

GPS Trail



**Chris Lenell, Deonvell Shed,
Luke Wittenkeller, and Fade Abdeljaber
For use in CS 440
at the
University of Illinois Chicago**

September 2018

Table of Contents

List of Figures	7
List of Tables	8
I Project Description	9
1 Project Overview	9
2 The Purpose of the Project	9
2a The User Business or Background of the Project Effort	9
2b Goals of the Project	9
2c Measurement	9
3 The Scope of the Work	9
3a The Current Situation	10
3b The Context of the Work	10
3c Work Partitioning	11
3d Competing Products	12
4 The Scope of the Product	12
4a Scenario Diagram(s)	12
4b Product Scenario List	12
4c Individual Product Scenarios	12
5 Stakeholders	12
5a The Client	12
5b The Customer	13
5c Hands-On Users of the Product	13
5d Maintenance Users and Service Technicians	13
5e Other Stakeholders	13
5f User Participation	13
5g Priorities Assigned to Users	13
6 Mandated Constraints	14
6a Solution Constraints	14
6b Implementation Environment of the Current System	14
6c Partner or Collaborative Applications	14
6d Off-the-Shelf Software	14
6e Anticipated Workplace Environment	14
6f Schedule Constraints	14
6g Budget Constraints	15
7 Naming Conventions and Definitions	15
7a Definitions of Key Terms	15

7b	UML and Other Notation Used in This Document	15
7c	Data Dictionary for Any Included Models	15
8	Relevant Facts and Assumptions	15
8a	Facts	15
8b	Assumptions	15
II	Requirements	16
9	Product Use Cases	16
9a	Use Case Diagrams	16
9b	Product Use Case List	17
9c	Individual Product Use Cases	17
10	Functional Requirements	19
11	Data Requirements	19
12	Performance Requirements	20
12a	Speed and Latency Requirements	20
12b	Precision or Accuracy Requirements	20
12c	Capacity Requirements	20
13	Dependability Requirements	21
13a	Reliability Requirements	21
13b	Availability Requirements	21
13c	Robustness or Fault-Tolerance Requirements	21
13d	Safety-Critical Requirements	22
14	Maintainability and Supportability Requirements	22
14a	Maintenance Requirements	22
14b	Supportability Requirements	22
14c	Adaptability Requirements	23
14d	Scalability or Extensibility Requirements	23
14e	Longevity Requirements	23
15	Security Requirements	23
15a	Access Requirements	24
15b	Integrity Requirements	24
15c	Privacy Requirements	24
15d	Audit Requirements	24
15e	Immunity Requirements	25
16	Usability and Humanity Requirements	25
16a	Ease of Use Requirements	25

16b	Personalization and Internationalization Requirements	25
16c	Learning Requirements	26
16d	Understandability and Politeness Requirements	26
16e	Accessibility Requirements	26
16f	User Documentation Requirements	27
16g	Training Requirements	27
17	Look and Feel Requirements	27
17a	Appearance Requirements	27
17b	Style Requirements	28
18	Operational and Environmental Requirements	28
18a	Expected Physical Environment	28
18b	Requirements for Interfacing with Adjacent Systems	28
18c	Productization Requirements	29
18d	Release Requirements	29
19	Cultural and Political Requirements	29
19a	Cultural Requirements	29
19b	Political Requirements	30
20	Legal Requirements	30
20a	Compliance Requirements	30
20b	Standards Requirements	30
21	Requirements Acceptance Tests	31
21a	Requirements – Test Correspondence Summary	31
21b	Acceptance Test Descriptions	31
III	Design	32
22	Design Goals	32
23	Current System Design	32
24	Proposed System Design	32
24a	Initial System Analysis and Class Identification	32
24b	Dynamic Modelling of Use-Cases	32
24c	Proposed System Architecture	32
24d	Initial Subsystem Decomposition	33
25	Additional Design Considerations	33
25a	Hardware / Software Mapping	33
25b	Persistent Data Management	33
25c	Access Control and Security	33

