***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.

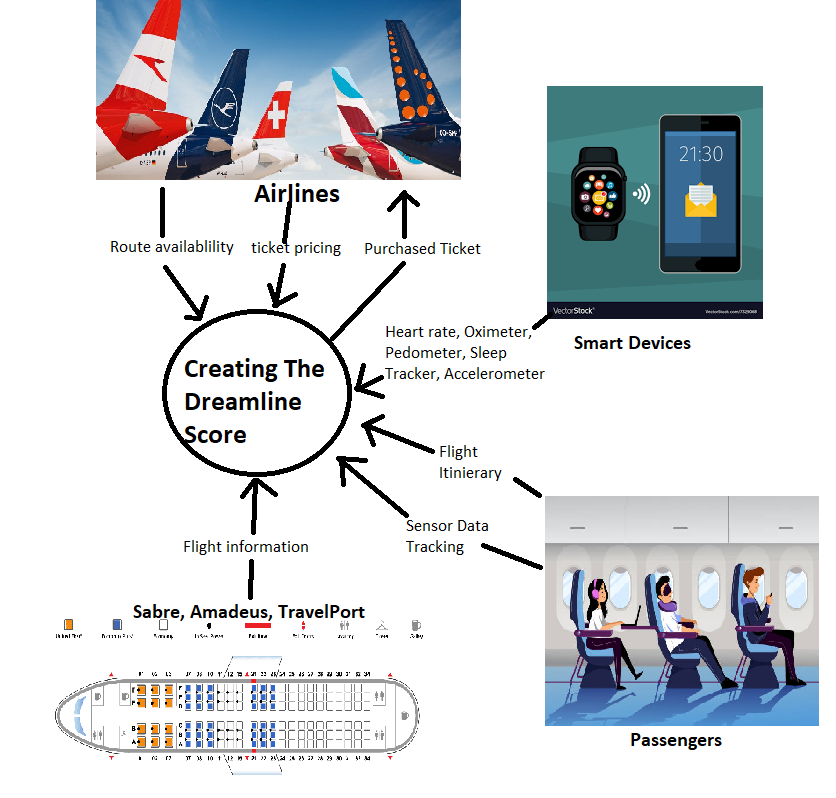
## 3 The Scope of the Work

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

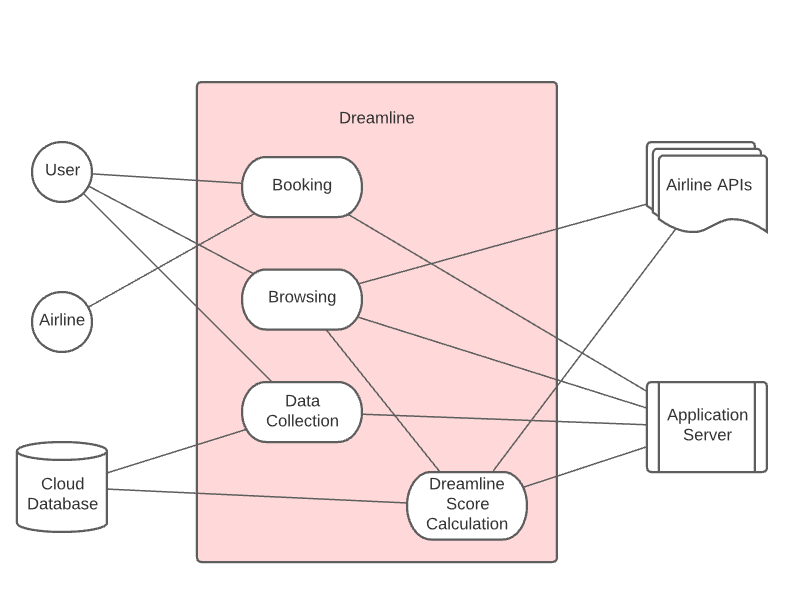
|  |  |  |
| --- | --- | --- |
| **Event Name** | **I/O** | **Summary** |
| The passenger describes the flight itinerary | Passengers (IN) | The passenger gives the origin and destination and the dates to build the Dreamline score on |
| Check Route Availability/ Get ticket pricing | Airlines(IN) | The itinerary will be checked for availability and the prices 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 indicates an uncomfortable price so the Dreamline score is adjusted down |
| Comfortable Flight | Passenger (IN)  Smart Devices(IN | The passenger review and the sensor data indicates 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 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.

## The Scope of the Product

### Scenario Diagram(s)



### 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 |

### 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.

## Stakeholders

### 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.

### 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.

### Hands-On Users of the Product

### 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*

### 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.

### 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.

### 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.

### 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.

## Mandated Constraints

### 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.

### 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.

### 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.

### Off-the-Shelf Software

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

### 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.

### 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.

### 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.

## Naming Conventions and Definitions

### 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.

### 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.

### 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)

## Relevant Facts and Assumptions

### Facts

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

### 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.

# Requirements

## Product Use Cases

### Use Case Diagrams

*SV: Use case diagrams list the use cases developed for a system, mark the boundary of what is internal or external to the system to be developed, and indicate which external entities ( actors ) are associated with each use case.*

*Use Case diagrams serve two purposes: As a form of graphical table of contents listing the individual use-cases, and also to define the boundary of what is included as part of the proposed system and what is not included.*

*A use case diagram identifies the boundaries between the users (actors) and the product. You arrive at the product boundary by inspecting each business use case and determining, in conjunction with the appropriate stakeholders, which part of the business use case should be automated (or satisfied by some sort of product) and what part should be done by the user. This task must take into account the abilities of the actors (section 3), the constraints (section 4), the goals of the project (section 1), and your knowledge of both the work and the technology that can make the best contribution to the work.*

*The use case diagram shows the actors outside the product boundary (the rectangle). The product use cases are the ellipses inside the boundary. The lines denote usage. Note that actors can be either automated or human.*

