Software Requirements Specification

for

Deluxe Menu Ordering System

Version 1.0 approved

Prepared by Manning Smith, Michael Lin, Vincent Ha, Rich Berger

TGG Inc.

November 12, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Vincent Ha	11/15/20	First Revision	1.0
Rich Berger	11/15/20	Adding to Sections	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to display a thorough description of the Deluxe Menu Ordering System. It is meant for stakeholders, potential investors, or people who are interested in our design. The document will showcase the features, design and interfaces used within our system.

1.2 Document Conventions

The Deluxe Menu Ordering System will be utilized by restaurants, retail stores, or online shops where the customer needs to place an order. The system is meant to increase productivity of the establishment as well as help keep the ordering process easy and simple for the customer.

This system will not only affect the customer's user experience it will affect the worker's as well. The Deluxe Menu Ordering System will alert customers of items that are out of stock as well as pass their orders on to the business seamlessly. This will create an automated environment where the establishment has more time to worry about improving their product instead of interacting heavily with customers.

1.3 Intended Audience and Reading Suggestions

This document is intended for investors as well as other people who are interested in our software. We will use this document to explain our system as well as the benefits of using it. By reading this document, we hope people will gain a better understanding of our goals as a design team and understand the convenience of using our software.

1.4 Product Scope

The Deluxe Menu Ordering System is a program that will help automate and simplify the ordering process for both the customer and the business. By creating a seamless ordering system we hope to bring more business to the establishments who use our product, boost workplace productivity, and offer a system to businesses that do not have an ordering system in place. In the future we wish to pitch this software to investors in order to improve and expand upon it. If we were

to receive more funds, we would look into developing additional features like a pre-ordering or scheduled delivery system

2. Overall Description

2.1 System Environment

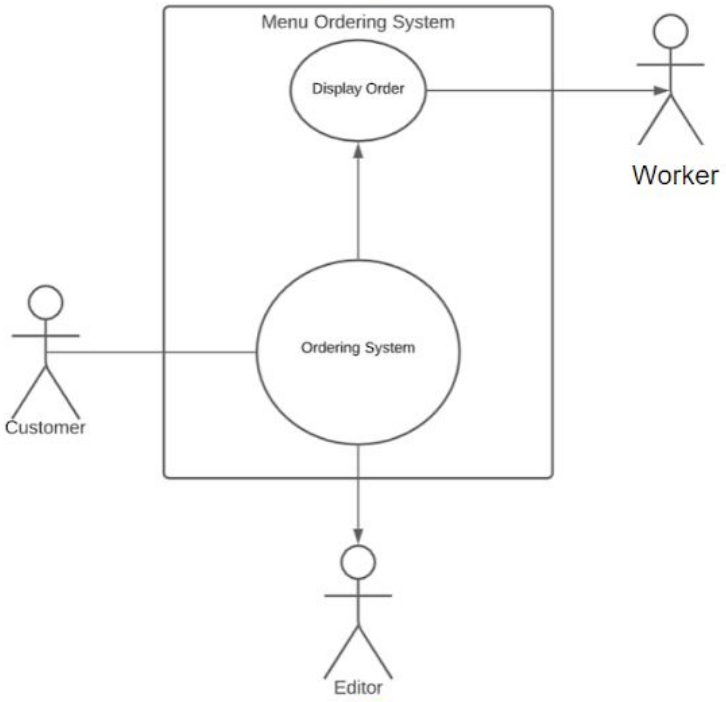


Figure 1 - System Environment

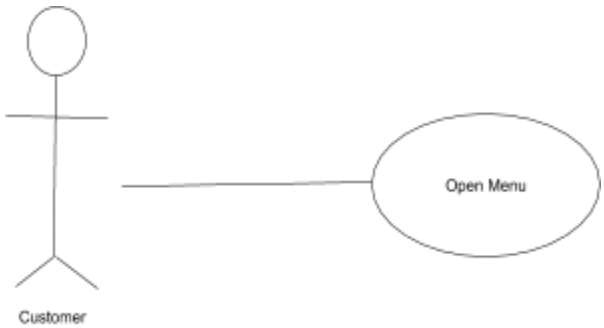
In the Menu Ordering System, there will be three active users all using a single cooperating system. These users are the customer, the editor and the chef. The customer will access the system through an electronic device and make a couple of choices. The editor has complete access to the system and is able to make changes accordingly or maintain the functionality of the system. The chef will take the information input by the customer and make their actions accordingly as well.

