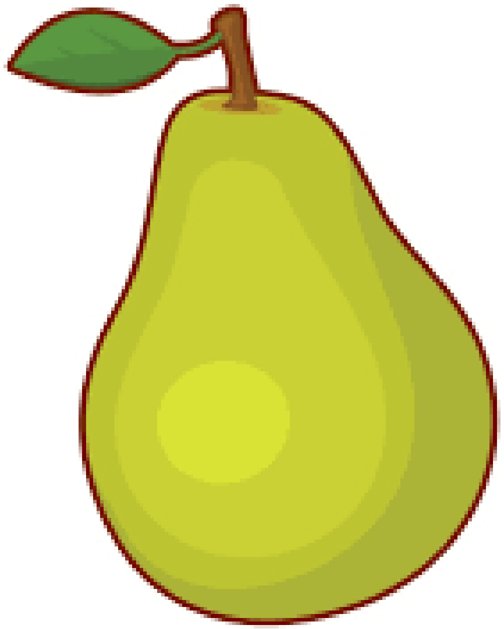
Fresh Food App



Prepared by Group 27:
Christian Gutierrez, Jordan Nguyen, Natalie Reyes, Wasay Ahmed

CS 440
University of Illinois Chicago
Spring 2022

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

The Fresh Food application will let the user know if food has gone bad by analyzing the chemical components of the odor released by food. Some chemical odors include methane (greenhouse gas given by rotting vegetation) or Ammonia (rotting fish and meat). The chemical makeup of the odor is read by external odor sensors and the data is sent to the app which will calculate edibility and estimate days til inedible depending on the type of food. This will remove the guesswork of deciding whether food has gone bad.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Food supply and vendor businesses like grocery stores display to customers to pick. Most of the food sold have sell-by and expiration dates that indicate the grocery store's limit of putting the food on display before getting thrown away. This is one of the main sources of food waste because the dates on these labels are not accurate. Restaurants buy food in large quantities to keep their business running and some of that food gets thrown out to prevent risks of customers getting sick. Food transporters receive the food from their sources and distribute it to the suppliers and vendors. The conditions of the environment the food is stored in, such as temperature and humidity, affects the rates of how quickly food goes bad. This lowers further the accuracy of the original expiration dates. Therefore shipping companies need specialized trucks to control their storage environment to keep the food fresh.

2b Goals of the Project

The goal of the Fresh Food app is to save money for ordinary consumers, restaurants, food suppliers, and food transporters by accurately determining if food is safe to eat by using gas/odor sensors. Over 108 billion pounds of food is wasted every year in the US which amounts to about $218 billion dollars. The app can also save people from food borne illnesses and reduce the number of related hospitalizations and people taking time off work to recover. The app would also have an impact on the environment by reducing the greenhouse gasses released by rotting food which contributes to about 6% of overall greenhouse emissions.

2c Measurement

Sensors for the detection of orders and gasses emitted from the food will be the source of all measurements of this project. Measurements such as methane and ammonia, depending on the type of food being monitored will be used to determine if the food is still edible and used to estimate the days until the food is inedible. The sensors will send their measurements by Bluetooth or Wi-Fi for analysis. Other sensors such as temperature and humidity sensors can also be factored in for the estimations.

3 The Scope of the Work

The work done by this app is to make accurate predictions of how long the food will be edible by analyzing the gas and odors emitted by the food or notify that the food is inedible. To do this work, the user manually sets up the sensors and makes sure that the measurements are being received successfully to the Fresh Food app. Then the user will see the estimates for the ediblity time left for the food.

3a The Current Situation

Currently there are no such widely used food freshness apps or devices. Most foods sold have either sell-by date or expiration date labels that are not entirely accurate. Food suppliers and vendors don’t want to risk selling expired food and throw out everything past those dates. This is one of the main issues that contributes to the massive amounts of food being wasted.

