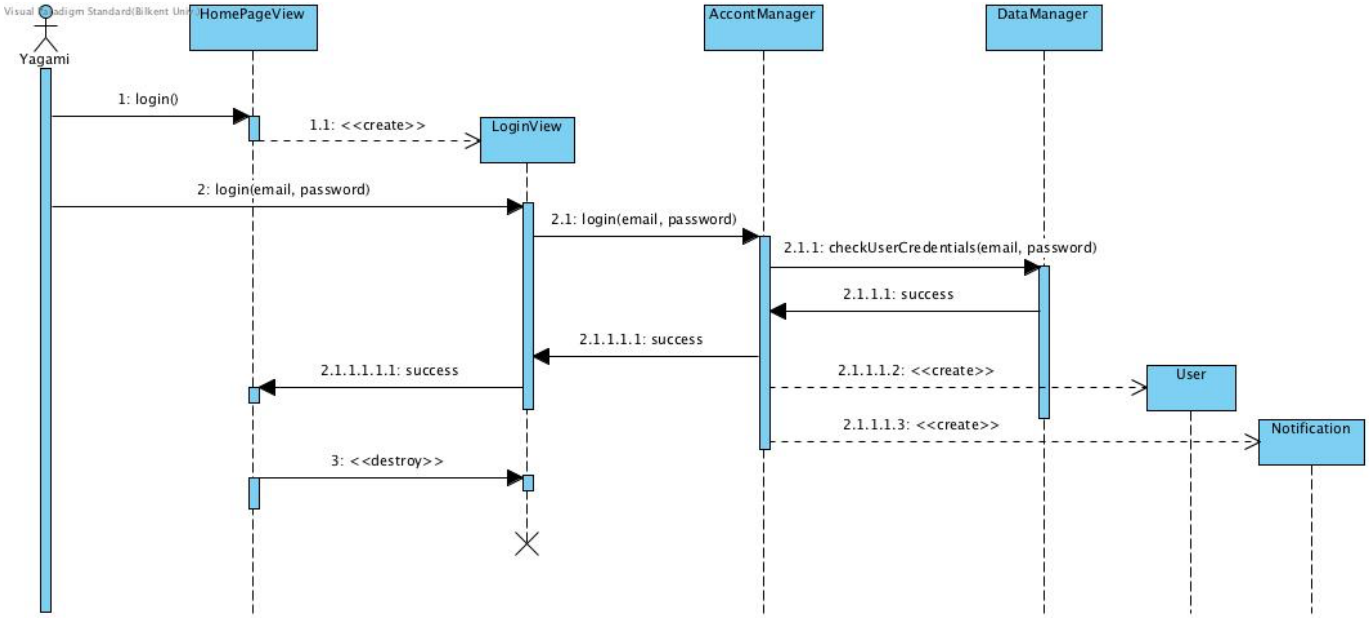

Figure 4 - Succssful Login Sequence Diagram

The sequence diagram represent the sequence of events for logging in AeroCast system successfully. Yagami starts to login in the *HomeView* by choosing the login option. The user fills and submits the login form. The necessary information is passed to the *DataManager* to check credentials of user. *DataManager* checks the credentials and signals the success to the *AccountManager*. *AccountManager* will create the user object and notification objects. *AccountManager* signals the success and login view page notifies the *HomeView* about the successful login operation. The *LoginView* is removed and the user is directed to the *HomeView*.

- **Failed Login Sequence Diagram**

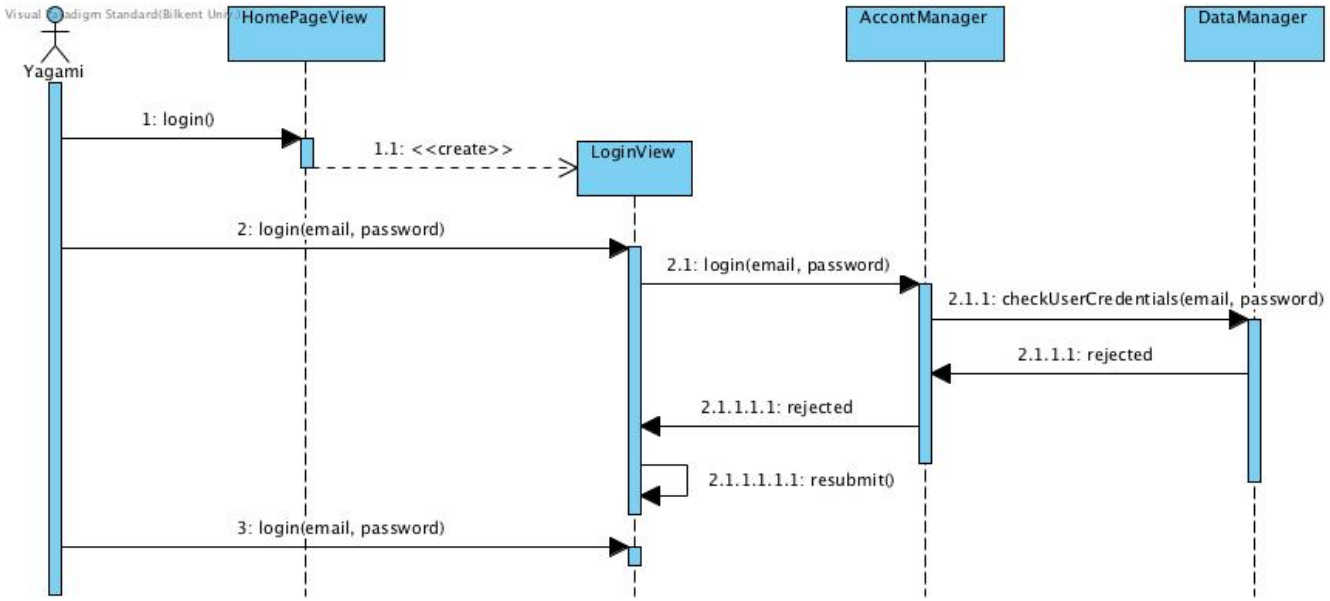

Figure 5 - Failed Login Sequence Diagram

The sequence diagram represent the sequence of events for logging in AeroCast system with wrong credentials. Yagami starts to login in the *HomeView* by choosing the login option. Yagami fills and submits the form in login form. The necessary information passed to data manager to check credential of user. *DataManager* checks the credentials and signals rejected to the *AccountManager*. *AccountManager* signals the rejected and *LoginView* notifies the home page view about the unsuccessful login operation. The error message is shown.

- **Update Profile Sequence Diagram**

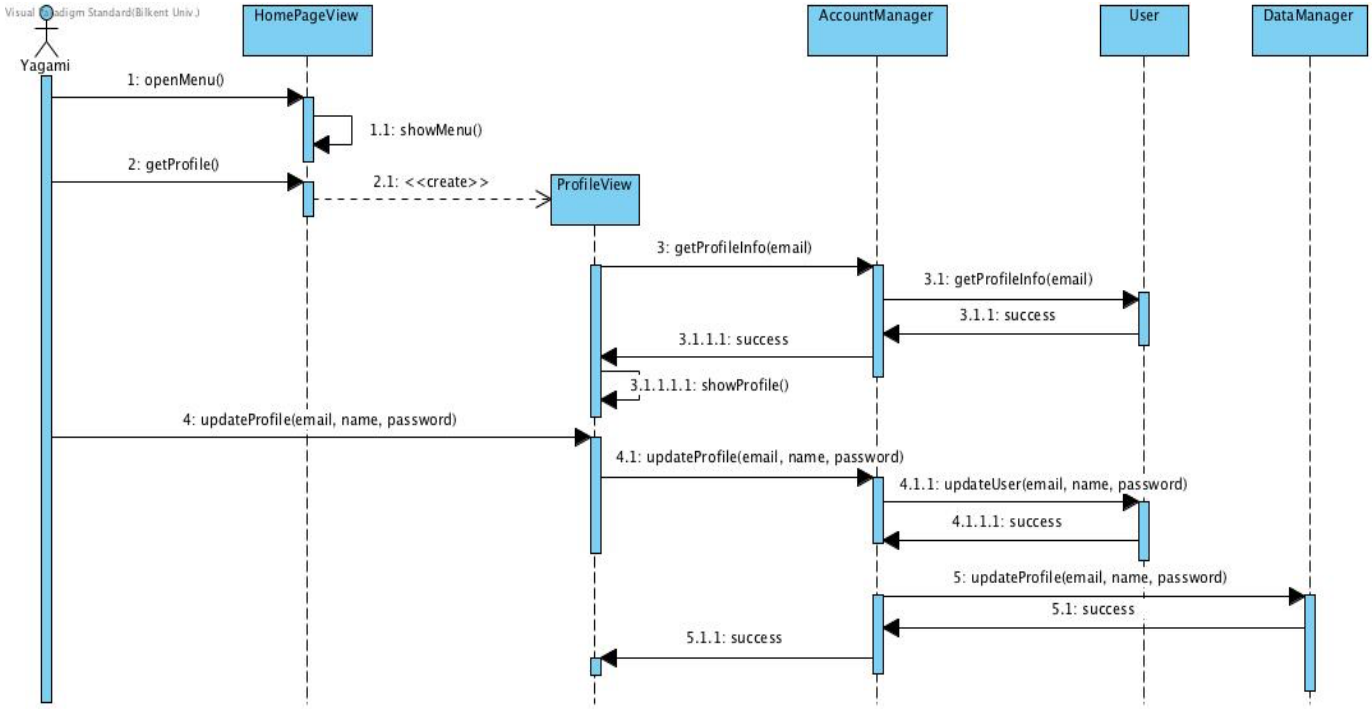

Figure 6 - Update Profile Sequene Diagram

The sequence diagram shows the actions taken by the user to display and update the profile information. Yagami first opens the menu in the *HomeView* that shows the relevant links to the profile, tracked routes and flights and logout pages. Yagami chooses the profile option and *ProfileView* is displayed in the screen. The *ProfileView* receives the profile information of the user from the *AccountManager*, which gets the profile information from the user entity in the system. *ProfileView* displays the profile information obtained from the *AccountManager*. Yagami chooses to update the profile information by changing the displayed values in the *ProfileView*. Yagami saves the changes and *ProfileView* sends the change to *AccountManager* to update the profile. *AccountManager* updates the user entity and requests the *DataManager* to update the data related to the user in the database. After the successful update of the user profile, the new profile information is shown in the *ProfileView*.

