Pac-Rats

A Browser Based Implementation of Multiplayer Pac-Man

Prepared by
Jonathon Repta, Jennifer Alonso, Edward Liang, Amal Syed
at the
University of Illinois Chicago

October 2020

REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

Pac-Rats is a multiplayer implementation of the arcade classic, Pac-Man. The purpose of Pac-Rats is to offer a unique twist on a classic game and, in doing so, reinvigorate the interest of players who are familiar with the original, but are looking for something fresh. In order to achieve our goal of revitalizing Pac-Man, we will introduce a number of new features. Features such as new game modes (co-op, free-for-all), an extensive leaderboard system, and community based features like map creation/sharing. Pac-Rats will offer an accessible experience that will be enjoyable to players of all skill levels.

Users can easily link up with friends and do not require a high end pc to play. Onced signed-up they will be registered into the database and can login anytime as long they have a computer and internet. This game is intended for all ages and since it is inspired by the classic pac-man game, the mechanics/controls are easy to learn. Even a child will be able to pick it quickly and play with each other.

2 The Purpose of the Project

The project aims to create a game that combines traditional Pac-Man style gameplay with in order to create a game that will be accessible to a wide range of players whilst still maintaining the attention of the serious "hardcore" gaming audience. The hope is that the project will be a profitable venture for the publisher whilst being an enjoyable experience for a variety of players.

2a The User Business or Background of the Project Effort

The business that the client is a part of is the gaming industry. Specifically the section of the games industry which is in charge of remaking/remastering classic games. This is relevant to the project since the developers need to understand the wants and needs of the average gamer and balance that with the requirements given by the stakeholders.

2b Goals of the Project

The goal of this project is to take a classic game and breathe new life into it by applying modern gaming practices. Furthermore, the project aims to make a classic game profitable again by retrofitting modern monetization schemes into a game that didn't have them originally. Essentially the project aims to take the addicting retro gameplay of Pac-Man and combine it with innovations in multiplayer gaming that have come since.

2c Measurement

Pac-Rats will keep detailed usage statistics about users in order to monitor audience retention. Things like the number of new users every day/month/week that log on, the amount of time the average user spends on the app, etc. The success of our product will be measured against the user retention of the average Steam game in order to see if our product is able to compete with comparable dedicated gaming products. If our

game is able to match the retention rate of the average Steam game (or better yet, exceed it) we will have achieved our goal.

