**SOFTWARE REQUIREMENTS SPECIFICATION**

**for**

***The Life Is Great Eatery***

**Version 1.0**

**Prepared By:**

Chiara DeMelfi

Felix Thomas

Jeremy Cruz

Ashlyn Welch

Lauren Turet

**The Snazzy Software Squad**

**10/23/17**

# Table of Contents

[Table of Contents](#_ij7fhxrg95r4)

[1.0 Introduction](#_90670tspppfs)

[1.1 Purpose](#_7essp0jemxm0)

[1.2 Document Conventions](#_e94887j5anof)

[1.3 Intended Audience and Reading Suggestions](#_f1jcedr548jr)

[1.4 Product Scope](#_65umeeay3pzv)

[2.0 Overall Description](#_odgh81rl1uir)

[2.1 Product Perspective](#_yxteoayld97o)

[2.2 Product Functions](#_crc8g8cj8enk)

[2.3 User Cases and Characteristics](#_syyygb6p4p04)

[2.3.1 Customer Use Case](#_3tf4pmkmkrkj)

[Use case: Choose Dining](#_rp1pu1j7mkdn)

[Use case: Pick Table](#_sm9o9fqua2lw)

[Use case: Order Food](#_kbvib39rsryz)

[Use case: Order Drink](#_5q4k5rl31nwd)

[Use case: Order Dessert](#_jhhx8u660dmu)

[Use case: Order Side](#_aut8ez56ytdv)

[Use case: Get Check](#_cin4eowlqiaw)

[Use case: Cook Food](#_clmejcwpw4cq)

[Use case: Inventory](#_gi0hz84hliei)

[Use case: Deliver Food](#_ezh4hleg51pz)

[Use case: Clear Table](#_ehdnat6ump6y)

[2.4 Operating Environment](#_nltopctjilv9)

[2.5 Design and Implementation constraints](#_53tkc79sr02o)

[2.6 User Documentation](#_gqx62te5bz1s)

[2.7 Assumptions and Dependencies](#_5zpgwotz1pb2)

[3.0 Requirements Specification](#_50awsurcpxlg)

[3.1 User Interfaces](#_ama52t54v92u)

[3.2 Hardware Interfaces](#_nxryx7lzd6ew)

[4.0 System Features](#_16yciwck5u8s)

[4.1 Personalized Seating Plan](#_fk12x7pt69oc)

[4.2 Customizable Menu Options](#_uetvrqhykflh)

[4.3 Cost Calculator](#_7bky9zbw0s5q)

[5.0 Other Requirements](#_x9b9tli7wka8)

[5.1 Legal Requirements](#_v7zu6nclxkht)

[5.2 Reuse Objective for this project](#_ow69bg3ys4p8)

[Appendix A: Glossary](#_d7akd0v76ney)

[Appendix B: To Be Determined List](#_72bk3h4sxwit)

# 

# 

# 1.0 Introduction

## 1.1 Purpose

The purpose of this document is to demonstrate the Snazzy Software Squad’s innovative, user-friendly restaurant technology for the fictional Life Is Great Eatery. This document will describe features of the system, such as the interface, processing, and data; software and hardware requirements for the system; applications of the system; constraints; and the functionality of this system. The document makes use of both written and visual means to achieve a thorough description.

## 1.2 Document Conventions

Many different standards and typographical conventions were used throughout the requirements specification document. We start each new section of the document with a different heading type to show a change of subject. We also bolded each subsection to show a transition within each section. Tables were used in both the glossary and functional requirements to help organize information. Diagrams of the use cases were also used in section 2 for better visual representation.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for the development team, the members of which will be involved in design, programming, testing, and writing documentation. This document may also be useful for the client requesting our product. All readers should follow the entire document to understand the functional and nonfunctional requirements and constraints of our program.

Section 2.0, Overall Description, utilizes a use case diagram. Each use case is described visually and textually to achieve a thorough description of each step of the program.

Section 3.0, Requirements Specification, specifies the interfaces and requirements, both functional and nonfunctional. The functional requirements are described in further detail using tables. Non-functional requirements are defined for the client wishing to use the system.

Section 4.0, System Features, describes each feature of the project. These features are those that are most important to the user of the system.