3b The Context of the Work

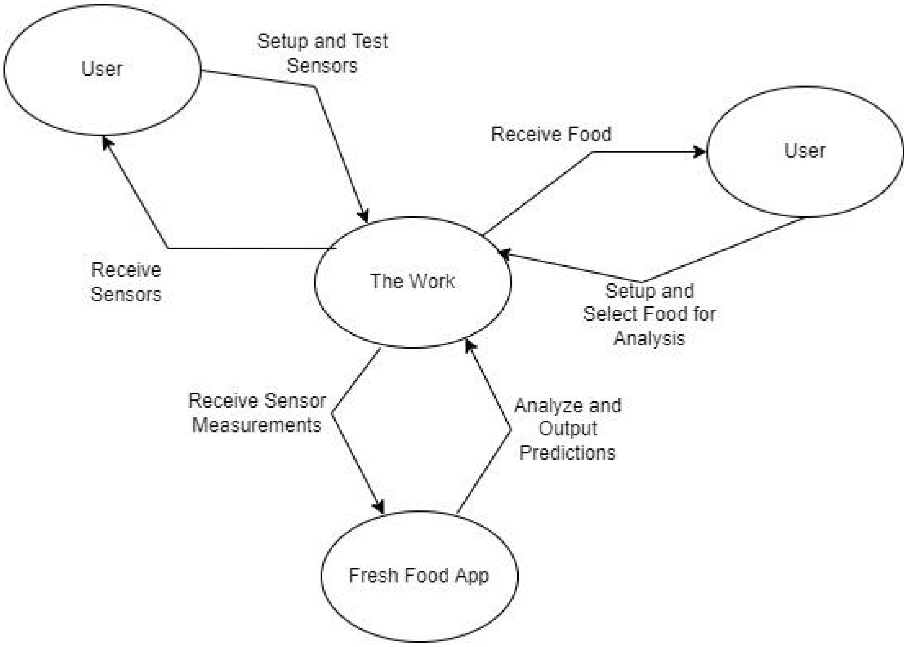

User
Setup and Test Sensors
Receive Food
User
Receive Sensors
The Work
Setup and Select Food for Analysis
Receive Sensor Measurements
Analyze and Output Predictions
Fresh Food App

Figure 1. *Work diagram of Fresh Food App.*

3c Work Partitioning

Table 1. *Work Partitioning table of Fresh Food App.*

Event Name	Input/Output
User Sensor Setup	Manually place sensors correctly for food gas/odor detection (Input) Test connection and data transfer with sensor and app (Input)
User Food Setup	Place food near sensors (input) Select food type for accurate calculations (input)
Sensors Measurements and Data Analysis	Measure gas/odor from the food (input) Analyze measurements and display estimated time left for food freshness (output)

3d Competing Products

Currently there is only one commercial product called FOODsniffer which is a handheld device that determines the freshness of meats. This device is limited to only meats. Therefore, the Fresh Food app has the advantage of measuring other types of foods and can be used for continuous monitoring anywhere the sensors are properly set up.

4 The Scope of the Product

- **Consumers** - This could be anyone who eats produce, meat, or dairy products. The Fresh Foods App would allow the consumer to check their produce, meat, or dairy products for any harmful chemicals and get an indication of whether it is okay to consume that food item or not. The consumer will also be able to view safety and expiration information regarding the product. Lastly, the consumer will be able to rate or review the place they got the product from.

- **Buyers** - This could be customers who shop for produce, meat, or dairy products. This could also be individuals or businesses who are buying produce, meat, or dairy products for their business. The Fresh Foods App will allow the buyer to check their produce, meat, or dairy products for any harmful chemicals and get an indication of whether it is okay to buy that food item or not. The buyer will also be able to view safety and expiration information regarding the product. Lastly, the buyer will be able to rate or review the food products for the place or company that the product is at.
- **Suppliers/Vendors** - This could be farms, companies, restaurants, business owners, markets, stores, employees, or anyone who supplies or sells fresh produce, meat, or dairy products. Suppliers and vendors can use the Fresh Foods App to check their produce, meat, or dairy products for any harmful chemicals and get an indication of whether it is okay to supply or sell that food item or not. The supplier and vendor will also be able to view safety and expiration information regarding the product. Lastly, the supplier or vendor can receive any ratings or reviews they have received from their shoppers, buyers, or consumers.
- **Shippers/Transporters** - This could be drivers, delivery employees, delivery companies, or anyone who transports goods from the supplier to their destination. This can also be the supplier if the supplier chooses to deliver their own products. Shippers/transporters can check to ensure the products did not go rotten during their time of being transported to their destination. The shipper/transport can communicate or inform the supplier and vendor of how the supply is doing.
- **System** - The system for the Fresh Foods App can process results from testing the food. The system can also use the internet (Google) to help present information regarding the results, expiration, and safety to the user.