25d	Global Software Control	33	
25e	Boundary Conditions	34	
25f	User Interface	34	
25g	Application of Design Patterns	34	
26	Final System Design		34
27	Object Design		34
27a	Packages	35	
27b	Subsystem I	35	
27c	Subsystem II	35	
27d	etc.	35	
IV	Project Issues		35
28	Open Issues		35
29	Off-the-Shelf Solutions		35
29a	Ready-Made Products	35	
29b	Reusable Components	35	
29c	Products That Can Be Copied	36	
30	New Problems		36
30a	Effects on the Current Environment	36	
30b	Effects on the Installed Systems	36	
30c	Potential User Problems	36	
30d	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	36	
30e	Follow-Up Problems	36	
31	Migration to the New Product		37
31a	Requirements for Migration to the New Product	37	
31b	Data That Has to Be Modified or Translated for the New System	37	
32	Risks		37
33	Costs		37
34	Waiting Room		37
35	Ideas for Solutions		37
36	Project Retrospective		38
V	Glossary		38

VI	References / Bibliography	38
VII	Index	38

List of Figures

Figure 2 - Sample Use Case Diagram from Bruegge & DuToit (modified)	16
Figure 3 - Sample Use Case Diagram from Robertson and Robertson	17

List of Tables

Table 2 - Requirements - Acceptance Tests Correspondence

31

I

Project Description

1 Project Overview

The product of this project is a mobile device application that is an accurate GPS mapping of trails in National Parks and forest preserves. A user will be able to view all of the trails in a given park and be able to select what trails they want to go on. They will also be able to see where all the landmarks are within the park and design a route that can bring them to those landmarks. We also want the users to help improve the parks and preserves, so the app will include a rating and commenting system for parks. Having this feedback will help parks focus on the things that are done well and things that are done poorly. The GPS system will be built using Google's GPS API that they use for their Maps application.

2 The Purpose of the Project

The purpose of this project is to improve the National Park system. Millions of people visit these parks every year and we want to make their experience the best it can be. By making it more accessible for all people, we can spur the growth of the National Parks community.

2a The User Business or Background of the Project Effort

The business that the application is being built for is the National Park Service. The people that work these parks certainly are familiar with all the routes in the park, but most visitors are not familiar at all. This application will give the users the tools to navigate just like a park ranger. The motivation for this project is to have more people come visit these places in an easier than ever way. Having more people wanting to travel to these places will generate more money for the parks and for the local businesses. Navigating unfamiliar places can be daunting for many people which is why having an easy-to-use tool like this would be important to create.

2b Goals of the Project

We want to improve the experience of traveling to and exploring a National park for our clients.

2c Measurement

The way we can measure the success of the project would be through the total number of visitors to National parks. By making the process easier for people to plan we should be able to directly see the increase in the number of people that come to the parks.

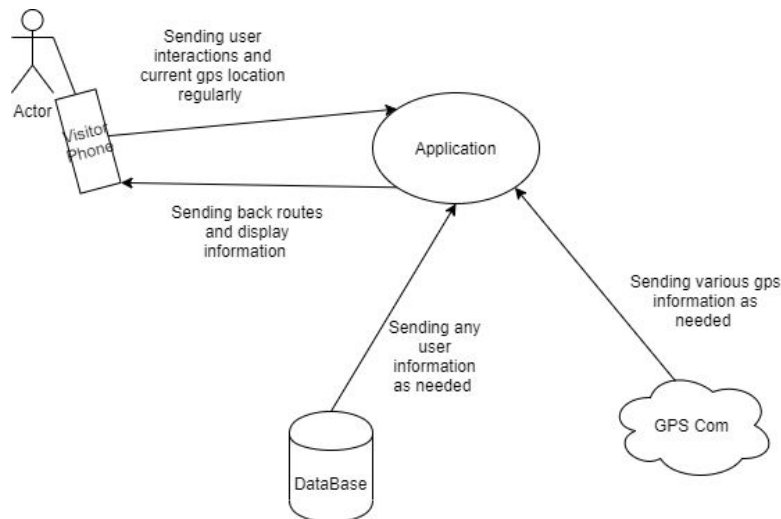
3 The Scope of the Work

The work addressed by this project will be for the entertainment and guidance of national park visitors. It would be a helpful tool that would improve the experience of the national park.

3a The Current Situation

Park visitors are now wasting time figuring out a route to best suit the areas they would want to visit at the park. Instead of using a paper map to figure it out, an application that could quickly be used will allow for more time to enjoy the park.

3b The Context of the Work



The actor will be any person such as a park visitor, park employees, or management. The management will also have the ability to send data that may change the database.