*Depending on the complexity of the product it may be necessary to use more than one diagram to list all of the use cases. When more than one diagram is required the use-cases can be divided up several ways: Normal operations versus exceptional cases, or daily tasks versus monthly tasks, or user tasks versus administration tasks, etc.*

*Examples*

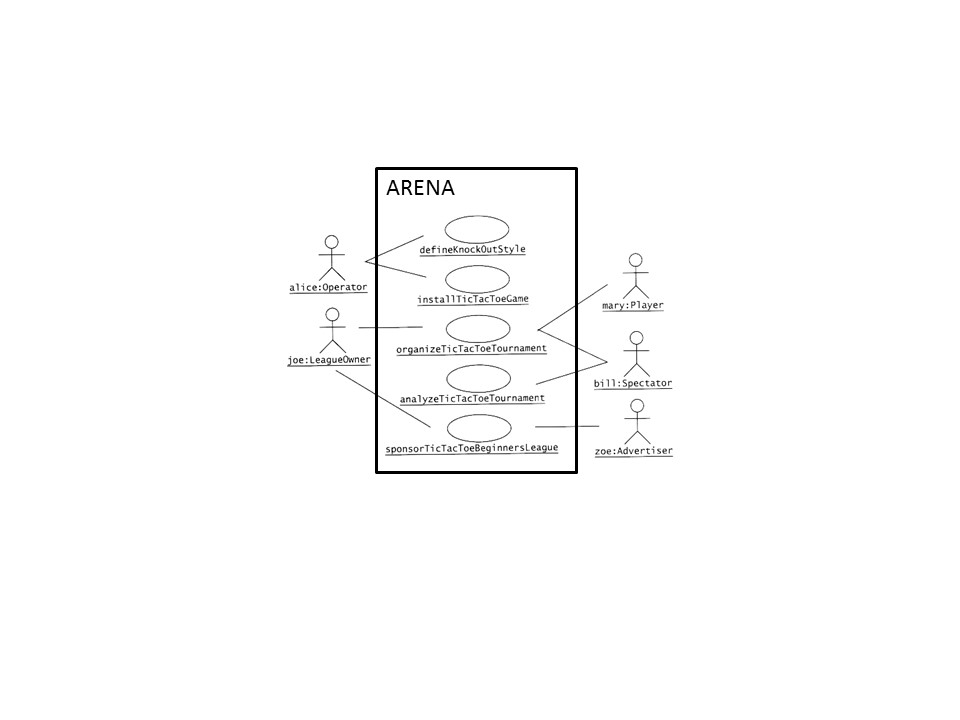


Figure 1 - Sample Use Case Diagram from Bruegge & DuToit ( modified )

**

Figure 2 - Sample Use Case Diagram from Robertson and Robertson

*Derive the product use cases by deciding where the product boundary should be for each business use case. These decisions are based on your knowledge of the work and the requirements constraints.*

### Product Use Case List

*SV: A list ( table ) of use cases is an alternative to the use case diagram, particularly when there are many use cases. There may be additional information in the table not found in the diagram, such as cross referencing to other sections or materials.*

*The use case diagram is a graphical way of summarizing the product use cases relevant to the product. If you have a large number of product use cases (we find 15–20 is a good limit), then it is better to make a list of the product use cases and model or describe each one individually.*

### Individual Product Use Cases

*Use cases are similar to scenarios, in that both tell the story of how the system interacts with the user(s) in response to some business event or while conducting some business task. The difference is that use-cases are much more formal, with certain pre-determined sections for each use-case, and that use-cases indicate clearly what action the system takes in response to what actions taken by the user.*

*SV: The following example was copied from “useCaseFormWithInstructions.docx”, available on the CS 440 web site. ( There is also a blank version available. )*

|  |
| --- |
| Use case ID: Name:  pre-conditions:  post-conditions:  Initiated by:  Triggering Event:  Additional Actors: |
| Sequence of Events:  Initiating event or action should be step 1, taken by initiating actor.   1. System response follows, indented right.   All external action steps are aligned with step 1. ( "stimulus" style )   1. All system responses are indented right, aligned with step 2. ( "response" style )   All steps should be expressed in the active voice, clearly indicating **who** performs each action   1. The sequence of events should show a back-and-forth stimulus-response relationship. |
| Alternatives: These would be normal and expected variations from the base case.  Exceptions: These would be unusual variations from the base case, often caused by problems. |

* *For all of the above, list as NA if not applicable.*
* *The following may be added if relevant, or omitted otherwise:*
  + *related use cases or scenarios*
  + *associated tests, systems, classes, etc.*
  + *revision history*
  + *references to other documents*
  + *author(s) / originator( s )*
  + *notes*
* *Alternatives and Exceptions may be listed either as separate use cases or as notes to a base case, depending on their significance and similarity.*
* *For regularly occurring periodic events, "time" can be listed as the initiating actor.*

*You may also want to view Figure 4.7 from "Object Oriented Software Engineering" by Bruegge and DuToit*

## Functional Requirements

*SV: Each requirement listed needs to have a unique identifier, a short name, a one- or two-sentence description, a rationale, a fit criteria, and reference to one or more acceptance tests to be used to confirm the completion of this particular requirement. The acceptance tests themselves are documented in section 0- See that section for further details. It is recommended to number the requirements according to their type, such as F-4 for the fourth functional requirement or U-2 for the second usability requirement. Functional requirements specifically deal with the functionality the system must have, and are generally derived directly from the steps the system takes during use cases.*

*Content*

*A specification for each functional requirement. A full explanation is included in this template’s introductory material.*

*Motivation*

*To specify the detailed functional requirements for the activity of the product.*

*Fit Criterion*

*Each functional requirement should have a fit criterion or a test case. In any event, the fit criterion is the benchmark to allow the tester to determine whether the implemented product has met the requirement.*