- **Search and Track Round Trip Flights Sequence Diagram**

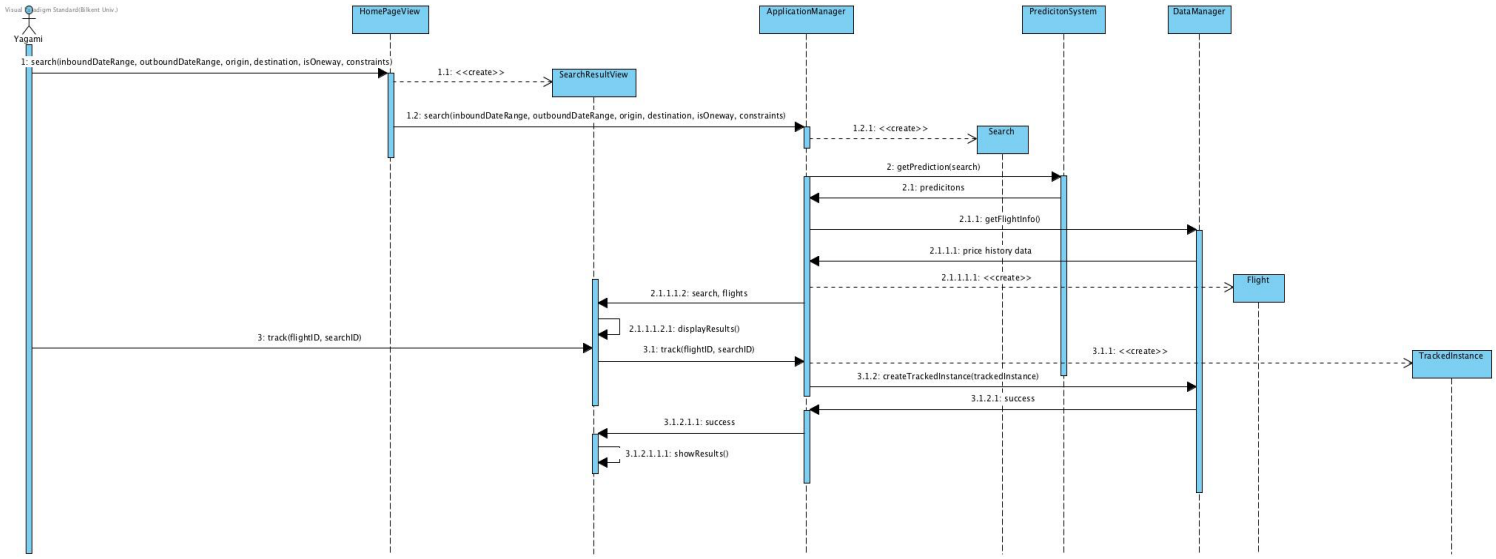

Figure 7 - Round Trip Sequence Diagram

The sequence diagram represents the sequence of events taken by the user to make a new search and track the results of the search. Yagami specifies the inputs to make a new search in the *HomePageView* and submits his specifications by clicking on the search button. The *SearchResultView* is created and the search made by the user is passed to the *ApplicationManager*. *ApplicationManager* creates the search entity and passes the search to the *PredictionSystem*. *PredictionSystem* generates the predictions for the search and gives back the predictions to the *ApplicationManager*. *ApplicationManager* gets the previous flight data from the *DataManager* and merges the predictions and flight data to create the flight entity. The content of the flight entity is passed to the *SearchResultView* by the *ApplicationManager* and the search results are displayed in the screen. After seeing the results, Yagami chooses to track one of the flights. *AppliationManager* creates the new *TrackedInstance* for the flight and requests the insertion of the instance to the database. *DataManager* adds the *TrackedInstance*. After the successful completion of the addition operation, the *SearchResultView* displays the page again.

- **View Tracked Flight and Untrack Sequence Diagram**

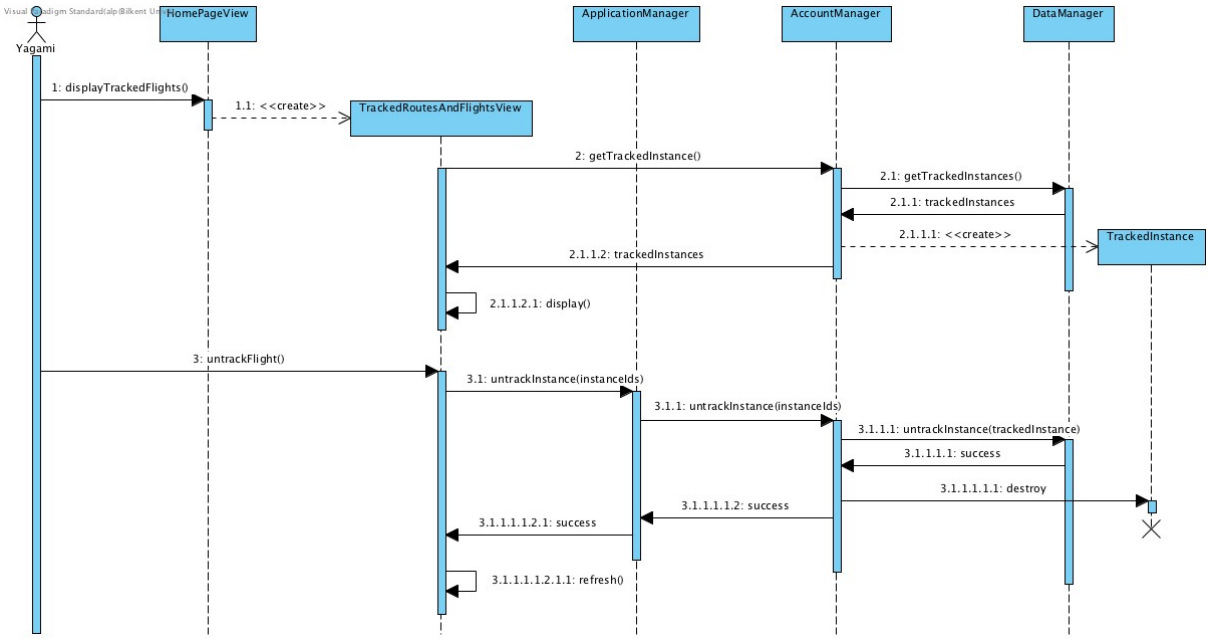

Figure 8 - View tracked Flight And Untrack Sequence Diagram

The diagram goes through the sequence of events when users view tracked flight and untrack one of the flights they track. For this case, a user, in this case Yagami, interacts with the home page screen named *HomePageView*. Upon clicking on the Tracked Routes and Flights button, Yagami will be interacting with *TrackedRoutesAndFlightsView* where he will see the list of flights tracked by the user. Subsequently, the *TrackedRoutesAndFlightsView* will ask *AccountManager* to give the tracked flights and *AccountManager* ask *DataManager* for the tracked instances of a particular user. After acquiring the related data, *AccountManager* creates *TrackedInstance* objects and delivers them to *TrackedRoutesAndFlightsView*. Moreover, in order for Yagami to untrack flights, he interacts with the *TrackedRoutesAndFlightsView*. He clicks on the untrack button, and *TrackedRoutesAndFlightsView* will ask *ApplicationManager* to untrack the instances whose ids are given. Similarly, *ApplicationManager* will ask *AccountManager* to untrack the instances. Consecutively, *AccountManager* asks *DataManager* to untrack the given

instances, destroys the *TrackedInstances* objects and return success code to the *TrackedRoutesAndFlightsView*.

- **View Flight Notification Sequence Diagram**

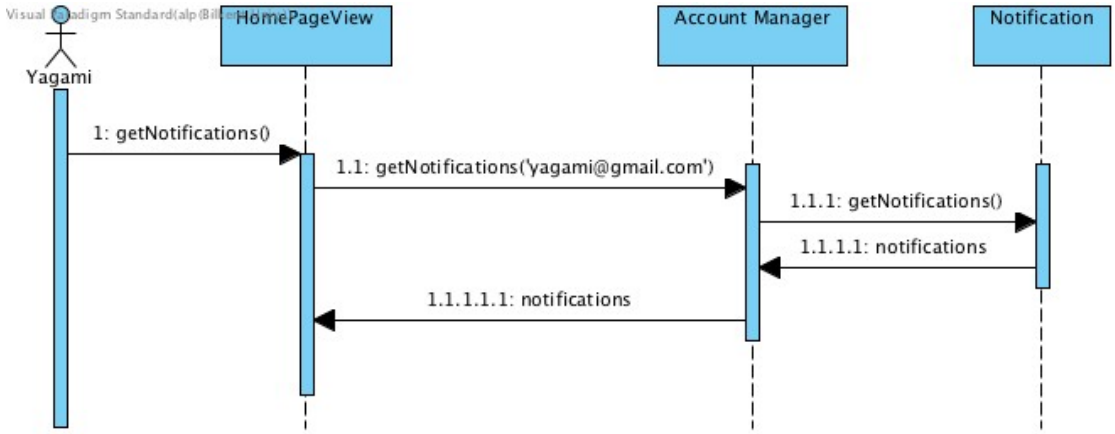

Figure 9 - View Flight Notification Sequence Diagram

The sequence diagram represents the sequences of events taken by the user to view notifications. Yagami sees the number of new notifications and their short versions on the homepage by clicking on notification panel. *ApplicationManager* gets notifications from *AccountManager* with the Yagami's email address. *AccountManager* returns notifications and *ApplicationManager* views Yagami's notifications on notification panel on the *HomePageView*.