2.2 Product Functions

This section of the document describes the use cases for the various people who interact with the Deluxe Menu Ordering System. These people include the customer, the editor, and the worker.

2.2.1 Customer Use Case

Use case: **Open Menu**
Diagram:

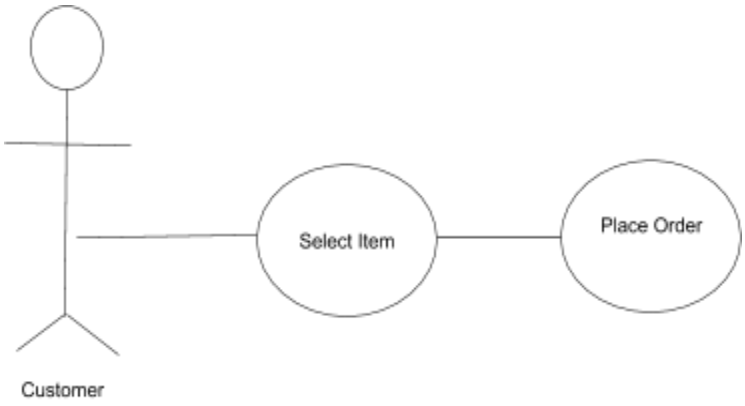


Brief Description
In a store, the Customer accesses the program, by clicking on the icon to open the menu. If they are on a personal device, they must download the app before opening it.

Initial Step-By-Step Description
Before this use case can be initiated, the Reader must have access to the program through an app on a personal device, or through a computer at a store.

1. The Customer opens their device
2. The Customer selects the application or goes to the appropriate website where they have access to the program
3. The Customer starts the program by selecting the icon
4. The system displays a menu of the current items in stock.

Use case: **Select Item & Place Order**
Diagram:



Brief Description
The Customer will select an image of the item they want, move it to the cart, and place their order.

Initial Step-By-Step Description
Before this use case can be initiated, the Customer must have opened the menu.

1. The Customer selects the item they want
2. The menu displays a description of the item
3. The menu asks if you would like to add it to your cart
4. If answered yes, The items are added to the cart
    a. if no, returns to start screen
5. Customer selects check out
6. Customer pays for order
7. The order is placed within the system

2.2.2 Editor Use Cases

The editor has the following use cases:

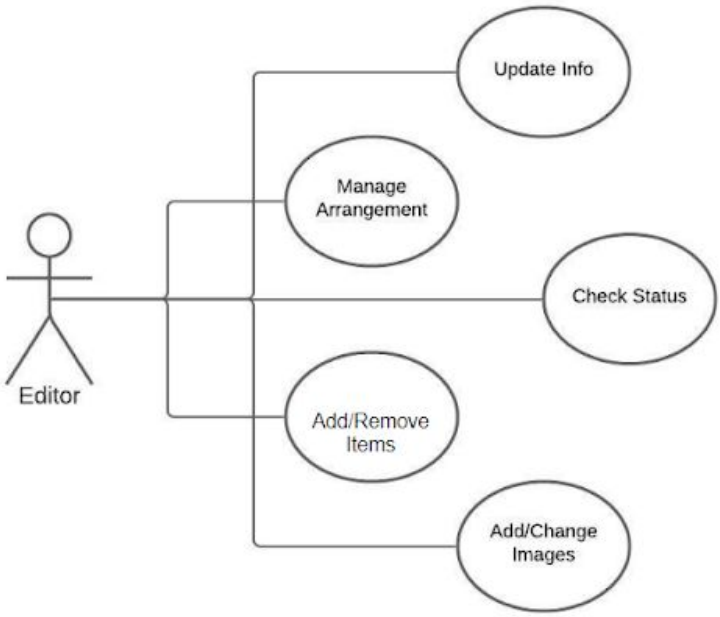
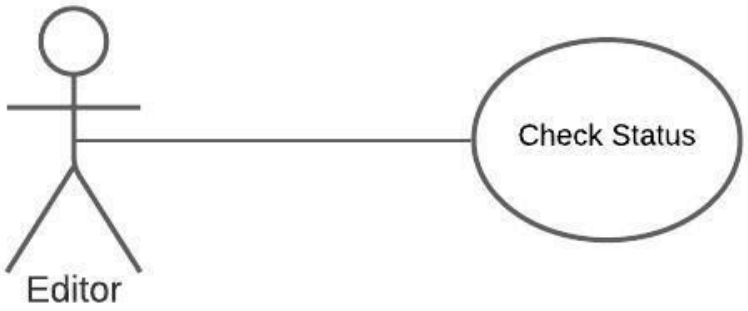