*Considerations*

*If you have produced an event/use case list (see sections 7b and 8a), then you can use it to help you trigger the functional requirements for each event/use case. If you have not produced an event/use case list, give each functional requirement a unique number and, to help with traceability, partition these requirements into event/use case–related groups later in the development process.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Data Requirements

*SV: Data requirements deal with requirements that are somehow related to data, such as the definition of what is included in a “student record” or the acceptable form of an e-mail address or allowable range of certain data items.*

*Content*

*A specification of the essential subject matter, business objects, entities, and classes that are germane to the product. It might take the form of a first-cut class model, an object model, or a domain model. Alternatively, these requirements might be described by defining the terms in the dictionary described in section 5.*

*Motivation*

*To clarify the system’s subject matter, thereby triggering recognition of requirements not yet considered.*

*Example*

*This is a model of the system’s business subject matter using the Unified Modeling Language (UML) class model notation.*

**

*You can use any type of data or object model to capture this knowledge. The issue is to capture the meaning of the business subject matter and the connections between the individual parts, and to show that you are consistent within your project. If you have an established company standard notation, use that, as it will help you to reuse knowledge between projects.*

*Considerations*

*Are there any data or object models for similar or overlapping systems that might be a useful starting point? Is there a domain model for the subject matter dealt with by this system?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Performance Requirements

### Speed and Latency Requirements

*SV: Requirements specifying how fast ( or slow ) the product must operate or how much lag is allowable between stimulus and either initial response or task completion. Other timing-related requirements could go in this section.*

*Content*

*Specifies the amount of time available to complete specified tasks. These requirements often refer to response times. They can also refer to the product’s ability to operate at a speed suitable for the intended environment.*

*Motivation*

*Some products—usually real-time products—must be able to perform some of their functionality within a given time slot. Failure to do so may mean catastrophic failure (e.g., a ground-sensing radar in an airplane fails to detect an upcoming mountain) or the product will not cope with the required volume of use (e.g., an automated ticket-selling machine).*

*Examples*

*Any interface between a user and the automated system shall have a maximum response time of 2 seconds.*

*The response shall be fast enough to avoid interrupting the user’s flow of thought.*

*The product shall poll the sensor every 10 seconds.*

*The product shall download the new status parameters within 5 minutes of a change.*

*Fit Criterion*

*Fit criteria are needed when the description of the requirement is not quantified. However, we find that most performance requirements are stated in quantified terms. The exception is the second requirement shown above, for which the suggested fit criterion is*

*The product shall respond in less than 1 second for 90 percent of the interrogations. No response shall take longer than 2.5 seconds.*

*Considerations*

*There is a wide variation in the importance of different types of speed requirements. If you are working on a missile guidance system, then speed is extremely important. By contrast, an inventory control report that is run once every six months has very little need for a lightning-fast response time.*

*Customize this section of the template to give examples of the speed requirements that are important within your environment.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Precision or Accuracy Requirements

*SV: Self-explanatory. How accurate or precise must the system be.*

*Content*

*Quantification of the desired accuracy of the results produced by the product.*

*Motivation*

*To set the client’s and users’ expectations for the precision of the product.*

*Examples*

*All monetary amounts shall be accurate to two decimal places.*

*Accuracy of road temperature readings shall be within ±2°C.*

*Considerations*

*If you have done any detailed work on definitions, then some precision requirements might be adequately defined by definitions in section 5.*

*You might consider which units the product is intended to use. Readers will recall the spacecraft that crashed on Mars when coordinates were sent as metric data rather than imperial data.*

*The product might also need to keep accurate time, be synchronized with a time server, or work in UTC.*

*Also, be aware that some currencies have no decimal places, such as the Japanese yen.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Capacity Requirements

*SV: Requirements regarding the largest “thing” the system must be able to handle, or perhaps how many things it can handle ( at once. ) Note: Requirements regarding how many things it can handle in a given time period would be a speed requirement, covered in section 12a above.*

*Content*

*This section specifies the volumes that the product must be able to deal with and the amount of data stored by the product.*

*Motivation*

*To ensure that the product is capable of processing the expected volumes.*

*Examples*

*The product shall cater for 300 simultaneous users within the period from 9:00 a.m. to 11:00 a.m. Maximum loading at other periods will be 150 simultaneous users.*

*During a launch period, the product shall cater for a maximum of 20 people to be in the inner chamber.*

*Fit Criterion*

*In this case, the requirement description is quantified, and thus can be tested.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Dependability Requirements

### Reliability Requirements

*SV: Reliability relates to how frequently the system fails, ( either by shutting down or by delivering erroneous results ), and the consequences of those failures. These requirements may also address the conditions under which it is allowed to fail ( or not. ), See also availability and robustness in the following sections.*

*Content*

*This section quantifies the necessary reliability of the product. The reliability is usually expressed as the allowable time between failures, or the total allowable failure rate.*

*Motivation*

*It is critical for some products not to fail too often. This section allows you to explore the possibility of failure and to specify realistic levels of service. It also gives you the opportunity to set the client’s and users’ expectations about the expected frequency and significance of potential failures.*

*Examples*

*The product shall not fail more than once per day.*

*No data shall be lost or damaged in the event of a failure. ( This is an example of a* ***fail-safe*** *requirement, which states that the product is allowed to fail, but it must do so safely. )*

*Considerations*

*Consider carefully whether the real requirement for your product is that it is available for use or that it does not fail at any time.*

*Consider also the cost of reliability and availability, and whether it is justified for your product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Availability Requirements

*SV: Availability addresses the amount of time the system is running and available for use. It is affected by how often the system goes down ( reliability ), but also by the time required to bring the system back up again, the availability lost due to regularly scheduled maintenance down times, and the ability of the system to offer at least partial functionality in the face of failures or resource shortages. See also reliability and robustness.*

*Content*

*This section quantifies the necessary availability of the product. The availability is usually expressed as the fraction of total time that the system is up and available for use.*