- **Send Feedback Sequence Diagram**

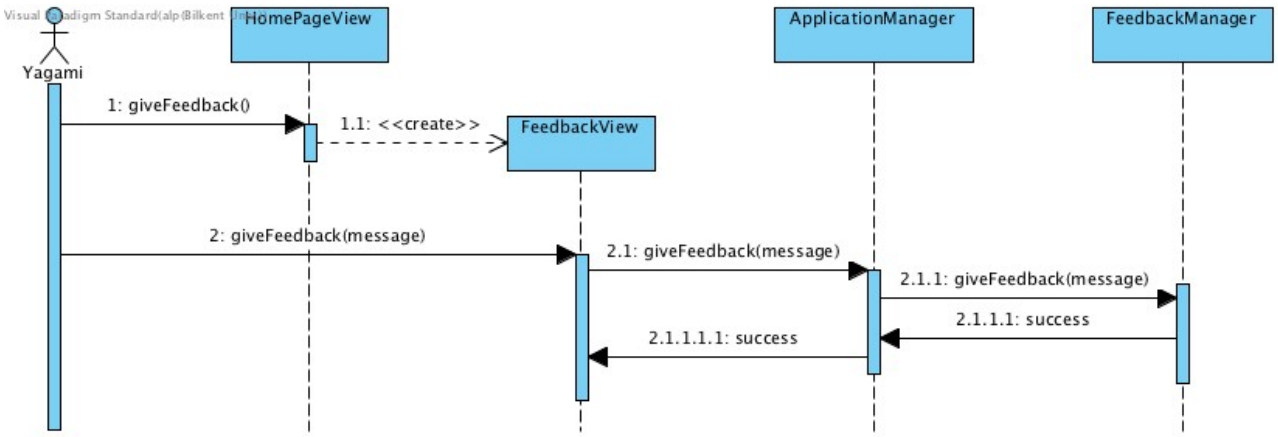

Figure 10 - Send Feedback Sequence Diagram

Yagami clicks on feedback link on the right-bottom of the page to request *FeedbackPage*. *ApplicationManager* shows the *FeedbackPage* to the user. Yagami fills the feedback message and submit it. *ApplicationManager* sends feedback submission to feedback manager. Feedback manager returns success result. *ApplicationManager* prompts a panel on the page to inform user for success of submission.

3.5.4.2. Activity Diagrams

- **Application Flow Activity Diagram**

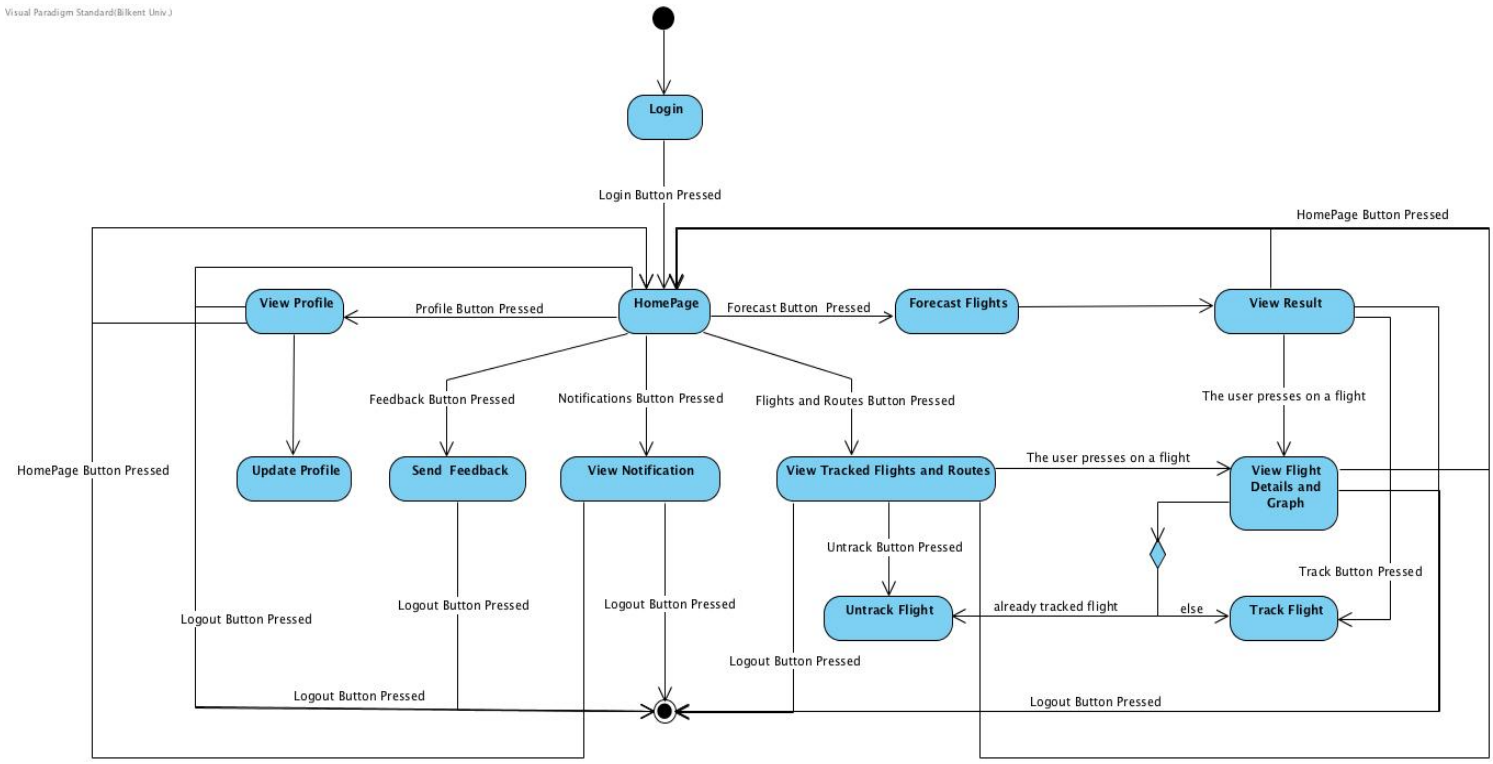

Figure 11 - Application Flow Activity Diagram

The flow of activities starts when the user logs into the system. The home page view is displayed after the log-in. The user can choose to get a forecast for a flight. He can also give feedback or view his profile. Besides, the user can choose to view his notifications or view his tracked routes and flights. After the user chooses to get a forecast, the user is directed to the results page. The user can click on a specific flight in the results page to view flight details and graph. The user can track a flight by clicking the track button in this page. After the user chooses to view tracked flights and routes, the user is directed to the tracked routes and flights page. The user can choose to untrack a flight in this page.

- **Log-in Flow Activity Diagram**

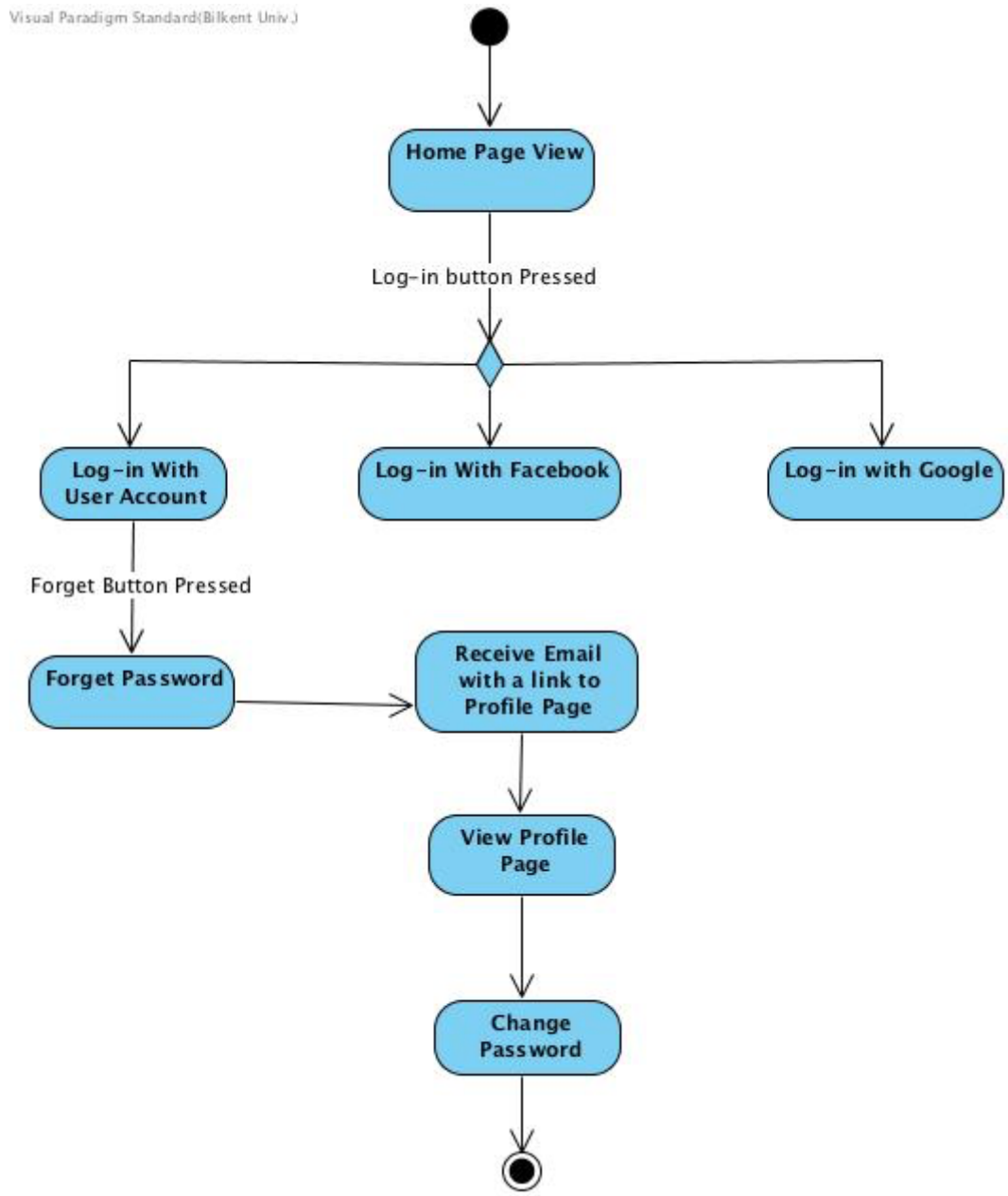

Figure 12 - Login Activity Diagram

The activity diagram represent the flow of event taken for the login operation. The user starts at the home page screen and clicks on the login button. After the button is pressed, the login screen is displayed. The user chooses to login either with a user account or facebook account or google account. When the user chooses to login with a user account, the user can click on the forgot password button to receive an email from

AeroCast that includes a link to the user's profile page. After the user clicks on the link, the profile page is displayed in the screen and the user changes his/her password in the profile page.

