National Parks Builder

A gaming application based on National Parks

Prepared by
Maaz Ahmed
Bryan Gutierrez
Ammar Idrees
Zaid Patel
for use in CS 440
at the
University of Illinois Chicago

Spring 2020

Table of Contents

List of Figures

List of Tables

Project Description

1 Project Overview

National Parks Builder is a gaming application that allows users to build and manage a national park from scratch. Users first get an option of picking an undeveloped land from a current National Park's location. So for example, if the user wanted to rebuild Sequoia National Park, the location would be California. Then the user would be in charge of rebuilding the park and deciding where features of the park would go. This can include features such as benches, campsites, trees, a welcome center, parking lot, etc. Once the user has built their initial design, then they will be in charge of the management aspects of running a National Park.

The management portion would contain duties such as fixing issues within the park, setting ticket prices, business decisions within the park, keeping employee morale high, etc. If you don't meet certain revenue and visitor population goals then there is a chance your park could get shut down. This application would allow you to play compete with your friends to see who can build and manage the better National Park. Also this game will have microtransactions and the funds from that would be donated to the National Parks Foundation.

Users can put tons of hours into this application because every time they log on there would be new tasks they would have to deal with that pertain to the National Park. There would also be a leaderboard within the application that would show which users have gained the most visitors and made the most revenue for their park. This entices competition within users and can help ensure they are enjoying the game.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Our application is most useful for the National Parks foundation especially those parks with lower attendance. The idea is that we feel National Parks are becoming less relevant in today's society. This is due to the fact that the usage of electronics and gaming is increasing significantly. So we felt that a gaming application like this would give exposure to National Parks and allow the user to appreciate its beauty and the effort that goes into managing one. We hope that this will encourage users to go experience these parks in person.

At the same time, we want this application to be equally entertaining as it is informative. We want to make sure that users get excited when logging into this application everyday. Furthermore, we want to convey the hidden message that users should be respectful to the parks and their management. We want them to become aware of the fact of how much time,effort and manpower goes into running a National Park. We have had experiences where we have seen people mess the trees and plants in National Parks because they think it'll just grow back easily.

2b Goals of the Project

We want to make sure that today's society is aware of all the National Parks that exist and the beauty within them.The goal of this project is to increase the number of people who visit National Parks. If the number of visitors increase then the hope is profit should increase as well. We hope to accomplish all this while making sure the user is being entertained as well.

2c Measurement

The simplest way to measure this is to track visitors monthly for a year before the application is released. Once the application is released, then the stats would be taken again monthly to see if visitors numbers increased monthly. To make sure that this application is the reason why, we could implement an idea that when visitors purchase their tickets they fill out a questionnaire. One of the questions would be Why have you decided to visit this National Park? Then we can have a drop down menu with options and one of the options would be because of the National Parks Builder application.

Furthemore, we can see if profit increased due to more people visiting the parks. We can count the revenue that came from people who bought tickets and put National Parks builder as the reason why. It would be difficult to track in park purchases such as the gift shop but it would be possible to track the sales of microtransactions. This can be tracked in monthly revenue. We would have to take a deeper dive into the revenue streams and keep track of the exact amount coming from microtransactions. We don't expect revenue to increase by a large margin from microtransactions, but something is better than nothing.

Additionally, we would want to try to see if management/employee morale increased since the user of National Parks Builder is becoming their own manager. We hope that they treat National Park managers with more respect and better manners when they visit themselves. This can be measured by an employee survey asking how their morale was this month. Also to see if less incidents/complaints are happening in National Parks due to this application.