Figure 2 - Use Cases for Editor

Use case: **Check Status Use Case**
Diagram:

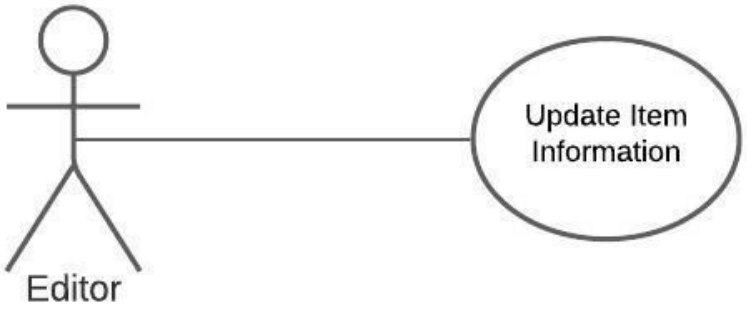


Brief Description
The Editor checks the status and functionality of the system

Initial Step-By-Step Description
For this case to occur, the Editor is already on the Menu Manager Page

1. Editor Selects to *Check Status*
2. System returns a list with all items with their description and image, also shows if system is up and running for the rest of customers
3. System returns the Editor to the Menu Manager Page

Use case: **Update Item Information Use Case**
Diagram:

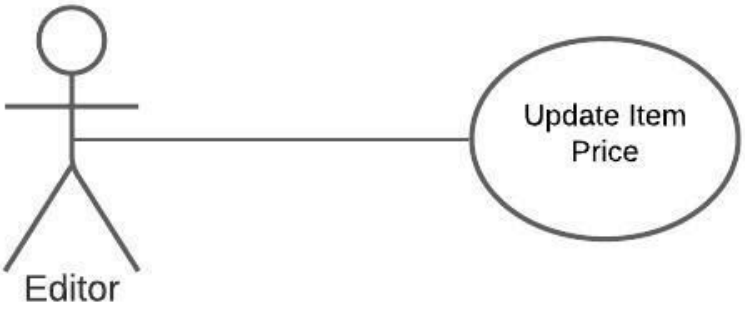


Brief Description
The editor changes the description of an item

Initial Step-By-Step Description

1. The Editor selects to *Update Item Information.*
2. The system prompts the Editor to select which item the Editor wants to make changes upon.
3. The Editor changes the text and description of the selected item and saves the changes to the system.
4. The system sends the Editor back to the Menu Manager Page

Use case: **Update Price**
Diagram:

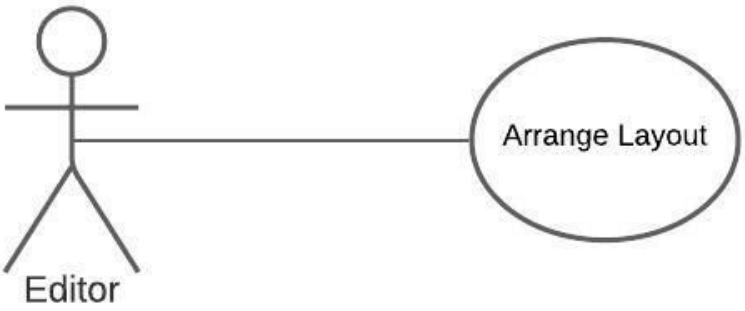


Brief Description
The editor changes the price of an item

Initial Step-By-Step Description

1. The Editor selects to *Update Item Price*
2. The system prompts the Editor to select which item the Editor wants to make changes upon.
3. The Editor changes the price of the selected item and confirms the changes to the system.
4. The system sends the Editor back to the Menu Manager Page

