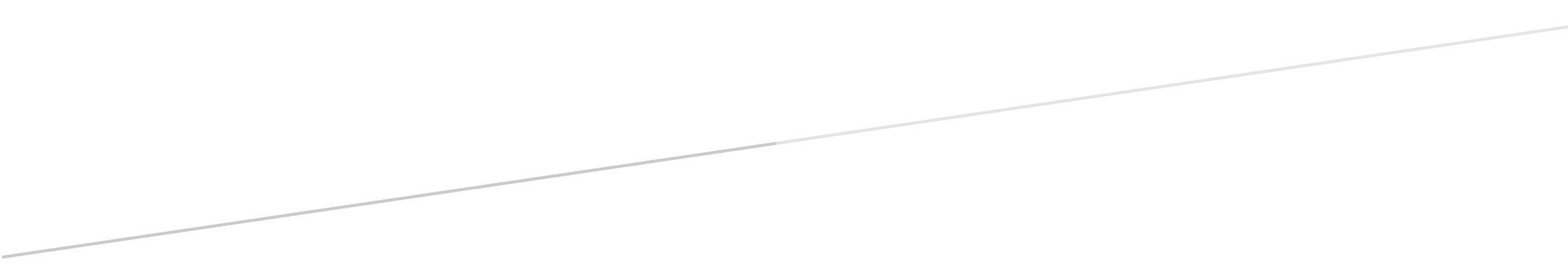
Anne Nguyen & David Leach

12/1/2022

Restaurant web app

Software Requirements Specification



Abstract

A web app that allows customers to order from their favorite  
restaurants seamlessly, providing a simple and fast ordering process.

# Table of Contents:

1. Functional Requirements

1. Login and Signup

2. Customer Functions

3. Admin Functions

4. Other Functions

2. Non-Functional Requirements

1. Security

2. Quality

3. External Interface Requirements

1. User Interfaces

2. Software Interfaces

|  |  |  |
| --- | --- | --- |
| **Name** | **Date** | **Reason for changes** |
|  |  |  |
|  |  |  |
|  |  |  |

# 1 Functional Requirements

## 1.1 Account Creation & Sign In

### 1.1.1 Account Creation

1.1.1.1 Form

* The user will have the following required and non-required form input options to fill out
  + Required:
    - Name
    - Email
    - Password
    - Confirm password
    - Checkbox – Agree to Terms and Conditions
  + Non-required:
    - Birthday
    - Phone
    - Checkbox – Subscribe to email offers

1.1.1.2 Password strength testing and minimum requirements

* Password will be given a strength rating
  + Sections are Weak, Okay, Good, and Great
  + Corresponding colors to these opens should be red, orange, yellow, and green
* Minimum requirements for the password will be shown
  + Each item should have its own icon showing whether or not it has met that requirement

1.1.1.3 Validation

* On clicking a Submit button at the end of the form, the client side should confirm the following:
  + All required boxes are filled
  + Email is correctly formatted
  + Password meets requirements
  + The Password and Confirm Password inputs match
* If any fail, alert user to error

1.1.1.4 Successful Account Creation

* User info should be stored in database
* User is sent to a page informing them of the success
* User is then redirected to the menu page

1.1.1.5 Return to Log In

* A button should be placed on the registration page that allows a user to navigate back to the log in page

### 1.1.2 Sign In

1.1.2.1 Form

* The user will be shown a form with an input for their email address and password

1.1.2.2 Validation & Authentication

* At the end of this form will be a Sign In button. When clicked:
  + Client side will confirm that boxes are both filled
    - If empty, error will show under the corresponding box informing the user
  + Database will be searched for the input email
    - If not found, error message will show under the email input box
    - If found, will then continue authentication
  + User input for password will be authenticated by comparing it to the stored password in the database

1.1.2.3 Successful Sign In

* After it has been confirmed that the user’s password is correct, the client will store a cookie that holds the user’s information for later visits
* User is sent to the menu page

1.1.2.4 Create Account

* A button will be placed under the login form to allow the user to be directed to the account creation page

## 1.2 Customer Functions

### 1.2.1 Select Items from menu

1.2.1.1 Add Item Button

Every item shall have a button that will be used to add that item to the user’s cart

On click, the item information should be collected and used to create that item as an element in the cart

On click, the add button will flash, showing “Added!” This will fade back into the normal button

Each click adds one item to the cart

### 1.2.2 Add & Remove Items From Cart

1.2.2.1 Cart Information Stored by Client

* The items that are in the user’s cart should be held in localStorage. This is so that the cart can persist even after the window has been closed.
* When the object holding the data for the cart contents is updated, that is then sent to localStorage to replace the current saved value, if there is any.