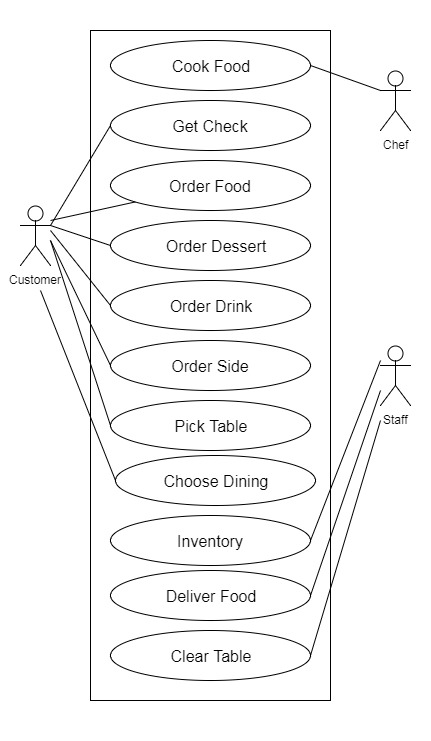
## 1.4 Product Scope

The system described in this document is a digital kiosk designed for a restaurant, which we named the Life Is Great Eatery. This system is designed to cut wait time for customers who want a sit-down restaurant experience without the wait. The system also caters to those wishing to take-out by offering a more customizable alternative to traditional fast food. This innovative, user-friendly system is indispensable to people looking for a customizable and efficient restaurant experience.

The simple, user-friendly system for Life Is Great Eatery is designed to give the customer full control over their seat, order, and payment with little interaction with staff. The system caters mostly to the customer, and the staff is involved only to be notified of the customer’s chosen seat and order. The customer chooses their seat on a screen, which is the described product. Then, they use buttons to select their food, and finally, they choose a payment option and pay for their food. The software eliminates the need for traditional cash registers and the frustrating yet frequent wait to be seated which often causes customers to leave.

# 2.0 Overall Description

## 2.1 Product Perspective



*The Life is Good Eatery* System is designed specifically for the use of *The Life is Good Eatery* restaurant*.* This is a new, self-contained product. There are three active actors: the Customer, the Cook, and the Staff. At the moment this system’s main focus is on the Customer’s perspective.

## 2.2 Product Functions

The Customer must be able to:

* Decide to eat-in or take-out
* See a seating chart and pick a table
* Select to order from “food”, “drink”, “side”, and/or “dessert” menu
* Keep accessing menus and ordering until finished
* Finish order and get receipt

Time permitting the Staff should be able to:

* Update the table status, i.e. clear a table
* Update status of food: if it was delivered or not
* Look at and update inventory list and in future menu

Time permitting the Cook should be able to:

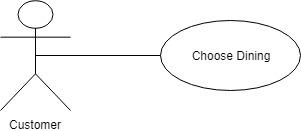
* See a list of food that was ordered
* Update status of food: if it was cooked or not

## 2.3 User Cases and Characteristics

### 2.3.1 Customer Use Case

#### Use case: Choose Dining

**Diagram:**

****

**Brief Description:**

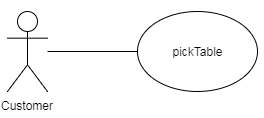
Customer will decide if they are eating in the restaurant or taking the food to go.

**Initial Step-by-step Description:**

1. This will be the first option our system will display on the screen
2. Customers will have the choice to either dine in or to take out
3. If customers choose to dine in then the table arrangement screen will appear
4. If customer chooses to take out the system will bring the customer immediately to the menu options

#### Use case: Pick Table

**Diagram:**

****

**Brief Description:**

The customer picks a table at which to sit based on location and party size.

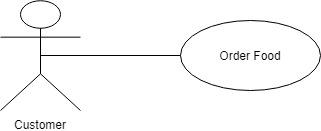
**Initial Step-by-step Description:**

Before this use case can be initiated, the customer has already decided to dine in.

1. The system displays a screen with table arrangements and the tables’ availability.
2. The customer chooses a table.
3. If a table is not available the customer will not be able to select that table.
4. The system verifies the information and returns the customer to the menu option

#### Use case: Order Food

**Diagram:**

****

**Brief Description:**

Customer will choose their meal from a variety of food choices.