Use case: **Arrange layout**
Diagram:



Brief Description
The Editor changes the aesthetics of the menu

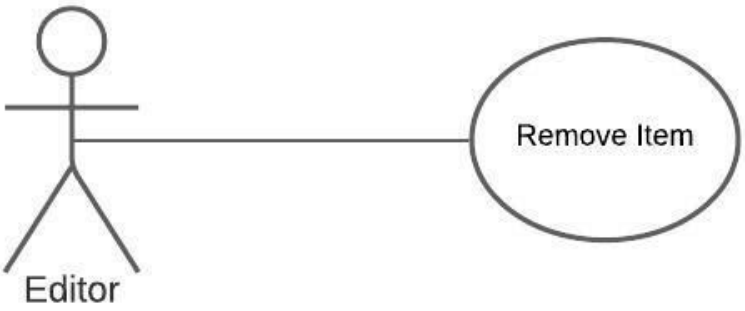
Initial Step-By-Step Description

1. The Editor selects the *Arrange Layout*

2. The system brings the Editor to a page where the Editor is able to make changes to the display's color, how the information is arranged, how much information is going to be displayed at once, etc.
3. The Editor makes the desired changes to the front end
4. The system sends the Editor back to the Menu Manager Page

Removal/Adding Use Cases

Use case: **Remove item**
Diagram:

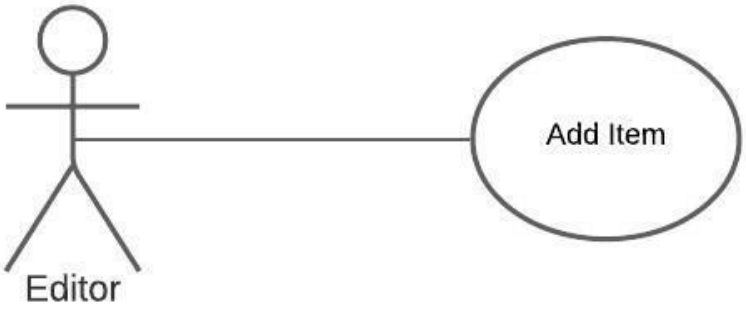


Brief Description
The Editor removes an item from the menu entirely

Initial Step-By-Step Description

1. The Editor selects the *Remove Item*
2. The system lists all the items that are currently in the system and allows the Editor to select items
3. The Editor selects the items to be removed
4. System will show the Editor how the menu will appear if the changes go through, if the Editor approves, the system will also show the Editor the option to *Arrange Layout.*
5. The system sends the Editor back to the Menu Manager Page

Use case: **Add item**
Diagram:

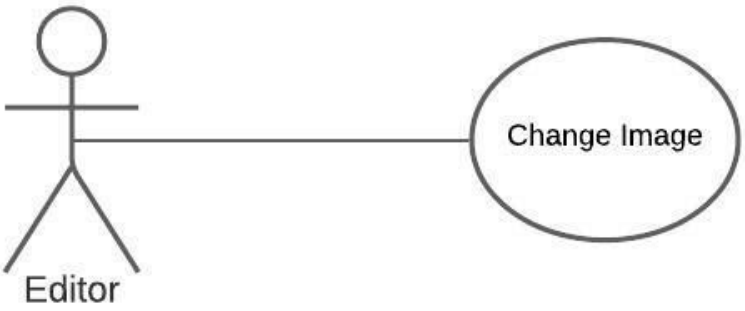


Brief Description
The Editor adds an item to the menu

Initial Step-By-Step Description

1. The Editor selects the *Add Item*
2. The system lists all the items that are currently in the system
3. The Editor selects or enters in the items to be added
4. Prompts the Editor to upload an image alongside the information
5. System will show the Editor how the menu will appear if the changes go through, if the Editor approves, the system will also show the Editor the option to *Arrange Layout.*
6. The system sends the Editor back to the Menu Manager Page