3c Work Partitioning

Event Name	Input and Output	Summary
1. Reading visitor/ employee/ management location	Application(in) Gps Com(out)	Get current user location at first and continually update location as needed
2. Management edit available locations	Database(in)	The database will edit the currently available locations
3. Application display list of locations	Application(out), Database(out)	Application sending the user the currently available locations
4. Planning new areas	Database(in)	Sending new information to

		database about new locations
5. Private Areas	Application(out), Gps (out)	Informing the users that they are in a private location

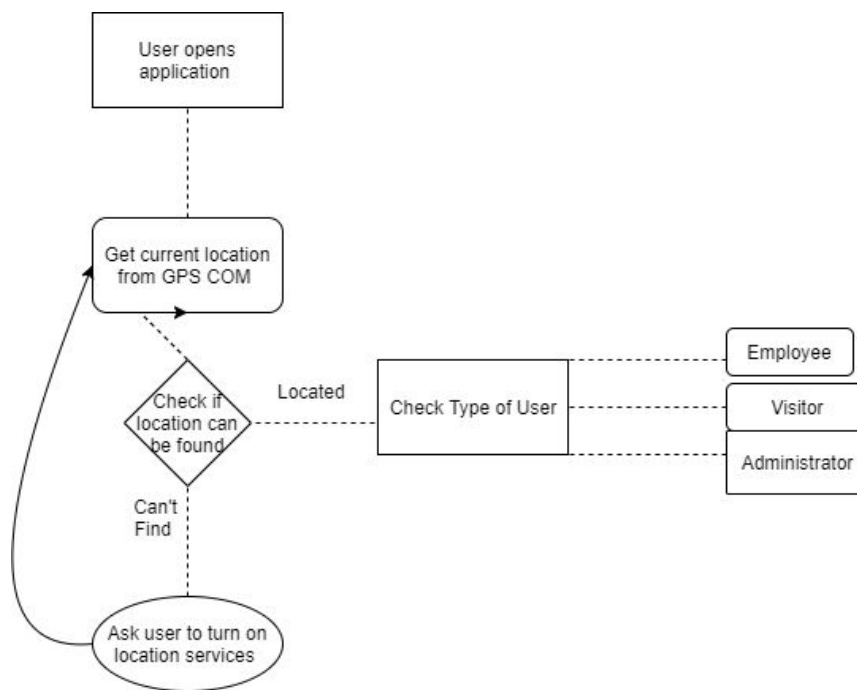
3d Competing Products

There are currently no competing products with ours. The reason for this is because there are no applications that use the current gps mapping to edit the current area of a national park.

4 The Scope of the Product

The application will be used to manage and guide through national parks. Only the administrator will be able to change the available locations and create new paths. The employees and visitors will have access to looking up paths and locations that are open to the public. While the visitors will have access to

4a Scenario Diagram(s)



4b Product Scenario List

- Find Best Route
- Edit Path
- Edit prohibited areas
- Recommend new routes

4c Individual Product Scenarios

Find Best Route- The user will select the option “Find Best Route”. In this option the user will select the areas of the park that they would like to visit. The application will then produce the most optimal route based on the areas picked. Then that route will be shown to the user and the user will have the option to edit the list if they wanted.

Edit Path- Only the key user profiles will have the option to edit a certain path. In this option they will have the ability to edit the path of their choice in a way that would see fit. After the user is done editing the path it can then be uploaded to the main database to be changed.

Edit Prohibited Paths- Only the key user profiles will have the option to change the current prohibited pathways for basic users. The application will look similar to the edit path option, only with a new option to change a current available path to change it to prohibited. After that is done in the database that path will change status to the off limits.

Recommend new Paths- All users will be able to use this feature. The feature will allow the user to recommend paths for others to follow at the park. When the user selects the option they will be able to add an description of why they would recommend that path. They will then choose to publish that path to the main database to be shared with others.

5 Stakeholders

5a The Client

The client is the National Park Service. They will pay for the development of the GPS trail application and provide input and guidance for the product.

5b The Customer

The end product will be distributed through the app store on phones. It is not intended that the customer will purchase the app, as the goal is to have as many people as possible hiking the trails.

5c Hands-On Users of the Product

The hands-on users will be those who download the app. They will be visitors in the park who wish to hike the trails. These users are expected to know the park’s features and attractions, but they are not necessarily expected to be familiar with the park’s geography. It is expected for the hands-on users to be at least 16 years of age.

5d Maintenance Users and Service Technicians

Maintenance will be necessary to ensure that the pathing algorithm is working properly and the trail map is up to date. Trails in the park may closed or new paths may open. This maintenance is expected to be done by the National Parks Service.

5e Other Stakeholders

There will be legal stakeholders. They will be interested in making sure the app complies with any relevant laws.