4a Scenario Diagram(s)

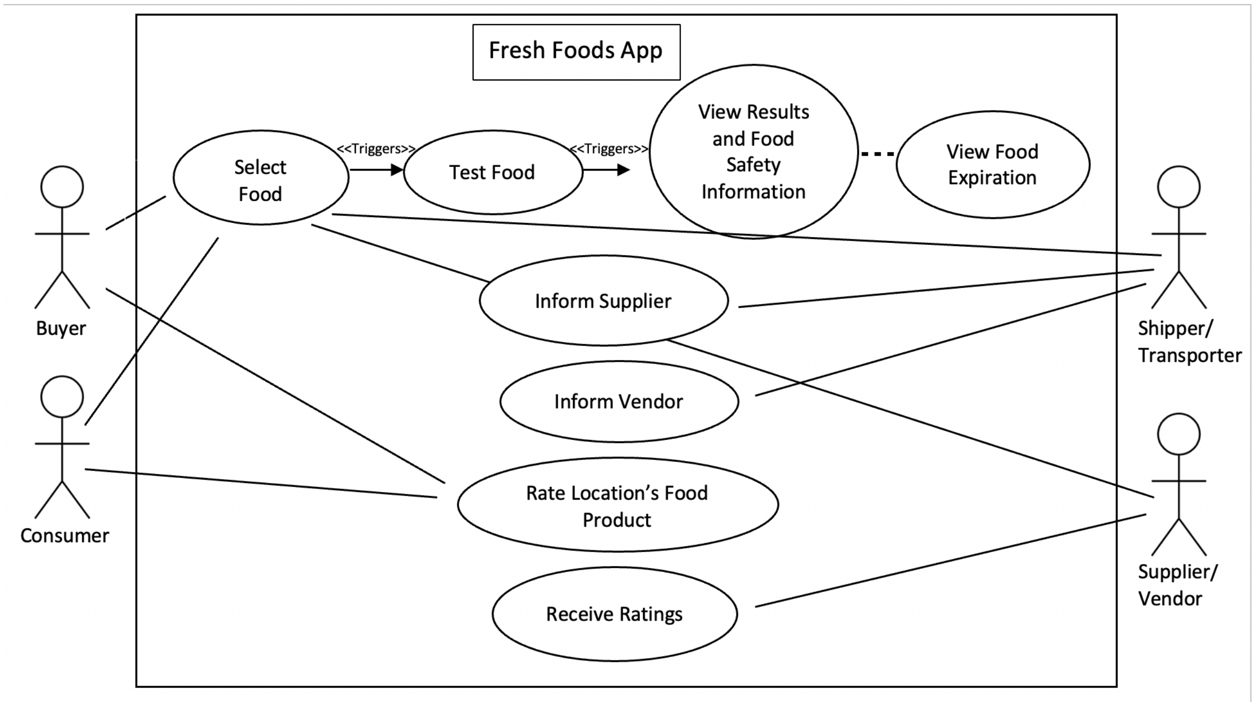


Figure 2. *Scenario Diagram of Fresh Food App.*

4b Product Scenario List

Table 2. *Product Scenario List of Fresh Food App.*

Scenario Name	External Actors
Select Food	A database of produce, meat, or dairy items can be used to identify the specific food item selected
Test Food	Sensor (separate hardware through bluetooth connection) will be used to collect data from food
View Results and Food Safety Information	Consumer, buyer, shipper/transporter, supplier/vendor can view results, and safety information regarding the food gathered from Google
View Food Expiration	Consumer, buyer, shipper/transporter, supplier/vendor can view the expiration date of the food determined by the

	amount of harmful chemicals detected by the sensor and expiration data on the specific food item from Google
Inform Supplier	Shipper/Transporter can separately communicate that the food product is rotten
Inform Vendor	Shipper/Transporter can separately communicate that the food product is rotten and can't be sold
Rate Location's Food Product	Consumer and buyer can separately rate or review the quality of the foods products from where they are seeing them or buying them from
Receive Ratings	Suppliers and vendors can receive ratings or reviews from people who have considered buying their food products or have bought them