*Availability is a function of the mean time between failures, the mean time required to bring the system back up after a failure, and the mean time the system is expected to be down for routine maintenance.*

*Motivation*

*There is a subtle distinction between how often a system goes down ( reliability ) and how much total time it spends being down ( availability ). This section allows you to specify realistic expectations about the amount of time that the product will be available for use.*

*Examples*

*The product shall be available for use 24 hours per day, 365 days per year.*

*The product shall be available for use between the hours of 8:00 a.m. and 5:30 p.m.*

*The escalator shall run from 6 a.m. until 10 p.m. or the last flight arrives.*

*The product shall achieve 99 percent uptime.*

*Considerations*

*Consider carefully whether the real requirement for your product is that it is available for use or that it does not fail at any time.*

*Consider also the cost of reliability and availability, and whether it is justified for your product.*

*The sections on reliability and availability can sometimes be combined.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Robustness or Fault-Tolerance Requirements

*SV: This section deals with the system’s ability to provide at least partial functionality in the face of failures or resource shortages, such as operating in offline mode when network connectivity is unavailable. See also reliability and availability.*

*Content*

*Robustness specifies the ability of the product to continue to function under abnormal circumstances.*

*Motivation*

*To ensure that the product is able to provide some or all of its services after or during some abnormal happening in its environment.*

*Examples*

*The product shall continue to operate in local mode whenever it loses its link to the central server.*

*The product shall provide 10 minutes of emergency operation should it become disconnected from the electricity source.*

*Considerations*

*Abnormal happenings can almost be considered normal. Today’s products are so large and complex that there is a good chance that at any given time, one component will not be functioning correctly. Robustness requirements are intended to prevent total failure of the product.*

*You could also consider disaster recovery in this section. This plan describes the ability of the product to reestablish acceptable performance after faults or abnormal happenings.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Safety-Critical Requirements

*SV: These requirements address potential harm to health, safety, or property, and may refer to relevant standards such as OSHA compliance.*

*Content*

*Quantification of the perceived risk of damage to people, property, and environment. Different countries have different standards, so the fit criteria must specify precisely which standards the product must meet.*

*Motivation*

*To understand and highlight the damage that could potentially occur when using the product within the expected operational environment.*

*Examples*

*The product shall not emit noxious gases that damage people’s health.*

*The heat exchanger shall be shielded from human contact.*

*Fit Criterion*

*The product shall be certified to comply with the Health Department’s standard E110-98. It is to be certified by qualified testing engineers.*

*No member of a test panel of [specified size] shall be able to touch the heat exchanger. The heat exchanger must also comply with safety standard [specify which one].*

*Considerations*

*The example requirements given here apply to some, but not all, products. It is not possible to give examples of every variation of safety-critical requirement. To make the template work in your environment, you should customize it by adding examples that are specific to your products.*

*Also, be aware that different countries have different safety standards and laws relating to safety. If you plan to sell your product internationally, you must be aware of these laws. A colleague has suggested that for electrical products, if you follow the German standards, the largest number of countries will be supported.*

*If you are building safety-critical systems, then the relevant safety-critical standards are already well specified. You will likely have safety experts on your staff. These experts are the best source of the relevant safety-critical requirements for your type of product. They will almost certainly have copious information that you can use.*

*Consult your legal department. Members of this department will be aware of the kinds of lawsuits that have resulted from product safety failure. This is probably the best starting place for generating relevant safety requirements.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Maintainability and Supportability Requirements

### Maintenance Requirements

*SV: This section deals with the ease with which the system can be maintained, and possibly who will perform system maintenance and under what conditions. The ease of evolving the system into future versions may also be addressed here, or in a separate section ( not included in this template ) if that is a major concern.*

*Content*

*A quantification of the time necessary to make specified changes to the product.*

*Motivation*

*To make everyone aware of the maintenance needs of the product.*

*Examples*

*New MIS reports must be available within one working week of the date when the requirements are agreed upon.*

*A new weather station must be able to be added to the system overnight.*

*Considerations*

*There may be special requirements for maintainability, such as that the product must be able to be maintained by its end users or by developers who are not the original developers. These requirements have an effect on the way that the product is developed. In addition, there may be requirements for documentation or training.*

*You might also consider writing testability requirements in this section.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Supportability Requirements

*SV: What ongoing support is to be provided, e.g. through a help desk. See also training requirements in section 16g below.*

*Content*

*This specifies the level of support that the product requires. Support is often provided via a help desk. If people will provide support for the product, that service is considered part of the product: Are there any requirements for that support? You might also build support into the product itself, in which case this section is the place to write those requirements.*

*Motivation*

*To ensure that the support aspect of the product is adequately specified.*

*Considerations*

*Consider the anticipated level of support, and what forms it might take. For example, a constraint might state that there is to be no printed manual. Alternatively, the product might need to be entirely self-supporting.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Adaptability Requirements

*SV: Description of other platforms or environments to which the product must be ported.*

*Content*

*Description of other platforms or environments to which the product must be ported.*

*Motivation*

*To quantify the client’s and users’ expectations about the platforms on which the product will be able to run.*

*Examples*

*The product is expected to run under Windows XP and Linux.*

*The product might eventually be sold in the Japanese market.*

*The product is designed to run in offices, but we intend to have a version running in restaurant kitchens.*

*Fit Criterion*

*Specification of system software on which the product must operate.*

*Specification of future environments in which the product is expected to operate.*

*Time allowed to make the transition.*

*Considerations*

*Question your marketing department to discover unstated assumptions that have been made about the portability of the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Scalability or Extensibility Requirements

*SV: The ease of expanding the system to a larger capacity as the business grows.*

*Content*

*This specifies the expected increases in size that the product must be able to handle. As a business grows (or is expected to grow), our software products must increase their capacities to cope with the new volumes.*