3.5.5. User Interface

3.5.5.1. Main Page Without Login

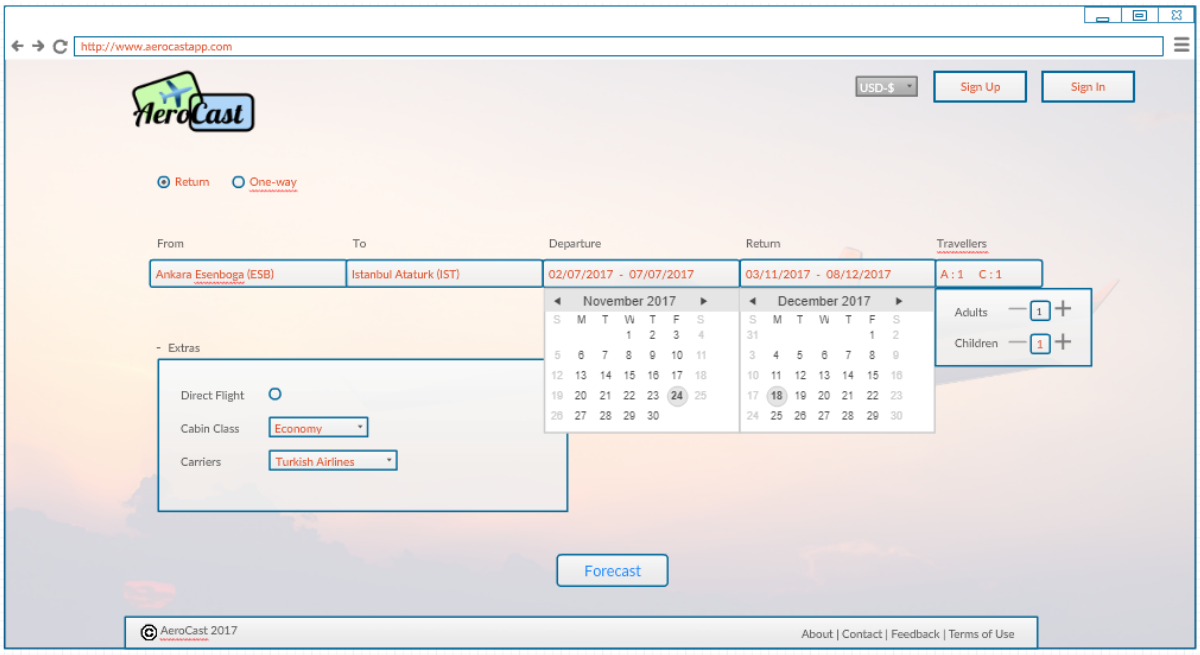

Figure 13 - Main Page Without Login

Main page is the first page user sees after enters the website. User can navigate all of the other pages from main page. Main page contains three main divisions.

At the top right of the screen there is user division. It has currency dropdown list which is chosen usd as default, "Sign Up" button, and "Sign In" button. User can change currency without login by choosing one of the currencies on the dropdown list. User can sign up by clicking on "Sign Up" button which is open sign up pop up panel. User can login by clicking on "Sign In" button which is open sign in pop up panel.

At the bottom of the page there is footer division. Left side of the footer contains the copyright of the website, and right side contains links which lead "About", "Contact", "Feedback", and "Term of use" pages.

At the center of the page there is a search division. Top of the search division user chooses one of the return or one-way routes. Under it, user sees options of the search. From left to right there are origin airport, destination airport, departure date, return date, and travelers respectively. The user clicks on the date panels and sees the calendars. Then he can choose date intervals on the calendars. Same as the dates, when user clicks on travelers panel a panel opens and closes. User enter traveler numbers after open the traveler box. Extras section is hidden by default. User clicks on "Extras" text to open the extras section. It contains extra options which are direct flight, cabin class, and carriers. User selects direct flight option by clicking on radio button. User chooses cabin class and carrier from the corresponding lists. User clicks on "Forecast Airfares" button at the bottom of the search division to submit the search with the selected options.

3.5.5.2. Sign Up

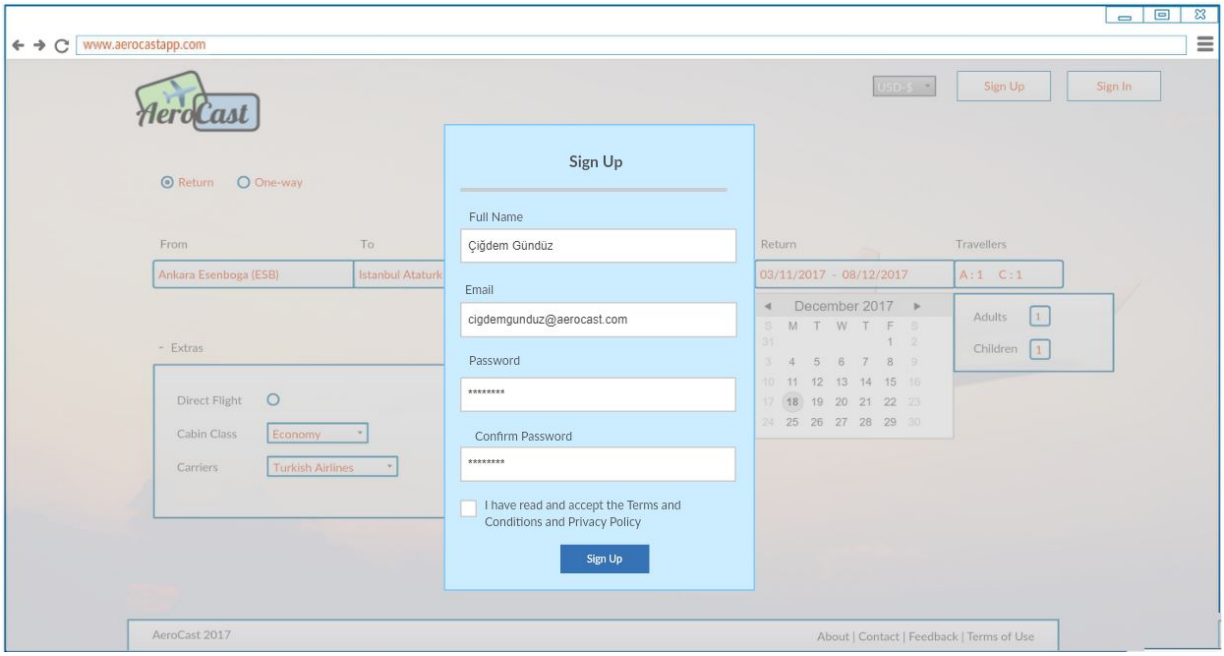

Figure 14 - Sign Up

"Sign Up Popup" appears when the "Sign Up" button on the Home Page or "Sign In Popup" is clicked. User enters Full Name, Email, Password and Confirmation Password

and click the checkbox for Terms and Agreements and clicks Sign Up button at the bottom.

