***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**

[*REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT* 2](#_Toc529138956)

[*How to Use This Document* 2](#_Toc529138957)

[List of Figures 9](#_Toc529138958)

[List of Tables 10](#_Toc529138959)

[I Project Description 11](#_Toc529138960)

[1 Project Overview 11](#_Toc529138961)

[2 The Purpose of the Project 11](#_Toc529138962)

[2a The User Business or Background of the Project Effort 11](#_Toc529138963)

[2b Goals of the Project 11](#_Toc529138964)

[2c Measurement 12](#_Toc529138965)

[3 The Scope of the Work 13](#_Toc529138966)

[3a The Current Situation 13](#_Toc529138967)

[3b The Context of the Work 13](#_Toc529138968)

[3c Work Partitioning 16](#_Toc529138969)

[3d Competing Products 18](#_Toc529138970)

[4 The Scope of the Product 18](#_Toc529138971)

[4a Scenario Diagram(s) 19](#_Toc529138972)

[4b Product Scenario List 19](#_Toc529138973)

[4c Individual Product Scenarios 19](#_Toc529138974)

[5 Stakeholders 20](#_Toc529138975)

[5a The Client 20](#_Toc529138976)

[5b The Customer 20](#_Toc529138977)

[5c Hands-On Users of the Product 21](#_Toc529138978)

[5d Maintenance Users and Service Technicians 22](#_Toc529138979)

[5e Other Stakeholders 22](#_Toc529138980)

[5f User Participation 23](#_Toc529138981)

[5g Priorities Assigned to Users 24](#_Toc529138982)

[6 Mandated Constraints 25](#_Toc529138983)

[6a Solution Constraints 25](#_Toc529138984)

[6b Implementation Environment of the Current System 26](#_Toc529138985)

[6c Partner or Collaborative Applications 27](#_Toc529138986)

[6d Off-the-Shelf Software 28](#_Toc529138987)

[6e Anticipated Workplace Environment 29](#_Toc529138988)

[6f Schedule Constraints 30](#_Toc529138989)

[6g Budget Constraints 31](#_Toc529138990)

[7 Naming Conventions and Definitions 31](#_Toc529138991)

[7a Definitions of Key Terms 31](#_Toc529138992)

[7b UML and Other Notation Used in This Document 32](#_Toc529138993)

[7c Data Dictionary for Any Included Models 33](#_Toc529138994)

[8 Relevant Facts and Assumptions 34](#_Toc529138995)

[8a Facts 34](#_Toc529138996)

[8b Assumptions 34](#_Toc529138997)

[II Requirements 36](#_Toc529138998)

[9 Product Use Cases 36](#_Toc529138999)

[9a Use Case Diagrams 36](#_Toc529139000)

[9b Product Use Case List 38](#_Toc529139001)

[9c Individual Product Use Cases 38](#_Toc529139002)

[10 Functional Requirements 40](#_Toc529139003)

[11 Data Requirements 41](#_Toc529139004)

[12 Performance Requirements 42](#_Toc529139005)

[12a Speed and Latency Requirements 42](#_Toc529139006)

[12b Precision or Accuracy Requirements 43](#_Toc529139007)

[12c Capacity Requirements 44](#_Toc529139008)

[13 Dependability Requirements 45](#_Toc529139009)

[13a Reliability Requirements 45](#_Toc529139010)

[13b Availability Requirements 46](#_Toc529139011)

[13c Robustness or Fault-Tolerance Requirements 47](#_Toc529139012)

[13d Safety-Critical Requirements 48](#_Toc529139013)

[14 Maintainability and Supportability Requirements 49](#_Toc529139014)

[14a Maintenance Requirements 49](#_Toc529139015)

[14b Supportability Requirements 50](#_Toc529139016)

[14c Adaptability Requirements 51](#_Toc529139017)

[14d Scalability or Extensibility Requirements 52](#_Toc529139018)

[14e Longevity Requirements 52](#_Toc529139019)

[15 Security Requirements 53](#_Toc529139020)

[15a Access Requirements 53](#_Toc529139021)

[15b Integrity Requirements 54](#_Toc529139022)

[15c Privacy Requirements 55](#_Toc529139023)

[15d Audit Requirements 56](#_Toc529139024)