4c Individual Product Scenarios

All users of the Fresh Foods App can test produce, meat, or dairy products to check whether they are okay to eat, view results, food safety information, and food expiration information. Buyers of produce, meat, or dairy products can use the Fresh Foods App to ensure that they are buying good quality and safe products. If they find a product is not safe to eat according to the sensor, then they can leave a negative rating or review to that vendor or supplier. If the buyer finds the products are safe and with good quality, they can leave a positive rating or review to that vendor or supplier. Consumers who are about to eat a food product can test the product before eating. This can be useful especially at restaurants to help prevent food poisoning. The consumer can leave a rating or review regarding the quality of the product from the location it's sold. Suppliers of food products can use the Fresh Foods App to ensure they are supplying good quality food products. Suppliers can also receive ratings from buyers or consumers to get further feedback on their products. Vendors can use the Fresh Foods App to ensure they are not selling harmful foods to anyone. They can receive ratings from consumers or buyers for further feedback on their products they sold. Shippers/Transporters can test the food products before and after shipping. They can inform the supplier and vendor if there is anything wrong with the food product.

5 Stakeholders

5a The Client

The clients of the Fresh Foods App will be the everyday consumer that shops for fresh produce, meat, and dairy products. The application will be used by the user to determine how fresh the products they have purchased are in the comfort of their own home. Other clients of the application could include vendors or suppliers of the food that is to be sold such as grocery stores , as well as the producers of the food. to ensure quality products. Instead of having to rely on taste , smell, and how the products appear visually we can reduce the risk by using an attachment near or directly inside product containers to detect and monitor the presence of harmful gasses that can come from food that has expired.

5b The Customer

Users of this application will have the responsibility of picking out which product they would like to purchase as inputting the correct settings for said chosen product. The user will be able to choose from a variety of food types and categories from dairy, products, meat, fruits, and vegetables. The customer can learn and read what the detection results mean and whether or not the food is edible or what stage the food product is in.

5c Hands-On Users of the Product

The clients and customers will be the hands-on-users for the app. They will use the application to test individual food items and after the testing is complete a log will be made to indicate when the product will need to be discarded using a calendar. The pages will be defined by the UI designer so that customers can navigate through and complete the action of testing.

5d Maintenance Users and Service Technicians

Our application will be maintained by the developers who will be in charge of testing the performance of the app and ensuring it is accurately updated from time to time. Similar to the maintenance the developers are responsible for, engineers will perform tasks to the database to make sure it runs smoothly. Both will need to continue checking and updating the product for the whole duration of the product's life.

5e Other Stakeholders

Other stakeholders will include legal experts to make sure everything runs smoothly if anything occurs. Another stakeholder would be a marketing team who can assist with making the product attractive to potential customers and users. A financial advisor will also most likely be used to figure out how to manage the money made from the application and budget accordingly.

5f User Participation

User participation will occur once the app has been established and the functional aspects have been implemented.

An example would be once the database of various foods have been inputted and data can be accurately read from the gasses emitted then user testing can be done. Once these steps have been completed and the interface and database is ready to go then the user can give feedback on how the app is designed.

5g Priorities Assigned to Users

The key users will be the at home users. The people who purchase fresh produce on a regular basis. It's very important to establish a good relationship with our main users that way we can eventually transition to more mainstream use.

Other users can include the vendors who sell the items such as a grocery store. Once the application has had some marketing and the main users are satisfied then there will be a smoother transition from at home use to everyday use for stores to use.

Users that are not as important as the product producers. More than likely it will be difficult for a fast paced environment with producing so many products that they already have their own maintenance checks for bigger amounts of inventory.