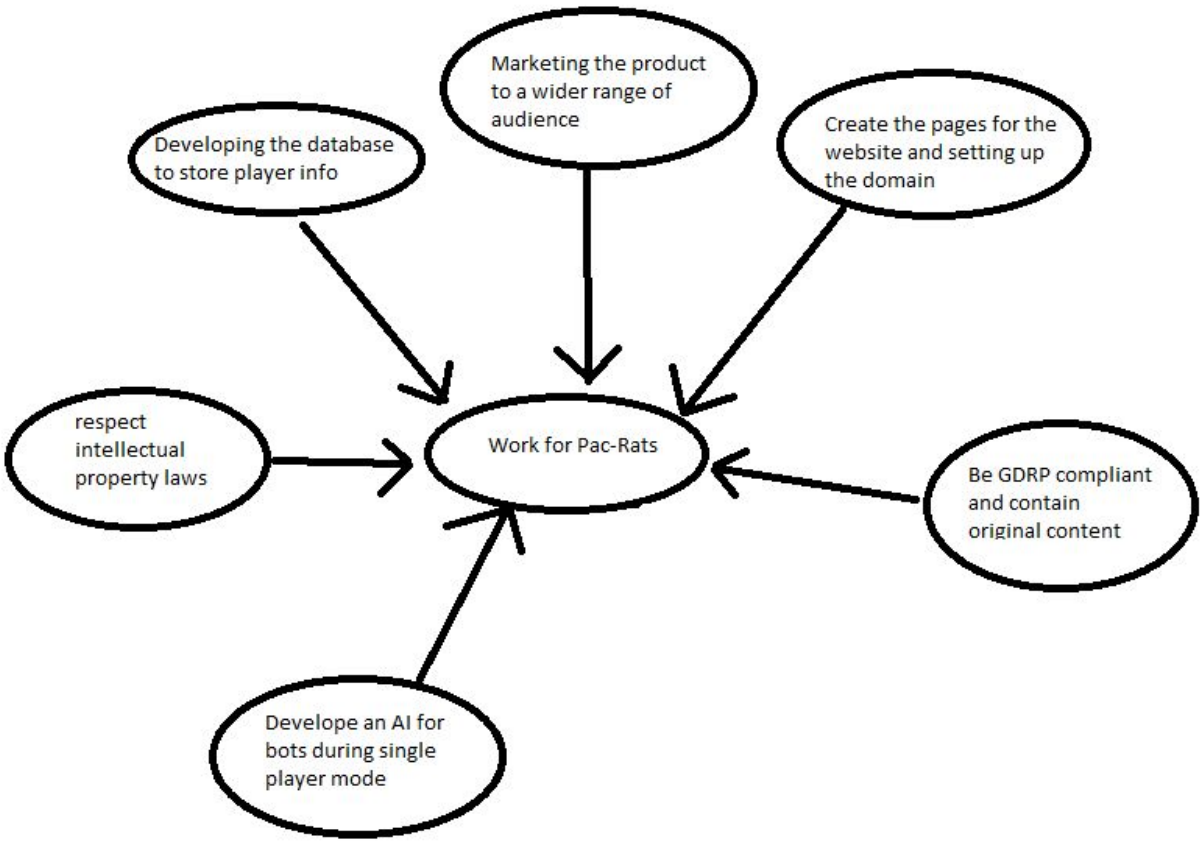
3 The Scope of the Work

The business is that of the games industry. Specifically, recreating old classical arcade games that everybody knows growing up and featuring new and original content. The work can be addressed with this project by recreating a traditional pac-man game from the arcade with pac-rats.