[15e Immunity Requirements 57](#_Toc529139025)

[16 Usability and Humanity Requirements 57](#_Toc529139026)

[16a Ease of Use Requirements 57](#_Toc529139027)

[16b Personalization and Internationalization Requirements 59](#_Toc529139028)

[16c Learning Requirements 60](#_Toc529139029)

[16d Understandability and Politeness Requirements 61](#_Toc529139030)

[16e Accessibility Requirements 62](#_Toc529139031)

[16f User Documentation Requirements 63](#_Toc529139032)

[16g Training Requirements 64](#_Toc529139033)

[17 Look and Feel Requirements 65](#_Toc529139034)

[17a Appearance Requirements 65](#_Toc529139035)

[17b Style Requirements 66](#_Toc529139036)

[18 Operational and Environmental Requirements 67](#_Toc529139037)

[18a Expected Physical Environment 67](#_Toc529139038)

[18b Requirements for Interfacing with Adjacent Systems 68](#_Toc529139039)

[18c Productization Requirements 69](#_Toc529139040)

[18d Release Requirements 70](#_Toc529139041)

[19 Cultural and Political Requirements 71](#_Toc529139042)

[19a Cultural Requirements 71](#_Toc529139043)

[19b Political Requirements 72](#_Toc529139044)

[20 Legal Requirements 73](#_Toc529139045)

[20a Compliance Requirements 73](#_Toc529139046)

[20b Standards Requirements 74](#_Toc529139047)

[21 Requirements Acceptance Tests 75](#_Toc529139048)

[21a Requirements – Test Correspondence Summary 75](#_Toc529139049)

[21b Acceptance Test Descriptions 76](#_Toc529139050)

[III Design 76](#_Toc529139051)

[22 Design Goals 76](#_Toc529139052)

[23 Current System Design 76](#_Toc529139053)

[24 Proposed System Design 76](#_Toc529139054)

[24a Initial System Analysis and Class Identification 77](#_Toc529139055)

[24b Dynamic Modelling of Use-Cases 77](#_Toc529139056)

[24c Proposed System Architecture 77](#_Toc529139057)

[24d Initial Subsystem Decomposition 77](#_Toc529139058)

[25 Additional Design Considerations 77](#_Toc529139059)

[25a Hardware / Software Mapping 77](#_Toc529139060)

[25b Persistent Data Management 78](#_Toc529139061)

[25c Access Control and Security 78](#_Toc529139062)

[25d Global Software Control 78](#_Toc529139063)

[25e Boundary Conditions 78](#_Toc529139064)

[25f User Interface 78](#_Toc529139065)

[25g Application of Design Patterns 79](#_Toc529139066)

[26 Final System Design 79](#_Toc529139067)

[27 Object Design 79](#_Toc529139068)

[27a Packages 79](#_Toc529139069)

[27b Subsystem I 79](#_Toc529139070)

[27c Subsystem II 79](#_Toc529139071)

[27d etc. 79](#_Toc529139072)

[IV Project Issues 80](#_Toc529139073)

[28 Open Issues 80](#_Toc529139074)

[29 Off-the-Shelf Solutions 80](#_Toc529139075)

[29a Ready-Made Products 81](#_Toc529139076)

[29b Reusable Components 81](#_Toc529139077)

[29c Products That Can Be Copied 81](#_Toc529139078)

[30 New Problems 82](#_Toc529139079)

[30a Effects on the Current Environment 82](#_Toc529139080)

[30b Effects on the Installed Systems 83](#_Toc529139081)

[30c Potential User Problems 83](#_Toc529139082)

[30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 84](#_Toc529139083)

[30e Follow-Up Problems 84](#_Toc529139084)

[31 Migration to the New Product 85](#_Toc529139085)

[31a Requirements for Migration to the New Product 85](#_Toc529139086)

[31b Data That Has to Be Modified or Translated for the New System 86](#_Toc529139087)

[32 Risks 86](#_Toc529139088)

[33 Costs 87](#_Toc529139089)

[34 Waiting Room 88](#_Toc529139090)

[35 Ideas for Solutions 89](#_Toc529139091)

[36 Project Retrospective 90](#_Toc529139092)

[V Glossary 90](#_Toc529139093)