Use Case:**Update Images Use Case**



Brief Description
The Editor changes an image of an item

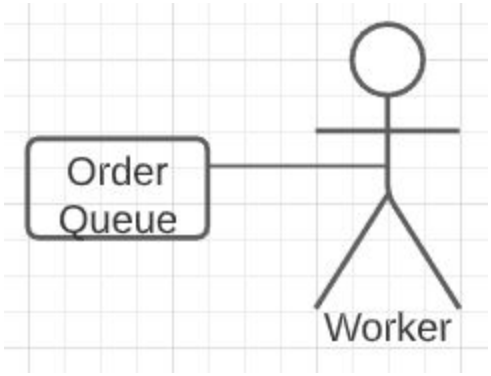
Initial Step-By-Step Description

1. The Editor selects the *Change Image.*

2. The system lists all the items that are currently in the system and allows the Editor to select items.
3. The Editor uploads new image for item
4. System will show the Editor how the menu will appear if the changes go through, if the Editor approves, the system will also show the Editor the option to *Arrange Layout.*
5. The system sends the Editor back to the Menu Man

2.2.3 Worker Use Cases

Use Case: **Bump Order Queue**
Diagram:



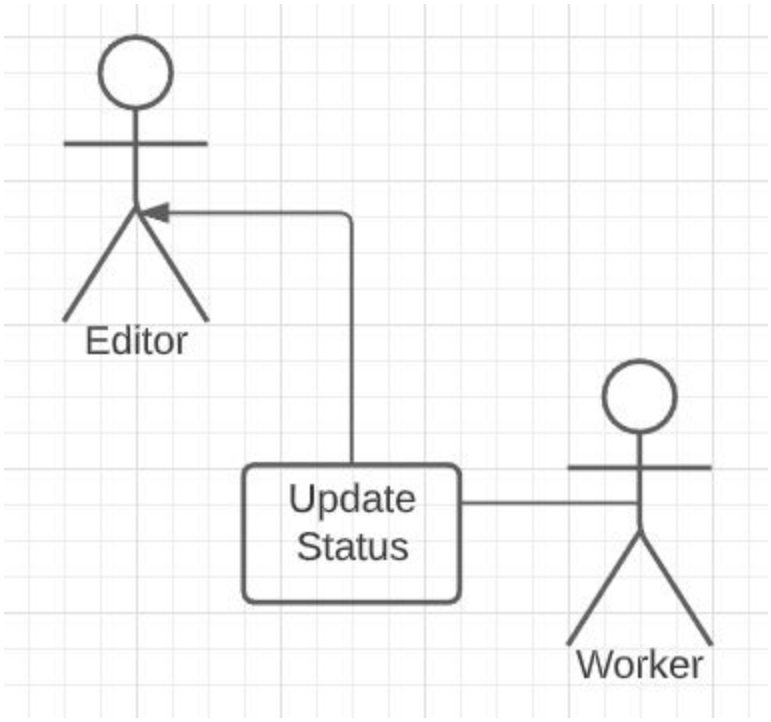
Brief Description

The Worker accesses the list of active orders. Selects the order they are working on. Once completed, they mark order as complete.

Initial Step-By-Step Description

1. The order queue automatically selects the oldest order.
2. The Worker selects this order.
3. The Worker completes the order.
4. The Worker marks the order as complete.
5. The order queue updates and removes completed orders from the queue.
6. The order queue bumps existing orders.