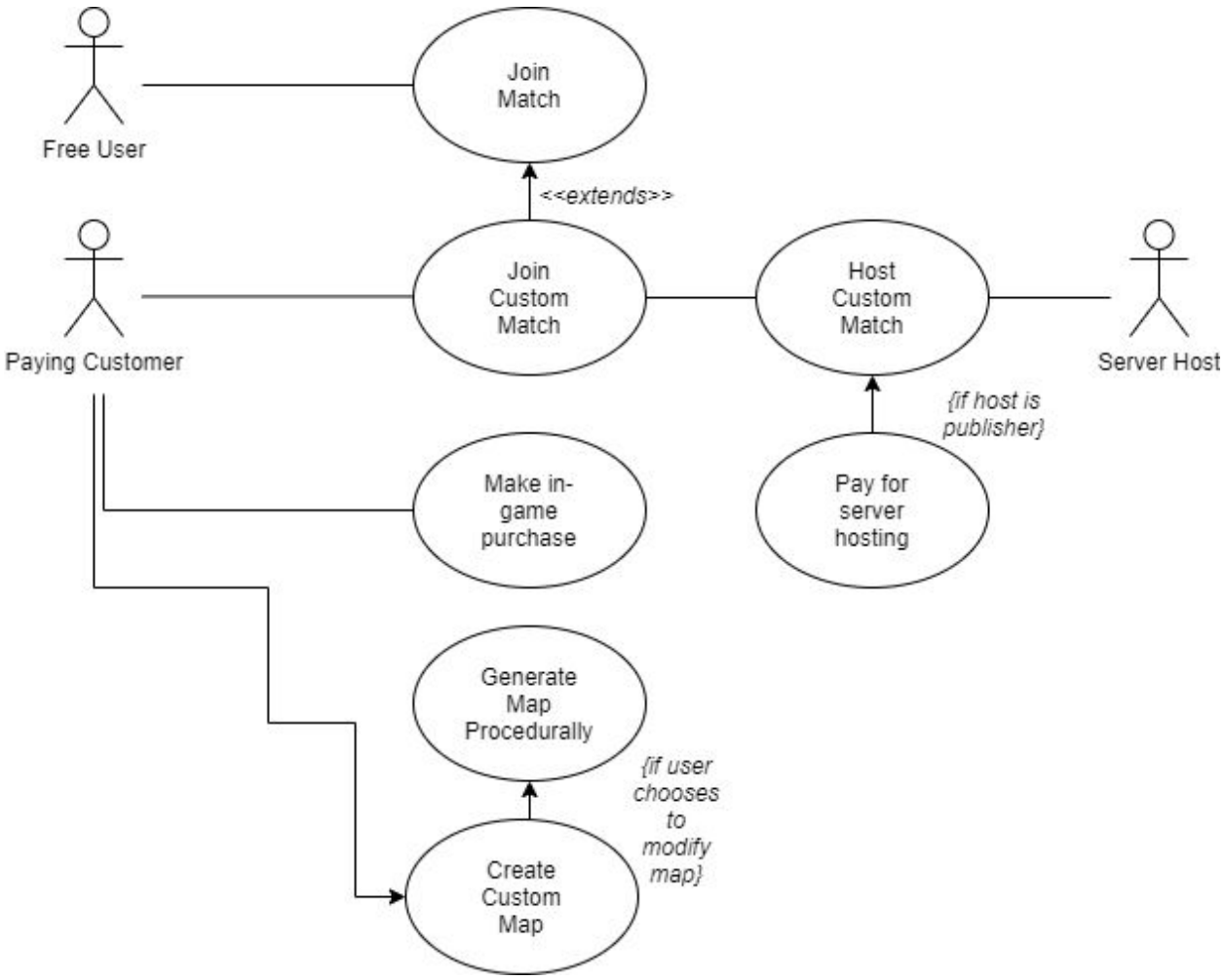
3a The Current Situation

Presently, the client has ported a standard version of Pac-Man to a wide variety of dedicated gaming consoles and mobile devices. All of these releases contain monetization schemes which either involve upfront payment for usage of the software, or are monetized via in game advertisements.

3b The Context of the Work



3c Work Partitioning



3d Competing Products

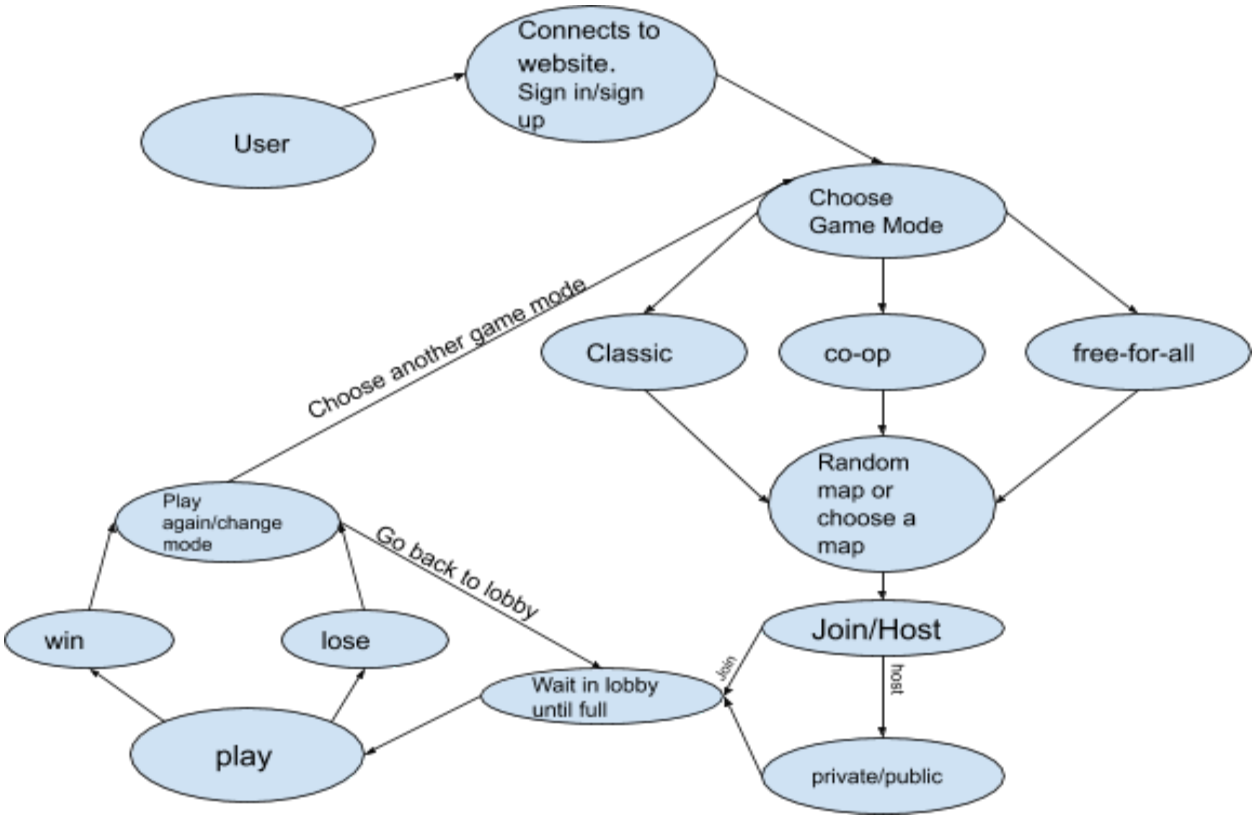
The game will be competing in the casual, free-to-play, market. Direct competition includes other Pac-Man style games, whereas indirect competition includes free to play mobile games in general. Our product is still useful as there is still room to innovate within the free-to-play market. By revilitazing a classic game we would be able to capture a market of players who would not otherwise be interested in a more serious/novel game.