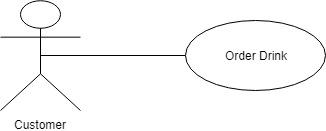
**Initial Step-by-step Description:**

Before this use case can be initiated, the customer has already decided to take out or picked a table.

1. After choosing the preferred dining option, the customer will be shown menu options.
2. The system will give customers the option to order various meal options such as food, beverages, sides, etc.
3. Customers will see their order listed in a mini box on the side of the screen which tracks the meals ordered along with the pricing of each item. They will have the option to return to the original menu as many times as desired if the customer would like to order more items.
4. After customer has completed the order, he or she will see the total price, and the order will be processed.

#### Use case: Order Drink

**Diagram:**

****

**Brief Description:**

Customers will choose what kind of drink they would want along with their meal

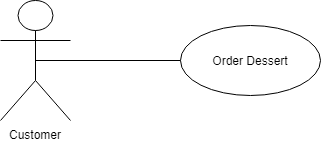
**Initial Step-by-step Description:**

Before this use case can be initiated, the customer must decide to order for take-out or pick a table.

1. As customers are ordering their meals, the system will also display a drink menu
2. Customers will have the choice to add various beverage options from the menu

#### Use case: Order Dessert

**Diagram:**

****

**Brief Description:**

Customers are given the choice to order a variety of desserts after their main course

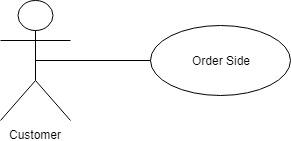
**Initial Step-by-step Description:**

Before this use case can be initiated, the Customer has already decided to take out or picked a table.

1. As the customer goes through the ordering process, the system will also display a dessert menu in the main menu screen.
2. Customers can add their dessert at any point in the ordering process
3. If the customer does not initially order dessert and decides to order later, customer will have to ask waiter/waitress for assistance.

#### Use case: Order Side

**Diagram:**

****

**Brief Description:**

Customers are given the choice to order a side of their choice with their meal, or simply get a side instead of a meal.

**Initial Step-by-step Description:**

Before this use case can be initiated, the customer must decide to order for take out or to pick a table.

1. In the beginning of the ordering process, the system will show the customer a “side order” option in addition to a “meal order”.
2. Customer has the option of getting a side on its own without getting a meal
3. Customer can choose not to get a side
4. Customer can add a side at any point in the order process
5. Once any of the decisions are made, the main menu screen will appear where they can add more to their order or they can end the order

#### Use case: Get Check

**Diagram:**

**.png**

**Brief Description:**

Customers after eating their meal will be given their check

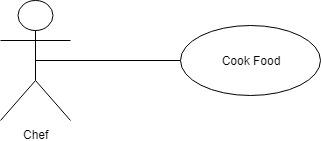
**Initial Step-by-step Description:**

Before this use case can be initiated the Customer has finished adding items to their order.

1. The customer selects the “Checkout” button on the bottom-right of the screen.
2. The system gives the customer a review of the order
3. The customer has the option to complete or cancel the order
4. Once the customer finishes the order, they are given a receipt and can pay for their meal with a staff member

#### Use case: Cook Food

**Diagram:**

****

**Brief Description:**

This is where the chef will make the food once each customer orders are submitted

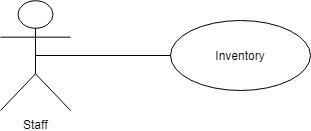
**Initial Step-by-step Description:**

The Chef will need to be logged into the system to access the orders.

1. Once the complete order has been processed, the chef will see the order on his own screen.
2. The chef will cook the food and then will notify the completion of order on their screen
3. The cooked order will be removed from the screen

#### Use case: Inventory

**Diagram:**

****

**Brief Description:**

This is where staff members will check to make sure that the inventory needed for various meal items are fully in stock or not. Staff members will need to restock the necessary items.

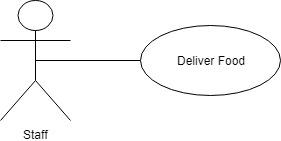
**Initial Step-by-step Description:**

The staff will need to be logged in to access the inventory sheet.

1. The staff will make sure that the necessary items needed for the various food options on the menu are in stock.
2. The inventory allows for tracking of what items are running low and need to be reordered
3. It is the staff’s responsibility to restock the inventory
4. The staff will be able to update the menu if an item is out of stock

#### Use case: Deliver Food

**Diagram:**

****

**Brief Description:**

Waitress/Waiter will serve the customer the meal they ordered at their designated table.