[VI References / Bibliography 91](#_Toc529139094)

[VII Index 91](#_Toc529139095)

### ****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.

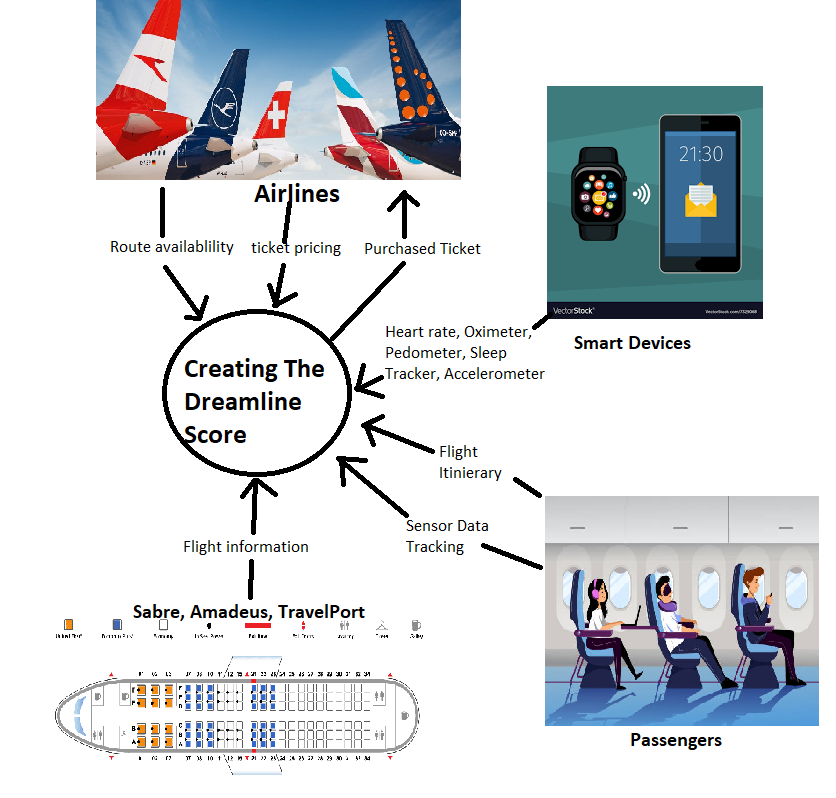
## 3 The Scope of the Work

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

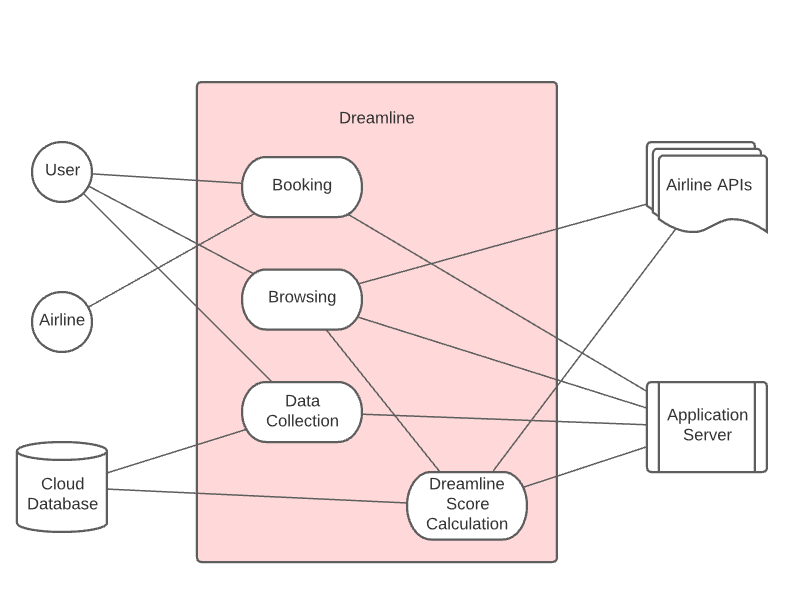
|  |  |  |
| --- | --- | --- |
| **Event Name** | **I/O** | **Summary** |
| The passenger describes the flight itinerary | Passengers (IN) | The passenger gives the origin and destination and the dates to build the Dreamline score on |
| Check Route Availability/ Get ticket pricing | Airlines (IN) | 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.

