***Dreamline Project Report***



**Written by Cole Pearson, Nausherwan Tirmizi, Brian Kopec, and Harshal Patel**

**for use in CS 440**

**at the**

**University of Illinois Chicago**

**February 2022**

**Table of Contents**

[List of Figures 6](#_Toc102255377)

[List of Tables 7](#_Toc102255378)

[I Project Description 8](#_Toc102255379)

[1 Project Overview 8](#_Toc102255380)

[2 The Purpose of the Project 8](#_Toc102255381)

[2a The User Business or Background of the Project Effort 8](#_Toc102255382)

[2b Goals of the Project 8](#_Toc102255383)

[2c Measurement 8](#_Toc102255384)

[3 The Scope of the Work 9](#_Toc102255385)

[3a The Current Situation 9](#_Toc102255386)

[3b The Context of the Work 9](#_Toc102255387)

[3c Work Partitioning 10](#_Toc102255388)

[3d Competing Products 11](#_Toc102255389)

[4 The Scope of the Product 11](#_Toc102255390)

[4a Scenario Diagram(s) 11](#_Toc102255391)

[4b Product Scenario List 12](#_Toc102255392)

[4c Individual Product Scenarios 12](#_Toc102255393)

[5 Stakeholders 13](#_Toc102255394)

[5a The Client 13](#_Toc102255395)

[5b The Customer 13](#_Toc102255396)

[5c Hands-On Users of the Product 13](#_Toc102255397)

[5d Maintenance Users and Service Technicians 14](#_Toc102255398)

[5e Other Stakeholders 14](#_Toc102255399)

[5f User Participation 15](#_Toc102255400)

[5g Priorities Assigned to Users 15](#_Toc102255401)

[6 Mandated Constraints 16](#_Toc102255402)

[6a Solution Constraints 16](#_Toc102255403)

[6b Implementation Environment of the Current System 17](#_Toc102255404)

[6c Partner or Collaborative Applications 17](#_Toc102255405)

[6d Off-the-Shelf Software 17](#_Toc102255406)

[6e Anticipated Workplace Environment 17](#_Toc102255407)

[6f Schedule Constraints 18](#_Toc102255408)

[6g Budget Constraints 18](#_Toc102255409)

[7 Naming Conventions and Definitions 18](#_Toc102255410)

[7a Definitions of Key Terms 18](#_Toc102255411)

[7b UML and Other Notation Used in This Document 19](#_Toc102255412)

[7c Data Dictionary for Any Included Models 19](#_Toc102255413)

[8 Relevant Facts and Assumptions 19](#_Toc102255414)

[8a Facts 19](#_Toc102255415)

[8b Assumptions 19](#_Toc102255416)

[II Requirements 20](#_Toc102255417)

[9 Product Use Cases 20](#_Toc102255418)

[9a Use Case Diagrams 20](#_Toc102255419)

[9b Product Use Case List 20](#_Toc102255420)

[9c Individual Product Use Cases 22](#_Toc102255421)

[10 Functional Requirements 28](#_Toc102255422)

[11 Data Requirements 30](#_Toc102255423)

[12 Performance Requirements 31](#_Toc102255424)

[12a Speed and Latency Requirements 31](#_Toc102255425)

[12b Precision or Accuracy Requirements 32](#_Toc102255426)

[12c Capacity Requirements 32](#_Toc102255427)

[13 Dependability Requirements 33](#_Toc102255428)

[13a Reliability Requirements 33](#_Toc102255429)

[13b. Availability Requirements 34](#_Toc102255430)

[13c Robustness or Fault-Tolerance Requirements 35](#_Toc102255431)

[13d Safety-Critical Requirements 36](#_Toc102255432)

[14 Maintainability and Supportability Requirements 36](#_Toc102255433)

[14a Maintenance Requirements 36](#_Toc102255434)

[14b Supportability Requirements 36](#_Toc102255435)

[14c Adaptability Requirements 37](#_Toc102255436)

[14d Scalability or Extensibility Requirements 38](#_Toc102255437)

[14e Longevity Requirements 38](#_Toc102255438)

[15 Security Requirements 38](#_Toc102255439)

[15b Integrity Requirements 39](#_Toc102255440)

[15c Privacy Requirements 40](#_Toc102255441)

[15d Audit Requirements 41](#_Toc102255442)

[15e Immunity Requirements 41](#_Toc102255443)

[16 Usability and Humanity Requirements 42](#_Toc102255444)

[16b Personalization and Internationalization Requirements 42](#_Toc102255445)

[16c Learning Requirements 43](#_Toc102255446)

[16d Understandability and Politeness Requirements 43](#_Toc102255447)

[16e Accessibility Requirements 43](#_Toc102255448)

[16f User Documentation Requirements 44](#_Toc102255449)

[16g Training Requirements 45](#_Toc102255450)

[17 Look and Feel Requirements 45](#_Toc102255451)

[17a Appearance Requirements 45](#_Toc102255452)

[17b Style Requirements 46](#_Toc102255453)

[18 Operational and Environmental Requirements 46](#_Toc102255454)

[18a Expected Physical Environment 46](#_Toc102255455)

[18b Requirements for Interfacing with Adjacent Systems 47](#_Toc102255456)

[18c Productization Requirements 47](#_Toc102255457)

[18d Release Requirements 48](#_Toc102255458)

[19 Cultural and Political Requirements 49](#_Toc102255459)

[19a Cultural Requirements 49](#_Toc102255460)

[19b Political Requirements 49](#_Toc102255461)

[20 Legal Requirements 49](#_Toc102255462)

[20a Compliance Requirements 49](#_Toc102255463)

[20b Standards Requirements 49](#_Toc102255464)

[21 Requirements Acceptance Tests 50](#_Toc102255465)

[21a Requirements – Test Correspondence Summary 50](#_Toc102255466)

[21b Acceptance Test Descriptions 51](#_Toc102255467)

[III Design 55](#_Toc102255468)

[22 Design Goals 55](#_Toc102255469)

[23 Current System Design 55](#_Toc102255470)

[24 Proposed System Design 55](#_Toc102255471)

[24a Initial System Analysis and Class Identification 55](#_Toc102255472)

[24b Dynamic Modelling of Use-Cases 57](#_Toc102255473)

[24c Proposed System Architecture 59](#_Toc102255474)

[24d Initial Subsystem Decomposition 59](#_Toc102255475)

[25 Additional Design Considerations 59](#_Toc102255476)

[25a Hardware / Software Mapping 59](#_Toc102255477)

[25b Persistent Data Management 60](#_Toc102255478)

[25c Access Control and Security 60](#_Toc102255479)

[25d Global Software Control 60](#_Toc102255480)

[25e Boundary Conditions 60](#_Toc102255481)

[25f User Interface 60](#_Toc102255482)

[26 Final System Design 62](#_Toc102255483)

[27 Object Design 62](#_Toc102255484)

[27a Packages 62](#_Toc102255485)

[IV Project Issues 62](#_Toc102255486)

[28 Open Issues 62](#_Toc102255487)

[29 Off-the-Shelf Solutions 62](#_Toc102255488)

[29a Ready-Made Products 62](#_Toc102255489)

[29b Reusable Components 63](#_Toc102255490)

[29c Products That Can Be Copied 63](#_Toc102255491)

[30 New Problems 63](#_Toc102255492)

[30a Effects on the Current Environment 63](#_Toc102255493)

[30b Effects on the Installed Systems 63](#_Toc102255494)

[30c Potential User Problems 63](#_Toc102255495)

[30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 64](#_Toc102255496)

[30e Follow-Up Problems 64](#_Toc102255497)

[31 Migration to the New Product 64](#_Toc102255498)

[32 Risks 64](#_Toc102255499)

[33 Costs 65](#_Toc102255500)

[34 Waiting Room 65](#_Toc102255501)

[35 Ideas for Solutions 65](#_Toc102255502)

[36 Project Retrospective 66](#_Toc102255503)

[References / Bibliography 66](#_Toc102255504)

[Bibliography 66](#_Toc102255505)

[Index 67](#_Toc102255506)

### ****List of Figures****

[Figure 2 - Sample Use Case Diagram from Bruegge & DuToit ( modified ) 37](#_Toc525544068)

[Figure 3 - Sample Use Case Diagram from Robertson and Robertson 37](#_Toc525544069)

### ****List of Tables****

[*Table 2 - Requirements - Acceptance Tests Correspondence* 75](#_Toc525544085)

# I Project Description

## 1 Project Overview

Dreamline is about finding the cheapest and most comfortable seat for a customer. The cheapest seat is found by comparing flight prices for an itinerary. Alongside that, Dreamline will find the most comfortable seat through flight information as well as sensor data. Flight information will describe the aircraft being used and sensors commonly seen in smart watches and phones will help determine the comfort level of the flights. Those sensors will help with sleep tracking, repositioning due to discomfort, undue turbulence, ambient noise levels, and cabin pressure. With these two key pieces of information, Dreamline will return a flight with the highest Dreamline score. The Dreamline score will find the best ticket on a price/comfort basis and allow the customer to book that ticket. Then continue to improve the Dreamline score accuracy through further comfort tracking on the booked flight.

## 2 The Purpose of the Project

### 2a The User Business or Background of the Project Effort

The airline industry has a problem of not being able to properly advertise the amenities offered by a certain flight. A customer will simply purchase the cheapest ticket for their itinerary and be stuck with a deeply uncomfortable seat. When in a lot of cases, an extra $10 can massively improve the situation. Determining the marginal utility of those $10 is key to Dreamline. There is currently nothing considering comfort in the travel planning process which means there is a market for the client to capitalize on with the development of Dreamline. The client’s customer (people booking tickets) can receive a much better flying experience and help augment the Dreamline score making process through sensors on their smart devices.

### 2b Goals of the Project

The goal for this project is to create a system that constantly monitors the pricing and comfort of their customers. Ensuring that their customers receive the best deal for their dollar when purchasing ticket prices. The project will provide the customer with a comfortable flying experience

### 2c Measurement

The success of the business will be increasing the customer base and generating increasing revenue. Growing the customer base will also improve the Dreamline score making process with the increased sensor data. The sensor data with the improved Dreamline score will also indicate a trend of the overall comfort increase for the customers. Trends like the increasing of REM Sleep through the sleep tracker, less repositioning from the accelerometer, higher oxygen levels from the oximeter, lower heart rate from the Heart Rate monitor etc. These would be followed by positive customer reviews of the flights.