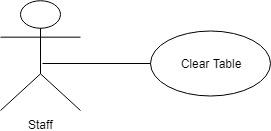
**Initial Step-by-step Description:**

Before this use case can be initiated a staff member must be logged in and the chef must have notified the system that the food has been cooked.

1. Staff member will be notified by the chef that the order is ready to be delivered.
2. Waiter/Waitress will then deliver the food to the specific table that the customer of the order has specified
3. The staff must then mark in the system that the food has been delivered

#### Use case: Clear Table

**Diagram:**

****

**Brief Description:**

Once the customer is finished with their meal, a staff member will come and clear off the table and make sure it is clean for the next customer.

**Initial Step-by-step Description:**

1. Once the customer has left the table, staff members will come and clear off the table.
2. Staff members will then clean and wipe down the table.
3. Once the specific table is cleaned, staff members will be able to mark the table as available in the system.
4. The system will then update the available tables displayed to the customer in the Pick Table use case.

## 2.4 Operating Environment

The system will be operating in three places - a location for each of the three actors. The customer will have access to one tablet in the front of the restaurant. The chef will have a tablet and the staff will have a tablet. Each tablet will be in a durable case and secured to the surface on which it is located but can be removed by restaurant personnel easily if needed. The chef’s tablet will be located in the kitchen area based on the kitchen’s design and accessibility but will not be located near water or open flames. The staff will also have access to a tablet in a back room but they will be able to take it out of the back room and bring it with them around the restaurant as needed. The server will also be located in a back room of the restaurant.

## 2.5 Design and Implementation constraints

*The Life is Good Eatery* will be responsible for alerting us with any problems or updates they may need to the system. If a problem should arise, they should read the user manual for help or contact Customer Service. *The Life is Good Eatery* is responsible for updating their menu and inventory. An order will timeout if the customer has not been active for five minutes. The words displayed on the screen will be in the English language. The system will not have to hold exact orders after they have been delivered to the customer but the inventory will need to be saved for at least three months for the restaurant’s personal record. The customers to the restaurant will only have access to one screen at the beginning and will need to be reassessed as the restaurant grows.

## 2.6 User Documentation

* User manual
* Online help
* Customer Service information

## 2.7 Assumptions and Dependencies

We assume that the restaurant will have a functioning staff of waiters, waitresses, chefs, etc. We assume that the restaurant will also have their own payment system in place. The restaurant will also need to have access to the internet and WiFi. The system depends on the fact that the restaurant has their own inventory that the have correctly supplied us with for their initial system to have the proper menu.

# 3.0 Requirements Specification

## 3.1 User Interfaces

All user interactions will be run within a GUI interface. The software will be presented on a touchscreen desktop device where the users can tap on options on the screen until they are satisfied with their order. Specifics about the device can be found in *3.2 Hardware Interfaces*. The first screen presented to the user will be a welcome and greeting to the eating establishment and the two choices of either “Eat In” or “Take Out”. If “Eat In” is selected, the user will be presented with a seating chart of the restaurant and all available seating for them to choose from by table number. Once a table is selected they will immediately be brought to the “Order” section of the interface.

If the user instead presses “Take Out”, the “Order” interface will be brought up automatically skipping the table selection. At the order menu a total price and order list will stay constant on the right side of the screen adding up their total and showing their current order items as they progress through. Two buttons “Cancel” and “Check Out” will also be at the bottom right-hand corner of the screen through this process along with their chosen table number at the top right. Once finished, the user can press “Check Out” and will be brought to the payment screen. The total price will be brought to the center of the screen and two more options of either paying with “Cash” or “Card”. Finally after choosing an option and paying a thank you message will appear and their table number once again.

## 3.2 Hardware Interfaces

In order for our software to run it must be used on a windows operating system. That being said, most tablets today use either Android or iOS, and both would be incompatible with our software. To make this work, we can use a smaller desktop computer that uses a touchscreen display instead of keyboard and mouse. This would make it more user friendly, while still allowing to run our software. A simple 12-inch display would suffice, and we could have multiple systems all connected to one desktop.