3.5.5.3. Sign In

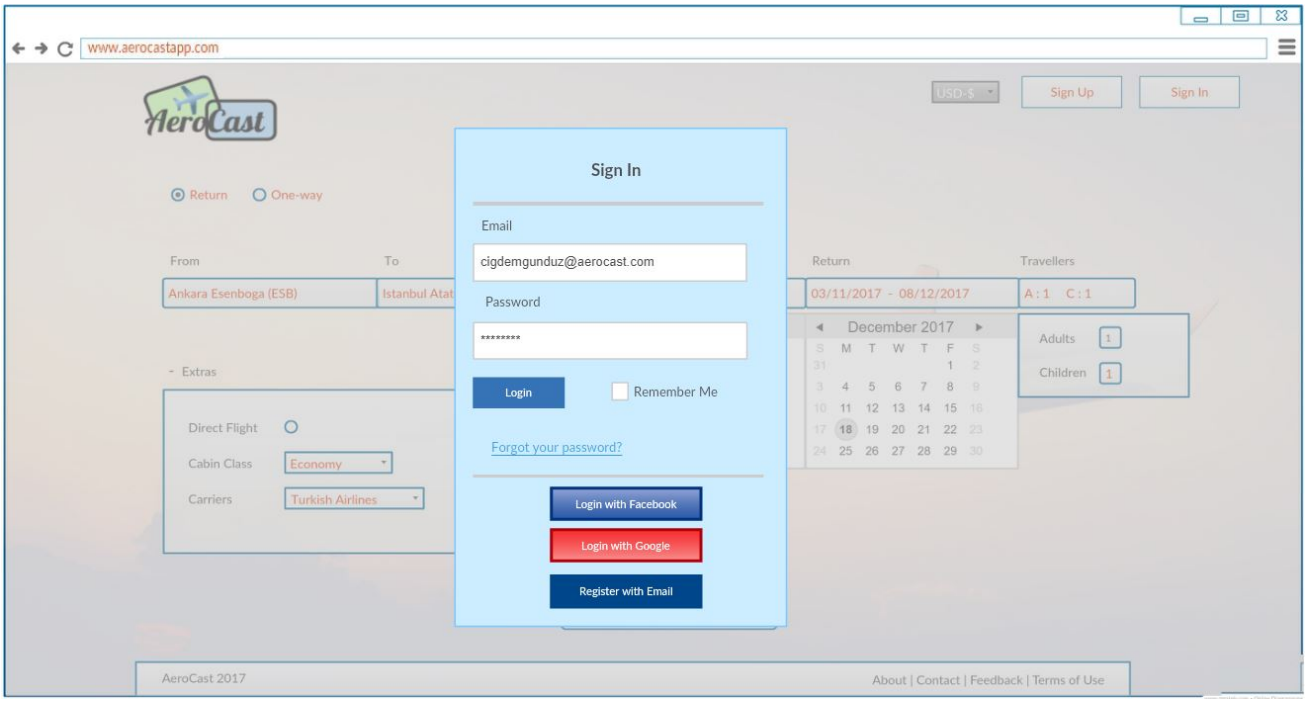

Figure 15 - Log In

"Sign In Popup"(Figure) appears when the sign in button on the Home Page is clicked or the website offers user to Sign In to have membership function. If user is Signed Up with his/her email, he/she can basically enters his/her credentials and click "Login" button. If the user does not want to enter the credentials every time, he/she can click "Remember Me" checkbox. Also there is an option for the users who forgot their passwords.

To offer users a quick shortcut, there are also options to sign in with Google or Facebook.

3.5.5.4. Main Page (Logged In)

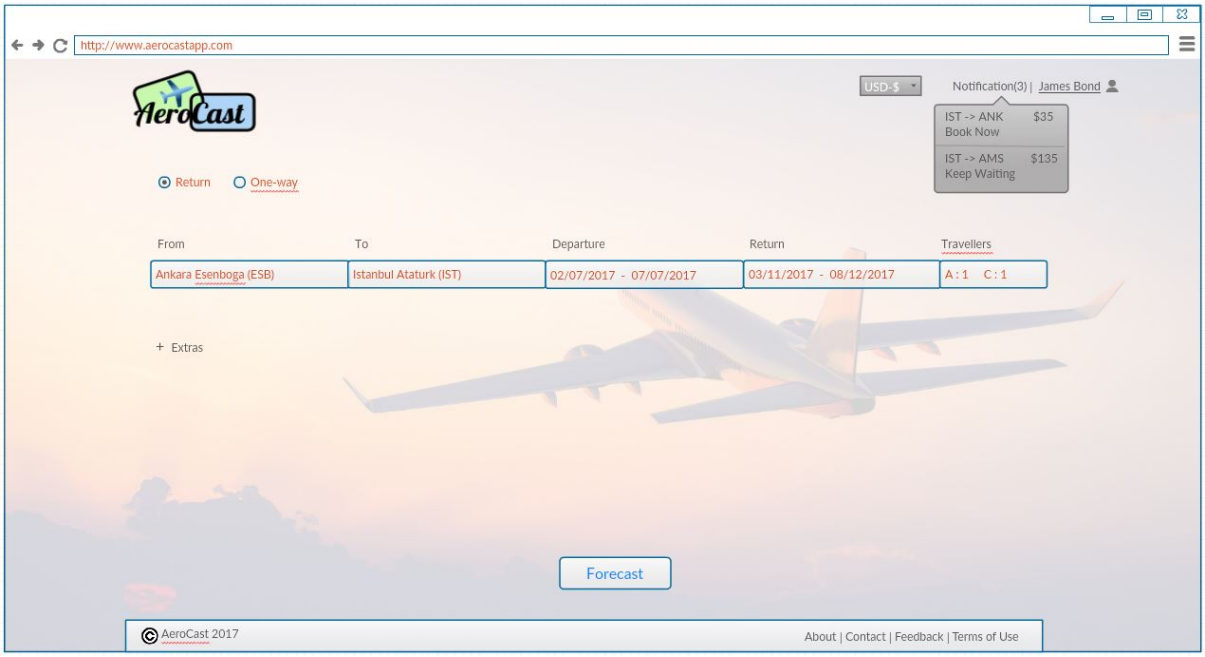

Figure 16 - Main Page After Log In

After logged in user sees the same page as the without logged in except the user division. In the user division sign up and sign in buttons replaced with notifications and profile. User can see notifications briefly by clicking on notifications text. And also user sees the number of the notifications next to it. User can go profile, tracked flights and routes pages and logout by clicking on username and select the choice from the opened panel.

3.5.5.5. Results of Two-Way Multiple Departure & Return Dates

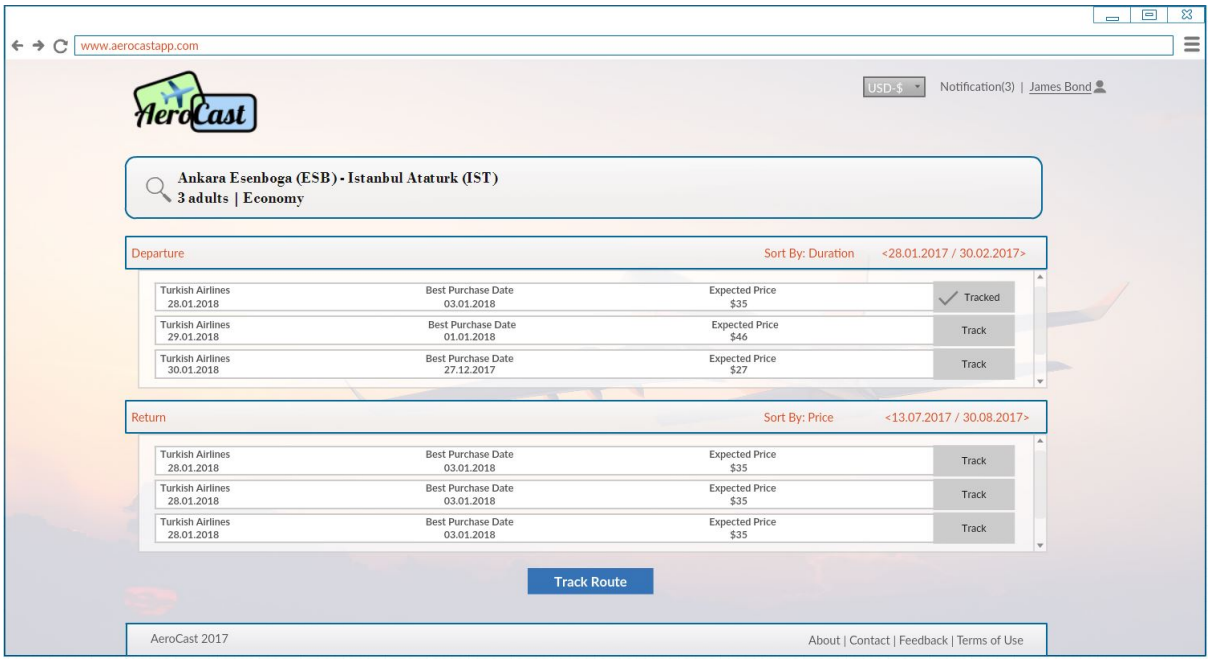

Figure 17 - Results Page With Multiple Departure Dates

This page is the sample result page which shows up when user search with the options “Two-Way, Multiple Departure & Return Dates”. At the top of the page, there will be information about the search such as Departure Airport, Destination Airport, number of passenger and extras options.

In the first scrollable division there will be prediction results of the first departure whereas in the second one there will be results for the return flight. In the both divisions, On their title section user will be able to sort the results according to options like duration and price. Also the range of the departure dates will be specified on the right of the title of the division. Inside of the divisions, There will be results which shows the carrier, departure date, best purchase date and expected price. Also user can track the results (each prediction) by clicking on “Track” button.

Additionally, user can track the whole search by clicking the “Track Route” button at the bottom of the page.

3.5.5.6. Results of Two-Way Single Departure & Return Date

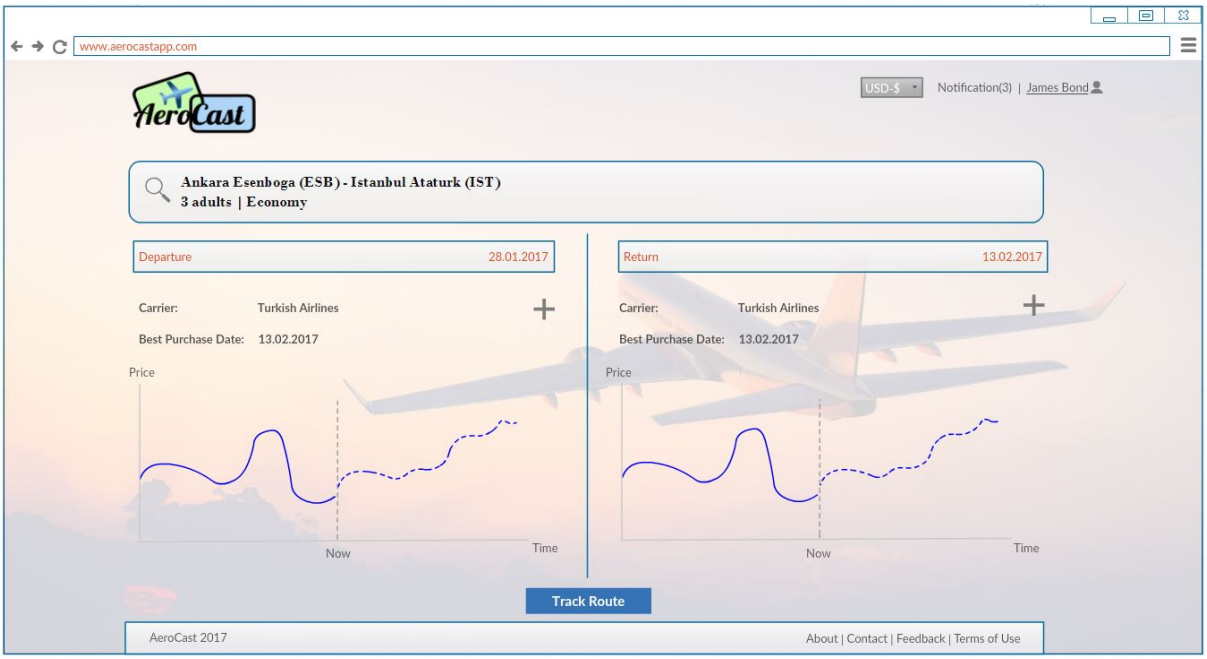

Figure 18 - Results Page With Round Trip Single Departure Date

This page is the sample result page which shows up when user search with the options "Two-Way, Single Departure & Return Date". At the top of the page, there will be information about the search such as Departure Airport, Destination Airport, number of passengers and extras options.