3 The Scope of the Work

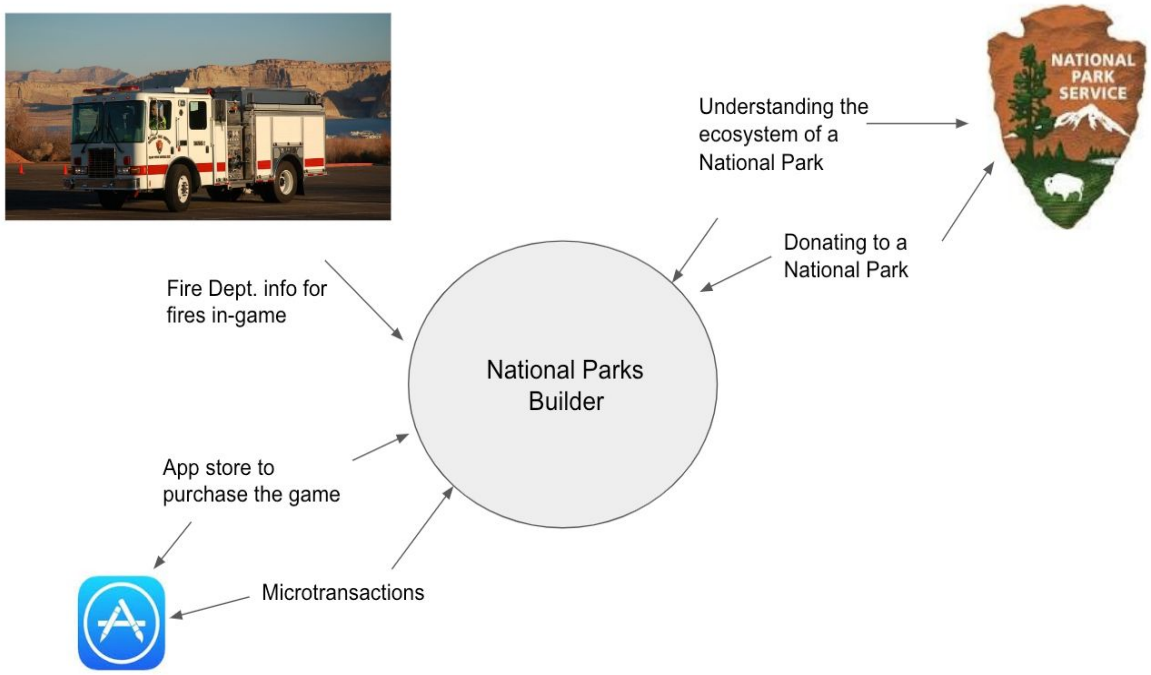
The business involved with this product is the National Park Service. Our product is a small part of this business that involves creating awareness for the struggles and every day tasks of taking care of a National park while also making it entertaining to play.

3a The Current Situation

This product does not necessarily replace or change how the business will be run directly. The National Park Builder is another form of awareness that is more interesting than the

current options. Currently, there are only traditional ways of showing difficulties and requesting aid. There are many articles that list the many problems that National Parks must face that are mostly due to a lack of financial aid. Another form of awareness of the National Park Service is their website, it exhibits the beauties in National Parks and has an option for donations to improve the parks.

3b The Context of the Work



The National Park Builder will primarily take information from the National Park Service website to interpret the data and create a game out of. Phone app stores will be used to manage the purchase of the game and microtransactions. Natural Disasters such as fires will be in the game so it is important to make it realistic.

3c Work Partitioning

Event Name	Input and Output	Summary
User adds natural Units	Unit placed in terrain(Input), unit is displayed on map(out)	The user can add a new unit(, tree,animals,etc) to the terrain
Fire	Not enough rainfall,	There are a few ways for a fire to

	random, or possibly accident by employee (in) Causes fire(out)	start within the park, the user must stop the fire from spreading
Microtransactions	User purchases in-game item(In), Item is added to inventory(out)	When an item is purchased, the user must be able to see and use it
User adds Building Unit	Building unit placed on terrain(in), building is in place(out)	The user can decide between different buildings to add depending on what they need.
Cleaning area	Clicking dirty sections of land (in), land will clean up(out)	The user's map/land can get messy with visits from tourists so the user can clean it up
Donation	User donates money to NPS(In), App sends money to NPS	The game will come with an option to donate money to the National Park Service to be used for renovations/repairs. They can pick which park to donate t.

