**Requirements**

**Requirement** #: 1 **Use Case** #: 1 N**ame of the Use Case**: User registers account

**Introduction**: The user is just beginning on the webpage and has certain requirements to complete to gain access to all products on the webpage.

**Rationale**: This requirement allows the system to verify who is accessing the webpage and who to send information to.

**Input**: User inputs their username, email, date of birth, and creates password

**Requirement description**: A welcome to the new webpage for a person who is interested into trying for their grocery shopping experience

**Output**: After registering on webpage an email is sent with further instructions to confirm the account or notifies the user if they do not meet the criteria for registering. All information is stored to identify each user.

**Test Cases**:

**Description**: User Registers Account

**Test Inputs**: Username, email, date of birth, password all meeting the new user requirements to begin webpage

**Expected Results**: confirmed account, user gains access to webpage for grocery shopping system

**Dependencies**: None

**Initialization**: Already sent instructions on confirming account to email

**Test Steps**:

1. Submit valid credentials to gain access to webpage.

**Requirement** #: 2 **Use Case** #: 2 **Name of the Use Case**: User Logs In

**Introduction**: A user is already finished setting up their account and is ready to access what they need for grocery shopping starting from the homepage.

**Rationale**: This requirement is so that the system can differentiate between each user.

**Input**: The user inputs their user name and password and clicks submit to redirect

**Requirement Description**: A way to let the database know that it is you who our signing in

**Output**: The database sends the user to the homepage then allows access

**Test Cases**:

**Description**: User Logs in

**Test Inputs**: Username and password that matches system

**Expected Results**: User brought to homepage

**Dependencies**: None

**Initialization**: Account already made and in system

**Test Steps**:

1. Submit username and password correctly

**Requirement** #: 3 **Use Case** #: 3 **Name of the Use Case**: Retrieve lost password

**Introduction**: A lot may not remember the password that they set upon registering their account, so they have the chance of remembering something simpler such as their email to gain back access

**Rationale**: This requirement is set so that a user won’t destroy any of the information they have inside of their account

**Input**: User inputs their email and is given a chance to restore their password with something that they can remember

**Requirement Description**: A help as there are a lot of different sites that require usernames and passwords that have different requirements

**Output**: They can get back into the homepage to access what they need

**Test Cases**:

**Description**: Retrieve lost password

**Test Inputs**: Emails associated with account

**Expected Results**: Gain access back to account

**Dependencies**: Remember email

**Initialization**: Already have an email inside of the system database

**Test Steps**:

1. Submit a valid email and confirm new password

**Requirement** #: 4 **Use Case** #: 4 **Name of the Use Case**: Registering Location

**Introduction**: Since a user will have access to all products through the webpage registering location helps to set a location for each user to have it shipped to them

**Rationale**: This requirement is set so a user can receive anything that they need since it is online

**Input**: The user inputs their city, state, street, and zip code

**Requirement Description**: To let the system know you are a valid user

**Output**: User now has a current location that is updated

**Test Cases**:

**Description**: Registering Location

**Test Inputs**: Current address linked with account

**Expected Results**: Have account up to date

**Dependencies**: None

**Initialization**: Moved into a valid address

**Test Steps**:

1. Correct information of city, state, street, and zip

**Requirement** #: 5 **Use Case** #: 5 **Name of the Use Case**: User Logs Out

**Introduction**: No one wants to have someone else in their account to gain their credentials, so this is an easy process to make sure it is safe

**Rationale**: This requirement is to keep a user information only for them

**Input**: Click log out button on the upper right part of the page

**Requirement Description**: sends a signal to the system to log out a user if before 24 hours which will log out automatically

**Output**: Takes you to the heart of the webpage

**Test Cases**:

**Description**: User Logs out

**Test Inputs**: logging out inside of webpage before exiting

**Expected Results**: required to log back in

**Dependencies**: none

**Initialization**: remembering to log out

**Test Steps**:

1. Logging out and watching screen take you to the heart of the webpage

**Requirement** #: 6 **Use Case** #: 6 **Name of the Use Case**: Select Current Location

**Introduction**: This is a tab inside of the webpage to help users stay up to date with where their current location is

**Rationale**: This requirement allows all products to be delivered to the correct place

**Input**: The user goes to their account page and selects current location tab to where there is a drop-down menu of registered locations to choose from, the user then clicks the current location and then submit for it to be updated