There will be 2 division side by side. The left one is for Departure and the right one is for Return flight. In the titles of the divisions there will be the date for flight. Inside of the division, carrier and best purchase date information will be shown. In addition to these default information there will be a graph which shows the flight price in time. The left of the graph shows the prices which are in the past and the right of the graph shows the predicted prices for future dates.

Also from the "+" sign at the top right of the division user can track each flight. If the user wants to track both of the flights he/she can click on "Track Route" button

3.5.5.7. Results of One-Way Single Departure Date

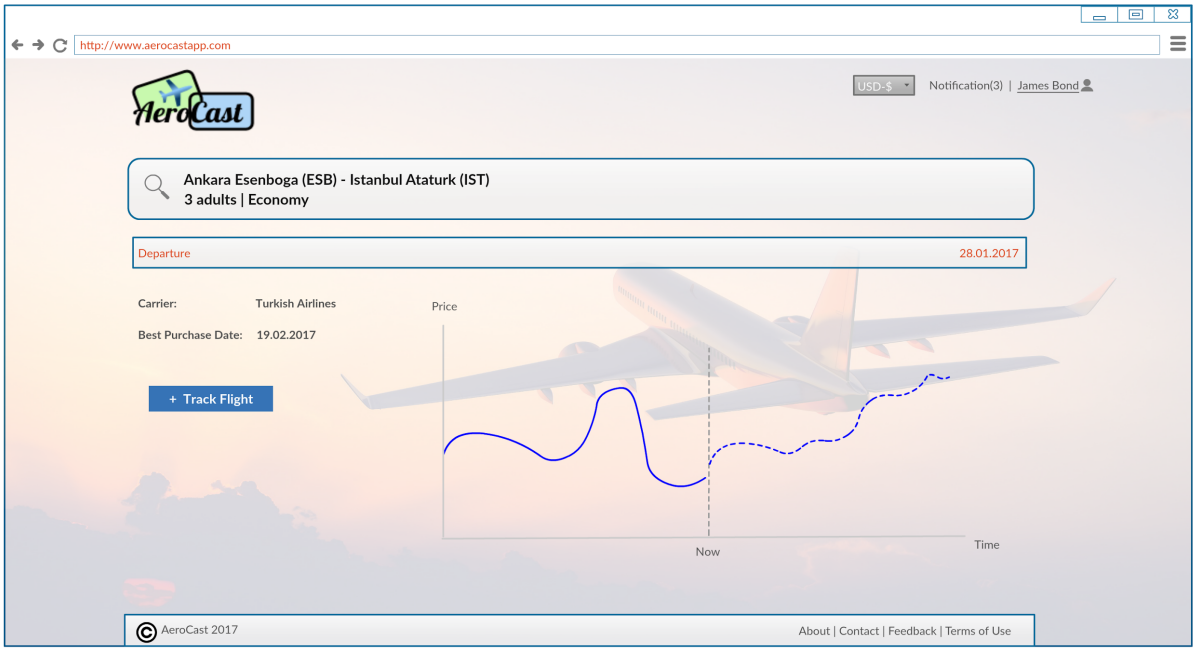

Figure 19 - Results Page With One Way Single Departure Date

This page is the sample result page which shows up when user search with the options "One-Way, Single Departure Date". At the top of the page, there will be information about the search such as Departure Airport, Destination Airport, number of passengers and extras options. At the center of the page the result division is shown. Left side of the result division contains carrier of the flight, estimated best purchase date, "Track Flight" button and feedback question. Right side of the result division contains the time-price graph of the flight. The graph of the flight shows past prices of the flight and after the "now" line it shows predicted prices. By hovering the mouse on the graph user sees the historical or predicted price of the flight on that day. User tracks the flight by clicking on the "Track Flight" button and sends feedback by choosing one of the "Yes" or "No" checkboxes.

3.5.5.8. Detailed Flight Graph

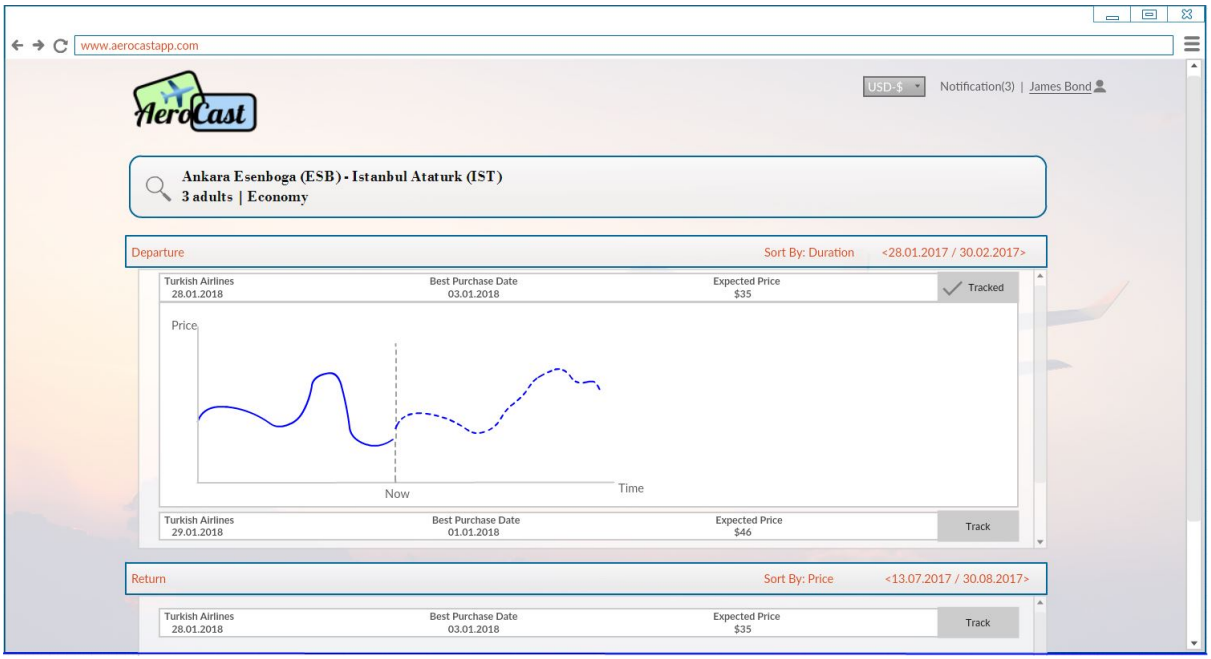

Figure 20 - Detailed Flight Graph

User clicks on a flight to show the time-price graph of the flight. The graph box open downward under the flight panel. The rest of the list continues after the graph box same as when graph box is hidden. The graph of the flight shows past prices of the flight and after the "now" line it shows predicted prices. By hovering the mouse on the graph user sees the historical or predicted price of the flight on that day. User hides the graph box by clicking on the flight panel once more.

3.5.5.9. Profile

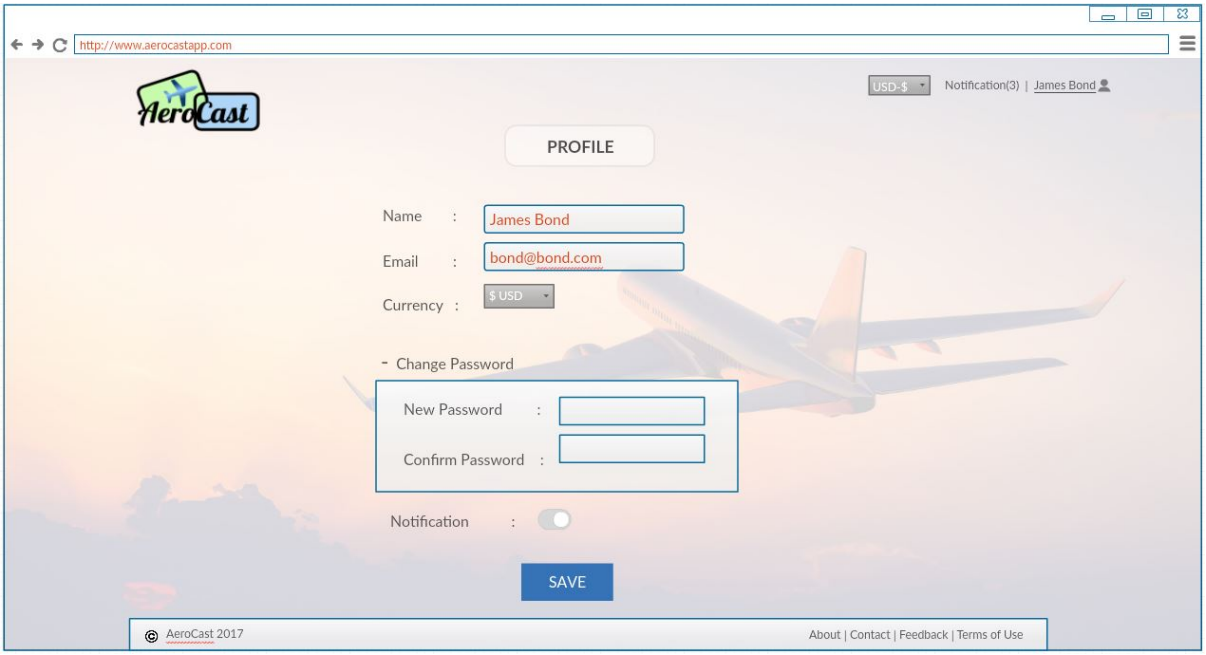

Figure 21 - Profile Page

Profile screen shows the current user informations. User changes the name, email by refill the corresponding areas, and currency by choosing one from the dropdown list. Change password panel is hidden when the page is rendered. By clicking on the "Change Password" text hidden panel is opened. User enters new password and confirm it by reentering to update the password. User can open and close the notifications by shifting the notification toggle button. At the bottom of the page there is a save button. By clicking on it, user update the current profile information with the entered informations.