*Motivation*

*To ensure that the designers allow for future capacities.*

*Examples*

*The product shall be capable of processing the existing 100,000 customers. This number is expected to grow to 500,000 customers within three years.*

*The product shall be able to process 50,000 transactions per hour within two years of its launch.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Longevity Requirements

*SV: This specifies the expected lifetime of the product.*

*Content*

*This specifies the expected lifetime of the product.*

*Motivation*

*To ensure that the product is built based on an understanding of expected return on investment.*

*Examples*

*The product shall be expected to operate within the maximum maintenance budget for a minimum of five years.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Security Requirements

*SV: Security requirements address who is allowed what type of access to the system, and what areas require special protection or diligence. In practice security requirements must often be written by security experts, and may refer to standards.*

### Access Requirements

*SV: These requirements address who has access to what ( data or functionality ) and under what conditions or restrictions.*

*Content*

*Specification of who has authorized access to the product (both functionality and data), under what circumstances that access is granted, and to which parts of the product access is allowed.*

*Motivation*

*To understand the expectations for confidentiality aspects of the system.*

*Examples*

*Only direct managers can see the personnel records of their staff.*

*Only holders of current security clearance can enter the building.*

*Fit Criterion*

*System function name or system data name.*

*User roles and/or names of people who have clearance.*

*Considerations*

*Is there any data that management considers to be sensitive? Is there any data that low-level users do not want management to have access to? Are there any processes that might cause damage or might be used for personal gain? Are there any people who should not have access to the system?*

*Avoid stating how you will design a solution to the security requirements. For instance, don’t “design a password system.” Your aim here is to identify the security requirement; the design will then come from this description.*

*Consider asking for help. Computer security is a highly specialized field, and one where improperly qualified people have no business. If your product has need of more than average security, we advise you to make use of a security consultant. Such consultants are not cheap, but the results of inadequate security can be even more expensive.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Integrity Requirements

*SV: These requirements address the protection of data(bases) from intentional or accidental corruption, loss, or theft.*

*Content*

*Specification of the required integrity of databases and other files, and of the product itself.*

*Motivation*

*To understand the expectations for the integrity of the product’s data. To specify what the product will do to ensure its integrity in the case of an unwanted happening such as attack from the outside or unintentional misuse by an authorized user.*

*Examples*

*The product shall prevent incorrect data from being introduced.*

*The product shall protect itself from intentional abuse.*

*Considerations*

*Organizations are relying more and more on their stored data. If this data should be come corrupt or incorrect—or disappear—then it could be a fatal blow to the organization. For example, almost half of small businesses go bankrupt after a fire destroys their computer systems. Integrity requirements are aimed at preventing complete loss, as well as corruption, of data and processes.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Privacy Requirements

*SV: These requirements address data that must remain confidential, such as medical records or other personally identifiable data. Laws often apply. (See also section 20.)*

*Content*

*Specification of what the product has to do to ensure the privacy of individuals about whom it stores information. The product must also ensure that all laws related to privacy of an individual’s data are observed.*

*Motivation*

*To ensure that the product complies with the law, and to protect the individual privacy of your customers. Few people today look kindly on organizations that do not observe their privacy.*

*Examples*

*The product shall make its users aware of its information practices before collecting data from them.*

*The product shall notify customers of changes to its information policy.*

*The product shall reveal private information only in compliance with the organization’s information policy.*

*The product shall protect private information in accordance with the relevant privacy laws and the organization’s information policy.*

*Considerations*

*Privacy issues may well have legal implications, and you are advised to consult with your organization’s legal department about the requirements to be written in this section.*

*Consider what notices you must issue to your customers before collecting their personal information. A notice might go so far as to warn customers that you intend to put a cookie in their computer. Also, do you have to do anything to keep customers aware that you hold their personal information?*

*Customers must always be in a position to give or withhold consent when their private data is collected or stored. Similarly, customers should be able to view any private data and, where appropriate, ask for correction of the data.*

*Also consider the integrity and security of private data—for example, when you are storing credit card information.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Audit Requirements

*SV: This section applies when a system must provide support for transaction auditing, such as some financial or medical systems.*

*Content*

*Specification of what the product has to do (usually retain records) to permit the required audit checks.*

*Motivation*

*To build a system that complies with the appropriate audit rules.*

*Considerations*

*This section may have legal implications. You are advised to seek the approval of your organization’s auditors regarding what you write here.*

*You should also consider whether the product should retain information on who has used it. The intention is to provide security such that a user may not later deny having used the product or participated in some form of transaction using the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Immunity Requirements

*SV: This section addresses the system’s ability to resist viruses, worms, Trojan Horses, etc.*

*Content*

*The requirements for what the product has to do to protect itself from infection by unauthorized or undesirable software programs, such as viruses, worms, and Trojan horses, among others.*

*Motivation*

*To build a product that is as secure as possible from malicious interference.*

*Considerations*

*Each day brings more malevolence from the unknown, outside world. People buying software, or any other kind of product, expect that it can protect itself from outside interference.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Usability and Humanity Requirements

*SV: This section is concerned with requirements that make the product usable and ergonomically acceptable to its hands-on users.*

### Ease of Use Requirements

*SV: This section addresses the ease with which the intended audience can use the system properly, and conversely the difficulty with which they can use it improperly.*

*Content*

*This section describes your client’s aspirations for how easy it is for the intended users of the product to operate it. The product’s usability is derived from the abilities of the expected users of the product and the complexity of its functionality.*

*The usability requirements should cover properties such as these:*

*●* *Efficiency of use: How quickly or accurately the user can use the product.*

*●* *Ease of remembering: How much the casual user is expected to remember about using the product.*

*●* *Error rates: For some products it is crucial that the user commits very few, or no, errors.*

*●* *Overall satisfaction in using the product: This is especially important for commercial, interactive products that face a lot of competition. Web sites are a good example.*