5f User Participation

Park workers will provide feedback during the development of the product. They will ensure that the generated trails are accurate and safe for the end-users of the product.

5g Priorities Assigned to Users

Park workers are the key users of the product. They have detailed knowledge of the park and can ensure the app is working correctly. The park visitors are secondary users. They will be the users the app is ultimately targeted towards.

6 Mandated Constraints

6a Solution Constraints

The program must be programmed to run on a mobile device. Creating the product to run on a desktop is not ideal because it's intended to be marketed towards hikers, joggers, and other clients who visit national parks. Creating a mobile application is the only viable option for a product that expects the users to be on the move.

6b Implementation Environment of the Current System

Because the product is going to be used in national parks, expecting the client to have a stable internet connection while in the park is asking too much. For this reason, the app will need to be made as native as possible. Designing the application so that it's as native as possible alleviates the reliance on a stable internet connection.

The maps of each national park will be downloaded during installation of the product. One of the few instances where the application will require an internet connection is when it's attempting to figure out which national park the user is in. Drawing the path the user will take will also require an internet connection.

6c Partner or Collaborative Applications

The application will primarily rely on Google's API in order to display the possible paths the user can take when the application is given a starting point. The maps for each national park will be stored as a .shp in order to ensure compatibility with Google maps.

Other collaborative applications include the users "phone" app, which can be used to call for help in the event of an emergency with a simple press of a button in the application.

6d Off-the-Shelf Software

Google's API and SDK allows the application to run by loading .shp and showing the user the entire map of the national park they are currently in. Google's API must be used in order for the application to run since its primary purpose is to show the user a possible path that they can take. The user is also expected to be running IOS or Android, as these are the only two operating systems that the application will support.

6e Anticipated Workplace Environment

Since this application is primarily marketed towards those who visit national parks, the environment of a national park does not allow for stable internet connection. For this reason, a primary goal during development should be to limit the reliance on an internet connection and to make things as native as possible. Having everything pre-downloaded from the beginning will ensure that the application remains working even when the user has no internet connection.

6f Schedule Constraints

The most popular times for national parks to receive visitors is during the summertime. For this reason, the beta version of the application should be finished by the beginning of April and the second version of the application should be finished by mid May.

The main features that should be implemented from get-go is the user interface and the ability to locate which national park the user is currently in. These are the two most basic things that should be added. From there, the ability for the application to read .shp files and load them depending on which national park the user is located in should be the next thing to be added. The final feature to be added is presenting the paths to the user based on the user's current location with the help of the phones location services and Google's API. The app with these basic features is considered in beta.

The second iteration of the application will allow the user to select their own starting point rather than relying on their location services in the event that they do not have an internet connection. Generating a path will also be possible natively when the user manually chooses a starting point and an ending point.

6g Budget Constraints

An application like this does not require a major budget. The biggest expense to keep the application running is the database that holds the national parks information. Paying Google to make API calls will also be a primary expense since the application will constantly be making calls in order to show the user's current location and their progress as they navigate through the path.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Actor: Any person utilizing the app.

Key users: Users critical to the continued success of the product

Secondary users: Users use the product, but their opinion is not critical to the product long-term

7b UML and Other Notation Used in This Document

This document follows the standard UML file format.

7c Data Dictionary for Any Included Models

There is no data dictionary for any of the models.

8 Relevant Facts and Assumptions

8a Facts

The time when national parks receive the most visitors is during the summer so the application should be completed by that time. Most national parks can also have multiple paths that the user will be able to select from. The application will use Google's API in order to generate a path for the user. The maps of the national parks will also be downloaded during the installation of the application. The user's device running the application will be either an iPhone running IOS or an Android running any android operating system that allows .apk files to be installed.

8b Assumptions

It's assumed that the user is not going to have a solid internet connection so the app will run as natively as possible to avoid the limitations of having no internet connection. It is also assumed that while the internet is not the best, it will still be good enough for Google API to run and to show the user their current location from the beginning of the path to the end.

II Requirements

SV: Sections 9 and 10 deal with functional requirements. Sections 11 to 20 are a very thorough list of possible non-functional requirements, not all of which apply to every project. You should think carefully about each of these, form requirements if applicable, or write “Not Applicable” otherwise. See section 10 for the format of individual requirements. Section 21 documents the acceptance tests planned to verify the requirements – See that section for further details, and be aware that every requirement needs at least one verifying acceptance test (though some tests may verify more than one requirement.)