3.5.5.10. Tracked Routes

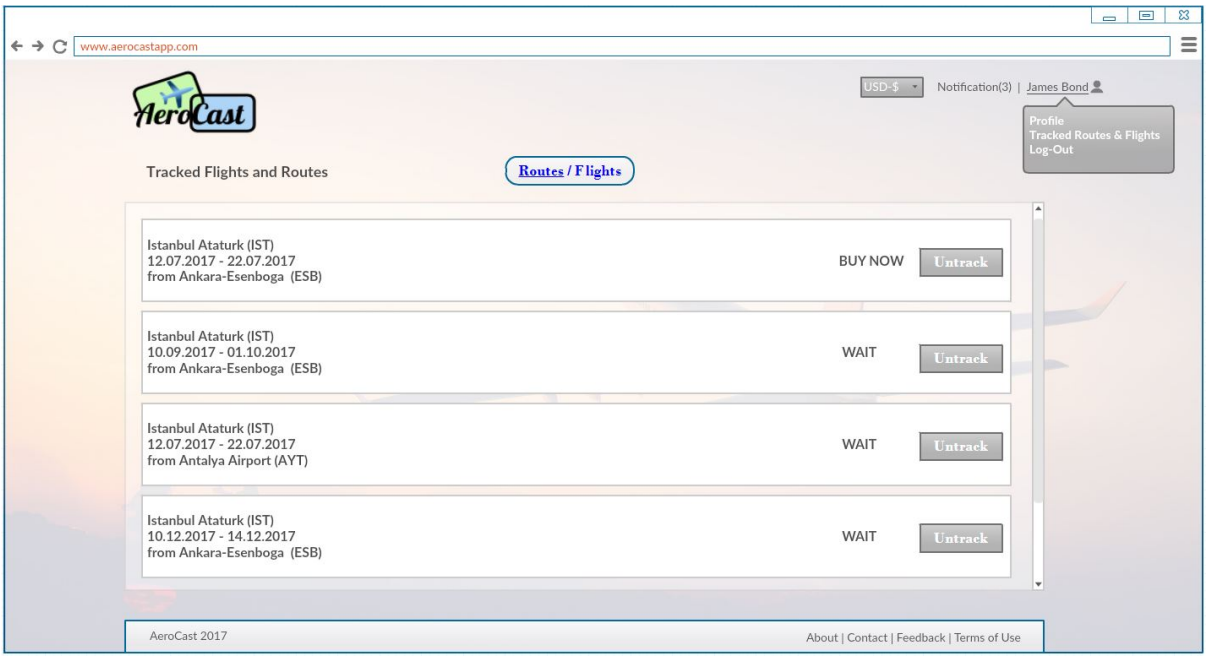

Figure 22 - Tracked Routes Page

Tracked routes screen lists the routes that user tracked beforehand. It lists flights by the order of track order. User sees the route information on the left side of the panel. On the right side of the panel user sees current recommendation of the app and "untrack" button. By clicking on the "untrack" button user can untrack the route and delete it from the list which is updated automatically.

3.5.5.11. Tracked Flights

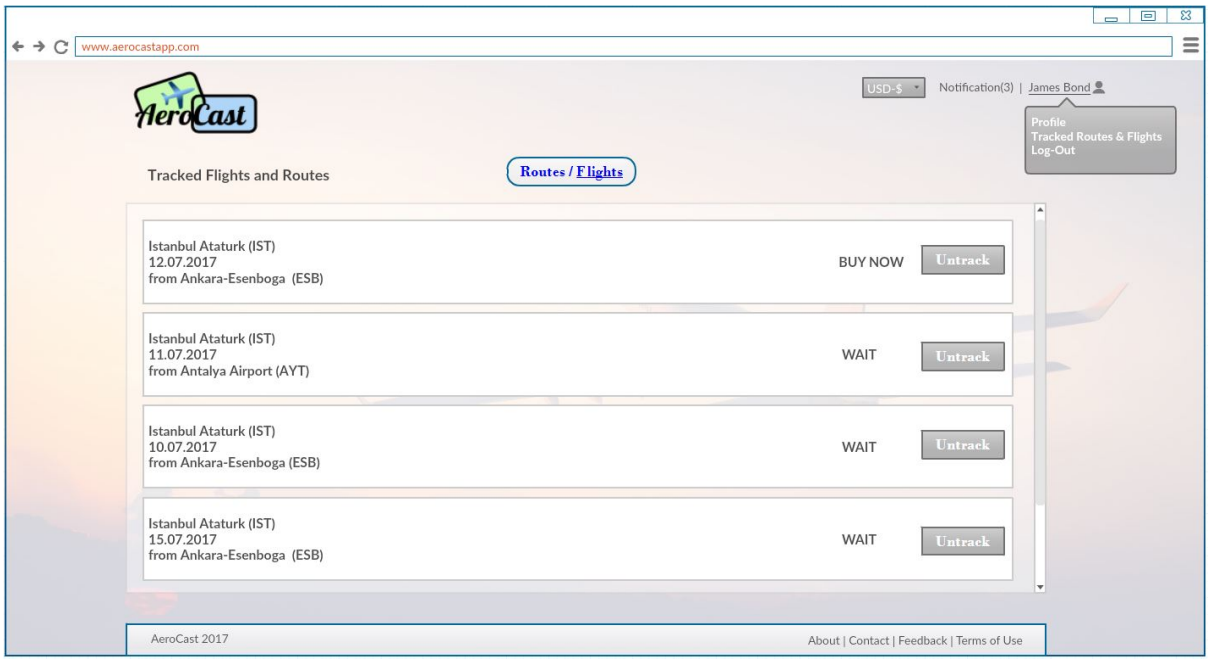

Figure 23 - Tracked Flights Page

Tracked flights screen lists the flights that user tracked beforehand. It lists flights by the order of track order. User sees the flight information on the left side of the panel. On the right side of the panel user sees current recommendation of the app and "untrack" button. By clicking on the "untrack" button user can untrack the flight and delete it from the list which is updated automatically.

3.5.5.12. Notifications

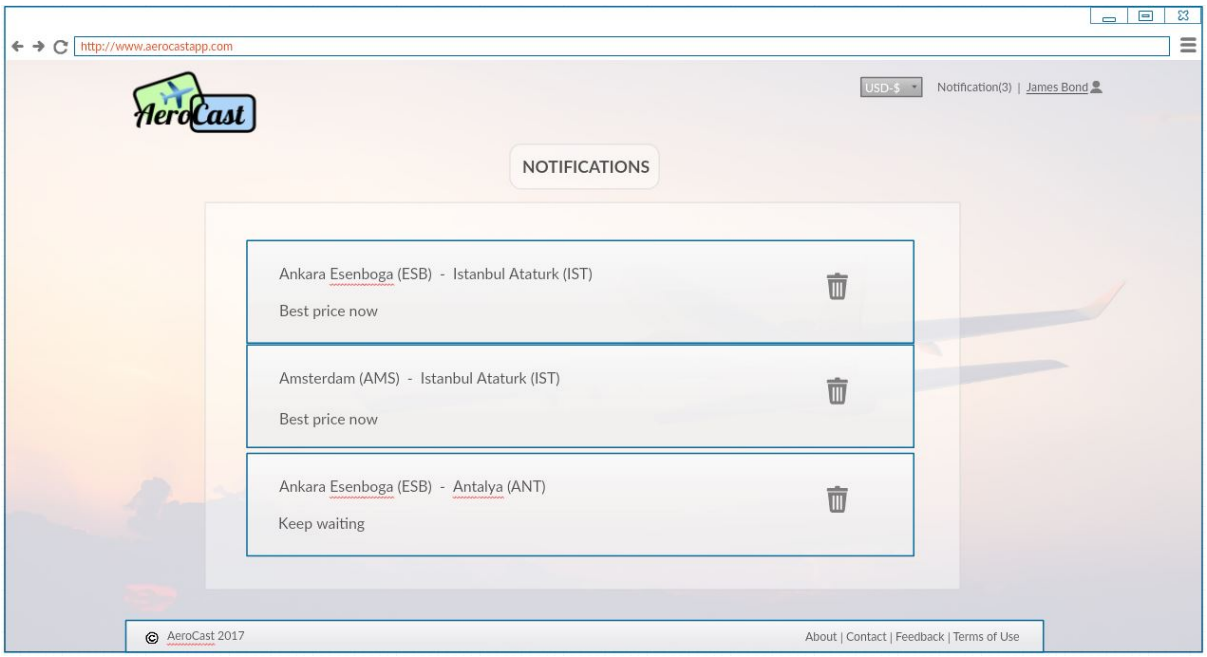

Figure 24 - Notifications Page

Notification screen list notifications of the user. It shows both the search and flight notifications. It orders notifications by their creation time. There is a trash icon right side of the notification panel. By clicking on trash icon user can delete the notification from the list which is updated automatically.