*●* *Feedback: How much feedback the user needs to feel confident that the product is actually accurately doing what the user expects. The necessary degree of feedback will be higher for some products (e.g., safety-critical products) than for others.*

*Motivation*

*To guide the product’s designers toward building a product that meets the expectations of its eventual users.*

*Examples*

*The product shall be easy for 11-year-old children to use.*

*The product shall help the user to avoid making mistakes.*

*The product shall make the users want to use it.*

*The product shall be used by people with no training, and possibly no understanding of English.*

*Fit Criterion*

*These examples may seem simplistic, but they do express the intention of the client. To completely specify what is meant by the requirement, you must add a measurement against which it can be tested—that is, a fit criterion. Here are the fit criteria for the preceding examples:*

*Eighty percent of a test panel of 11-year-old children shall be able to successfully complete [list of tasks] within [specified time].*

*One month’s use of the product shall result in a total error rate of less than 1 percent.*

*An anonymous survey shall show that 75 percent of the intended users are regularly using the product after a three-week familiarization period.*

*Considerations*

*Refer to section 3, Users of the Product, to ensure that you have considered the usability requirements from the perspective of all the different types of users.*

*It may be necessary to have special consulting sessions with your users and your client to determine whether any special usability considerations must be built into the product.*

*You could also consider consulting a usability laboratory experienced in testing the usability of products that have a project situation (sections 1–7 of this template) similar to yours.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Personalization and Internationalization Requirements

*SV: This section addresses the ease with which the system can be configured for personal preferences, and for things such as language, currency, units, symbols, etc.*

*Content*

*This section describes the way in which the product can be altered or configured to take into account the user’s personal preferences or choice of language.*

*The personalization requirements should cover issues such as the following:*

*●* *Languages, spelling preferences, and language idioms*

*●* *Currencies, including the symbols and decimal conventions*

*●* *Personal configuration options*

*Motivation*

*To ensure that the product’s users do not have to struggle with, or meekly accept, the builder’s cultural conventions.*

*Examples*

*The product shall retain the buyer’s buying preferences.*

*The product shall allow the user to select a chosen language.*

*Considerations*

*Consider the country and culture of the potential customers and users of your product. Any out-of-country users will welcome the opportunity to convert to their home spelling and expressions.*

*By allowing users to customize the way in which they use the product, you give them the opportunity to participate more closely with your organization as well as enjoy their own personal user experience.*

*You might also consider the configurability of the product. Configurability allows different users to have different functional variations of the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Learning Requirements

*SV: Requirements related to how easy it is for the intended audience to learn to use the product.*

*Content*

*Requirements specifying how easy it should be to learn to use the product. This learning curve ranges from zero time for products intended for placement in the public domain (e.g., a parking meter or a web site) to a considerable amount of time for complex, highly technical products. (We know of one product where it was necessary for graduate engineers to spend 18 months in a training program before being qualified to use the product.)*

*Motivation*

*To quantify the amount of time that your client feels is allowable before a user can successfully use the product. This requirement guides designers to understand how users will learn the product. For example, designers may build elaborate interactive help facilities into the product, or the product may be packaged with a tutorial. Alternatively, the product may have to be constructed so that all of its functionality is apparent upon first encountering it.*

*Examples*

*The product shall be easy for an engineer to learn.*

*A clerk shall be able to be productive within a short time.*

*The product shall be able to be used by members of the public who will receive no training before using it.*

*The product shall be used by engineers who will attend five weeks of training before using the product.*

*Fit Criterion*

*An engineer shall produce a [specified result] within [specified time] of beginning to use the product, without needing to use the manual.*

*After receiving [number of hours] training a clerk shall be able to produce [quantity of specified outputs] per [unit of time].*

*[Agreed percentage] of a test panel shall successfully complete [specified task] within [specified time limit].*

*The engineers shall achieve [agreed percentage] pass rate from the final examination of the training.*

*Considerations*

*Refer to section 3, Users of the Product, to ensure that you have considered the ease of learning requirements from the perspective of all the different types of users.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Understandability and Politeness Requirements

*SV: These requirements relate to how intuitively the intended audience understands what the program does, what its messages mean, and how to use it. Definitely related to ease of use, ( section 16a ), but more specifically addressing comprehension of the program output, instructions, and other messages.*

*This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end users.*

*Content*

*This specifies the requirement for the product to be understood by its users. While “usability” refers to ease of use, efficiency, and similar characteristics, “understandability” determines whether the users instinctively know what the product will do for them and how it fits into their view of the world. You can think of understandability as the product being polite to its users and not expecting them to know or learn things that have nothing to do with their business problem.*

*Motivation*

*To avoid forcing users to learn terms and concepts that are part of the product’s internal construction and are not relevant to the users’ world. To make the product more comprehensible and thus more likely to be adopted by its intended users.*

*Examples*

*The product shall use symbols and words that are naturally understandable by the user community.*

*The product shall hide the details of its construction from the user.*

*Considerations*

*Refer to section 3, Users of the Product, and consider the world from the point of view of each of the different types of users.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Accessibility Requirements

*SV: Requirements related to use of the product by individuals with disabilities.*

*Content*

*The requirements for how easy it should be for people with common disabilities to access the product. These disabilities might be related to physical disability or visual, hearing, cognitive, or other abilities.*

*Motivation*

*In many countries it is required that some products be made available to the disabled. In any event, it is self-defeating to exclude this sizable community of potential customers.*

*Examples*

*The product shall be usable by partially sighted users.*

*The product shall conform to the Americans with Disabilities Act.*

*Considerations*

*Some users have disabilities other than the commonly described ones. In addition, some partial disabilities are fairly common. A simple, and not very consequential, example is that approximately 20 percent of males are red-green colorblind.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### User Documentation Requirements

*SV: List of the user documentation to be supplied as part of the product.*

*Content*

*List of the user documentation to be supplied as part of the product.*

*Motivation*

*To set expectations for the documentation and to identify who will be responsible for creating it.*