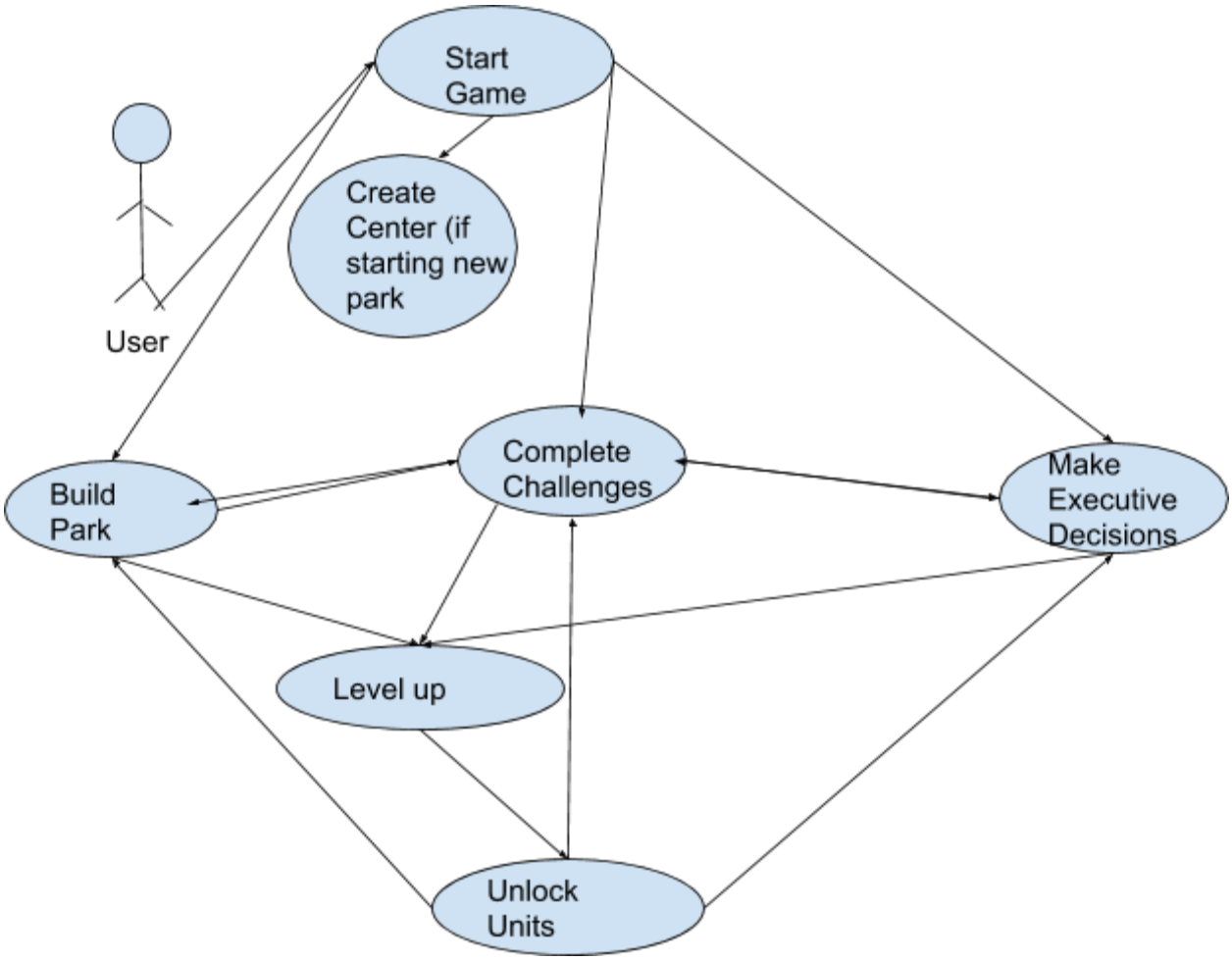
3d Competing Products

There are no other products similar to this one that relate to the same topic of national parks. There are similar building games such as Clash of Clans or The Sims but they do not involve national parks.