3.5.5.13. Feedback

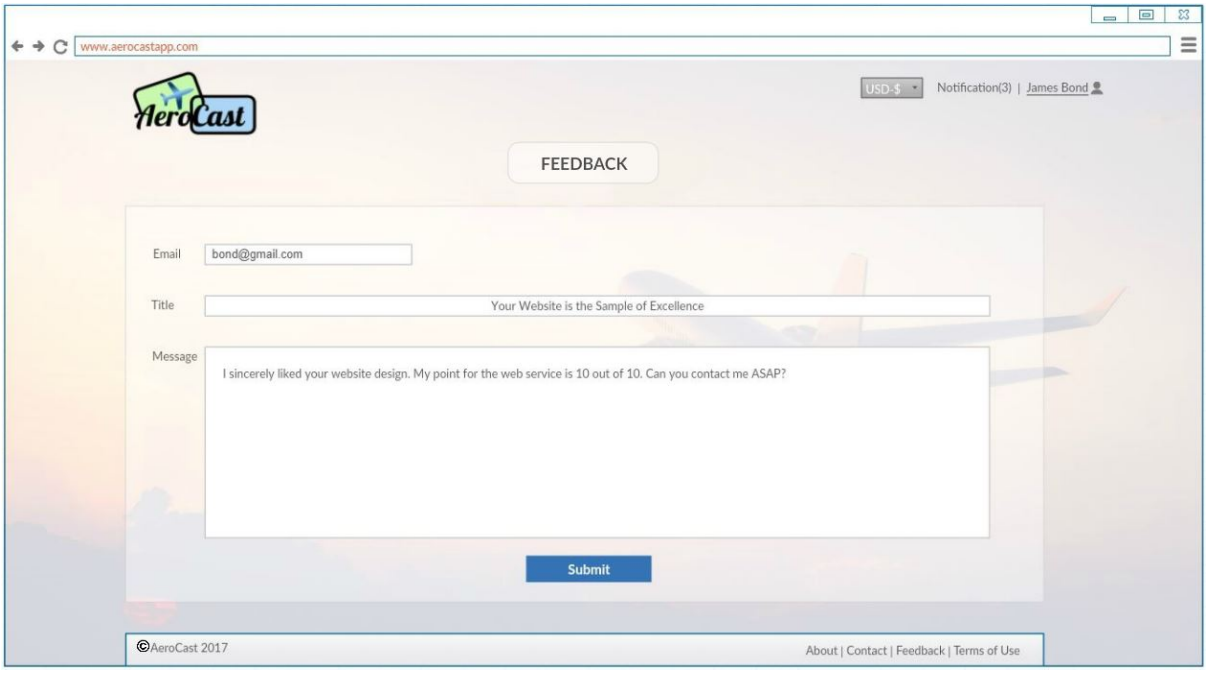

Figure 25 - Feedback Page

Feedback page is for the users who wants to contact with us or give their feedback. There are 3 sections such as Email, Title and Message in the middle of the page and one "Submit" button to send the feedback.

3.5.5.14. About

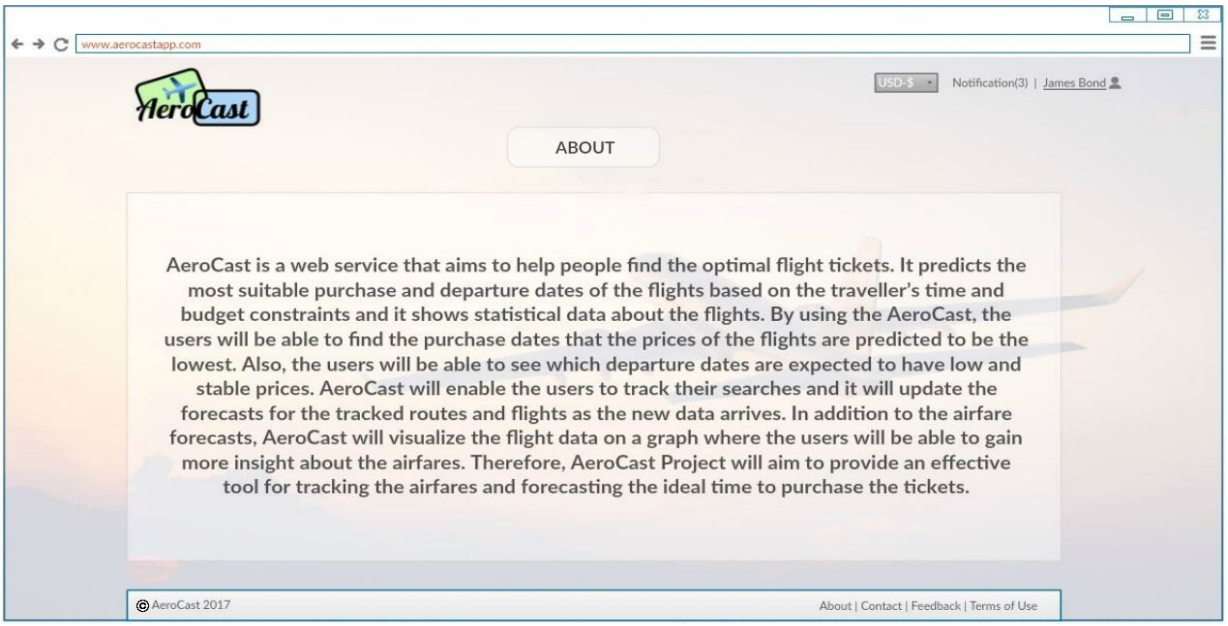

Figure 26 - About Page

This page is to give information about the project AeroCast.

3.5.5.15. Contact

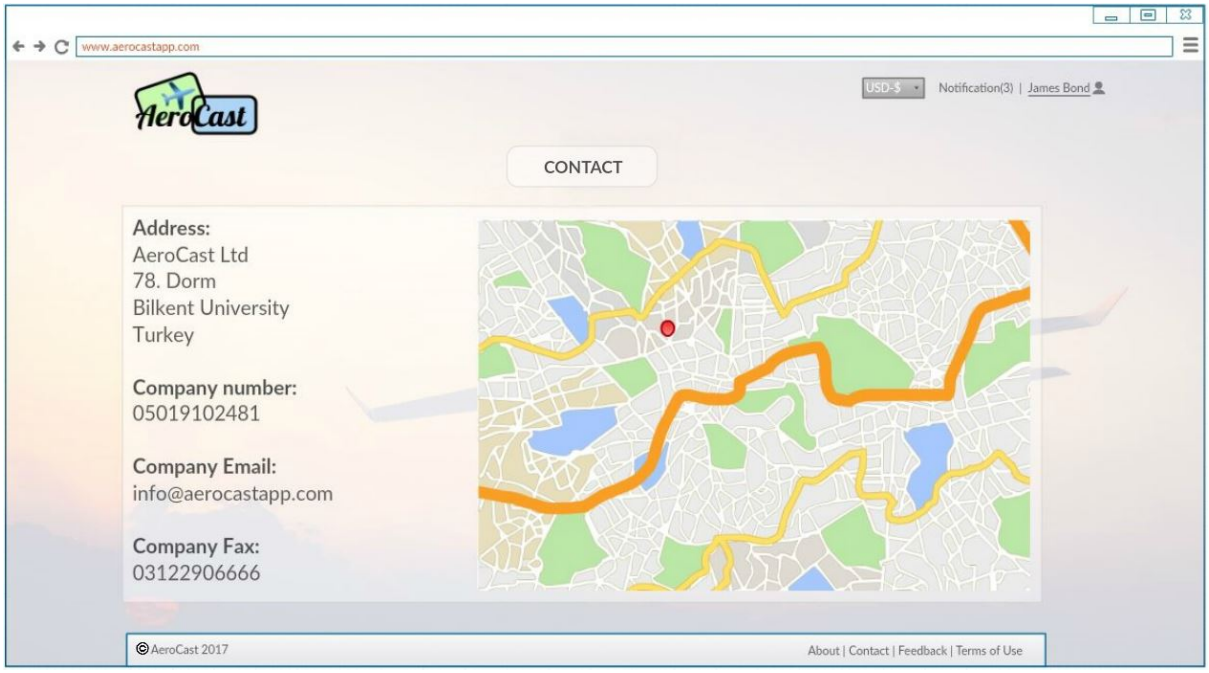

Figure 27 - Contact Page

This page is to give the contact information of AeroCast to the user.

4. Glossary

Flight : An entity, which is created by using the user's input and has an origin, destination, departure date, carrier code, purchase date forecast and expected price.

Route : Result of the user search that consists a list of flights. It created with the chosen constraints.

Track : Following a flight or route including getting updates and notifications for the tracked flight or route.

Expected Price : The expected minimum price, which prediction mechanism finds among the future predicted prices.

Best Purchase Date : The found date from the prediction mechanism that has the minimum price.

Panel : A part of the view that contains components in it.

Division : A combination of components that have same objective.

5. References

[1] "About Skyscanner". https://www.skyscanner.net/aboutskyscanner.aspx. [Accessed: Nov 05, 2017].

[2] "AirFareWatchDog". https://www.airfarewatchdog.com. [Accessed: Nov 01, 2017].

[3] "Expedia Customer Service". https://www.expedia.com/service. [Accessed: Nov 03, 2017].

[4] "TripAdvisor Help Center". https://www.tripadvisorsupport.com/hc/en-us. [Accessed: Oct 25, 2017].

[5] "Hopper About". http://www.hopper.com/corp/about.html. [Accessed: Oct 30, 2017].

[6] "FairFly Corporate Travel". http://www.fairfly.com/corporate-travel. [Accessed: Oct 14, 2017].

[7] "Kayak". https://www.kayak.com. [Accessed: Nov 02, 2017].

[8] "SOLID Object Oriented Design". https://en.wikipedia.org/wiki/SOLID_(object-oriented_design). [Accessed: Oct 29, 2017].

[9] "IEE Code of Ethics". http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Oct 10, 20177ae].