*Examples*

*Technical specifications to accompany the product.*

*User manuals.*

*Service manuals (if not covered by the technical specification).*

*Emergency procedure manuals (e.g., the card found in airplanes).*

*Installation manuals.*

*Considerations*

*Which documents do you need to deliver, and to whom? Bear in mind that the answer to this questions depends on your organizational procedures and roles.*

*For each document, consider these issues:*

*●* *The purpose of the document*

*●* *The people who will use the document*

*●* *Maintenance of the document*

*What level of documentation is expected? Will the users be involved in the production of the documentation? Who will be responsible for keeping the documentation up-to-date? What form will the documentation take?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Training Requirements

*SV: A description of the training needed by users of the product.*

*Content*

*A description of the training needed by users of the product.*

*Motivation*

*To set expectations for the training. To identify who is responsible for creating and providing that training.*

*Considerations*

*What training will be necessary? Who will design the training? Who will provide the training?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Look and Feel Requirements

### Appearance Requirements

*SV: These requirements address things such as the colors, fonts, and logos used, often to reflect corporate branding or similarity to related products. See also style in the next section.*

*Content*

*The section contains requirements relating to the spirit of the product. Your client may have made particular demands for the product, such as corporate branding, colors to be used, and so on. This section captures the requirements for the appearance. Do not attempt to design it until the appearance requirements are known.*

*Motivation*

*To ensure that the appearance of the product conforms to the organization’s expectations.*

*Examples*

*The product shall be attractive to a teenage audience.*

*The product shall comply with corporate branding standards.*

*Fit Criterion*

*A sampling of representative teenagers shall, without prompting or enticement, start using the product within four minutes of their first encounter with it.*

*The office of branding shall certify the product complies with the current standards.*

*Considerations*

*Even if you are using prototypes, it is important to understand the requirements for the appearance. The prototype is used to help elicit requirements; it should not be thought of as a substitute for the requirements.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Style Requirements

*SV: Style requirements address the impression the product makes upon users, such as professionalism for a tax accounting package, friendliness for a children’s game, or how “cool” it is for a teenage audience. Product packaging may also be addressed here, and/or appearance in the previous section.*

*Content*

*Requirements that specify the mood, style, or feeling of the product, which influences the way a potential customer will see the product. Also, the stakeholders’ intentions for the amount of interaction the user is to have with the product.*

*In this section, you would also describe the appearance of the package if this is to be a manufactured product. The package may have some requirements as to its size, style, and consistency with other packages put out by your organization. Keep in mind the European laws on packaging, which require that the package not be significantly larger than the product it encloses.*

*The style requirements that you record here will guide the designers to create a product as envisioned by your client.*

*Motivation*

*Given the state of today’s market and people’s expectations, we cannot afford to build products that have the wrong style. Once the functional requirements are satisfied, it is often the appearance and style of products that determine whether they are successful. Your task in this section is to determine precisely how the product shall appear to its intended consumer.*

*Example*

*The product shall appear authoritative.*

*Fit Criterion*

*After their first encounter with the product, 70 percent of representative potential customers shall agree they feel they can trust the product.*

*Considerations*

*The look and feel requirements specify your client’s vision of the product’s appearance. The requirements may at first seem to be rather vague (e.g., “conservative and professional appearance”), but these will be quantified by their fit criteria. The fit criteria give you the opportunity to extract from your client precisely what is meant, and give the designer precise instructions on what he is to accomplish.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Operational and Environmental Requirements

### Expected Physical Environment

*SV: These requirements relate to the physical environment in which the product will operate.*

*Content*

*This section specifies the physical environment in which the product will operate.*

*Motivation*

*To highlight conditions that might need special requirements, preparations, or training. These requirements ensure that the product is fit to be used in its intended environment.*

*Examples*

*The product shall be used by a worker, standing up, outside in cold, rainy conditions.*

*The product shall be used in noisy conditions with a lot of dust.*

*The product shall be able to fit in a pocket or purse.*

*The product shall be usable in dim light.*

*The product shall not be louder than the existing noise level in the environment.*

*Considerations*

*The work environment: Is the product to operate in some unusual environment? Does this lead to special requirements? Also see section 11, Usability and Humanity Requirements.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Requirements for Interfacing with Adjacent Systems

*SV: This section describes the requirements to interface with partner applications and/or devices that the product needs to successfully operate.*

*Content*

*This section describes the requirements to interface with partner applications and/or devices that the product needs to successfully operate.*

*Motivation*

*Requirements for the interfaces to other applications often remain undiscovered until implementation time. Avoid a high degree of rework by discovering these requirements early.*

*Examples*

*The products shall work on the last four releases of the five most popular browsers.*

*The new version of the spreadsheet must be able to access data from the previous two versions.*

*Our product must interface with the applications that run on the remote weather stations.*

*Fit Criterion*

*For each inter-application interface, specify the following elements:*

*●* *The data content*

*●* *The physical material content*

*●* *The medium that carries the interface*

*●* *The frequency*

*●* *The volume*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Productization Requirements

*SV: Requirements related to the distribution and/or installation of the product.*

*Content*

*Any requirements that are necessary to make the product into a distributable or salable item. It is also appropriate to describe here the operations needed to install a software product successfully.*

*Motivation*

*To ensure that if work must be done to get the product out the door, then that work becomes part of the requirements. Also, to quantify the client’s and users’ expectations about the amount of time, money, and resources they will need to allocate to install the product.*

*Examples*

*The product shall be distributed as a ZIP file.*

*The product shall be able to be installed by an untrained user without recourse to separately printed instructions.*

*The product shall be of a size such that it can fit on one CD.*

*Considerations*

*Some products have special needs to turn them into a salable or usable product. You might consider that the product has to be protected such that only paid-up customers can access it.*