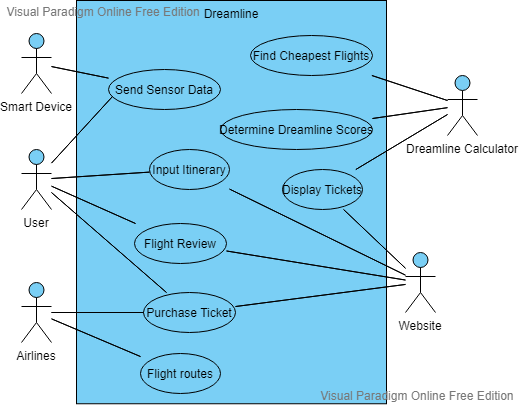
From a business perspective, we are assuming that people are willing and able to seek out a method to weigh their options when buying plane tickets. Furthermore, a reasonable portion of these people (15%) must also be willing to allow us to track them over the course of their flight.

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. |

### 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

*SV: Identify the important design goals that are to be optimized in the proposed design.*

*Content*

*Design goals are important properties of the system to be optimized, and which may affect the overall design of the system. For example, computer games place a higher priority on speed than accuracy, and so the physics engine for a computer game may make some rough approximations and assumptions that allow it to run as fast as possible while sacrificing accuracy, whereas the physics calculations performed by NASA must be much more rigorously correct, even at the expense of speed.*

*Note an important difference between design goals and requirements: Requirements include specific values that must be met in order for the product to be acceptable to the client, whereas design goals are properties that the designers strive to make "as good as possible", without specific criteria for acceptability. (Note also that the same property may appear in both a requirement and a design goal, so a design goal may be to make the system run as fast as possible, with a requirement that says any speed below a certain specified threshold is unacceptable.)*

Your text goes here . . .

## Current System Design

*SV:* ***IF*** *the proposed new system is to replace an existing system, then the current system should be described here. Otherwise insert a brief statement that there is no pre-existing system.*

Your text goes here . . .

## Proposed System Design

*This section will make heavy use of class diagrams, and also sequence and deployment diagrams where noted. However, don’t overlook finite state, activity, communication, or other diagram types as needed for effective communication.*

### Initial System Analysis and Class Identification

*SV: Perform grammatical and similar analyses to identify the most import and obviously needed classes, and to organize them into an initial class structure. An initial class diagram is appropriate, containing few if any internal details.*

Your text goes here . . .

### Dynamic Modelling of Use-Cases

*SV: Insert sequence diagrams of (at least the most important) use-cases, as a means of identifying other needed classes.*

*Content*

*Include sequence diagrams of each important use-case here. This is a first step towards identifying preliminary objects. (If the sequence diagram would be too big to fit, then it can either be broken down into pieces or a communication diagram can be used in its place.)*

Your text goes here . . .

### Proposed System Architecture

*SV: Identify the Software Architecture to be applied to this project, such as Client-Server, Repository, MVC, etc., along with justification for the choice.*

Your text goes here . . .

### Initial Subsystem Decomposition

*SV: A slightly more detailed class diagram, showing the classes identified in sections 24a, 24b, and 0 above, partitioned into subsystems. For each subsystem provide a brief description of the subsystem, including its key responsibilities. There should still be few if any internal details.*

Your text goes here . . .

## Additional Design Considerations

*SV: The sections listed here do not need to be presented in the order given and may not all be relevant for any particular project. Those that are relevant can help identify additional classes that are needed as a result.*

### Hardware / Software Mapping

*SV: This is particularly important for distributed systems, such as those employing a client-server architecture. Use a deployment diagram to indicate which subsystems are mapped onto which piece(s) of hardware, and what communication subsystems need to be added to the system as a result.*

Your text goes here . . .

### Persistent Data Management

*SV: Document the classes and perhaps subsystems necessary to store persistent data when the system shuts down, and to restore that data when the system starts back up again.*

*Reiterate key data structures and information as necessary for the understanding of this design phase. Refer the reader back to the data dictionary in section I7c above to avoid undue repetition, while reviewing only the most relevant items here.*

Your text goes here . . .