4 The Scope of the Product

The product must handle all aspects pertaining to the product's gameplay. This means that the product must allow for the user to join a server, host their own custom server, browse community made maps, create their own maps, browse the leaderboards and share their creations with the community. The product will not cover the creation of

things outside of the scope of the game. This includes the creation of a payment platform, the infrastructure needed for server hosting, etc.

4a Scenario Diagram(s)



4b Product Scenario List

Scenario name	External Actors	Relevant information
Account creation	users, customers	user creation will be the same regardless of whether the user is a customer
Gameplay mode selection	users, customers	
Hosting/Joining game	host, publisher	The user can opt to host the game themselves, or pay for hosting from the publisher
Waiting in Lobby	users, customers	
Game Starts	users, customers	
Returning to lobby or start new game	users, customers	

4c Individual Product Scenarios

1. New users create signup through the website and returning players sign in.
2. Choose one of the many game modes to play (co-op, free for all)
3. Enter a seed to randomly generate a map for the match or choose from one of the many community made maps or one of your self made maps from map creator.
4. Choose to host or join a game. If hosting, choose to make it private for friends to join or public for random players. If joining a game, wait in the lobby.
5. Wait for the lobby to fill up to the required number of players and host to start.
6. Play game
7. Choose to remain in the current lobby or return to the main menu to try another game mode.

5 Stakeholders

5a The Client

The client would be a game company/publisher such as Bandai Namco, Koei Tecmo, Capcom, etc. Any company interested in revitalizing one of their existing IPs and applying modern gameplay mechanics and monetization schemes in order to capitalize on a growing market would be interested in our product.

5b The Customer

The customer of the product would be a user who is invested in the game to a point where they would be willing to spend money upgrading their experience. Upgrades

such as private servers hosted by the publisher, cosmetic upgrades like player skins, and the removal of in game advertisements would allow customers to differentiate themselves and enhance their experience.

5c Hands-On Users of the Product

Hands on users of the product would be both customers and people who play the game free of charge. In contrast to our paying customers, “free” users are less involved in the game and are more likely to be a casual gamer who plays in order to kill some time -- rather than to be involved competitively. This is relevant as customers interests may conflict with the interests of the average user.

5d Maintenance Users and Service Technicians

Since the game will run entirely in the browser, the game will require the user to do absolutely no installations/updates/etc whatsoever. If the player wishes, they will have the option to host their own private server or pay a subscription for hosting from the publisher.

5e Other Stakeholders

Beta-testers: This group will test out the game before it is fully released for the general public. They test out the game for still remaining glitches and bugs that still need to be addressed.

Marketing Manager: They are responsible for advertising the product. This game can only reach so many players by itself and from friends/family. Marketing Managers help spread it to other players from various backgrounds and create a diverse community.

5f User Participation

User participation is not required for development of the project, however, the developers may find it useful to provide polls for the community to determine which direction the game should head in in order to improve player retention and increase the quality of the product.