6 Mandated Constraints

6a Solution Constraints

1.) The app will need to keep an updated database to be able to accept different kinds of products within the app. Each kind of food will have its own section and will be listed in alphabetical order.

The best way to sort out each kind of food is to alphabetize them in each of their respective categories to make them easily accessible.

After choosing the food item you can then test for gasses such as:

- Methane (greenhouse gas given by rotting vegetation)
- Hydrogen sulfide (emitted from dead plant and animal matter)
- Noxious solanine gas (produced by rotting potatoes, dangerous if inhaled in large amounts)
- Ammonia (rotting fish and meat)

2.) The Fresh Foods App will be available on android and iOS

Making the app downloadable by a wider variety of people who have different devices.

In order to be downloadable it will first need to be evaluated so it is available in the app store for both apple and android products.

6b Implementation Environment of the Current System

The app will be on the Android and iOS systems. The sensors can be connected to Arduinos or other microcontrollers that communicate with the app via Bluetooth and WiFi.

6c Partner or Collaborative Applications

- Database: access different foods for testing
- Cloud: access safety tips for food storage

6d Off-the-Shelf Software

- Sharing API
    - Connect and share data to google calendar or with others when getting gas detection results.
- Smartphone
    - Needed by the client in order to have access to the application.
- Login API
    - Will allow clients to easily login using gmail accounts.

6e Anticipated Workplace Environment

The product is developed to be easily accessible due to the versatility when it comes to users of the product. This application can be used at the clients home and even accessible when grocery shopping at the grocery store.

6f Schedule Constraints

In order to distribute the application we must have consumers who are able and ready to utilize it and hopefully implement it into their routine. Whether the routine be shopping for food at a grocery store, the grocery store employees maintaining and checking their own products, or the distributor of the food.

6g Budget Constraints

The product itself will be an application that will not require a large amount of funds besides changing and updating the database, maintaining the UI, and other expenses related to the development or infrastructure of the application.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Buyer: An individual, shopper, or business that is planning on purchasing or has purchased produce, meat, or dairy product(s).

Consumer: An individual who is about to eat or has eaten produce, meat, or dairy product(s).

Shipper/Transporter: An individual, or employee of a business who ships/transports produce, meat, or dairy products from the supplier to the intended destination.

Vendor: Farms, companies, restaurants, business owners, markets, stores, employees, or anyone who sells fresh produce, meat, or dairy products.

Supplier: Farms, companies, restaurants, business owners, markets, stores, employees, or anyone who supplies fresh produce, meat, or dairy products to a buyer.