### Access Control and Security

*SV: Identify the access control and security concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### Global Software Control

*SV: Identify the global software control concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### Boundary Conditions

*SV: Identify the boundary condition concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns. In particular consider startup, shutdown (normal or abnormal), and the creation and/or maintenance of any configuration files, databases, or similar supporting data files.*

Your text goes here . . .

### User Interface

*SV: Include a preliminary user interface design here, possibly as a rough sketch or other mockup, in order to identify additional classes needed to implement the interface.*

*The final user interface design will normally be developed by appropriate experts in that area. However, it is appropriate to include an initial design here, including possibly a low- or high- fidelity sketch/mockup, in order to identify key classes necessary to implement the user interface, such as forms and dialog windows. It may also go towards addressing usability and/or look-and-feel requirements, and/or identifying other overlooked components.*

Your text goes here . . .

### Application of Design Patterns

*SV: Any design patterns applied as a result of previous sections should have been addressed there and identified as such at the time. Use this section to document only the additional design patterns that were not previously covered elsewhere. (If any.)*

Your text goes here . . .

## Final System Design

*SV: Include here the final version of the overall system design, incorporating all the subsystems and classes added as a result of additional design considerations. Multiple diagrams may be needed, possibly starting with an overall package diagram showing all the different subsystems and the (important) classes contained within each one. Still not a lot of internal details.*

Your text goes here . . .

## Object Design

*This section documents the internal details of each class, to the extent that they can be designed at this time. Included should be the class interfaces (public method signatures and responsibilities) and constraints. It is probably best to break this section up into subsections corresponding to subsystems as documented above, and/or by (Java) packages if those are designed. It may also be appropriate to address additional design pattern considerations here, but not to the point of being redundant of previous documentation.*

*Certain methods, such as simple getters, setters, and constructors are not always documented, unless there is something special about them such as in the Singleton or Factory Method design patterns.*

### Packages

*SV: If the design involves assigning classes to packages (. e.g., Java packages), then the packages to be created should be documented here.*

Your text goes here . . .

### Subsystem I

Your text goes here . . .

### Subsystem II

Your text goes here . . .

### etc.

Your text goes here . . .

# Project Issues

## Open Issues

*SV: Issues that have been raised and do not yet have a conclusion.*

*Content*

*A statement of factors that are uncertain and might make significant difference to the product.*

*Motivation*

*To bring uncertainty out in the open and provide objective input to risk analysis.*

*Examples*

*Our investigation into whether the new version of the processor will be suitable for our application is not yet complete.*

*The government is planning to change the rules about who is responsible for gritting the motorways, but we do not know what those changes might be.*

*Considerations*

*Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations or systems on your context diagram? Are there any legislative changes that might affect your system? Are there any rumors about your hardware or software suppliers that might have an impact?*

Your text goes here . . .

## Off-the-Shelf Solutions

*SV: Discussion of products or components currently available that could either be incorporated into the new solution or simply used instead of developing (parts of) the new solution.  The distinction between sections 35 a, b, and c is subtle, and not very important.*

Your text goes here . . .

### Ready-Made Products

*SV: Products available for purchase that could be used either as part of a solution or instead of (a part of) a solution.*

*Content*

*List of existing products that should be investigated as potential solutions. Reference any surveys that have been done on these products.*

*Motivation*

*To consider whether a solution can be bought.*

*Considerations*

*Could you buy something that already exists or is about to become available? It may not be possible at this stage to make this determination with a lot of confidence, but any likely products should be listed here.*

*Also consider whether some products must not be used.*

Your text goes here . . .

### Reusable Components

*SV: Similar to 35a, but for components such as libraries or toolkits instead of fully blown products.*

*Content*

*Description of the candidate components, either bought from outside or built by your company, that could be used by this project. List libraries that could be a source of components.*

*Motivation*

*Reuse rather than reinvention.*

Your text goes here . . .