Use case: **Worker Check Status**
Diagram:

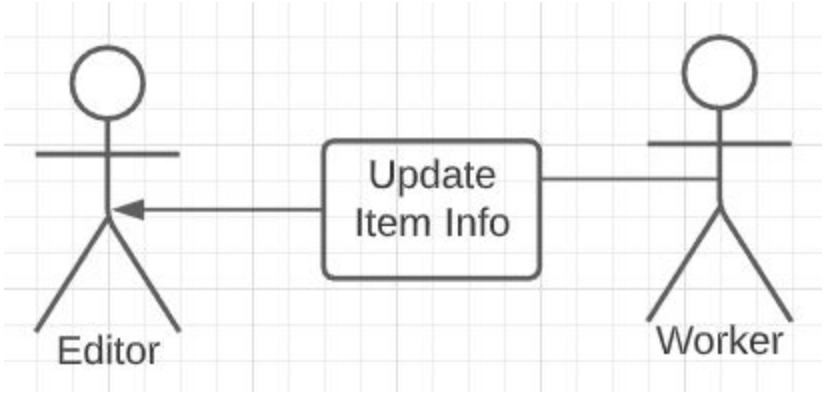


Brief Description
The Worker updates the order whenever they are done to let the Editor know that

Initial Step-By-Step Description
Before this use case can be initiated, there must be an existing order in place.

1. The Worker finishes order
2. The Worker toggles the order status from unfinished/finished
3. The Editor reads it

Use case: **Worker Update Item Info**
Diagram:



Brief Description

The Worker can change the availability of items on the menu and then Update Item Info can be viewed by the Editor

Initial Step-By-Step Description
Before this use case can be initiated, there must be existing items on the menu and is toggled when a change in availability happens

1. Item in menu becomes available/unavailable
2. The Worker toggles it from available to unavailable or unavailable to available
3. The Editor reads it and is informed of the change

2.3 User Classes and Characteristics

The customer should be technologically literate in order to use the Deluxe Menu Ordering System. If the customer is using the software remotely, they must have access to the internet as well as knowledge on how to surf the web. If they are within a business who uses our program, they do not have to worry about a stable internet connection. Our system should be usable by all customers no matter their age or literacy. We will utilize pictures as well as a text to speech option in order to appeal to all audiences. The only people that may have trouble using the Deluxe Menu Ordering System are those who are both visually and audibly impaired.

Similar to the customer, the Editor must also be technologically literate. They should also be able to interact with our interface and communicate with the workers at the business.

The Workers must be able to read the order to determine what the customer wants. In order to do so, they should be able to interact with our system in order to pull out the necessary information.

2.4 Operating Environment

The software should be able to run on most smartphones, tablets, laptops,PCs and most other electronic devices. It would also be on large touch screens within the stores so it is more convenient for the customer, whether not having to wait at a large line or not having to make the customer navigate the whole facility to find that one object that he or she wants to purchase. Even though the project is mainly graphically based, the devices do not need high specs since the software will not be as intense as other software out on the market. For example, an iPhone 4 would be able to run the software pretty easily.

2.5 Design and Implementation Constraints (Optional)

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

The developers may face limitations when it comes to connecting to our servers. If the user does not have stable internet connection or some other safe way of utilizing the internet, they may run into problems. Security Considerations may also be an issue if our data storage gets hacked. Our program will store Customer data such as name, order, and possibly payment method. If this information was to be tampered with, numerous problems could arrive.

2.6 User Documentation

There will be an online manual accessible to the customers in order to assist them if there are any problems. Our design team would also like to create a mobile helpline for any problems that our users may experience.

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

Our team assumes that the customer's internet service provider is reliable and that their device will be able to run our software. If they are in an area without any internet connection, we will not be able to take their order. We also assume that the customer will understand our software and be able to use it easily. If they are having problems, we will offer methods to guide and assist them. The customer must also be at a technologically literate age where they are able to pay for their items.

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces (Optional)

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by ***system features, the major services provided by the product.*** *You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

[Note- Students of CS275 Will most likely organize by bolded topics]

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software should be able to run on most smartphones, tablets, laptops,PCs and most other electronic devices. It would also be on large touch screens within the stores so it is more convenient for the customer, whether not having to wait at a large line or not having to make the customer navigate the whole facility to find that one object that he or she wants to purchase. Even though the project is mainly graphically based, the devices do not need high specs since the software will not be as intense as other software out on the market. An iPhone 4 would be able to run the software pretty easily for a comparison

(More needs to be added)

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>