1.2.2.2 Item Quantity Display

* There shall be an input box for each item. This will display the current selected quantity for that specific item.
* A minus (-) and plus (+) button shall be on either side of the input box. These will allow the user to adjust the amount of each item from the cart menu.
* This item number will not be able to go below 0 or above 10.

### Update Total Cost

1.2.3.1 Update Total Button

* A button will be placed at the bottom of the cart list that will allow the user to update the total cost of the cart contents
* This is to avoid unnecessary performance loss that could be caused by a system that automatically updates as the user changes item quantities

### 1.2.4 Payment Option Selection

1.2.4.1 Card or Cash Selector

* (decide on a selector style) will be available at checkout to choose whether the user will be paying with card or in person with cash
* The card option should be selected by default

1.2.4.2 Card selection

* When the card option is selected, a form shall be displayed for the user to enter their card information
* The card number input will require 16 numbers to pass a validity check
* There will be a cardholder name input
* There will be an input for the card’s expiration date. This must be set to the current month as the soonest option available
* There will be a CVV number input, this will require 3 digits to pass a validity check
* A tip option will be available only while this payment option is selected (See 1.2.5)
* The user will only be able to attempt to complete their purchase if all of these input fields are filled and meeting their requirements.

1.2.4.3 Cash Selection

* When cash option is selected, a string will be displayed to inform the user to show their receipt to the cashier when they arrive

### 1.2.5 Option to Leave a Tip

1.2.5.1 Display

* The option for the user to leave a tip shall only be available when the card option is selected. This is because the user will be provided to option to tip when paying in person.

1.2.5.2 Pre-Defined Tip Options

* The user shall be given pre-defined options for tips. These will include 3 percentages, being 15%, 20%, and 25%, as well as an option to provide their own amount.
* Each of the 3 pre-defined options will display the calculated tip amount at that percentage.

### 1.2.6 Successfully Place Order

1.2.6.1 Order Timer

* When the user first enters the checkout page, a timer will begin
* This timer will not persist if user leaves checkout screen, but will start from 0 again
* This timer is used to keep track of the time it takes for the user to successfully complete their order

1.2.6.2 Order Submission

* Cart information will be stored for use in creating the receipt
* Cart content should be cleared
* User should be shown a “Order complete” message, which will redirect to the finalized order page, which will contain the receipt

### 1.2.7 Order Receipt

1.2.7.1 Items Ordered

* There will be an itemized list of every item in the menu
* Each item will show its quantity
* For items with longer names, abbreviations should be used

1.2.7.2 Customer Information

* Customer information will be shown at the top of the receipt
* This will include their name, their payment selection, time that the order was completed, and the time taken from start to end of checkout
* If the card payment option was selected, it will show a preview of the card number, with the first 12 digits hidden

1.2.7.3 Total Cost

* A subtotal will be given below the item list
* If tip was added, the tip should be shown below the subtotal
* The tax fee should also be shown based on the state that the store is located
* A the sum of these numbers will then be shown as the total cost

## 1.3 Admin Functions

### 1.3.1 Admin Account

1.3.1.1 Hard Coded Account

* In order to ensure that the admin will always have access, the sign in information for the admin account will be written in the code
* The admin account information should be obfuscated

## 1.4 Other Functions

### 1.4.1 Animate.css

### 1.4.2 jQuery

### 1.4.3 Dexie.js

### 1.4.4 Ionicons

# 2 Non-Functional Requirements

## 2.1 Security

### 2.1.1 Admin Account Information

2.1.1.1 Obfuscation

* The admin sign in information should be obfuscated to help hide it within the code.
* Any other potential security-related code should also be obfuscated.
* Resource: <https://www.obfuscator.io/>

### 2.1.2 Cookies

2.1.2.1 Client Side User Information

* When a user logs in, a cookie will be stored to save that user’s sign in within their browser
* Cookies provide a higher level of security than other options such as localStorage and sessionStorage
* Cookies can also be set to expire after a set amount of time and be used for authorization

## 2.2 Quality

Description

# 3 External Interface Requirements

## 3.1 Customer Interfaces

## 3.2 Admin Interfaces

Description