### Products That Can Be Copied

*SV: Products that could legally be copied would typically be past projects developed by the same development group, provided there were no restrictions that would prevent their reuse.*

*Content*

*List of other similar products or parts of products that you can legally copy or easily modify.*

*Motivation*

*Reuse rather than reinvention.*

*Examples*

*Another electricity company has built a customer service system. It’s hardware is different from ours, but we could buy its specification and cut our analysis effort by approximately 60 percent.*

*Considerations*

*While a ready-made solution may not exist, perhaps something, in its essence, is similar enough that you could copy, and possibly modify, it to better effect than starting from scratch. This approach is potentially dangerous because it relies on the base system being of good quality.*

*This question should always be answered. The act of answering it will force you to look at other existing solutions to similar problems.*

Your text goes here . . .

## New Problems

*SV: The proposed new system certainly has its benefits, but it could also raise new problems.  It is a good idea to identify any such potential problems early on, rather than being surprised by them later.*

### Effects on the Current Environment

*SV: Could the new system have any adverse effects on the working environment, e.g., the way people do their jobs?*

*Content*

*A description of how the new product will affect the current implementation environment. This section should also cover things that the new product should not do.*

*Motivation*

*The intention is to discover early any potential conflicts that might otherwise not be realized until implementation time.*

*Examples*

*Any change to the scheduling system will affect the work of the engineers in the divisions and the truck drivers.*

*Considerations*

*Is it possible that the new system might damage some existing system? Can people be displaced or otherwise affected by the new system?*

*These issues require a study of the current environment. A model highlighting the effects of the change is a good way to make this information widely understandable.*

Your text goes here . . .

### Effects on the Installed Systems

*SV: Could the new system have any adverse effects on other hardware or software systems?*

*Content*

*Specification of the interfaces between new and existing systems.*

*Motivation*

*Very rarely is a new development intended to stand completely alone. Usually, the new system must coexist with some older system. This question forces you to look carefully at the existing system, examining it for potential conflicts with the new development.*

Your text goes here . . .

### Potential User Problems

*SV: Could the new system have any adverse effects on the users of the software? Could users possibly have a negative response to the new system?*

*Content*

*Details of any adverse reaction that might be suffered by existing users.*

*Motivation*

*Sometimes existing users are using a product in such a way that they will suffer ill effects from the new system or feature. Identify any likely adverse user reactions and determine whether we care about those reactions and what precautions we will take.*

Your text goes here . . .

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

*SV: Are there any (physical) limitations in the expected environment that could inhibit the proposed product?  (e.g., weather, electrical interference, radiation, lack of reliable power, etc.)*

*Content*

*Statement of any potential problems with the new automated technology or new ways of structuring the organization.*

*Motivation*

*The intention is to make early discovery of any potential conflicts that might otherwise not be realized until implementation time.*

*Examples*

*The planned new server is not powerful enough to cope with our projected growth pattern.*

*The size and weight of the new product do not fit into the physical environment.*

*The power capabilities will not satisfy the new product’s projected consumption.*

*Considerations*

*This requires a study of the intended implementation environment.*

Your text goes here . . .

### Follow-Up Problems

*SV: Basically, any other possible problems that could occur.*

*Content*

*Identification of situations that we might not be able to cope with.*

*Motivation*

*To guard against situations where the product might fail.*

*Considerations*

*Will we create a demand for our product that we are not able to service? Will the new system cause us to run afoul of laws that do not currently apply? Will the existing hardware cope?*

*There are potentially hundreds of unwanted effects. It pays to answer this question very carefully.*

Your text goes here . . .

## Migration to the New Product

*SV: This section only applies when there is an existing system that is being replaced by a new system, particularly when data must be preserved and possibly translated / reformatted.  Otherwise just write "Not Applicable" under section 38 and remove sections 38a and 38b.*

### Requirements for Migration to the New Product

*SV: These are a list of requirements relevant to the migration procedures.  For example, a requirement that the two systems be run in parallel for a time until the client is satisfied with the new system and the users know how to use it.*

*Content*

*A list of the conversion activities. Timetable for implementation.*

*Motivation*

*To identify conversion tasks as input to the project planning process.*

*Considerations*

*Will you use a phased implementation to install the new system? If so, describe which requirements will be implemented by each of the major phases.*

*What kind of data conversion is necessary? Must special programs be written to transport data from an existing system to the new one? If so, describe the requirements for these programs here.*

*What kind of manual backup is needed while the new system is installed?*

*When are each of the major components to be put in place? When are the phases of the implementation to be released?*

*Is there a need to run the new product in parallel with the existing product?*

*Will we need additional or different staff?*

*Is any special effort needed to decommission the old product?*

*This section is the timetable for implementation of the new system.*

Your text goes here . . .