4 The Scope of the Product

The business involved with this product is national parks but the main focus of the product is to create more ways to raise awareness of the struggles that national park workers deal with. The few forms of awareness are mostly articles/readings but this product creates a way to entertain users while also teaching them.

4a Scenario Diagram(s)



4b Product Scenario List

1.User adds natural units
2.User adds Building Unit
3.Fire
4.Cleaning area
5.Microtransactions

4c Individual Product Scenarios

A user will be able to select between “natural” units such as trees, bushes, a variety of animals, and many more units. The units will be purchased using currency earned from achievements and other forms of production. Similar to the natural units, a user will be to add Building units to their land that each serve their own purpose. When the game initially starts up, the user will need to create a “Town Hall” for the center of their national park. Fires can

occur for many reasons that do not necessarily involve the user to be at fault. Fires can occur for lack of rain(random) or for putting units too closely together. Cleaning an area is a task that the user will have in order to maintain a clean park for guests/tourists. The user will simply have to click the dirty areas and select clean but this will take time and resources. Microtransactions will be used by the user in order to purchase items that they can use in-game to better their national park.

5 Stakeholders

The stakeholders that are interested in this product would be the private landowners who own a major portion of the parks land. Furthermore, in an indirect sense activists that are in support of the success and future of national parks would also have an interest in the product being developed.

5a The Client

The client in this case would be the the land owners, and different investors who are interested in owning a stake within the parks. Alternatively, some investors who are avid in terms of keeping the longevity and sustainability of the parks could invest in our product. For those interested in the sustainability of the parks, they would want to spread awareness across society through the game trying to educate a mass amount of people about the efforts and financial needs that go into merely running the park.

5b The Customer

The customers will be those looking to gain experience on national parks, or whoever is interested in downloading it within the app store. There could possibly be a deluxe edition for those actually visiting National Parks, via a QR code given at the beginning entrance to customize the experience within the app during their visit. The app would be given on mobile devices for customers, and would be able to choose the maps between different national parks.

5c Hands-On Users of the Product

The hand on users for this product could be used by non profit organizations or as an educational tool within schools. Non profit organizations could use this product as a way to get people informed about the app, and could be a method to bring in donations for the parks to use. The directors of the parks could also be an effective user for this product, and they could possibly have enhanced features regarding the product, being able to view which playing customers are at their current parks, and if possible be able to alter or bring on some additions for their tour throughout the park.

5d Maintenance Users and Service Technicians

The app itself would be free for users looking to download it within the app store. Overall, the National Park members themselves who are enrolled as premium members may have to pay a

beginning fee to have the app linked to their parks, and possibly monthly recurring payments to fix patches and have updates within the app. The app itself could possibly have in game purchases for the customers, depending on how they want to customize the experience, and monthly updates would be free of charge to them.

5e Other Stakeholders

Technology Experts: These members will be in charge of offering the best service for the app, adding the best services for the app. They could possibly translate the app into a CRM, trying to normalize the relationship between customer and the application. This team will also be in charge of gathering all the fixes needed monthly, from bug patches, to updates, to marketing. They are basically the focal point that gathers all the data from all the other teams and gets it into production.

Software Developers/Quality Testers: This team will be in charge of the weekly springs enabling the back end, database, gameplay, and overall mechanics of the whole application. All the heavy load programming is done here, and so obviously they have a big stake on whether the application is successful or not. Overall, development is constantly needed with applications such as these, so this team will be having to constantly update the app, think of new ideas to implement, and be in charge of providing the best experience for the customers. Furthermore, the quality testers are in charge of constantly making improvements and completing unit tests possible utilizing maven.

Strategic Analysts: This team will be implementing critical solutions through reviews and large amounts of testing. Furthermore, they will be trying to begin to increase profitability and growth for the application as a whole. Overall, this team will be trying to gain the analytics of the customers themselves, and have a large impact on the stakes if they want the app to be successful. They will have to analyze the current market for similar applications, and try and optimize the sales of the game.