3.3 Functional Requirements

**3.3.1 Dine In**

|  |  |
| --- | --- |
| **Use Case Name** | dineIn |
| **Trigger** | No pre existing trigger, this is one of the first options on the main screen. |
| **Precondition** | The user has accessed the main screen. |
| **Basic Path** | 1.) The user is presented with either “Dine In” or “Take Out” with the first two options.  2.) If “Dine In” is selected then the user is brought to the table selection screen and can start their ordering process. |
| **Alternate Paths** | In step 2, if “Take Out” is selected then the user is taken directly to the menu and can start the ordering process without choosing a seat. |
| **Postcondition** | The user is able to continue with their order. |
| **Exception Paths** | The user can cancel the order or pay at any time. |
| **Other** |  |

**3.3.2 Take Out**

|  |  |
| --- | --- |
| **Use Case Name** | takeOut |
| **Trigger** | No pre existing trigger, this is one of the first options on the main screen. |
| **Precondition** | The user has accessed the main screen. |
| **Basic Path** | 1.) The user is presented with either “Dine In” or “Take Out” with the first two options.  2.) If “Take Out” is selected then the user completely skips the table selection process and is brought directly to the order screen. |
| **Alternate Paths** | In step 2 the user can instead decide to “Eat In” and will instead be brought to the table selection screen. |
| **Postcondition** | The user is able to order their food. |
| **Exception Paths** | Cancel or Checkout the order at any time. |
| **Other** |  |

**3.3.3 Select Table**

|  |  |
| --- | --- |
| **Use Case Name** | selectTable |
| **Trigger** | The user selects to “Dine In” from the main menu. |
| **Precondition** | The user has accessed the main menu. |
| **Basic Path** | 1.) The user is given a selection of 8 different tables to choose from.  2.) The user selects the table of their choice.  3.) They are immediately brought the the ordering screen after their selection. |
| **Alternate Paths** |  |
| **Postcondition** | The user can now order their food after a selection for a table has been made. |
| **Exception Paths** | The user can cancel their order. |
| **Other** |  |

**3.3.4 Order Food**

|  |  |
| --- | --- |
| **Use Case Name** | orderFood |
| **Trigger** | The user selects “Take Out” from the menu or they they are coming from the “Select Table” screen. |
| **Precondition** | The user has accessed the main screen and selected a dining option. |
| **Basic Path** | 1.) The user selects “Order Food” from the list of different ordering options.  2.) A menu comes up with all foods that can be ordered.  3.) The users can customize their food to their liking. |
| **Alternate Paths** | In step 1, if the user also has 3 other ordering options they can choose from. |
| **Postcondition** | The user is able to order and customize their food. |
| **Exception Paths** | The user may decide to cancel or checkout at any time. |
| **Other** |  |

**3.3.5 Order Drink**

|  |  |
| --- | --- |
| **Use Case Name** | orderDrink |
| **Trigger** | The user selects “Take Out” from the menu or they they are coming from the “Select Table” screen. |
| **Precondition** | The user has accessed the main screen and selected a dining option. |
| **Basic Path** | 1.) The user selects “Order Drink” from the list of different ordering options.  2.) A menu comes up with all drinks that can be ordered.  3.) The users can select which drink they would like. |
| **Alternate Paths** | In step 1, the user also has 3 other ordering options they can choose from. |
| **Postcondition** | The user is able to order their drink of choice. |
| **Exception Paths** | The user may decide to cancel or checkout at any time. |
| **Other** |  |

**3.3.6 Order Sides**

|  |  |
| --- | --- |
| **Use Case Name** | orderSide |
| **Trigger** | The user selects “Take Out” from the menu or they they are coming from the “Select Table” screen. |
| **Precondition** | The user has accessed the main screen and selected a dining option. |
| **Basic Path** | 1.) The user selects “Order Side” from the list of different ordering options.  2.) A menu comes up with all sides that can be ordered.  3.) The users can select which side they would like. |
| **Alternate Paths** | In step 1, the user also has 3 other ordering options they can choose from. |
| **Postcondition** | The user is able to order a side of their choice. |
| **Exception Paths** | The user may decide to cancel or checkout at any time. |
| **Other** |  |