### Data That Has to Be Modified or Translated for the New System

*SV: This section specifically addresses****data****that must be preserved and/or translated / reformatted during the migration process.*

*Content*

*List of data translation tasks.*

*Motivation*

*To discover missing tasks that will affect the size and boundaries of the project.*

*Fit Criterion*

*Description of the current technology that holds the data.*

*Description of the new technology that will hold the data.*

*Description of the data translation tasks.*

*Foreseeable problems.*

*Considerations*

*Every time you make an addition to your dictionary (see section 5), ask this question: Where is this data currently held, and will the new system affect that implementation?*

Your text goes here . . .

## Risks

*SV: Consideration of the potential risks that could cause the project to fail / underperform.*

*All projects involve risk—namely, the risk that something will go wrong. Risk is not necessarily a bad thing, as no progress is made without taking some risk. However, there is a difference between unmanaged risk—say, shooting dice at a craps table—and managed risk, where the probabilities are well understood, and contingency plans are made. Risk is only a bad thing if the risks are ignored, and they become problems. Risk management entails assessing which risks are most likely to apply to the project, deciding a course of action if they become problems, and monitoring projects to give early warnings of risks becoming problems.*

*This section of your specification should contain a list of the most likely risks and the most serious risks for your project. For each risk, include the probability of that risk becoming a problem. Capers Jones’s Assessment and Control of Software Risks (Prentice-Hall, Englewood Cliffs, N.J., 1994) gives comprehensive lists of risks and their probabilities; you can use these lists as a starting point. For example, Jones cites the following risks as being the most serious:*

*• Inaccurate metrics*

*• Inadequate measurement*

*• Excessive schedule pressure*

*• Management malpractice*

*• Inaccurate cost estimating*

*• Silver bullet syndrome*

*• Creeping user requirements*

*• Low quality*

*• Low productivity*

*• Cancelled projects*

*Use your knowledge of the requirements as input to discover which risks are most relevant to your project.*

*It is also useful input to project management if you include the impact on the schedule, or the cost, if the risk does become a problem.*

Your text goes here . . .

## Costs

*SV: An estimate of what it will cost to complete this project.  Think not only in terms of dollars, but also time, resources, lost opportunities, etc.*

*For details on how to estimate requirements effort and costs, refer to Appendix C Function Point Counting: A Simplified Introduction*

*The other cost of requirements is the amount of money or effort that you have to spend building them into a product. Once the requirements specification is complete, you can use one of the estimating methods to assess the cost, expressing the result as a monetary amount or time to build.*

*There is no best method to use when estimating. Keep in mind, however, that your estimates should be based on some tangible, countable artifact. If you are using this template, then, as a result of doing the work of requirements specification, you are producing many measurable deliverables. For example:*

*●* *Number of input and output flows on the work context*

*●* *Number of business events*

*●* *Number of product use cases*

*●* *Number of functional requirements*

*●* *Number of nonfunctional requirements*

*●* *Number of requirements constraints*

*●* *Number of function points*

*The more detailed the work you do on your requirements, the more accurate your deliverables will be. Your cost estimate is the amount of resources you estimate each type of deliverable will take to produce within your environment. You can create some very early cost estimates based on the work context. At that stage, your knowledge of the work will be general, and you should reflect this vagueness by making the cost estimate a range rather than a single figure.*

*As you increase your knowledge of the requirements, we suggest you try using function point counting—not because it is an inherently superior method, but because it is so widely accepted. So much is known about function point counting that it is possible to make easy comparisons with other products and other installations’ productivity.*

*It is important that your client be told at this stage what the product is likely to cost. You usually express this amount as the total cost to complete the product, but you may also find it advantageous to point out the cost of the requirements effort, or the costs of individual requirements.*

*Whatever you do, do not leave the costs in the lap of hysterical optimism. Make sure that this section includes meaningful numbers based on tangible deliverables.*

Your text goes here . . .