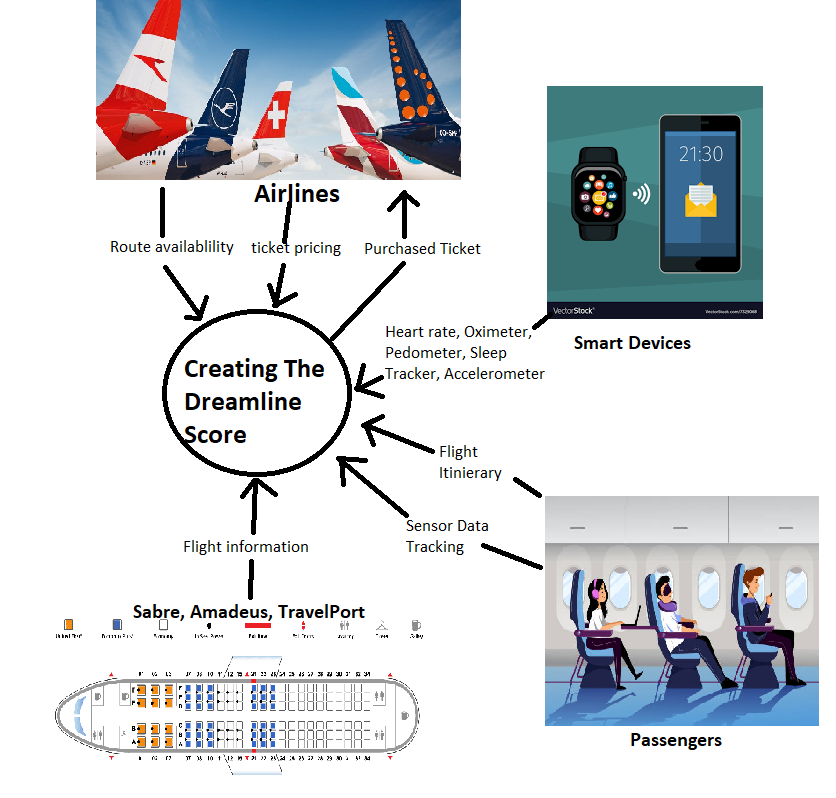
## 3 The Scope of the Work

The project aims to index ticket prices and monitor passenger comfort. Then balance those two data points to create a Dreamline score and provide the customer with the best flying experience

### 3a The Current Situation

The current ticketing process is completely reliant on price. The cheapest ticket is sold and once the purchase is complete, that is the end of the interaction with the customer. But this aims to continuously track the passengers' comfort level even after the ticket has been purchased. The customer’s smart devices have sensors that will help with understanding how comfortable a flight route was on that airline. This will serve to benefit both the service and the customer.

### 3b The Context of the Work



### 3c Work Partitioning

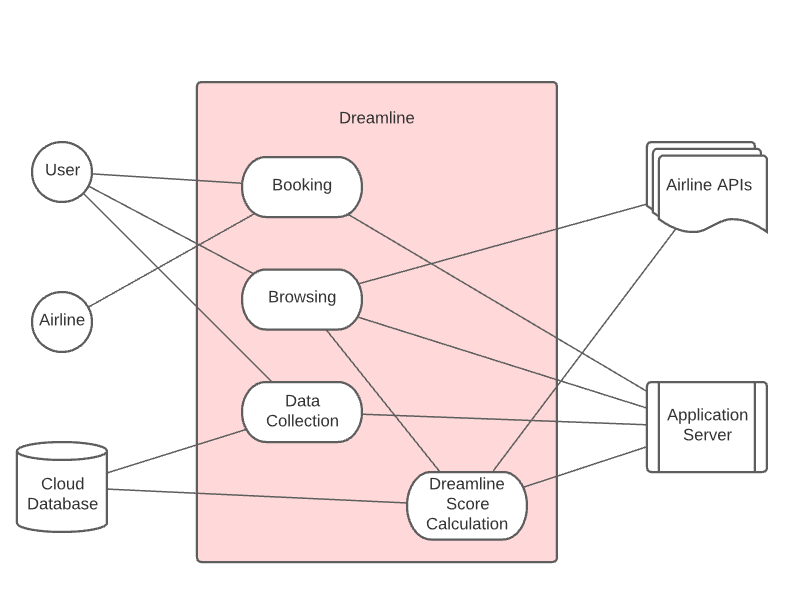
|  |  |  |
| --- | --- | --- |
| **Event Name** | **I/O** | **Summary** |
| The passenger describes the flight itinerary | Passengers (IN) | The passenger gives the origin and destination and the dates to build the Dreamline score on |
| Check Route Availability/ Get ticket pricing | Airlines (IN) | The itinerary will be checked for availability and the price on all airlines is revealed |
| Match the available routes with flight information | Sabre, Amadeus, Travelport (IN) | The available flights are checked for legroom and included amenities |
| The ticket is booked | Passengers (IN)  Airlines (OUT) | The passenger selects one of the tickets to purchase and the ticket is purchased from the airline |
| Passenger Agrees to provide sensor data from their smart devices | Passenger (IN)  Smart Devices (IN) | Oximeter, pedometer, sleep tracker, heart rate monitor, and GPS are tracked and stored to measure comfortability |
| Uncomfortable Flight | Passenger (IN)  Smart Devices (IN) | The passenger review and the sensor data indicate an uncomfortable price, so the Dreamline score is adjusted down |
| Comfortable Flight | Passenger (IN)  Smart Devices (IN | The passenger review and the sensor data indicate an uncomfortable price, so the Dreamline score is adjusted up |
|  |  |  |

### 3d Competing Products

There is a lot of competition in the flight ticket sale market. One Google search will reveal dozens of solutions to find the lowest price of a ticket. They are all useful tools for finding the cheapest price. Which one you choose is a matter of personal choice, there isn’t really anything that differentiates them. The project can incorporate one of those companies entirely for the price indexing part. Google Flights API () and SkyScanner API (Skyht) are all available for purchase to handle those tasks. None of the competitors factor comfort in any capacity. Dreamline has something unique to its ticketing solution, The Dreamline Score.

## 4 The Scope of the Product

### 4a Scenario Diagram(s)



### 4b Product Scenario List

|  |  |  |
| --- | --- | --- |
| **Name** | **Involved Parties** | **Information** |
| Finding a flight ticket. | User (OUT), Application Server (IN), Airline API’s (IN) | User browses through airline tickets by applying certain filters and makes decision based on suggestions by the Dreamline score provided by the application server. |
| Booking a flight | Users (IN), Airlines (OUT), Application Server (OUT)  Users (IN), Cloud Database (OUT), Application Server (OUT) | User would purchase a flight from a given airline through Dreamline. This will allow the app to start collecting data once that flight takes off |
| Collecting data. | Users (IN), Cloud Database (OUT), Application Server (OUT) | This will help us gather information about the flight, including smoothness, air quality, and comfortability. Using that information, Dreamline score will be updated and displayed |

### 4c Individual Product Scenarios

Fred would like to book a flight to Colorado for a week-long vacation with his family. He feels like he does a good job finding good prices, but his flight always ends up as a disaster. His kids get sick, his wife gets a headache, and he never feels like he can get any rest on the four-hour flight from New York. This time he uses Dreamline, so he can factor in his comfort based on real, user-collected data. Fred is willing to spend a few extra dollars on his ticket, as long as the flight goes smoothly, so he decides to use the Dreamline browsing tool to factor in better air quality/pressure so his family feels better, and more legroom so he can more comfortably rest. Even though Fred will have to pay $5.99 extra for these tickets compared to what he normally would, he decides he will buy these anyway for the peace of mind offered.

Vaiva used Dreamline last night to find a ticket to DC, but she didn’t get paid until this morning, so she wrote the ticket information down and brought it to work with her. Vaiva could go to the website of the airline that she wrote down and purchase the ticket directly off of their website, but instead goes back to Dreamline. This way, she can confirm that she is still getting the deal she wants, and that no better deal has popped up. She will also be able to consent to our data collection request this way, which would not have been possible had she chosen to go to the airline’s website.

Dianne has just booked a ticket through the Dreamline application and was prompted to consent to have her data collected from her smart watch. She always immediately pressed ‘No’ but this time she thought about it first. Dianne figured with all her recent excellent flight experiences; she had been reaping the benefits of data collection without contributing at all, so she decided to allow the app to track her for this flight. Once the plane takes off, the GPS, accelerometer, heart rate monitor and all the other sensors attached to her smart watch immediately start getting recorded and saved to our cloud database. Because of her choice to share data, our server now has more information to factor in when any future user wants to find a ticket.

## 5 Stakeholders

### 5a The Client

The client will be whoever is interested in investing in this product. The client will pay up front for the website and software to be developed. The client will work directly with the programmers so they can receive feedback on the website while it's in development.

### 5b The Customer

Customers that will be most likely to use this product are people who tend to fly more often. It will attract people who are interested in a comfortable and affordable flight. It will also attract people who are looking for certain accommodation, like increased leg room, reduced jetlag, and a smoother ride.

### 5c Hands-On Users of the Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Category** | **User Role** | **Subject Matter Experience** | **Technological Experience** | **Age Group** |
| Regular User | End user | Journeyman-Master | Journeyman- Master | Young Adults to Adults |
| Casual User | End user | Novice | Novice - Journeyman | Young Adults to Adults |
| Developer | Software/Website developer | Master | Master | Primarily Adults |
| Beta Tester | Development Testing | Novice - Master | Novice | Young Adults to Adults |
| Marketing | Advertiser/Critic/End user | Journeyman-Master | Novice - Master | Primarily Adults |
| Airlines | Information provider | Master | Journeyman-Master | Primarily Adults |

*Table 3 - Hands-On Users of the Product*

### 5d Maintenance Users and Service Technicians

The developers will setup the website and software. They will also provide any updates or bug fixes throughout the life of the product. There will also be technical service support available via email through the website.

### 5e Other Stakeholders

* + - Marketing experts – marketing experts will work with the client to help advertise this product. They will have a knowledge of the airline's business and advertising experience. These marketers will show this product to the public in many ways such as making advertisements for television and online content.
    - Business analysts – They will work with the developers to make smarter decisions that will benefit the client and the customer. They will express to the developers what a customer may want from the application and how to make it profitable for the client. They will have experience and knowledge of airlines.
    - Web developers – They will oversee making the website and application. They will work with the client and implement the type of website they want. They will also work with the other developers to implement the features into the software. They will have a good understanding of how to make an appealing user interface. They will have a large influence since this will be the first thing the customer sees when they use the software.
    - Legal experts – They will provide legal information during the development. They will work with the developers and client to make sure they don’t have to worry about getting in legal trouble. They will also help with privacy laws while developing the features. They will also help with the release of the software to make it smooth and efficient.
    - Translator – They will play a big role because we plan on having this software international. There will be many types of people from around the world flying so we can assume they will speak multiple languages. The translator will help make the software appeal to a broader group of people.
    - Investors – These will be the people who believe in our product and are willing to invest in it. They will help fund the advertising and development of the software.