**3.3.7 Order Dessert**

|  |  |
| --- | --- |
| **Use Case Name** | orderDessert |
| **Trigger** | The user selects “Take Out” from the menu or they they are coming from the “Select Table” screen. |
| **Precondition** | The user has accessed the main screen and selected a dining option. |
| **Basic Path** | 1.) The user selects “Order Dessert” from the list of different ordering options.  2.) A menu comes up with all desserts that can be ordered.  3.) The users can select which dessert they would like. |
| **Alternate Paths** | In step 1, the user also has 3 other ordering options they can choose from. |
| **Postcondition** | The user is able to order a dessert of their choice. |
| **Exception Paths** | The user may decide to cancel or checkout at any time. |
| **Other** |  |

**3.3.8 Add Item**

|  |  |
| --- | --- |
| **Use Case Name** | addItem |
| **Trigger** | The user selects a food item, drink, side or dessert to order. |
| **Precondition** | The user has accessed the main screen and selected a food option. |
| **Basic Path** | 1.) The user selects what type of food they would like to order first from the screen.  2.) The user customizes their selected food.  3.) Once they are done customizing, they are given the option to add another item.  4.) The user is taken back to the food selection screen. |
| **Alternate Paths** | In step 4, the user can also choose the option to checkout or cancel their order. |
| **Postcondition** | The user can add another item to their current order. |
| **Exception Paths** | Cancel or checkout at any time. |
| **Other** |  |

**3.3.9 Cancel Order**

|  |  |
| --- | --- |
| **Use Case Name** | cancelOrder |
| **Trigger** | The user has accessed the food selection screen. |
| **Precondition** | The user has chosen a “Dine In” or “Take Out” option. |
| **Basic Path** | 1.) The user has selected a dining option, but now needs to cancel their order.  2.) The user can press the “Cancel” button on the interface.  3.) A message appears asking if they are sure they want to cancel with a “Yes” or “No” option.  4.)The user presses yes and their order will be canceled taking them back to the home screen. |
| **Alternate Paths** | In part 3, if the user presses “No” they can continue with their order instead. |
| **Postcondition** | The user successfully cancels their order. |
| **Exception Paths** |  |
| **Other** |  |

**3.3.10 Checkout Order**

|  |  |
| --- | --- |
| **Use Case Name** | checkoutOrder |
| **Trigger** | The user has accessed the food selection screen. |
| **Precondition** | The user has chosen a “Dine In” or “Take Out” option. |
| **Basic Path** | 1.) The user has selected a dining option and ordered their food.  2.) The user now needs to pay for their food.  3.) The user can press the “Checkout” button on the interface.  4.) A message appears asking if they are sure they want to Checkout with a “Yes” or “No” option.  5.)The user presses yes and they will be taken to the checkout screen. |
| **Alternate Paths** | In step 4, if the user presses “No” they can continue with their order. |
| **Postcondition** | The user is taken to the checkout screen and can now pay for their order. |
| **Exception Paths** | The user can cancel at anytime. |
| **Other** |  |

**3.3.11 Payment**

|  |  |
| --- | --- |
| **Use Case Name** | payOrder |
| **Trigger** | The user has pressed “Checkout” on the order screen. |
| **Precondition** | The user has accessed the menu and placed and order. |
| **Basic Path** | 1.) The user selects “Yes” after pressing checkout.  2.) The user has the option to either pay with “Cash” or “Card”.  3.) After selecting a payment method a thank you message will appear. |
| **Alternate Paths** |  |
| **Postcondition** | The user has successfully placed an order and purchased their food. |
| **Exception Paths** | The user can still cancel at the checkout screen. |
| **Other** |  |

3.4 Detailed Non-Functional Requirements

**3.4.1 Performance Requirements**

The system needs to be able to handle multiple different users ordering at once from different computers. It has to be able to deal with high traffic times such as lunch and dinner without the system crashing. For example, if multiple different groups want to dine in at once, we need to be able to make sure they do not pick the same table. The system also needs to be able to handle multiple different payments happening at once. Furthermore, the system will not allow users to place orders within 15 minutes of the restaurant closing. For example, if we close at 9PM, the system will only allow users to place orders up until 8:45PM. After this it will shut down.