## Waiting Room

*SV: This is a place to record ideas or wishes that will not be included in the current release of the product, but which might be worth reconsidering at a later date.*

*Requirements that will not be part of the next release. These requirements might be included in future releases of the product.*

*Content*

*Any type of requirement.*

*Motivation*

*To allow requirements to be gathered, even though they cannot be part of the current development. To ensure that good ideas are not lost.*

*Considerations*

*The requirements-gathering process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section holds these requirements in waiting. The intention is to avoid stifling the creativity of your users and clients, by using a repository to retain future requirements. You are also managing expectations by making it clear that you take these requirements seriously, although they will not be part of the agreed-upon product.*

*Many people use the waiting room as a way of planning future versions of the product. Each requirement in the waiting room is tagged with its intended version number. As a requirement progresses closer to implementation, then you can spend more time on it and add details such as the cost and benefit attached to that requirement.*

*You might also prioritize the contents of your waiting room. “Low-hanging fruit”—requirements that provide a high benefit at a low cost of implementation—are the highest-ranking candidates for the next release. You would also give a high waiting room rank to requirements for which there is a pent-up demand.*

Your text goes here . . .

## Ideas for Solutions

*SV: When developing requirements only, it is not the role of the business analyst to dictate the implementation of the solution.  However, they can pass along any ideas they have here as suggestions to the developers.  For CS 440 this report includes system and object design, so this section would make suggestions for implementation and testing that would come after design, such as the use of a particular language, IDE, library, or other tools.*

*When you gather requirements, you focus on finding out what the real requirements are and try to avoid coming up with solutions. However, when creative people start to think about a problem, they always generate ideas about potential solutions. This section of the template is a place to put those ideas so that you do not forget them and so that you can separate them from the real business requirements.*

*Content*

*Any idea for a solution that you think is worth keeping for future consideration. This can take the form of rough notes, sketches, pointers to other documents, pointers to people, pointers to existing products, and so on. The aim is to capture, with the least amount of effort, an idea that you can return to later.*

*Motivation*

*To make sure that good ideas are not lost. To help you separate requirements from solutions.*

*Considerations*

*While you are gathering requirements, you will inevitably have solution ideas; this section offers a way to capture them. Bear in mind that this section will not necessarily be included in every document that you publish.*

Your text goes here . . .

## Project Retrospective

*SV: At the conclusion of the (CS 440) project, reflect back on what worked well and what didn't, and how the process could be improved in the future.*

*Content*

*At the end of every project, you should reflect upon what methods were used that worked out well and should be repeated in the future, and also what methods did not work out well and should be avoided. Any recommendations, suggestions, or ideas for how to do things better in the future should also be documented*

*Motivation*

*To learn from experience, and to continually strive for process improvement.*

*Considerations*

*When things don't go well, it is important to distinguish whether the methods themselves were poor, or simply poorly implemented in this particular case, or whether they just weren't right for this particular project / group of engineers.*

Your text goes here . . .

# Glossary

*SV: The glossary is a completer and more inclusive dictionary of defined terms than that found in section I.7.a, the latter of which only covered the most important key terms needed to understand the report.*

*The glossary defines terms that may not be familiar to all readers. This is especially important if the document is expected to reach a wide and varied audience, such as school children. The glossary may be placed at either the beginning or the end of the document.*

***Flotsam:*** *Any part of a ship or its cargo found floating on the water, whether it was deliberately or accidentally lost by its original owners.*

***Jetsam:*** *Any part of a ship or its cargo that is deliberately cast off (jettisoned) by its original owners, generally in order to lighten the ship, whether it floats or sinks.*

Your text goes here . . .

# References / Bibliography

*This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the “Insert Citation” and “Bibliography” buttons in the “Citations & Bibliography” section under the “References” tab of MS Word. Creating new citations will not update this list unless you click on it and select “Update Field”. You may need to reset the style for this paragraph to “normal” after updating.*

(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

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.

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

*This section provides an index to the report. The sample below was generated using the “Mark Entry” and “Insert Index” items from the “Index” section on the “References” tab and can be automatically updated by right clicking on the table below and selecting “Update Field”. To remove marked entries from the document, toggle the display of hidden paragraph marks (the paragraph button on the “Home” tab), and remove the tags shown with XE in {curly braces.}*

Design 61, 63

Requirements 35, 51, 58

Test 64, 65