### 5f User Participation

Users will play a big role in the development of this application. They will provide feedback of what types of features are important to them. Users will also be able to try beta/alpha versions of the application and give feedback of what they’d like to see. During the alpha, it will be a smaller select group of people, with lots of family and friends. The beta version will be open to a bigger part of the public. This will help us find bugs and add a few features before the full release. This will also help us find potential memory leaks while its being used.

### 5g Priorities Assigned to Users

**Key users:**

* + The development team will be in charge of creating the software user interface.
  + Marketing experts will help attract customers to use the application. They will advertise on many platforms and be able to attract a broad group of people.
  + Alpha/Beta testers will help spot bugs and potential improvements early in the development process. They will have a say in what type of features make it and what gets cut out.
  + The client will essentially have the final say to anything that goes on in the software.
  + Active players will be able to provide feedback and reviews on features they liked or would like to see implemented. They can help find bugs. They can be a good indicator to see how well the application is doing.
  + The legal team will help the process go smoothly and help avoid any legal trouble. This is important because lawsuits and legal issues can be a big drawback.

**Secondary Users:**

* + Users of the application that do not offer feedback about any features or bugs.

**Unimportant Users:**

* + These will be anyone who doesn’t use the application or doesn’t contribute. They're feedback will not be helpful towards the software.

## 6 Mandated Constraints

### 6a Solution Constraints

Description: The application should be a Mobile application that can connect to smartwatches. There must also be a Web application.

Rationale: The mobile application will collect sensor data from phones and smartwatches as well as give data about what flight tickets will give you the most comfortability. These results are also displayed in the web application for a better user interface and accessibility.

Fit Criterion: The web application must be accessible from multiple web browsers on any OS. The mobile application must be accessible from an android or apple device as well as a smartwatch that can collect necessary data.

Description: The Dreamline Score must be based on comfort

Rationale: The point of the application compared to other competitors is to not to provide the cheapest flight but to provide which flight would give you the most comfort for the price.

Fit Criterion: The Dreamline score would be greater for a flight that includes a higher overall comfort level at a slightly higher price than a ticket that is cheaper but has signs of discomfort such as higher movement reading from an accelerometer, higher heart rate from the Heart Rate monitor etc. It would not provide a higher score to the cheapest ticket every time.

Description: The application must display relevant comfort information

Rationale: The customers should know what data and information is leading to the Dreamline score, so they can determine if the comfort level is right for them

Fit Criterion: The Dreamline score would be accompanied by a series of data points calculated by the software using the multitude of sensors that help determine the comfort level and would be displayed on the mobile and web applications. It would provide heart rates, resting heart rates, amount of repositioning collected by accelerometers, and disturbances in REM sleep to name a few. This would also be accompanied by a short explanation that helps the customer understand what the data points represent and what it means for the comfort level of that flight.

### 6b Implementation Environment of the Current System

The mobile application must be able to run on all android and apple devices. The device must have at minimum an accelerometer, GPS, and pedometer. The application will also be used in tandem with smartwatches to collect additional data from sensors such as, oximeter, heart rate sensor, and sleep tracker.

The web application must be able to run on any OS that is capable of running a web browser. The application must be compatible with Mac OS, Chrome OS, Windows 7 and up. The application will require an internet connection to be able to run.

The mobile application will require an internet connection to send the data back to the cloud before and after flights, or during the flight if possible.

### 6c Partner or Collaborative Applications

We will be collaborating with Google AdSense to run ads on our page to generate additional revenue. The actual service using the application will be free, however, we will run ads on empty space on the website to generate revenue.

We will collaborate with cloud services to store sensor data from particular flights and organize data to run data processing, as well as run web services. These cloud services will be the backbone of the software and will allow us to scale the application especially when the application will generate more users and in turn generate more data to provide accurate calculations.

We will also collaborate with airlines to include sponsored listings to provide advertisements from competing airlines.

### 6d Off-the-Shelf Software

Dreamline intends to use API calls from Amadeus (Amaht), Sabre (Sabht), and TravelPort (Traht), to collect aircraft information and well as real time ticket price information. There is no other OTS software use in our application.

### 6e Anticipated Workplace Environment

The application will only collect data when on a flight and will not collect data when customer is simply browsing through recommended flights. This suggests that if a user books a flight through our application, the app will start collecting data at the time of the flight.

The user may move around the airplane that may not particularly represent discomfort. This suggests that the application must ignore outlier data collected from sensors, typically when there is a large amount of movement detected by the accelerometer and pedometer, such as walking to the bathroom.

### 6f Schedule Constraints

The API calls must provide up to date price and aircraft information. If the application fails to meet this expectation, it will skew the Dreamline score and provide false information

Dreamline score must be updated with changing prices as well as additions in comfort data. As more data is collected, the accuracy of the scores will increase and therefore must be constantly updated.

The application must be fully tested before it is released, since false information or bugs would cost the customers a lot of money and likely result in negative reviews.

The application must include data from all domestic airlines before it is released, which means there will be a beta testing phase that may last 6 months.

Once the application is stable and running smoothly, updates to include international flight should be started, which will also require a longer period of beta testing.

### 6g Budget Constraints

The majority of the will be allocated to using cloud services, API usage, and the initial design of the software, since most of the maintenance and testing for bugs will be done before releasing the final product to the public. A good portion of the budget will also go into the research and beta testers who will go on flights to collect initial data on comfort levels.

## 7 Naming Conventions and Definitions

### 7a Definitions of Key Terms

Smart device: Refers to an electronic device with a base set of capabilities (cellular and/or wireless connection, various sensors, etc.) that, with the consent of the customer, we can connect to and receive data from.

Dreamline score: This is the score that we show the user for a given airline ticket based on all the factors we use, such as price, comfort (calculated based on data gathered from multiple users), and user preference.

Sensor: A device that gathers and records information (heart rate, temperature, etc.) for the purpose of providing us with valuable date we can further use to improve our algorithm. This device’s existence be exclusive to gathering information, or it may be secondary to a different device, such as a smart device.

### 7b UML and Other Notation Used in This Document

This document follows standard UML format as described by Fowler in *UML Distilled*, 3rd edition. Exceptions noted when made.

### 7c Data Dictionary for Any Included Models

Dreamline score = (Above average seat qualities + above average flight/cabin conditions) / ((Below average seat qualities + above average seat qualities) x ticket price)

## 8 Relevant Facts and Assumptions

### 8a Facts

We know that at least 85% of Americans own a smartphone (Pew Research Center). We also know about 1/5th of Americans owns a smart watch or other device capable of using sensors to gather data from the user (Pew Research Center).

### 8b Assumptions

We are assuming that the customer has a smart watch/wearable smart device, or at the very least a smartphone, as the user needs to have some way to use the sensors, we require to record data.

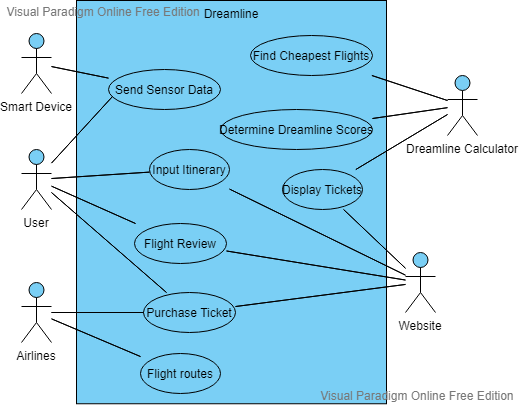
From a business perspective, we are assuming that people are willing and able to seek out a method to weigh their options when buying plane tickets. Furthermore, a reasonable portion of these people (15%) must also be willing to allow us to track them over the course of their flight.

Finally, we are assuming all flights to be domestic, as we will only be dealing with domestic airline ticket sales.

# II Requirements

## 9 Product Use Cases

### 9a Use Case Diagrams



### 9b Product Use Case List

|  |  |  |
| --- | --- | --- |
| **Use Case** | **Users(s)** | **Description** |
| Find Cheapest Flight | Dreamline Calculator | Indexes the cheapest flight for a certain itinerary |
| Determine Dreamline Score | Dreamline Calculator | With the existing review score, sensor data, and flight info determine Dreamline score |
| Send Sensor Data | User  Smart Device | The sensor data of the user’s smart device is sent |
| Input Itinerary | User  Website | The website allows the user to enter the itinerary they want the Dreamline score for |
| Flight Review | User  Website | The website asks for the user’s review score for a certain to flight to further augment the Dreamline score accuracy |
| Display Ticket | Websites | The website displays the tickets alongside the calculated Dreamline scores from the calculator |
| Flight Routes | Airlines | Airlines create their flight routes and that determines availability for a certain itinerary |
| Purchase Tickets | User  Airlines  Website | The user confirms a ticket to purchase. The website then purchases that ticket from the airline. |

### 9c Individual Product Use Cases

|  |
| --- |
| Use case ID: 1 Name: Find Cheapest Flight  pre-conditions: User has entered an itinerary  post-conditions: More than 0 flights were returned  Initiated by: Dreamline calculator  Triggering Event: All flight details are completed  Additional Actors: |
| Sequence of Events:  The user has entered all their flight details   1. The system finds flights related to that itinerary   The system indexes them all and orders them on price   1. The system returns the list of tickets to the calculator for Dreamline score calculation |
| Alternatives: If no flights are found for the date, then an empty list is returned.  Exceptions: Itinerary has logical errors (e.g.: return date before depart date) |

|  |
| --- |
| Use case ID: 2 Name: Determine Dreamline Score  pre-conditions: A non-empty list of flights  post-conditions: Every flight has a Dreamline (DL) score  Initiated by: Dreamline calculator  Triggering Event: A flight list where one or more flights do not have a DL score  Additional Actors: |
| Sequence of Events:   1. The system receives a list of flights 2. The system retrieves flight comfort information 3. The system calculates a DL score based on price and comfort 4. The system assigns the DL score to that ticket |
| Alternatives: Price ticket, or comfort data changes for a ticket  Exceptions: An uncomfortable or unreasonably expensive ticket is given a high DL score |

|  |
| --- |
| Use case ID: 3 Name: Send Sensor Data  pre-conditions: The user has a smart device  post-conditions:  Initiated by: User  Triggering Event: Flight lands and user is reconnected to data  Additional Actors: Smart Device |
| Sequence of Events:   1. The user boards a plane with a smart device that has one or more of the following sensors: sleep tracker, accelerometer, oximeter, heart rate monitor 2. The smart device tracks the sensor data on the flight 3. The system’s local instance, the app, is linked to the smart device through Bluetooth 4. The user’s flight lands and connects to an internet connection 5. The system uploads the data for future DL calculation |
| Alternatives: Price ticket, or comfort data changes for a ticket  Exceptions: An uncomfortable or unreasonably expensive ticket is given a high DL score |