1 Product Use Cases

SV: Product Use Cases are very similar to Product Scenarios, but in more formal detail. They serve as a first step towards developing functional requirements, and can aid in organizing requirements according to the use case(s) from which they were developed. See the CS 440 web site for a sample use-case form, with instructions.

1a 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.

Examples

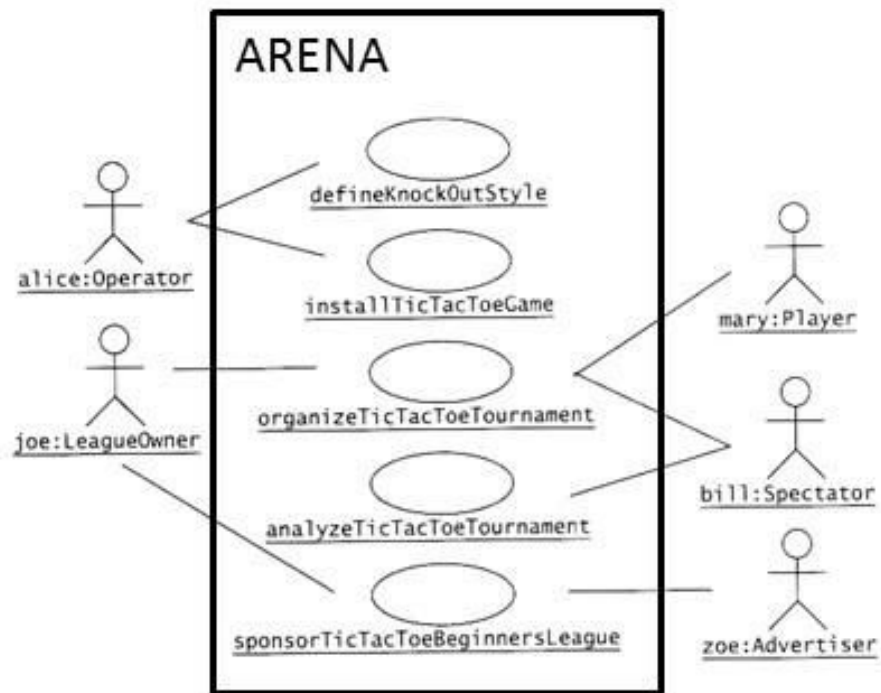


Figure 1 - Sample Use Case Diagram from Bruegge & DuToit (modified)

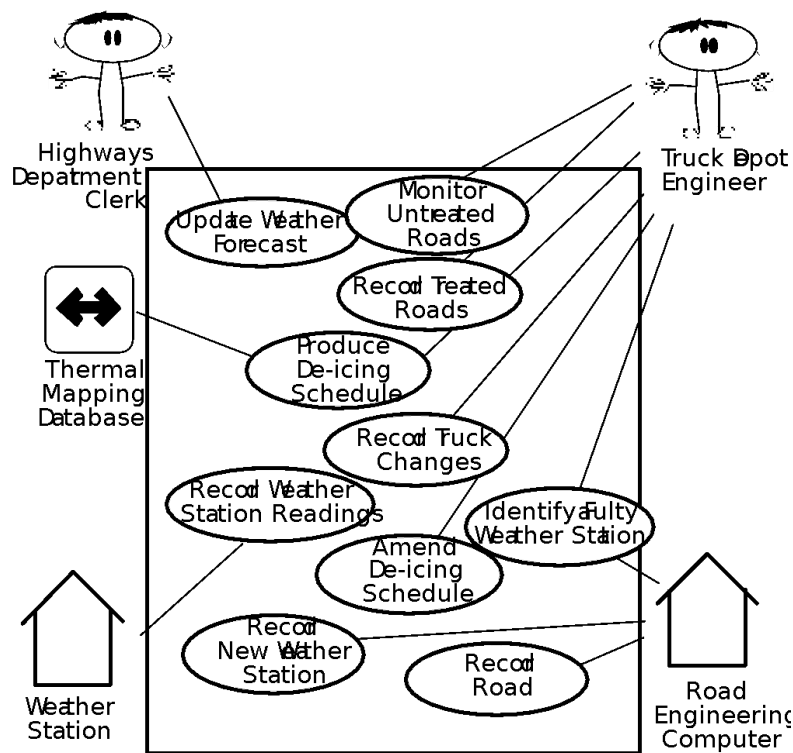


Figure 2 - Sample Use Case Diagram from Robertson and Robertson

1b 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.