*Ask questions of your marketing department to discover unstated assumptions that have been made about the specified environment and the customers’ expectations of how long installation will take and how much it will cost.*

*Most commercial products have some needs in this area.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Release Requirements

*SV: Specification of the intended release cycle for the product and the form that the release shall take.*

*Content*

*Specification of the intended release cycle for the product and the form that the release shall take.*

*Motivation*

*To make everyone aware of how often you intend to produce new releases of the product.*

*Examples*

*The maintenance releases will be offered to end users once a year.*

*Each release shall not cause previous features to fail.*

*Fit Criterion*

*Description of the type of maintenance plus the amount of effort budgeted for it.*

*Considerations*

*Do you have any existing contractual commitments or maintenance agreements that might be affected by the new product?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Cultural and Political Requirements

### Cultural Requirements

*SV: This section contains requirements that are specific to the sociological factors that affect the acceptability of the product. If you are developing a product for foreign markets, then these requirements are particularly relevant. Bear in mind that “cultural groups” may also apply to population subgroups such as teenagers, the elderly, or ironworkers.*

*Content*

*This section contains requirements that are specific to the sociological factors that affect the acceptability of the product. If you are developing a product for foreign markets, then these requirements are particularly relevant.*

*Motivation*

*To bring out in the open requirements that are difficult to discover because they are outside the cultural experience of the developers.*

*Examples*

*The product shall not be offensive to religious or ethnic groups.*

*The product shall be able to distinguish between French, Italian, and British road-numbering systems.*

*The product shall keep a record of public holidays for all countries in the European Union and for all states in the United States.*

*Considerations*

*Question whether the product is intended for a culture other than the one with which you are familiar. Ask whether people in other countries or in other types of organizations will use the product. Do these people have different habits, holidays, superstitions, or cultural norms that do not apply to your own culture? Are there colors, icons, or words that have different meanings in another cultural environment?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Political Requirements

*SV: Requirements included strictly to make “the boss” happy, either internally to the development company, or internally to the client company, or possibly an external third party.*

*Content*

*This section contains requirements that are specific to the political factors that affect the acceptability of the product.*

*Motivation*

*To understand requirements that sometimes appear irrational.*

*Examples*

*The product shall be installed using only American-made components.*

*The product shall make all functionality available to the CEO.*

*Considerations*

*Did you intend to develop the product on a Macintosh, when the office manager has laid down an edict that only Windows machines are permitted?*

*Is a director also on the board of a company that manufactures products similar to the one that you intend to build?*

*Whether you agree with these political requirements has little bearing on the outcome. The reality is that the system has to comply with political requirements even if you can find a better, more efficient, or more economical solution. A few probing questions here may save some heartache later.*

*The political requirements might be purely concerned with the politics inside your organization. However, in other situations you may need to consider the politics inside your customers’ organizations or the national politics of the country.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Legal Requirements

### Compliance Requirements

*SV: A statement specifying the legal requirements for this system, often referring to relevant laws and/or requiring approval by the legal department.*

*Content*

*A statement specifying the legal requirements for this system.*

*Motivation*

*To comply with the law so as to avoid later delays, lawsuits, and legal fees.*

*Examples*

*Personal information shall be implemented so as to comply with the Data Protection Act.*

*Fit Criterion*

*Lawyers’ opinion that the product does not break any laws.*

*Considerations*

*Consider consulting lawyers to help identify the legal requirements.*

*Are there any copyrights or other intellectual property that must be protected? Conversely, do any competitors have copyrights on which you might be in danger of infringing?*

*Is it a requirement that developers have not seen competitors’ code or even have worked for competitors?*

*The Sarbanes-Oxley (SOX) Act, the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act may have implications for you. Check with your company lawyer.*

*Might any pending legislation affect the development of this system?*

*Are there any aspects of criminal law you should consider?*

*Have you considered the tax laws that affect your product?*

*Are there any labor laws (e.g., working hours) relevant to your product?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Standards Requirements

*SV: These requirements specify documented standards to which the product must conform, as opposed to legal regulations.*

*Content*

*A statement specifying applicable standards and referencing detailed standards descriptions. This does not refer to the law of the land—think of it as an internal law imposed by your company.*

*Motivation*

*To comply with standards so as to avoid later delays.*

*Example*

*The product shall comply with MilSpec standards.*

*The product shall comply with insurance industry standards.*

*The product shall be developed according to SSADM standard development steps.*

*Fit Criterion*

*The appropriate standard-keeper certifies that the standard has been adhered to.*

*Considerations*

*It is not always apparent that there are applicable standards because their existence is often taken for granted. Consider the following:*

*●* *Do any industry bodies have applicable standards?*

*●* *Does the industry have a code of practice, watchdog, or ombudsman?*

*●* *Are there any special development steps for this type of product?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Requirements Acceptance Tests

*SV: Every requirement must have one or more acceptance tests associated with it, to confirm that the requirement has been met. At this point these tests are not yet completely specified – A one- or two-sentence description of each test will suffice. Note that some tests may verify more than one requirement, and that some requirements may require multiple tests for their confirmation.*

### Requirements – Test Correspondence Summary

*SV: The following sample table is available from the CS 440 web site as “Sample Requirement Test Correspondence Table.xlsx” It is recommended that you work with the table in Excel, and then drag it into the document when it is completed. Depending on the number of requirements and/or tests included, it may be necessary to use multiple tables, and/or use landscape mode. Every row and every column of the table should include at least one X. Below the table list the ID #, name, and short description of each individual acceptance test.*

**

*Table 1 - Requirements - Acceptance Tests Correspondence*

### Acceptance Test Descriptions

*SV: Provide a brief description of each acceptance test. Detailed test specifications will appear in a separate document, which may be referenced here when available.*

**ID # - Name**

**Description:** Your description here . . .

# Design

## 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 give consideration to 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. Its 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 more complete and 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.*

# 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