|  |
| --- |
| Use case ID: 4 Name: Input Itinerary  pre-conditions:  post-conditions: The itinerary logic is valid  Initiated by: User  Triggering Event: User opens website for itinerary entry  Additional Actors: |
| Sequence of Events:   1. The system asks the user to input their itinerary details 2. The user enters their origin, destination, departure date, return date 3. The system checks to see if the return date is not before the departure date and that the origin and destination are not the same 4. The system stores this information for use by other functionalities |
| Alternatives: Itinerary logic does not work, no return date, flexible dates  Exceptions: Invalid dates, origin and destination are the same |

|  |
| --- |
| Use case ID: 5 Name: Flight Review  pre-conditions: The user has a taken a flight  post-conditions: The review is received by the system  Initiated by: User  Triggering Event: User selects “rate your flight” button  Additional Actors: Website |
| Sequence of Events:   1. The user rates different aspects of their flight 2. The system receives the flight review 3. The system uses the flight review to increase the accuracy of the DL score    1. The reviews and the dl score are contradictory, so the DL score is lowered for future flights    2. If the reviews exceed DL score expectations, increase DL score    3. If the reviews meet DL expectation, change nothing |
| Alternatives: User does not want to review  Exceptions: |
|  |

|  |
| --- |
| Use case ID: 6 Name Display Flights  pre-conditions: DL calculations have been run on user entered flight itinerary  post-conditions: all flights routes displayed are purchasable  Initiated by: System  Triggering Event: DL calculations complete |
| Sequence of Events:   1. The system displays a list of purchasable flights to the user 2. The system allows the user to select any of the flights for further details 3. The system takes the user to the purchasing portal once they are ready to purchase their ticket |
| Alternatives: User does not want to purchase  Exceptions: Flights are not purchasable; DL scores are incorrect |
|  |

|  |
| --- |
| Use case ID: 7 Name: Purchase Tickets  pre-conditions: The ticket is valid  post-conditions: Ticket is purchased  Initiated by: User  Triggering Event: User selects a ticket to purchase |
| Sequence of Events:   1. The user has selected a ticket and is in the purchase portal 2. The systems allow the user to enter a payment method to buy their ticket 3. The system purchases the ticket from the airline and gives confirmation to the user |
| Alternatives: User cancels purchase  Exceptions: Communication with airline fails payment authentication fails |
|  |

## 10 Functional Requirements

**ID#** FUNC-1 **Name** Receive Itinerary

**Description:** The system must have a way for the user to input their flight itinerary details

**Rationale:** The itinerary is crucial in understanding which tickets to get the price and DL score of

**Fit Criterion:** The system receives the user’s input and the final itinerary received matches what the user inputted

**Acceptance Tests:** T8

**ID#** FUNC-2 **Name** Ticket Lookup

**Description:** The system must be able to make API calls for ticket availability from different airlines. Travelport (Traht) and DirectConnect (Alvht) are API sources for ticket availability

**Rationale:** The ticket must be available for the user to book, and the system must be able to look up that information

**Fit Criterion:** The system can communicate with different airlines to receive ticket information

**Acceptance Tests:** T9

**ID#** FUNC-3 **Name** Price Indexing

**Description:** The system can compile all the ticket prices and order them in ascending order

**Rationale:** The system will use the price information to understand if the ticket is a good deal relative to other prices. This plays a role in calculating the DL score

**Fit Criterion:** The prices received are matched with the actual ticket price. The list is correctly sorted

**Acceptance Tests:** T10

**ID#** FUNC-4 **Name** Tracker Data

**Description:** The system tracks the sensor data during the flight and uploads it once it reconnects

**Rationale:** The sensor data is essential for the system to calculate the Dreamline score

**Fit Criterion:** There is a local instance of the system through the app. The user takes the smart device on the phone and the app tracks the sensor data from the various sensors. The data is uploaded once the user has an internet connection

**Acceptance Tests:** T11

**ID#** FUNC-5 **Name** Reviews

**Description:** The system has a way for users to enter reviews for their flight

**Rationale:** The reviews are used in the Dreamline score

**Fit Criterion:** The review can be entered and then it will be stored for that specific flight route so it can be used by

**Acceptance Tests:** T12

**ID#** FUNC-6 **Name** Dreamline Score

**Description:** The Dreamline calculator can intake the price, review, and sensor data to create the Dreamline score

**Rationale:** Core of the project

**Fit Criterion:** A list of tickets have their Dreamline score calculated and assigned

**Acceptance Tests:** T13

## 11 Data Requirements

**ID#** DATA-1 **Name** Sensor Data

**Description:** The system must be able to store the sensor data from the smart devices

**Rationale:** This information is used for the Dreamline score

**Fit Criterion:** The system is storing data from the Sleep tracker, oximeter, heart rate sensor, and accelerometer

**Acceptance Tests:** T11

**ID#** DATA-2 **Name** Review Data

**Description:** The system must be able to store the review data from the user reviews

**Rationale:** This information is used for the Dreamline score

**Fit Criterion:** The system is storing the following review categories: seat comfort, legroom, service, and sleep

**Acceptance Tests:** T12

**ID#** DATA-3 **Name** Ticket Data

**Description:** The system must be able to store ticket data from the airlines

**Rationale:** This information is used for the Dreamline score and purchasing of the ticket

**Fit Criterion:** The system can store a ticket’s price, destination, and origin

**Acceptance Tests:** T9

## 12 Performance Requirements

### 12a Speed and Latency Requirements

**ID#** SLR-1 **Name** Real time Dreamline Score

**Description:** Updating the Dreamline score from the moment more comfort data for a specific flight is added to updating the app and web application to display it must be done in a matter of no more than 10 seconds.

**Rationale:** Users must have live updates to the Dreamline score to ensure they don’t receive old data that may have changed due to sudden changes in aircraft quality.

**Fit Criterion:** any additional comfort level data that is updated must result in an updated Dreamline score in no more than 10 seconds.

**Acceptance Tests:** T13**,** T14

**ID#** SLR-2 **Name** Server to Application speed

**Description:** When a user uses the web or mobile application do filter and search for flight, retrieving the results from the cloud should take no more than 500ms.

**Rationale:** Users should not have to wait for results to load each time a filter is applied and users with slow internet speeds can benefit from faster retrieval speeds.

**Fit Criterion:** Any action that results in data being accessed from the cloud and analyzed to provide some result to the web and mobile application must produce and display such results within 500ms.

**Acceptance Tests:** T1, T2, T3, T9, T10

### 12b Precision or Accuracy Requirements

**ID#** PAR-1 **Name** Numerical Sensor Data

**Description:** All numerical sensor data must be precise to 5 decimal space when collecting and must be rounded to the nearest whole number when displaying.

**Rationale:** The system must collect precise data since small changes in the sensor data can be signs of discomfort. Though web and mobile application must be visually appealing, so when displaying such data, it would be easier to understand and would look cleaner if the values are rounded to the nearest whole number.

**Fit Criterion:** Any sensor data collected will be stored in the cloud with 5 decimal places. Same will be true when it is used for calculations. Data values that are to be displayed for the user to understand what determines the Dreamline score, must be rounded to the nearest whole number.

**Acceptance Tests:** T15

**ID#** PAR-2 **Name** Data collection frequency

**Description:** Sensors must continuously be collecting data.

**Rationale:** The system must not miss any important fluctuations in sensor data since any minor change can be considered as an indication of comfort level.

**Fit Criterion:** Data that is collected from the sensors must be a continuous line of data where each piece of data is taken in 10ms increments.

**Acceptance Tests:** T16

### 12c Capacity Requirements

**ID#** CAPR-1 **Name** Data storage

**Description:** There must be enough data storage to allow for millions of units of data from multiple sensors to be collected from a single user.

**Rationale:** Since data is continuously being collected from a sensor over a couple of hours of flight, there must be enough storage to hold that information.

**Fit Criterion:** Cloud storage must never be completely full or get too close to being full at any point in time.

**Acceptance Tests:** T16

**ID#** CAPR-2 **Name** Data compression

**Description:** The application must compress data once it has been recorded

**Rationale:** Since there will be a large amount of data collected on a user’s phone over a couple of hours of flight before it is sent over to the cloud, it can clog up the user’s phone storage. The application should not take up too much space on the user’s device.

**Fit Criterion:** Application data should not be over 500MB while the application collects data.

**Acceptance Tests:** T5, T17

**ID#** CAPR-3 **Name** Application download size and app data storage

**Description:** The application must not be greater 500MB and any data that is collected must not stay in the user’s device

**Rationale:** The user’s device will have no use for the data once it has been collected and so the user should not be responsible for storing that data. The application should not take up too much space on the user’s device since that may prevent the user from using the app.

**Fit Criterion:** The download size for the mobile application should not be over 500MB and the all the data collected from sensors in a period of time must be sent to the cloud and removed from the device when the device reconnects to the internet

**Acceptance Tests:** T5, T12

## 13 Dependability Requirements

### 13a Reliability Requirements

**ID#** RELR-1 **Name** Cloud server failures

**Description:** The cloud servers must not fail at all or at most once per year. If failures do occur, data must be backed up and no data must not be lost.

**Rationale:** Failures can cause old data to be lost and any incoming data to be prevented from being stored. This data is key to providing customers up to data comfort levels in flights. Old data will give incorrect information, thus leading to customer dissatisfaction.

**Fit Criterion:** There must be copies of data stored elsewhere in case of a failure in the cloud system, and only 1 largescale failure is allowed per year.

**Acceptance Tests:** T1, T2, T12

**ID#** RELR-2 **Name** Web application and mobile application failures

**Description:** The web and mobile applications must not crash, nor should they be down more than once every 3 months.

**Rationale:** Web and mobile applications undergo changes and updates which may result in the application being down for a couple hours, however, largescale crashes and network traffic must be managed, since clients may buy flight tickets at any moment and may send sensor data at any moment.

**Fit Criterion:** The system must be able to handle many connections and a large amount of incoming data. As a result, the applications must not fail more than 4 times a year.

**Acceptance Tests:** T1, T2, T3, T12

### 13b. Availability Requirements

**ID#** AVR-1 **Name** Web and mobile applications availability

**Description:** The applications must be able to be accessed at any time of the day to browse flights and buy tickets

**Rationale:** User’s must be able to buy a flight at any time of the day as there are flights that occur overnight. Sensor data may come from these overnight flights as well and so the application must be updated with that data at that time of the day.

**Fit Criterion:** The applications must be available to browse and buy tickets at any time of the day, as well as receive sensor data from devices at any time of the day.

**Acceptance Tests:** T3, T12

**ID#** AVR-2 **Name** Mobile application and connected devices with sensors

**Description:** Any device with sensors that collect data, and the mobile application must always be running in the background only during a flight starting at take off until landing.

**Rationale:** The sensor data must only be collected while the user is on a flight as it is only determining levels of discomfort while on the flight. Any other movements or activities will cause inconsistencies in data such as very different heart rates and accelerometer readings that don’t correspond to sitting in the flight itself.