7b UML and Other Notation Used in This Document

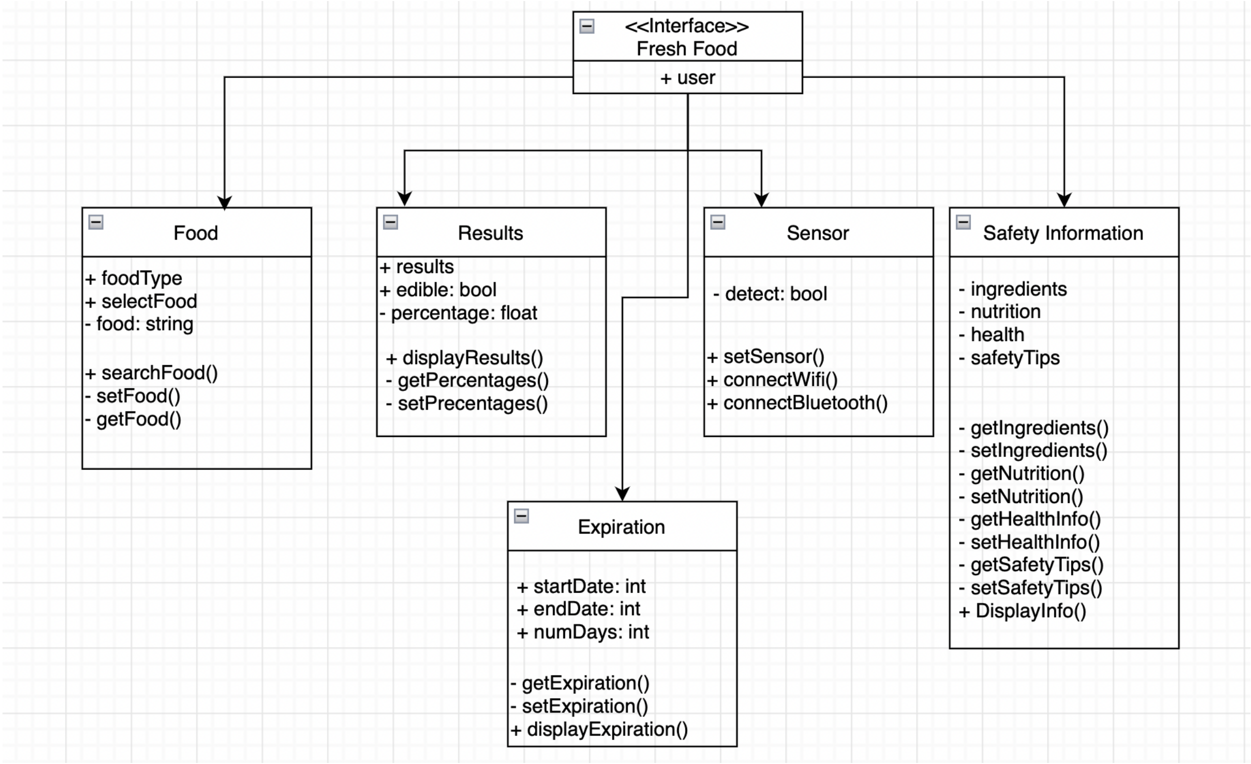


Figure 3. *UML diagram of Fresh Food App.*

7c Data Dictionary for Any Included Models

Select Food:

- Food type
- Specific food name
- Search
- Select

Test Food:

- Sensor detection
- Sensor connection to bluetooth
- Sensor connection to wifi
- Set up sensor

View Results:

- Percentages on gasses and chemicals emitted from food
- Mention whether safe to eat or not

View Safety Information:

- Nutrition information
- Ingredients
- Health information regarding food
- Safety tips on that particular food item

Food Expiration:

- Expiration date
- Number of days until safe to eat

Buyer:

- Identifier: company, business owner, customer, individual

Consumer:

- Identifier: customer, individual

Shipper/Transporter:

- Identifier: employee, company

Vendor:

- Identifier: seller, company, employee, owner, business

Supplier:

- Identifier: company, employee, individual, owner, business

8 Relevant Facts and Assumptions

8a Facts

Every year, an estimated 1 in 6 Americans (or 48 million people) get sick, 128,000 are hospitalized, and 3,000 die from foodborne diseases.

Each year, 108 billion pounds of food is wasted in the United States. That equates to 130 billion meals and more than $408 billion in food thrown away each year. EPA estimates $218 billion dollars wasted.

Food wastage is responsible for around 6% of total global greenhouse gas emissions.

The share of Americans that own a smartphone is now 85%(2021)

https://www.cdc.gov/foodsafety/food-poisoning.html

https://www.feedingamerica.org/our-work/our-approach/reduce-food-waste

https://ourworldindata.org/food-waste-emissions

https://www.pewresearch.org/internet/fact-sheet/mobile/

https://www.epa.gov/international-cooperation/international-efforts-wasted-food-recovery#:~:text=In%20the%20United%20States%2C%2040,emissions%20in%20the%20United%20States.

8b Assumptions

We assume that the users can obtain the odor sensors and set them up correctly to get accurate samples.

We assume that the amount of specific odors can correlate to edibility.

We assume that all food that rots will emit a traceable odor. If not, we can list specific foods for the user that cannot be checked by the app.

We assume odor sensors are affordable for any user.