5f User Participation

Users will have a large influence over the product, being able to share their thoughts within the reviews, or have a suggestion forum on the applications landing page. The customers basically have the ability to ask for new features, or different aspects of the app that they want changed and so forth. Overall, customers feedback will be very important as it will also affect National Parks, thus by listening to their voice it will only strengthen the relationship between them and the parks themselves. Furthermore, after users are able to visit the Parks they will be able to meet with the office to share their experience, in which their recommendations will take higher priority.

5g Priorities Assigned to Users

The users should simply have access to the application, basically just able to access the game, in a programming realm just have the read option. The management of the specific National Parks

that are related to the application could possibly also have a write option for certain aspects of the app, having permission to be able to change certain types of the product. The customers would also have the ability to voice their concerns to developers, who have access to write and can make subtle changes.

6 Mandated Constraints

6a Solution Constraints

Some solution constraints regarding this project is that the application follows the click and drag approach as that will be the majority of the functionality for the free version. Overall, there will be a pane regarding the application, imitating a full map of the National Park. There will be event handlers for each area of the map. For example, a player for example has a budget of a few million, and must allocate it in a way of keeping the park at a substantial level. There will be many conflicts within the game such as pollution in the water reservoirs, or high pollution levels affecting wildlife. The players will have to allocate resources in these areas to keep the park running. The game will have a real time health bar, possibly within javascript that will slowly decrease if the issues within the park are not being solved. The developers will need to have to link all the different aspects within the game such as wildlife, forestry, water reservoirs, tourists, income all within the health bar. If the player decides to visit a National Park with the app in hand, the developers will need to implement a version of currency that can be collected throughout the application with the camera. The user will give access to their camera to the app, and will walk around the Park until they find a token which could be converted to currency within the game. This will require some use of computer vision and a form of AI such as tensor flow.

6b Implementation Environment of the Current System

The product will need to be runned on mobile devices, and will need to follow a particular design pattern overall. The game will need to purchase its own server, to have all the different players be able to run on it. The game should be able to run with library graphics, and could be done optimizing with the application fusion. Furthermore, some data encryption will need to be implemented within the app, to keep a secure platform and save all information. There will need to be a form of SPIM which could provide an emulator if needed.

6c Partner or Collaborative Applications

Some collaborative applications that could be utilized would be obviously setting up some form of payment reaching out to a form of CTO commerce to obtain rights. Furthermore, security and hashing important information within the application would be important, so a possible partnership with a security solutions team would be essential in keeping the privacy of the customers who have given their information to the app.

6d Off-the-Shelf Software

There are quite a few off shelf softwares that would be needed to complete the application. Certain outside computer vision softwares such as YOLO could be utilized for the feature

utilizing the camera for then the user decides to engage with the camera. Furthermore, the application needs to be able to run for both IOS and Android so a few options between Xamarin or Flutter could be utilized to ensure the application is properly published.

6e Anticipated Workplace Environment

The product will mainly be used on mobile devices, either indoor or outdoor depending on the user. The application should not take up too much storage, as it might scare customers away. Furthermore, depending on if the customer wants to try the premium version which is able to correspond with real National Parks, a live location will be needed, with possible access to the camera of the device.

6f Schedule Constraints

There are not many schedule constraints regarding the development and launch of the product. However, for the premium edition of the application, where users will have a personalized experience based on the National Park that they visit, could vary depending on the availability of the park. If some parks are closed seasonal, or have certain times when they are not open to admission, it could somewhat complicate that feature of the application.

6g Budget Constraints

There may be some budget constraints for this project, especially if we would want to implement a feature that would entail combining it with actually visiting National Parks. The National Parks are usually federally funded, and it is highly unlikely they have the means to be willing to invest in the application in itself. Furthermore, some non profit organizations may be apart, if they are able to conform with the overall interior plan for why the app is being created. Furthermore, being able to hire all the needed members, such as marketing, analysts, developers would be a process in creating such a large team.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Park - The area that the user will be building throughout the game.