**Fit Criterion:** The application must start collecting data at the scheduled take off time and stop collecting data about 5 hours after landing.

**Acceptance Tests:** T18

### 13c Robustness or Fault-Tolerance Requirements

**ID#** RFTR-1 **Name** Delayed flights

**Description:** The mobile application and connected devices with sensors, must start collecting data at the scheduled take off time and stop collecting data 5 hours after landing time to account for minor delays.

**Rationale:** In the case where flights get delayed or the flight itself takes longer than the scheduled time, the application must be collecting data in a large enough time frame to make sure it is collecting data while they are on the flight.

**Fit Criterion:** The application will stop collecting data after 5 hours after landing, and later determine when the actual take off time and landing was to cutoff unnecessary data.

**Acceptance Tests:** T18

**ID#** RFTR-2 **Name** Canceled flights

**Description:** The mobile application and connected devices with sensors, must not collect data when there are major delays over 2 hours or flight cancellations.

**Rationale:** The flight cancellation themselves would cause incorrect data to be collected and therefore must not be collected at all. Major delays may cause incorrect data since users may feel discomfort after waiting long hours at the airport which effect the comfort level of the delayed flight.

**Fit Criterion:** Any cancellations in a flight or a major delays must cause the application to not collect data for that flight.

**Acceptance Tests:** T18

### 13d Safety-Critical Requirements

**ID#** SCR-1 **Name** Overheating

**Description:** Mobile applications must not cause overheating of the user’s device.

**Rationale:** User’s devices must not overheat as this can cause additional levels of discomfort which can skew sensor data points as well as cause damage to users with devices in their pockets or on their wrist.

**Fit Criterion:** The application must be optimized to limit the CPU usage

**Acceptance Tests:** T4, T11, T12

## 14 Maintainability and Supportability Requirements

### 14a Maintenance Requirements

**ID#** MAINR-1 **Name** Server failures and updates

**Description:** Any updates to the server or updates to the application must be done in off peak hours

**Rationale:** Any maintenance done should not affect the user’s experience, and should affect the least number of users, when there is the least number of users on the applications

**Fit Criterion:** Updates or fixes to the web and mobile applications as well as the servers must be done in off peak hours, typically overnight.

**Acceptance Tests:** T3**,** T12

### 14b Supportability Requirements

**ID#** SUPR-1 **Name** Customer Support

**Description:** There will be an option to chat with customer support in the case of technical issues and payment issues as well as any other miscellaneous issues with the mobile or web applications.

**Rationale:** If customers come across any issues, they must be handled since buying flight tickets can be fairly expensive and the transaction process must run as smoothly as possible.

**Fit Criterion:** There should be a customer chat option on the web and mobile applications where users can chat with a customer representative to resolve any issues.

**Acceptance Tests:** T3**,** T12, T19

**ID#** SUPR-2 **Name** Bug Reports

**Description:** Any bugs within the systems must be reported through a bug reporting system on the web and mobile applications.

**Rationale:** Any bugs or technical issues with the system must be reported in a simple manner which will allow developers to quickly resolve the conflicts.

**Fit Criterion:** Users can input in 1000-word description of the issue and be able to add images that show the bug.

**Acceptance Tests:** T3, T12, T19

### 14c Adaptability Requirements

**ID#** ADAPTR-1 **Name** Platform compatibility

**Description:** The web application must run on any browser on any operating system. The mobile application must be able to run on Android and Apple devices especially smartwatches.

**Rationale:** The mobile application requires the use of sensors such as sleep tracker, accelerometer, Heart Rate monitor and other sensors, which means it must run on Android and Apple devices that have such sensors

**Fit Criterion:** Product should be fully functional on any web browser, Android device, and Apple device of any screen size and must be able to use the sensors mentioned.

**Acceptance Tests:** T1, T2, T3, T6

### 14d Scalability or Extensibility Requirements

**ID#** SER-1 **Name** Product sustainability

**Description:** The cloud servers must be able to handle at least 100000 users per day. This number is expected to double in two years of product release.

**Rationale:** The application must not crash due to network overflow and give all users a lag free experience.

**Fit Criterion:** The servers must grow in proportion to the highest number of users there are at any point in time and never be full.

**Acceptance Tests:** T1, T2, T3, T14

### 14e Longevity Requirements

**ID#** LNGR-1 **Name** Post Release

**Description:** After 5 years of close maintenance and fixing bugs with the developers, the system must be able to run on its own with minimal technical support or issues.

**Rationale:** The systems must be able to run independently and be automated to allow the client to manage it on their own without any issues.

**Fit Criterion:** Any support needed by customers should mainly be for payment related issues or any issues related to the business. There should be minimal issues with the applications and servers.

**Acceptance Tests:** T20

## 15 Security Requirements

**15a Access Requirements**

**ID#** ACC-1 **Name** Access to user data

**Description:** Developers will have access to information obtained from users about flight trips.

**Rationale:** Developers need access to this information to be able to decide which flights are best. They will also need to show important information to other users.

**Fit Criterion:** There should be a save state that is backed up online on another server in case of data loss.

**Acceptance Tests:** T1, T2

**ID#** ACC-2 **Name** General data displayed to client

**Description:** Clients will have access to some general data that was obtained from other users.

**Rationale:** Some information, like leg room and comfort, will be necessary to show to the clients to allow them to decide which flight is best for them.

**Fit Criterion:** Users will not have access to personal information that another user might not want to share.

**Acceptance Tests:** T1, T2

### 15b Integrity Requirements

**ID#** IR-1 **Name** Verify data entry

**Description:** The application will check data that is entered and received from users.

**Rationale:** This is to prevent malicious intent sent to the application. This is also to confirm that the information that is obtained from a user is accurate.

**Fit Criterion:** Allow accurate data to enter into database

**Acceptance Tests:** T1, T2

**ID#** IR-2 **Name** Data is protected

**Description:** The data from the database will be safely backed up on another server.

**Rationale:** This is to prevent loss of data and resources in the case of data loss or physical destruction.

**Fit Criterion:** Makes sure data is safe and protected.

**Acceptance Tests:** T1, T2

### 15c Privacy Requirements

**ID#** PR-1 **Name** Terms and Conditions

**Description:** The application must present Terms and conditions to users before they use it.

**Rationale:** Users must be aware of what information will be collected when they use this application.

**Fit Criterion:** Users are aware of what is being collected when using the application.

**Acceptance Tests:** T3

**ID#** PR-2 **Name** Updates to terms and conditions

**Description:** The application will notify users when there is a change to the terms and conditions.

**Rationale:** Users must be aware of what is being changed in the terms and conditions before continuing to use this app.

**Fit Criterion:** Users are aware of what is being collected and what changed when using the application.

**Acceptance Tests:** T3

**ID#** PR-3 **Name** Use of client’s information

**Description:** The developers will consult the legal team before collecting user’s information.

**Rationale:** Developers will need to be aware of what they are legally allowed to have access to prevent them from infringing on user’s rights and prevent lawsuits.

**Fit Criterion:** Developers are aware of what they are allowed to collect.

**Acceptance Tests:** T4

**ID#** PR-4 **Name** Use of airline’s information

**Description:** The developers will consult the legal team before using airlines information.

**Rationale:** Developers will need to be aware of what information they can use from airlines without getting into legal trouble.

**Fit Criterion:** Developers will be aware of what information they can use from airlines.

**Acceptance Tests:** T4

### 15d Audit Requirements

Not Applicable.

### 15e Immunity Requirements

**ID#** IMMR-1 **Name** Server upkeep

**Description:** The developers will run regular maintenance on the application to protect it from malicious attacks and issues.

**Rationale:** This will be necessary to prevent issues from causing damage. This will help prevent bugs. This will also keep information up to date.

**Fit Criterion:** The application will be up to date with information.

**Acceptance Tests:** T1, T2

**ID#** IMMR-2 **Name** Cyber Security team doing regular maintenance

**Description:** There will be a cyber security team that regularly checks the application for issues and prevents security breaches.

**Rationale:** This is necessary to prevent malicious intent done to the application and protect data.

**Fit Criterion:** The application will be secure from unwanted threats.

**Acceptance Tests:** T1, T2

## 16 Usability and Humanity Requirements

**16a** **Ease of Use Requirements**

**ID#** EUR-1 **Name** Ease of Use for Acquiring Dreamline Score

**Description:** The user must be able to easily navigate and use the application to acquire a Dreamline score with minimal resistance.

**Rationale:** A poor user experience will deter customers from current use and future us of the product.

**Fit Criterion:** 90% of a test panel of ages 16-80 must be able to get a Dreamline score for a flight within 5 minutes.

**Acceptance Tests:** T-21

### 16b Personalization and Internationalization Requirements

**ID#** PIR-1 **Name** Currency Conversion

**Description:** The user should be able to convert the prices of the plane tickets into whichever currency they desire.

**Rationale:** International customers will not always be using USD.

**Fit Criterion:** 100% of users must be able to convert prices into their personally used currency.

**Acceptance Tests:** T-22

**ID#** PIR-2 **Name** Language Translation

**Description:** The user should be able to translate the language of the website into whichever language they desire.

**Rationale:** Not all customers can speak/read English.

**Fit Criterion:** 99% of users must be able to translate the application text into a language they understand.

**Acceptance Tests:** T-23

### 16c Learning Requirements

**ID#** LR-1 **Name** Learnability for new users

**Description:** This product must be easily learned by new users of the application who are unfamiliar with it.

**Rationale:** The product is unusable if the end user cannot learn it.

**Fit Criterion:** 90% of a test panel of new users of ages 16-80 must be able to get a Dreamline score for a flight within 5 minutes.

**Acceptance Tests:** T-21

### 16d Understandability and Politeness Requirements

**ID#** UPR-1 **Name** Dreamline Score Understandability

**Description:** The user must be able to comprehend and understand the meaning of the Dreamline score given to them for a particular flight

**Rationale:** The user cannot utilize the product effectively if they do not know what the Dreamline score means

**Fit Criterion:** 97% of a test panel of ages 16-80 must be able to adequately describe what the Dreamline score means after reading the in-application description.

**Acceptance Tests:** T-24

### 16e Accessibility Requirements

**ID#** AR-1 **Name** Accessibility Of Dreamline Calculation

**Description:** The process of acquiring the Dreamline score should be compatible with the standard accessibility functionalities packaged with common operating systems (i.e., Mac, Windows, Linux).

**Rationale:** The accessibility options will be at the very least up to industry standards.

**Fit Criterion:** 100% of common operating system accessibility tools can be used to navigate and operate the application

**Acceptance Tests:** T-25

### 16f User Documentation Requirements

**ID#** UDR-1 **Name** Adequate Instructions

**Description:** The product should come with adequate instructions for most users to be able to use the application to get a dream score.