5g Priorities Assigned to Users

Paying customers are given much higher priority than the average user as they are the ones who are most invested in the game. Furthermore, they are the ones who are most likely to continue to spend money on the game . Whilst a free user may install the game and play it for a week, a premium user is likely to continue playing the game for a far longer period of time.

6 Mandated Constraints

6a Solution Constraints

The game must be color blind friendly in order to be accessible for color blind users. The game must not rely on auditory cues in order to accommodate users who are hard of hearing or are in an environment where they cannot hear audio cues (public transit)

6b Implementation Environment of the Current System

This game operates exclusively within the browser, meaning that it is inherently compatible with any device that is capable of running a web browser. Because of the web based nature of our product, developers are afforded the ability to easily create *native* versions of the product through the use of software like React Native, Electron, etc. In regards to hardware requirements, since the game will retain a retro aesthetic it won't require a powerful machine to run. All the user needs is a stable internet connection and a way to interface with the game.

6c Partner or Collaborative Applications

The product must be compatible with all major browsers. Specifically, it should be compatible with browsers that account for at least 80% of the world's internet users. This means no use of outdated software like Flash, and no cutting edge software like WebAssembly. The game should be written using a game engine that utilizes JavaScript and HTML5 -- to maximize compatibility.

6d Off-the-Shelf Software

No commercial off the shelf software is required for our product. The product will instead rely entirely on free, open source, software for auxiliary code outside of what is needed for the game itself. Relying on free software will allow development costs to stay down whilst still utilizing high quality libraries.

6e Anticipated Workplace Environment

The product is intended for a wide variety of environments, in accordance with the games philosophy of accessibility. Because of this, information must be conveyed to the user in a clear and concise way that doesn't necessarily require any sort of auditory feedback.

6f Schedule Constraints

The developers must focus on the most fundamental elements of the application before moving onto smaller issues. This means getting the foundations of the program functional before focusing on niceties. Graphical fidelity comes second to a functioning game and in game purchases come second to graphical fidelity. If the game is too broken, players will not be interested in the product. If the product is too frustrating to look at/use, players will not be interested in spending money on the

product. Because of this, our priorities must be gameplay, then aesthetics, then monetization.

6g Budget Constraints

The budget given will be relatively small due to several factors. Firstly, since the developers will not be required to write any auxiliary structural code, the development team will not need as many members. Secondly, the aim of this project is not to push graphical fidelity or story but instead create a fun game that appeals to a wide audience.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Pac-Rats: Equivalent to Pac-Man in a typical game of Pac-Man, only in *Pac-Rats* there will be multiple player controlled Pac-Men on a single board.

Foxes: Natural predators of Pac-Rats. They are the equivalent to ghosts in Pacman and work collectively to hunt down the Pac-Rats.

Free-for-all: A game mode in which every rat is pitted against one another. There are no foxes on the board, instead, Pac-Rats compete with one another.

Co-op: A game mode in which multiple Pac-Rats play alongside one another to achieve the highest score possible without any opponents getting in their way.

7b UML and Other Notation Used in This Document

The database(s) must be blueprinted thoroughly through the use of an ER-diagram. Use case diagrams must be employed to model both casual player interaction as well as dedicated user interaction.

7c Data Dictionary for Any Included Models

The project will use a relational database to store all of the data required for gameplay. The specific flavor of the database is down to whichever implementation the developers are most comfortable with, however, they may not use a non-relational database. Map recommendations will not be on a star system, but instead be either "liked" by the user or "disliked". This is because both Youtube and Netflix have discovered that almost every user rates their experience either 5 stars or 1 star.

8 Relevant Facts and Assumptions

8a Facts

- The game must comply with the privacy laws of every market the game is intended for. Most notably, the game must be GDPR compliant.
- The game must not infringe on any other publishers intellectual property. This means no stolen assets, copyrighted terms, gameplay systems, software, etc.