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

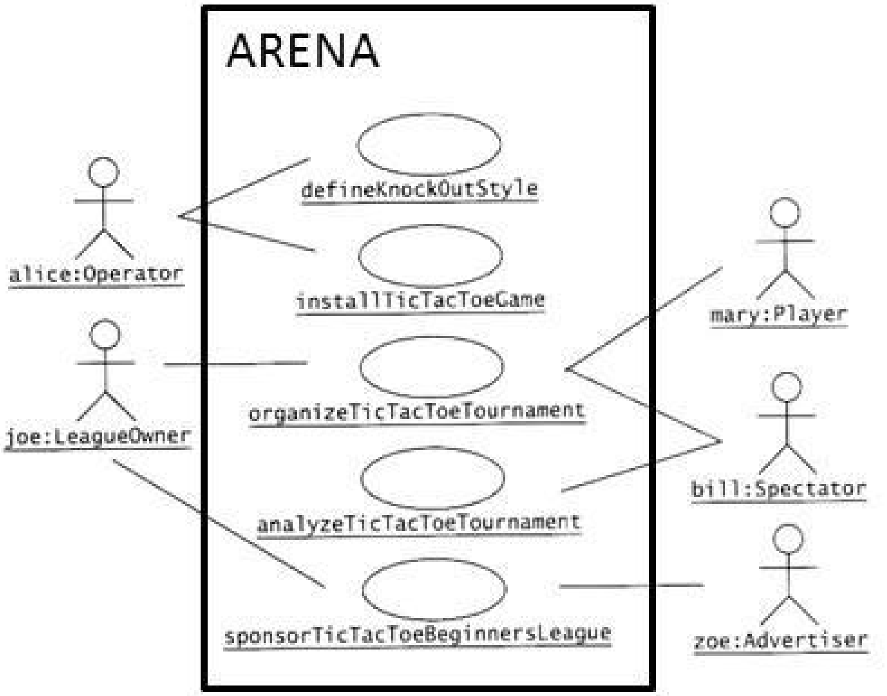


Figure 1 - Sample Use Case Diagram from Bruegge & DuToit (modified)

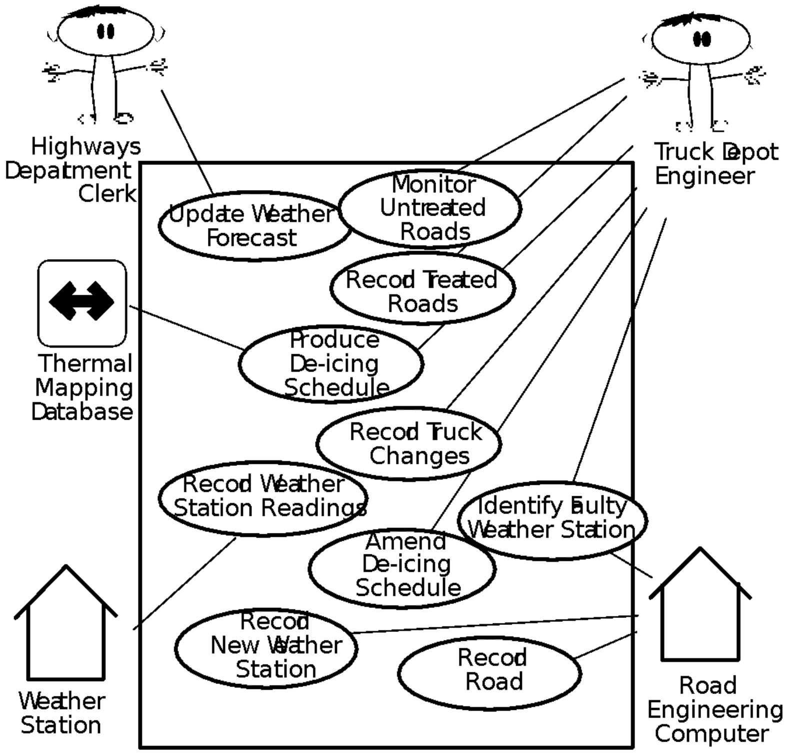


Figure 2 - Sample Use Case Diagram from Robertson and Robertson

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.

<u>ID# - Name</u>

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.

<u>ID# - Name</u>

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.

<u>ID# - Name</u>

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. }