**Rationale:** The product is unusable if the end user cannot learn it, and some users may require instruction.

**Fit Criterion:** 95% of a test panel of ages 16-80 must be able to utilize the product to acquire a Dreamline score after reading the user instructions.

**Acceptance Tests:** T-26

**ID#** UDR-2 **Name** Adequate Description

**Description:** The product should come with an adequate description for users to be able to understand the product.

**Rationale:** The product is unusable if the end user cannot understand its functionality or purpose, and some users may require a deliberate description of the product.

**Fit Criterion:** 97% of a test panel of ages 16-80 must be able to adequately describe what the Dreamline score means after reading the in-application description.

**Acceptance Tests:** T-27

**ID#** UDR-3 **Name** Data Collection Instructions

**Description:** The product should come with adequate instructions for most users to be able to use the application on their device to gather data.

**Rationale:** The product is unusable if the end user cannot gather data on their flight.

**Fit Criterion:** 80% of a test panel of ages 16-80 must be able to utilize the product to effectively collect data.

**Acceptance Tests:** T-28

### 16g Training Requirements

**ID#** TR-1 **Name** Minimal Training

**Description:** Users should need no or virtually no training to use the product.

**Rationale:** This product is an application in the public domain for anybody to use.

**Fit Criterion:** 90% of a test panel of ages 16-80 must be able to get a Dreamline score for a flight within 5 minutes.

**Acceptance Tests:** T-21

## 17 Look and Feel Requirements

### 17a Appearance Requirements

**ID#** LF-1 **Name** User friendly appearance

**Description:** The application will be appealing and user friendly.

**Rationale:** If the product is complicated or unappealing, it will repel users from trying to use it.

**Fit Criterion:** The office of branding will make sure the application complies with current standards.

**Acceptance Tests:** T3, T5

**ID#** LF-2 **Name** Modern appearance

**Description:** The application will have a modern appearance to attract users.

**Rationale:** The target audience for this application will be young adults/adults. We will need a modern appearance to attract this audience and compete against any other applications.

**Fit Criterion:** The office of branding will make sure the application complies with current standards.

**Acceptance Tests:** T3, T5

### 17b Style Requirements

**ID#** SR-1 **Name** The product will look professional

**Description:** The mood of the application will be professional and simple.

**Rationale:** This influences users by making the application look well put together. This will help develop trust with the user, which is needed since we will use their personal data.

**Fit Criterion:** After a user encounters the product, we expect that at least 75% of users will trust and enjoy the product.

**Acceptance Tests:** T3, T5

**ID#** SR-2 **Name** Light and dark mode

**Description:** The application will have a light and dark mode to choose from.

**Rationale:** Many applications and browsers are now offering a dark mode. This attracts many users that enjoy this feature and will help attract more users to this application.

**Fit Criterion:** After a user encounters the product, we expect that they’ll enjoy being able to choose between a light and dark mode.

**Acceptance Tests:** T3, T5

## 18 Operational and Environmental Requirements

### 18a Expected Physical Environment

**ID#** EPE-1 **Name** Application usability

**Description:** The app will primarily be used on mobile devices. It can also be used in an internet browser but with limited functionality.

**Rationale:** The app can be used to see prices and options through an internet browser, but in order to obtain information while flying it must be using through a mobile device.

**Fit Criterion:** This app will be functional on any preferred mobile device.

**Acceptance Tests:** T3, T5, T6

**ID#** EPE-2 **Name** Application environment

**Description:** The app will primarily be used during plane travel. The app will record information while someone is on an airplane or in an airline.

**Rationale:** The app needs user information to be able to have accurate and useful information for other users.

**Fit Criterion:** This app will be most often used on a plane.

**Acceptance Tests:** T3, T5

### 18b Requirements for Interfacing with Adjacent Systems

**ID#** IAS-1 **Name** Mobile software

**Description:** The app will be compatible with the newest versions of IOS and Android.

**Rationale:** This will allow almost anyone to use this application who own a device with IOS or Android.

**Fit Criterion:** A fully working application on any android or iPhone.

**Acceptance Tests:** T3, T5, T6

**ID#** IAS-2 **Name** Internet browsers

**Description:** The app will be compatible on the latest versions of popular browsers like google chrome, Microsoft edge, and internet explorer.

**Rationale:** Compatibility with these web browsers will allow anyone to use it and optimize the functionality on each browser.

**Fit Criterion:** A fully working application on popular browsers.

**Acceptance Tests:** T3, T5, T6

### 18c Productization Requirements

**ID#** PRO-1 **Name** App store

**Description:** The application will be easily accessible through any iOS or android app store.

**Rationale:** This will allow a user to easily find and use the app with just the name of the app.

**Fit Criterion:** Users will access the app through an app store.

**Acceptance Tests:** T6, T5

**ID#** PRO-2 **Name** Domain name

**Description:** The application will be easily accessible through a domain name.

**Rationale:** This will allow a user to easily find and use the app with just a domain name.

**Fit Criterion:** Users will access the app through any browser with just the name.

**Acceptance Tests:** T6, T5

### 18d Release Requirements

**ID#** RREQ-1 **Name** Development cycle

**Description:** The development cycle of this application will be one year.

**Rationale:** The application will have a budget that will correctly fund the team to finish the project in a year.

**Fit Criterion:** This project will be done in a year without going over the set budget.

**Acceptance Tests:** T7

**ID#** RREQ-2 **Name** New Updates

**Description:** There will be new updates added to the application at least once a year.

**Rationale:** The application will have room for improvements based on user’s and client's feedback.

**Fit Criterion:** The project will adapt and become better over time.

**Acceptance Tests:** T5, T6

## 19 Cultural and Political Requirements

### 19a Cultural Requirements

Not applicable

### 19b Political Requirements

Not applicable

## 20 Legal Requirements

### 20a Compliance Requirements

**ID#** COMR-1 **Name** Data Collection

**Description:** The application must only collect data from those who consent to data collection.

**Rationale:** It is a violation of privacy to gather information without consent.

**Fit Criterion:** Data collection can only be collected after the filling out of a consent form

**Acceptance Tests:** T4

**ID#** COMR-2 **Name** Airline Collection

**Description:** The application must only collect data within the bounds of what is allowed by each airline.

**Rationale:** It is a violation of privacy to take private data from a company without the consent of the company

**Fit Criterion:** Information from airlines will be gathered within legal bounds

**Acceptance Tests:** T4

### 20b Standards Requirements

**ID#** STANR-1 **Name** Airline Reservation System

**Description:** The product must be compliant with airline reservation system industry standards.

**Rationale:** In order to be a successful product, the product must be up to par with its competitors in the airline reservation system industry.

**Fit Criterion:** The product must function as efficiently as its competitors (e.g., Expedia, Booking.com)

**Acceptance Tests:** T20, T19, T17, T12, T10, T9, T8, T6, T5, T4, T3, T2, T1

## 21 Requirements Acceptance Tests

### 21a Requirements – Test Correspondence Summary

**

*Table 1 - Requirements - Acceptance Tests Correspondence*



*Table 2 - Requirements - Acceptance Tests Correspondence Cont.*

### 21b Acceptance Test Descriptions

**ID #** T1 **Name** Test save data files

**Description:** The test will pass if the application’s files can be correctly saved and loaded. It will also check to see if this can be done within half a second.

**ID #** T2 **Name** Test local and server files

**Description:** The test will pass if the application’s local and server files can correctly be saved and loaded. It will also test and make sure it can be loaded within 2 seconds.

**ID #** T3 **Name** Test interface

**Description:** The test will pass if the interface is visible to the user once the application is opened.

**ID #** T4 **Name** Test legal constraints

**Description:** This test will pass if the legal team doesn’t have any issues and there are no repercussions after the application is released.

**ID #** T5 **Name** Test application usability

**Description:** This test will pass if user is able to download and use the application without issue.

**ID #** T6 **Name** Test multi-platform usability

**Description:** This test passes if the application works on every platform (iOS, android, browsers, etc.).

**ID #** T7 **Name** Budget constraint

**Description:** This test passes if the application was finished within set budget.

**ID #** T8 **Name** Itinerary

**Description:** This test passes if the system correctly receives a valid itinerary from the user

**ID #** T9 **Name** Ticket

**Description:** This test passes if the system correctly receives valid ticket information from airlines

**ID #** T10 **Name** Price Sorting

**Description:** This test passes if the system correctly receives and sorts tickets based on price

**ID #** T11 **Name** Sensor Data

**Description:** This test passes if the system correctly receives sensor data from the user’s smart device

**ID #** T12 **Name** Review

**Description:** This test passes if the system correctly receives a valid review from the user

**ID #** T13 **Name** Dreamline Score

**Description:** This test passes if the system correctly calculates and assigns a Dreamline score to every ticket. Expected behavior like a decrease in price resulting in an increase in score. Increase in review scores and sensor comfort causing an increase in the DL score

**ID #** T14 **Name** Dreamline Score latency

**Description:** This test passes if the system can receive sensor data and update the Dreamline score within 10 seconds.

**ID #** T15 **Name** Data precision

**Description:** This test passes if the data collected from sensors must be precise to 5 decimal places

**ID #** T16 **Name** Data storage

**Description:** This test passes if the cloud storage can hold millions of data points without ever getting full where all the data points for a particular flight for a particular user are a stream of data with 10ms increments.

**ID #** T17 **Name** Data compression

**Description:** This test passes if the data files that are received by the cloud are all less than 500MB.

**ID #** T18 **Name** Data Collection

**Description:** This test passes if the sensors collect data at a certain time period that represents the range in which the user is flying.

**ID #** T19 **Name** Technical Support

**Description:** This test passes if the system is able to respond to customer reviews and bugs, as well as connect with users to resolve technical issues.

**ID #** T20 **Name** Sustainability

**Description:** This test passes if the system is still operational and providing the client with thousands of active users long after 5 years.

**ID #** T21 **Name** Panel #1

**Description:** This test passes if at least90% of a test panel made up of individuals unexperienced with the product from ages 16-80 can get a Dreamline score within 5 minutes.

**ID #** T22 **Name** Currency Conversion

**Description:** This test passes if the application can convert a price correctly into every used currency

**ID #** T23 **Name** Language Translation

**Description:** This test passes if the application contains the correct translation for the 30 most commonly spoken languages.

**ID #** T24 **Name** Panel #2

**Description:** This test passes if at least 97% of a test panel of individuals unexperienced with the product from ages 16-80 are able to adequately describe what the Dreamline score means after reading the in-application description.

**ID #** T25 **Name** Accessibility

**Description:** This test passes if all accessibility accommodations provided by common operating systems are compatible and function normally when used with our product.

**ID #** T26 **Name** Panel #3

**Description:** This test passes if at 95% of a test panel of individuals from ages 16-80 are able to utilize the product to acquire a Dreamline score after reading the user instructions.