1c Individual Product Use Cases

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: pre-conditions: post-conditions: Initiated by: Triggering Event: Additional Actors:	Name:
<p>Sequence of Events:</p> <ol style="list-style-type: none"> 1. Initiating event or action should be step 1, taken by initiating actor. 2. System response follows, indented right. 3. All external action steps are aligned with step 1. ("stimulus" style) 4. All system responses are indented right, aligned with step 2. ("response" style) 5. All steps should be expressed in the active voice, clearly indicating <u>who</u> performs each action 6. The sequence of events should show a back-and-forth stimulus-response relationship. 	
<p>Alternatives: These would be normal and expected variations from the base case.</p> <p>Exceptions: These would be unusual variations from the base case, often caused by problems.</p>	

- *For all of the above, list as NA if not applicable.*
- *The following may be added if relevant, or omitted otherwise:*
 - o related use cases or scenarios*
 - o associated tests, systems, classes, etc.*
 - o revision history*
 - o references to other documents*

- o *author(s) / originator(s)*
- o *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.*

2 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

3 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

4 Performance Requirements

4a 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

4b Precision or Accuracy Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

4c 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

5 Dependability Requirements

5a 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

5b 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

5c 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

5d Safety-Critical Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

6 Maintainability and Supportability Requirements

6a 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

6b Supportability Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

6c Adaptability Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

6d Scalability or Extensibility Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

6e Longevity Requirements

SV: This specifies the expected lifetime 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 . . .

7 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.

7a Access Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

7b Integrity Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

7c 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.)

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

7d Audit Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

7e Immunity Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

8 Usability and Humanity Requirements

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

8a 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

8b 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

8c Learning Requirements

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

8d 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

8e Accessibility Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

8f User Documentation Requirements

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

8g Training Requirements

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

9 Look and Feel Requirements

9a 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

9b 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

10 Operational and Environmental Requirements

10a Expected Physical Environment

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

10b 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

10c Productization Requirements

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

10d Release Requirements

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

11 Cultural and Political Requirements

11a 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

11b 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

12 Legal Requirements

12a 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.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

12b Standards Requirements

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

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

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

13 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.

13a 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.

Test	Requirements																			
	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9	Req 10	Req 11	Req 12	Req 13	Req 14	Req 15	Req 16	Req 17	Req 18	Req 19	Req 20
Test 1	X																			
Test 2		X				X														
Test 3			X	X																
Test 4					X	X														
Test 5																				
Test 6																				
Test 7																				
Test 8																				
Test 9																				
Test 10																				
Test 11																				
Test 12																				
Test 13																				
Test 14																				
Test 15																				

Table 1 - Requirements - Acceptance Tests Correspondence

13b 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 . . .

III Design

1 Design Goals

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

Your text goes here . . .

2 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 . . .

3 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.

3a Initial System Analysis and Class Identification

SV: Perform grammatical and similar analyses to identify the most important 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 . . .

3b 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.

Your text goes here . . .

3c 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 . . .

3d 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 . . .

4 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.

4a 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 . . .

4b 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 **Error! Reference source not found.** to avoid undue repetition, while reviewing only the most relevant items here.*

Your text goes here . . .

4c 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 . . .

4d 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 . . .

4e 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 . . .

4f 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.

Your text goes here . . .

4g 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 . . .

5 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 . . .

6 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.

6a 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 . . .

6b Subsystem I

Your text goes here . . .

6c Subsystem II

Your text goes here . . .

6d etc.

Your text goes here . . .

IV Project Issues

1 Open Issues

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

Your text goes here . . .

2 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 . . .

2a 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.

Your text goes here . . .

2b Reusable Components

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

Your text goes here . . .

2c 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.

Your text goes here . . .

3 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.

3a 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?

Your text goes here . . .

3b Effects on the Installed Systems

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

Your text goes here . . .

3c 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?

Your text goes here . . .

3d 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.)

Your text goes here . . .

3e Follow-Up Problems

SV: Basically any other possible problems that could occur.

Your text goes here . . .

4 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.

4a 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.

Your text goes here . . .

4b 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.

Your text goes here . . .

5 Risks

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

Your text goes here . . .

6 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.

Your text goes here . . .

7 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.

Your text goes here . . .

8 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.

Your text goes here . . .

9 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.

Your text goes here . . .

V 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.

Your text goes here . . .

VI 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.

[1] Robertson and Robertson, Mastering the Requirements Process.

- [2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.
- [3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.
- [4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

VII 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