**3.4.2 Safety Requirements**

The actual database holding all information will be kept away from the public in a locked room where only certified personnel have access. The touchscreen themselves will be bolted to a countertop with the computers locked away beneath them. The screens will also have protective cases on them. We will also have protection for our payment system so customers payment methods and identities can be kept safe from outside hackers. All health and sanitation codes will also be met since we are serving food.

**3.4.3 Software Quality Attributes**

The software will need to be easily changed in the future. If we ever want to remove or add menu items, we should be able to do so with no problems. If we run out of a certain food we will have to remove it from the menu as quickly as possible and with ease so no one orders it. Our system can easily adapt and update to what we have at hand. It can also be constantly reused since we only have one main purpose with our business: to serve food.

**3.4.4 Business Rules**

All customers have full access to the menu and can place orders. Only specially trained IT employees have access to the database and computers. Another rule with the system would be ordering times which we previously mentioned. No orders can be placed 15 minutes before closing.

# 4.0 System Features

## 4.1 Personalized Seating Plan

4.1.1 Description and Priority

This feature allows customers to select the seating according to preference on the basis of location and size. This feature is a medium priority because it greatly aids in the efficiency of the restaurant, while tending to the needs of the customers, allowing each customer to personalize their dining experience. By granting the freedom to choose the number of seats and the location of the seats at which they would like to dine, more staff will be free to help in other areas of the restaurant. Furthermore, customers will have access to the updated seating availability before finalizing the type of dining they wish to choose. However, this is not a high priority because our main goal is to serve food to our customers, which we would be able to still accomplish without this feature.

4.1.2 Stimulus/Response Sequences

Customers will be given the opportunity to select a button that reads either “Dine

In” or “Take Out.” Those who choose to dine at the Eatery will select the “Dine

In” button, which will send them to the seating chart. This chart will display the

current seating situation at the Life Is Great Eatery, showing which seats are

currently available, and allowing the customers to choose the number of seats

and the location of the seats as they wish. If the seats are entirely or mostly

unavailable, then the customers will be given the option to select the “Back”

button, allowing them to return to the previous screen and opt for “Take Out”

instead.

## 4.2 Customizable Menu Options

4.2.1 Description and Priority

This feature allows customers to personalize their dining options in that they

have the opportunity to mix and match various orders as they please, while

having the option to revert back to the original menu after each selection. This

way, the customer is able to order as many items as desired.

4.2.2 Stimulus/Response Sequences

As the general food category options are displayed on the screen, the

customer will select the category from which he or she wants to order.

Once a button representing a category of menu options is selected, the

system will show more specific options pertaining to that category, and the

customer may then select from the list. There will be the option at the bottom

of the screen to cancel the order, to finalize the order and submit to check out, or

to return to the original menu and order additional items.

## 4.3 Cost Calculator

4.3.1 Description and Priority

This feature calculates the total cost of the customer’s entire order. This is

a high priority, as it ensures that the customer will pay the accurate

amount of money for the services offered at the Life Is Great Eatery.

4.3.2 Stimulus/Response Sequences

As the customer selects the items from the menu, the system will keep

track of the cost of every item. Only once the customer submits the final

order will the system add up the total cost of the order and display it.

# 5.0 Other Requirements

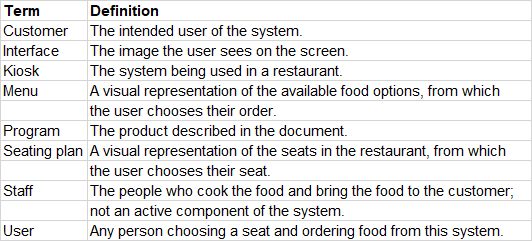
## 5.1 Legal Requirements

This software is only to be used with *“Life Is Great Eatery”* establishments. No employee may distribute any part of the code without company approval. The software is considered company property and any illicit distribution is punishable by law. Only certified employees and software creators are allowed access to the code.

## 5.2 Reuse Objective for this project

The software is designed to be reused and modified as the restaurant and brand grow. If the company were to expand, the project could be modified and reused for a new restaurant. Components of the project could be reused as a basis for future project should the brand venture outside of strictly the restaurant business.

# Appendix A: Glossary



# Appendix B: To Be Determined List

1. Type of database that will be necessary with this system
2. The code necessary to implement the project
3. The visual representation (GUI)
4. The layout and organization of the menu options
5. The various methods of payment
6. The variety of tables with regard to size/number of seats