**ID #** T27 **Name** Panel #4

**Description:** This test passes if at least 97% of a test panel of individuals from ages 16-80 are able to adequately describe what the Dreamline score means after reading the in-application description.

**ID #** T28 **Name** Panel #5

**Description:** This test passes if at least 80% of a test panel of individuals ages 16-80 are able to utilize the product to effectively collect data with no data collection errors.

# III Design

## 22 Design Goals

Dreamline strives to provide accurate information to guarantee users receive the best option for a flight based on comfort and price. This means that the system must go through multiple verification processes of data to make sure its accurate before storing it and displaying it to the user. Since browsing for a flight can often times be a lengthy process, this will give us enough time to make sure we provide the best data without having to sacrifice speed or efficiency

## 23 Current System Design

There is no pre-existing system.

## 24 Proposed System Design

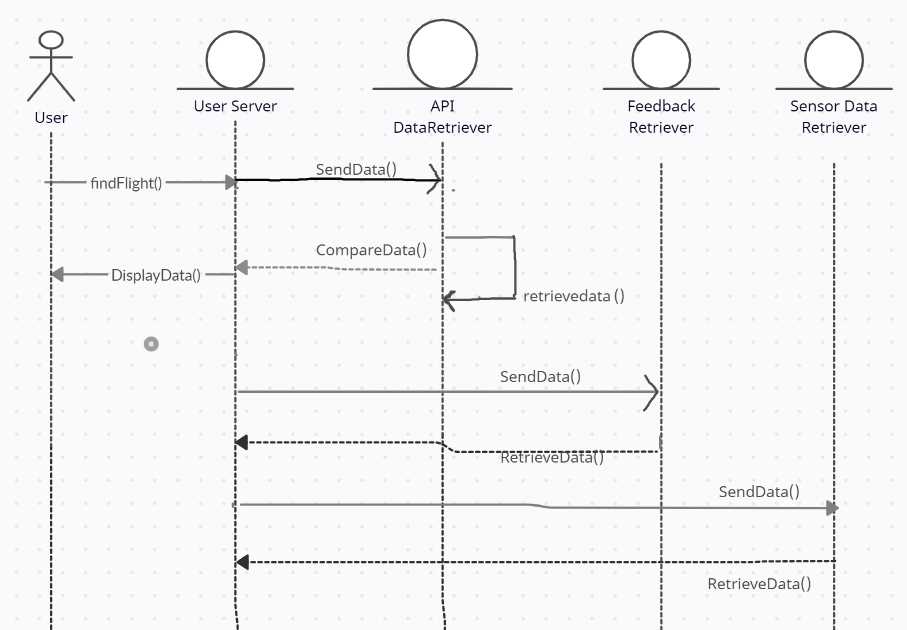
### 24a Initial System Analysis and Class Identification

* User – The user will class itself will only have an instance of TripInfo class which holds the users trip information. This includes departure and arrival information called departDate, returnDate, fromAirportCode, toAirportCode which are all Strings. Then we will have a boolean that determines whether it is a round trip or one way call roundTrip. Finally, there will be an integer representing the number of passengers called numPassengers. There will be a function called findFlight which returns a list of the best flights that match their trip information. A function called startTracking which will start collecting sensor data.
* UserServer – has 3 functions called sendData 1,2, and 3 which retrieves ratings for the list of flights and creates a list of the best flights. It will also have a function to display the list called displayData.
* DataServer – a function that collects the feedback data and tracks and collect the sensor data called collectData()
* APIDataRetriever – has a function to retrieve raw data in the form of a SensorData instance called retrieveData. It has a function to compare the data to a standard to create a list of FlightInfoRatings
* FeedbackRetriever – has a function to simply retrieve a list of feedback ratings from the database in the form of an instance of the Form class called retrieveData.
* SensorDataRetriever – has a function to retrieve a list of SensorRatings from the database called retrieveData
* SensorData - holds raw sensor data collected from smart devices. A long called sleepTrackerData and Maps of String, double called accelerometerData, heartRateMonitorData, and oximeterData.
* Form – holds the ratings from the feedback given from users. There will be integers called sleepRating, fulfillmentRating, amenityRating, comfortRating which represent the comfort of the flight.
* APIData - hold raw flight information taken from APIs. 2 doubles called price and legroom. 2 longs called departureTime and returnTime. It will also have a Map of Strings and booleans called amenities, which represent what amenities the flight includes.

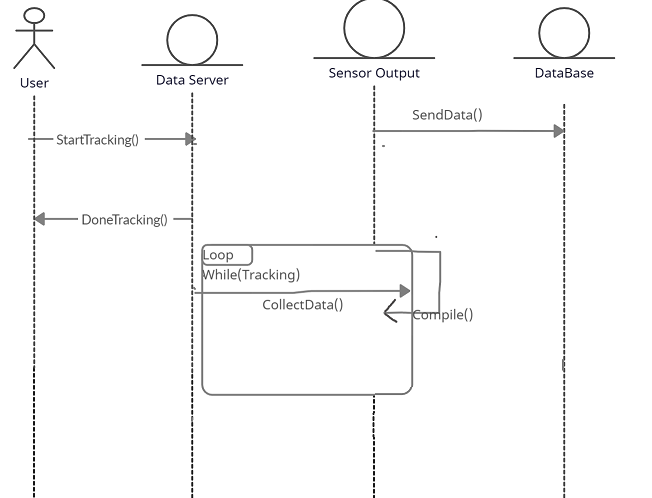
Diagram

Description automatically generated

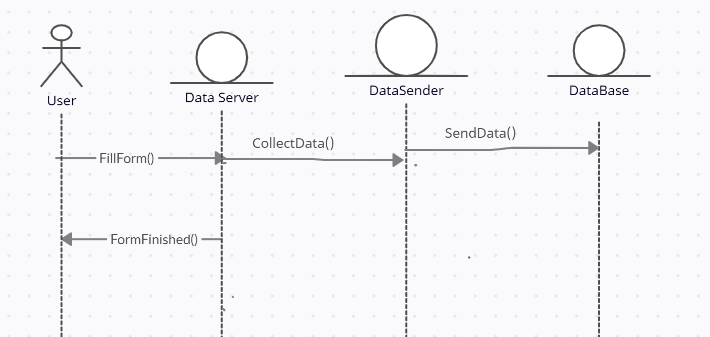
### 24b Dynamic Modelling of Use-Cases



This sequence diagram shows Use Case ID-1,2,4, and 6: This diagrams shows the process of the user inputs their itinerary, and the UserServer will retrieve a list of the cheapest flights. It also retrieves ratings of feedback and comfort levels from flights. It uses these 3 pieces of information to calculate a Dreamline score for each flight that matches the itinerary and displays it to the user.



This sequence diagram shows Use Case ID - 3. This diagram shows how that data is tracked from the smart devices of the user once they are on the flight. Once the DataServer is done collecting data, it uses a SensorOutput instance to compile the raw data into ratings and sends the data to the database.



This sequence diagram shows Use Case – 5. This diagram shows how the user leaves feedback about a flight which is then stored into the database. The User fills a form which is received by the DataServer which sends it to the database using an instance of DataSender.

### 24c Proposed System Architecture

The proposed system architecture will be Client-Server, which will allow users to seamlessly interact with the application and browse for flights as well as send sensor data, without having to do much of the backend work. This will prevent a heavy load on the user’s device and won’t take up much memory or space. This will also help remove performance issues when scaling the application.

### 24d Initial Subsystem Decomposition

**Timeline

Description automatically generated**

## 25 Additional Design Considerations

### 25a Hardware / Software Mapping

We will have a client-server architecture that will require hardware and software functionality. The first point of access for the user would be their cellular device which will have the Dreamline app that can display Dreamline scored flights and allow purchasing of them. For them to get the data it will need a connection with the server that will host the flight information on a database it receives by calling the flight request APIs. The server will perform Dreamline calculations to get the score and return that to the User’s smart device through their App. Then once the flight has started, the user’s app will start receiving smart device data. The app will receive sensor data from the phone itself and any connected smart devices. Once the flight is completed the sensor data will be uploaded to the Server for Dreamline score calculation.

### 25b Persistent Data Management

Storage of raw flight data is not of importance due to it being constantly changing and easily requestable from airlines through APIs calls. It is of extreme importance that the Dreamline data is stored in a database and a cloud system to make sure it is never lost. The Dreamline data would include user, sensor and review score data. The application aims for 24/7 functionality but in the event of failure retaining this data is of utmost importance because it is essential in the determining of the Dreamline score. Amazon Web Services, Microsoft Azure or Google Cloud are potential solutions to this.

### 25c Access Control and Security

Customer data security is of utmost importance. The sensor data that they share with Dreamline are done so with a level of trust that this information will not be shared or compromised. As a result, individual comfort tracking is not necessary but data aggregation. So once the data has been received, it should not be attached to a user. It should solely be used for comfort calculation. To further protect the data, the sensor data should only be uploaded at one point, after the flight. At that point when it is sent back it should be encrypted and safely received by the server to ensure there are no data leaks.

### 25d Global Software Control

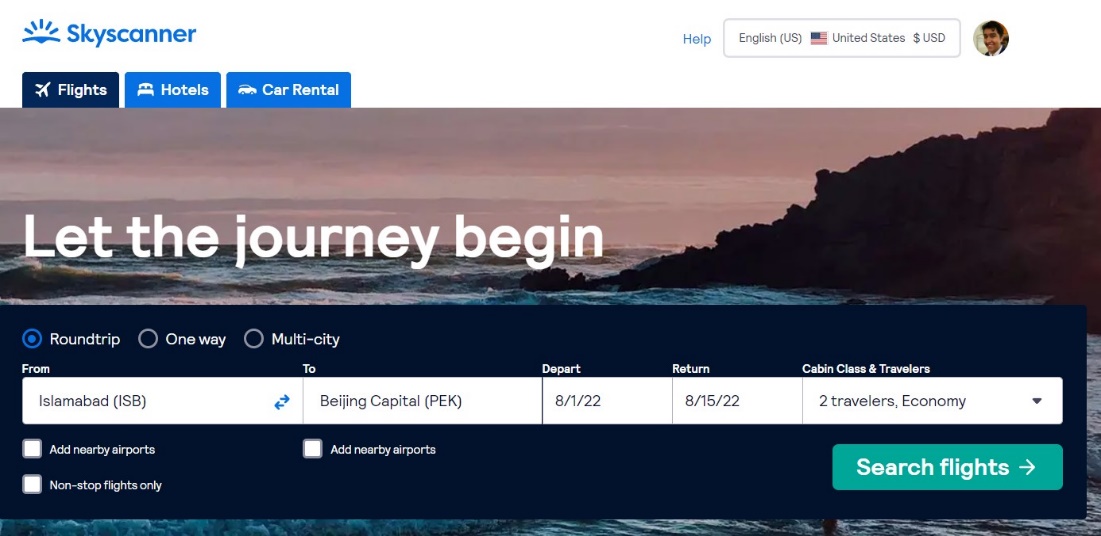
Flight information and access are standardized around the globe through various aviation councils. As a result, the ticket purchasing procedure can already meet those requirements when going through airline purchasing APIs. When it comes to gathering user data, restrictions vary around the world. As a result, the app should meet European Union regulations which are one of the strictest. If those standards are met, it will satisfy other countries' requirements as well.