Center - The main hub of the park. Will be the main source of organization for the entire map. Will be required to build at the start of the game.

Land - Unused space around the national park.

Unit - Some sort of object, building, or piece of nature that the user can place on the land.

7b UML and Other Notation Used in This Document

Any of the UMLs or notations that will be made in this project will be made based on the descriptions and diagrams throughout the project.

7c Data Dictionary for Any Included Models

User - Will require a server to work on the national park. Will need to be able to view currency, employee morale, and average visits.

Currency - Integer that will be a minimum of 0 (can't be negative).

Employee morale - Integer that will be displayed as some graphic.

Average visits - Will contain integers for age of park and total visits. Will calculate and display a double.

8 Relevant Facts and Assumptions

8a Facts

We want to compile facts and data about national parks so that they can be displayed to the user. They can be used as "fun facts" to give awareness. Furthermore, we can utilize data from actual national parks as a way to create challenges. Doing so would allow the challenges to mimic real life issues that national parks may face.

8b Assumptions

We would assume the user is able to perform basic tasks such as selecting certain options or navigate the menu and game. We would also assume the user has the necessary requirements to run the game.

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.)

9 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.

9a 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

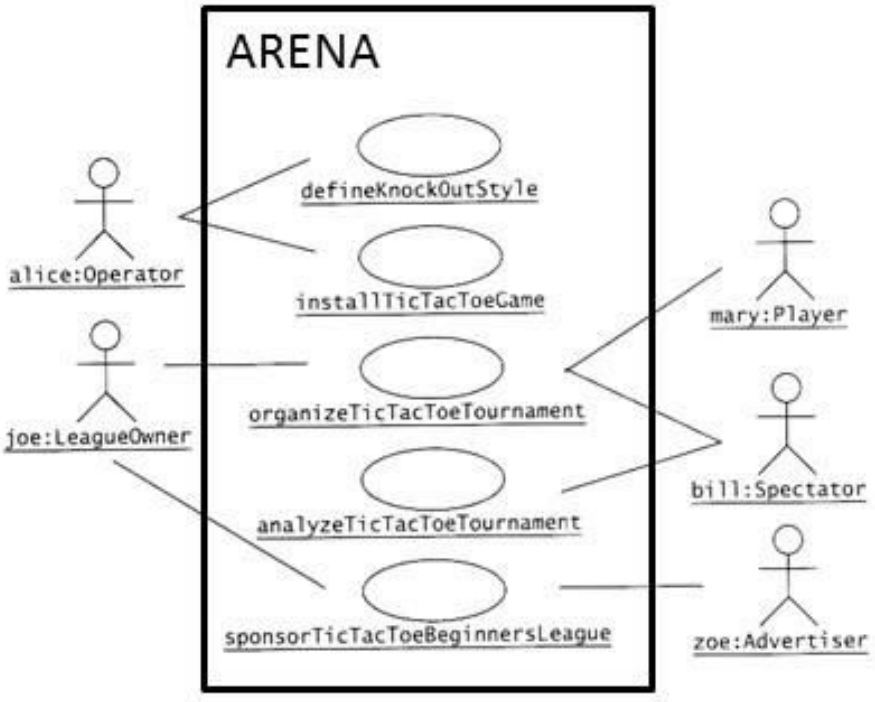


Figure 1 - Sample Use Case Diagram from Bruegge & DuToit (modified)