8b Assumptions

- 90+% of the players will not spend any money on in game purchases of any variety
- The player is familiar with how a standard game of Pac-Man
- The player has a stable internet connection

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

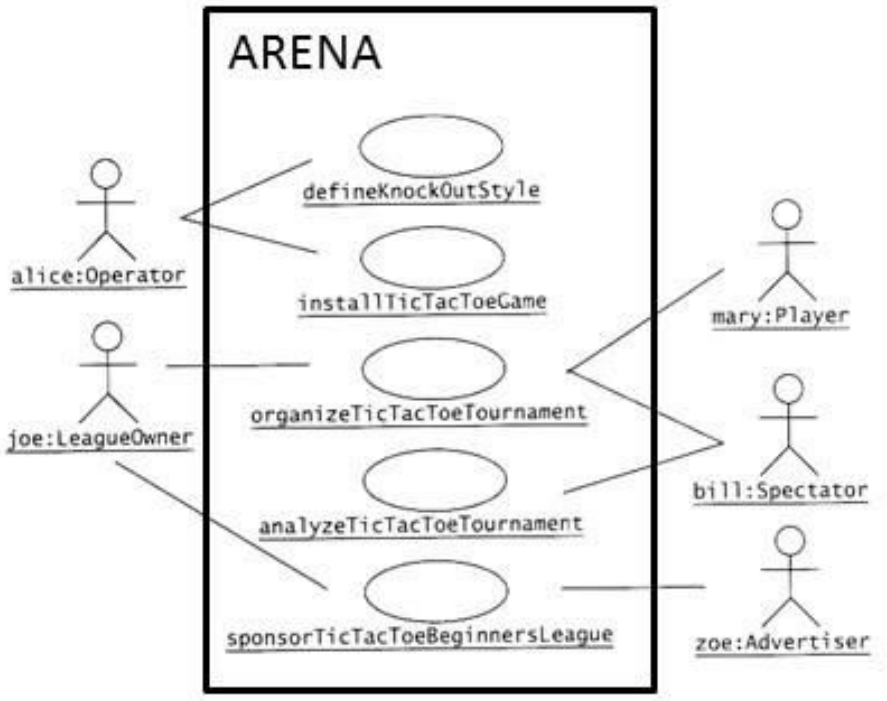


Figure 1 - Sample Use Case Diagram from Bruegge & DuToit (modified)

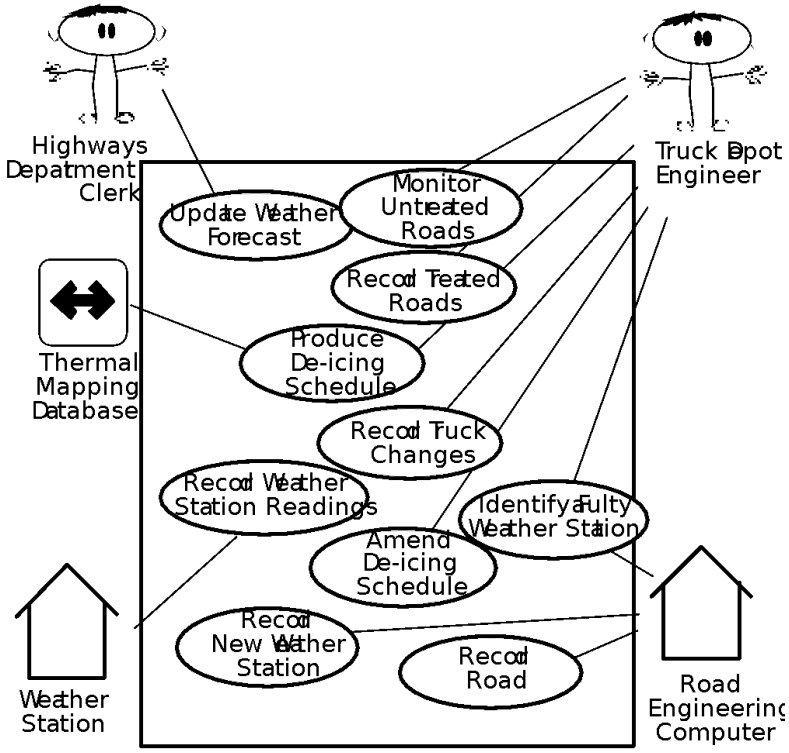


Figure 2 - Sample Use Case Diagram from Robertson and Robertson

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: 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.*

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.

	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

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 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 . . .

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. }