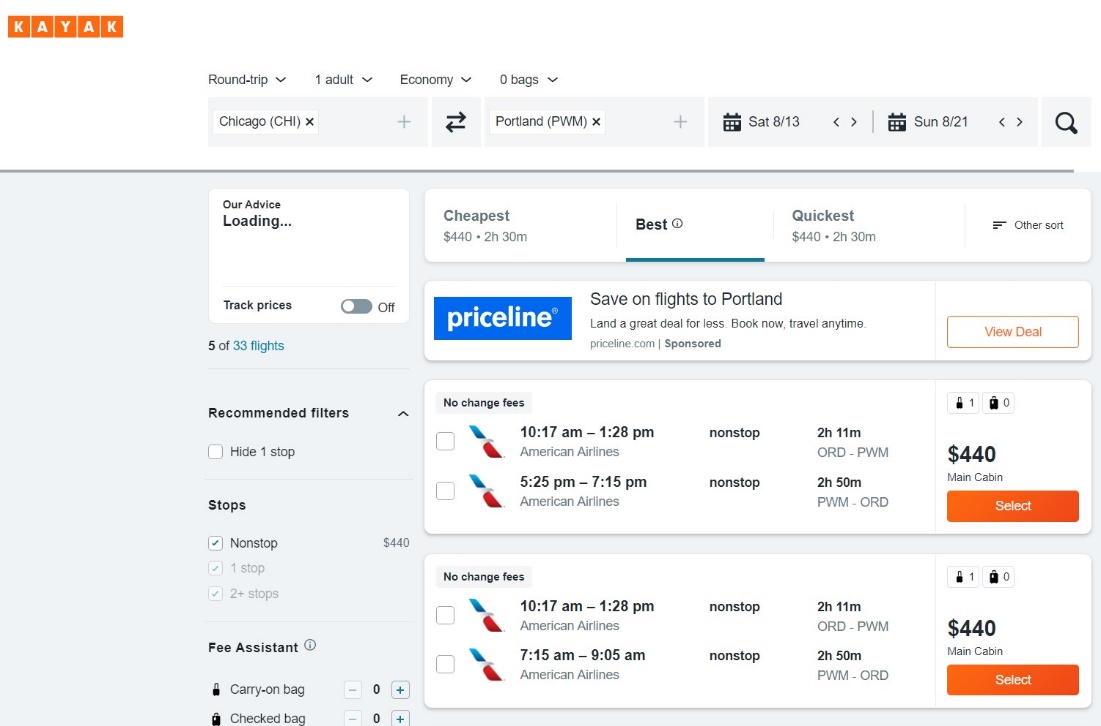
### 25e Boundary Conditions

The system aims to run constantly with no downtime for any scenario, including maintenance. As a result, any changes committed to the website must be able to occur without any interruption. In case of an abnormal shutdown, the system data must be stored on a secure database that is routinely uploaded to the cloud. The maintenance team will try to resume service as soon as possible. At that point all data should be usable and the application accessible by the user.

### 25f User Interface

User interface particulars are flexible. The only requirement is that it needs to be simple so any flyer can easily purchase their flight and upload their sensor information. Similar competitor UIs for reference





## 26 Final System Design

Diagram

Description automatically generated

## 27 Object Design

### 27a Packages

Google Flights API, TravelPort API, and smart device SDKs will be utilized. Refer to section 24d for more information on subsystems.

# IV Project Issues

## 28 Open Issues

* + We do not have knowledge of airlines’ allowance of the use of sensors to gather information about the plane while mid-flight.
  + Not every sensor is available on every smart device (e.g. oximeter).

## 29 Off-the-Shelf Solutions

Airline APIs and their fetching systems are prevalent and well documented allowing for a good cut to production time.

### 29a Ready-Made Products

* + Smart device SDKs

### 29b Reusable Components

* + Airline APIs are freely available and used extensively by our product.

### 29c Products That Can Be Copied

* + Other airline ticket aggregators such as Expedia and Booking.com can be studied for easier implementation of the API price fetching subsystem

## 30 New Problems

### 30a Effects on the Current Environment

The new system can cause the staff and employees to be more aware of their actions since they will be considered in the score of a plane. The system can cause airlines to take into consideration the planes they use since they know it will affect their sales. Any changes made to airplanes and airlines will affect the score of the plane and will have to be noted in the database. The new system shouldn’t be switched to a different operating system, it will be designed to work for android and iOS. Switching to a different operating system will result in the program needing to be reconstructed. The new system might also negatively affect other review companies and force them to reconstruct their software.

### 30b Effects on the Installed Systems

This application will be developed for mobile/wearable technology. It must be able to work on any android or IOS smart phone and smart watch with no issues. This application must be optimized for these platforms so that issues like memory leaks and bugs are avoided. This would otherwise cause the application to be slow and could produce inaccurate results. It must also be able to communicate with a database with no issues, Otherwise the results will be inaccurate. The application will also be required to be optimized with the user's hardware in mind. The app must work efficiently on any IOS/android device. If it isn’t, it can cause some devices to slow down, and can cause overheating. The app must also be efficient to preserve the battery life of the user’s device.

### 30c Potential User Problems

There can be some features that will negatively affect people with photosensitive epilepsy. Some patterns/photos can trigger seizures for people with this condition. There will be a warning because of this and a setting that will allow a more basic version for people. There can be issues with old smart devices not being able to track data with our application. This is because they may not have any of the sensors needed and can be too slow to handle the app.

### 30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Dreamline will have minimum requirements for each user’s android or IOS device. If the user’s device does not meet these requirements, then the application may work poorly. The application may also not work on certain devices, e.g., Fitbit and Gizmo watch. We also cannot assume that every user will have access to Wi-Fi during the entire flight. We will not be able to upload all user data until the data is reconnected to Wi-Fi. On many flights, there is also a lack of reliable power. This can cause some user’s device to have the battery drained while in use, causing the application to fail during flight. Bad weather/high altitude can also cause issues with some smart devices, which can cause issues with the application. The bad weather can also interfere with the sensors, causing inaccurate data to be sent to the database. If our application performs better than some of the competitors, like Expedia and Cheapflights, then the planned server and database will not be sufficient for the new users.

### 30e Follow-Up Problems

* + Bugs can appear after the launch of the application. This can cause many issues like a slow app, the app not working, and inaccurate results.
  + There is a possibility of going over the budget for this application.
  + Lack of comfort options for specific set of users.
  + Issues with design choices for this application.
  + Having short deadlines to create the application, this can cause issues with developers and a less functioning application.
  + The application can be delayed if it is not finished on time.

## 31 Migration to the New Product

Not Applicable

## 32 Risks

**32a Inaccurate Metric Risks**

Some of the metrics can be unprecise or inaccurate. There will be information coming from many users and there will need to be a way to filter through this information productively to produce accurate results.

**32b Inadequate Measurement**

There's a risk that many users will not have the sensors needed in their device to provide data. We will need to assume that enough users with smart devices that contain these sensors will use the application.

**32c Excessive Schedule Pressure**

It is unknown exactly how long it will take for this application to be developed. The developers can underestimate the time that is needed to develop this project. This can cause some features not being developed into the application at the first launch.

**32d Low Quality and Productivity Risks**

There is a risk that the application may not be completed in the given time frame. This can be caused by low productivity which will lead to a lower quality application.

**32e Creeping User Requirements**

This app will need to accommodate the needs of many users. If this application fails to meet the needs of enough users, then it may not be profitable.

## 33 Costs

Most funding for this project will go towards the actual engineering of the application, including frontend, backend and user experience. A team of 5-7 engineers would likely be able to complete this product over the course of 9-12 months. According to indeed.com (Indeed, 2022) the average yearly salary of a software engineer is $95,000 leading to a total cost of $356,000 – $665,000.

## 34 Waiting Room

In the future we would like to include more sensors to for users to collect data on the plane, for example:

* + Temperature (ambient and individual)
  + Blood pressure
  + Air quality

## 35 Ideas for Solutions

We would likely need to have users use separate peripherals with sensors built in in order to meet these requirements. For example, a blood pressure sensor could be exclusive to diabetic users, in which they can opt in to connect their blood pressure monitor to our application in order to keep track of it over the course of the flight.

In the future, more sensors may be added to smart devices such as air quality monitoring. This would make it easier on our end to track such statistics. However, as of right now, these sensors would likely require external sensors to be given to users. We could fix this problem by offering to loan long-time users of the application (especially frequent fliers and those with a history of collecting data for us) sensor-containing peripherals that they can bring on the flight to collect data.

## 36 Project Retrospective

This project started with some road bumps. Initially the design relied on price tracking as its sensor data since it would upload live. That was an incorrect assumption as it needed to be sensor data. We quickly pivoted to smart device data to augment the comfort level tracking process. That met the requirements as well as being a more accurate way of creating the Dreamline score. When it comes to individual participation, it was highly successful. Different sections of the document were grouped together by similarity, at which point people picked sections to work on. Communication was constant through Discord which made sure everyone aligned with a central vision. The final project is now actualizable.

# References / Bibliography

# Bibliography

(n.d.). (Skyscanner Ltd.) Retrieved from https://www.partners.skyscanner.net/affiliates/travel-apis

(n.d.). (Amadeus IT Group SA) Retrieved from https://developers.amadeus.com/

(n.d.). (Sabre) Retrieved from https://developer.sabre.com/home

(n.d.). (Travelport) Retrieved from https://www.travelport.com/products/api

(n.d.). (Alvarado Mfg. Co., Inc.) Retrieved from https://www.alvaradomfg.com/directconnect-api/

Bell, J. (2012). *Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports.* Chicago: Underwater Archaeological Society of Chicago.

Fowler, M. (2004). *UML Distilled, Third Edition.* Boston: Pearson Education.

*Indeed*. (2022, April 26). Retrieved from https://www.indeed.com/career/software-engineer/salaries/IL

Lindsey. (2020, JULY 29). *Google Flights API: Incorporate Travel Data into Your App*. Retrieved from rapidapi.com: https://rapidapi.com/blog/google-flights-api-incorporate-travel-data-into-your-app/

Robertson, & Robertson. (n.d.). *Mastering the Requirements Process.*

Silberschatz, A., Galvin, P. B., & Gagne, G. (2013). *Operating System Concepts* (Ninth ed.). Wiley.

# Index

Design 61, 63

Requirements 35, 51, 58

Test 64, 65