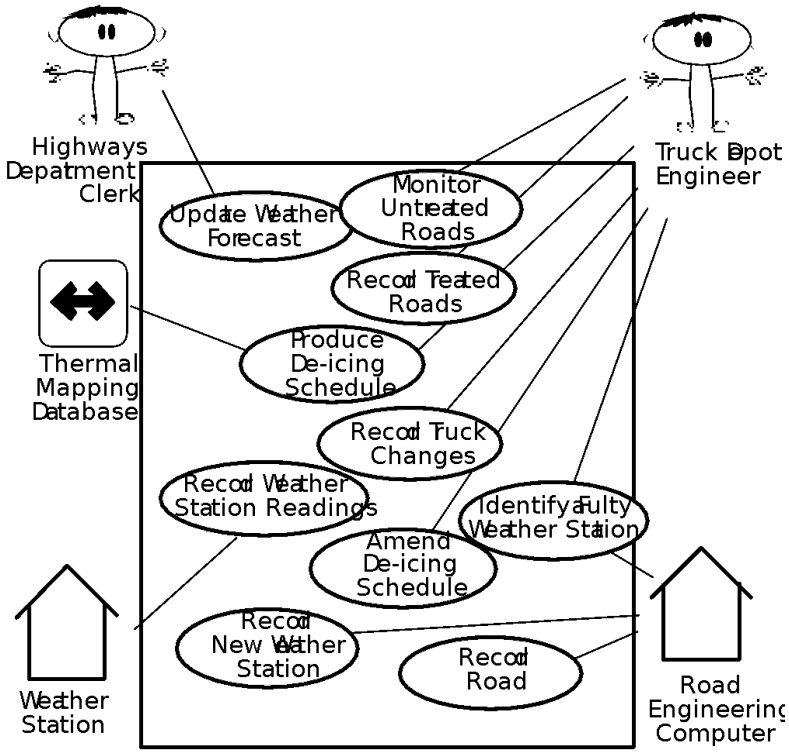


Figure 2 - Sample Use Case Diagram from Robertson and Robertson

9b 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.

9c 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: Name: pre-conditions: post-conditions: Initiated by: Triggering Event: Additional Actors:
Sequence of Events: 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 **who** performs each action 6. 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.*

10 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 . . .

11 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 . . .

12 Performance Requirements

12a 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 . . .

12b 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 . . .

12c 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 . . .

13 Dependability Requirements

13a 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 . . .

13b 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 . . .

13c 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 . . .

13d 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 . . .

14 Maintainability and Supportability Requirements

14a 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 . . .

14b 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 . . .

14c 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 . . .

14d 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 . . .

14e 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 . . .

15 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.

15a 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 . . .

15b 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 . . .

15c 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 . . .

15d 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 . . .

15e 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 . . .

16 Usability and Humanity Requirements

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

16a 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 . . .

16b 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 . . .

16c 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 . . .

16d 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 . . .

16e 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 . . .

16f 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 . . .

16g 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 . . .

17 Look and Feel Requirements

17a 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 . . .

17b 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 . . .

18 Operational and Environmental Requirements

18a 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 . . .

18b 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 . . .

18c 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 . . .

18d 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 . . .

19 Cultural and Political Requirements

19a 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 . . .

19b 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 . . .

20 Legal Requirements

20a 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 . . .

20b 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 . . .

21 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.

21a 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.

	Requirements																			
Test	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

21b 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

22 Design Goals

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

Your text goes here . . .

23 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 . . .

24 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.

24a 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 . . .

24b 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 . . .

24c 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 . . .

24d 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 . . .

25 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.

25a 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 . . .

25b 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 . . .

25c 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 . . .

25d 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 . . .

25e 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 . . .

25f 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 . . .

25g 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 . . .

26 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 . . .

27 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.

27a 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 . . .

27b Subsystem I

Your text goes here . . .

27c Subsystem II

Your text goes here . . .

27d etc.

Your text goes here . . .

Project Issues

28 Open Issues

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

Your text goes here . . .

29 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 . . .

29a 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 . . .

29b Reusable Components

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

Your text goes here . . .

29c 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 . . .

30 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.

30a 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 . . .

30b Effects on the Installed Systems

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

Your text goes here . . .

30c 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 . . .

30d 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 . . .

30e Follow-Up Problems

SV: Basically any other possible problems that could occur.

Your text goes here . . .

31 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.

31a 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 . . .

31b 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 . . .

32 Risks

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

Your text goes here . . .

33 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 . . .

34 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 . . .

35 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 . . .

36 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 . . .

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 . . .

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.

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. }