**Requirement Description**: set for people who have more than one home address on the account

**Output**: User is notified that they have set a new location for their account

**Test Cases**:

**Description**: Select Current Location

**Test Inputs**: Select an address if multiple addresses are available

**Expected Results**: Switch to address preferred

**Dependencies**: none

**Initialization**: Already have address stored before switching

**Test Steps**:

1. Selecting current location drop down menu
2. Clicking the right address and then clicking submit to update

**Requirement** #: 7 **Use Case** #: 7 **Name of the Use Case**: Delete a registered location

**Introduction**: This option is set for if a location is no longer needed to be used

**Rationale:** This requirement is so that a user will not be confused and can just delete it from the account

Input: Press a button on the right of the location on the listed labeled with a trash can

**Requirement Description**: Informs the system to remove the registered location from this user’s list of registered locations

**Output**: The page is reloaded to show that it has been removed from the list

**Test Cases**:

**Description**: Deleting registered location

**Test Inputs:** labeled trash can clicker

**Expected Results**: Address disappear

**Dependencies**: none

**Initialization**: multiple addresses stored

**Test Steps**:

1. Check location
2. Hit trash can on side
3. Page refresh and removed from list

**Requirement** #: 8 **Use Case** #: 8 **Name of the Use Case**: Comparing Prices

Introduction: Lets a user know what the best deal is between different locations

**Rationale**: This requirement is so one will know what the best decision to make for a product they wish to use

**Input**: The product they want in the search engine and enter

**Requirement Description**: Price comparisons between various local grocery stores and who likes them

**Output**: List of comparable prices for the product

**Test Cases**:

**Description**: Comparing Prices

**Test Inputs**: A product name desired to use

**Expected Results**: List of comparisons

**Dependencies**: search engine

**Initialization**: account for grocery shopping system

**Test Steps**:

1. Enter product name
2. Check price and distance

**Requirement** #: 9 **Use Case** #: 9 **Name of the Use Case**: Comparing Distance

**Introduction**: Though a product may be cheaper than the next you still must factor in gas because if a product is more expensive but is closer it may be the same

**Rationale**: This requirement is set to give the user another option of choosing their product

**Input**: User goes to the homepage and clicks on “Grocers near me.”

**Requirement Description**: Gives you the number of miles that you must travel

**Output**: List the closest grocery stores that have the product available

**Test Cases**:

**Description**: Comparing Distance

**Test Inputs**: Select “Grocers near me”

**Expected Results**: A list of times expected

**Dependencies**: search engine

**Initialization**: product desired

**Test Steps**:

1. Homepage
2. View list

**Requirement** #: 10 **Use Case** #: 10 **Name of the Use Case**: Get Directions

**Introduction**: A user gets guide to where they are trying to go which makes it easier

**Rationale**: This requirement is set so that the user knows how to get to the product

**Input**: Click on the name of the grocery store and you are directed to google maps

**Requirement Description**: Gives you directions from route 1

**Output**: Have directions for the desired store

**Test Cases**:

**Description**: Get Directions

**Test Inputs**: Google maps

**Expected Results**: get street for street directions

**Dependencies**: none

**Initialization**: a store you prefer

**Test Steps**:

1. Click on name of grocery store
2. Be directed to google maps

**Requirement** #: 11 **Use Case**#: 11 **Name of the Use Case**: Change Password

**Introduction**: A user may want to pick something easier to gain access to their account or may have been hacked

**Rationale**: This requirement is set to give the user what they prefer

**Input**: User fills out three fields, one with the current password and the other two fields being the new password then confirming the new password

**Requirement Description**: checks to see if you meet password requirements before granting a user with a new password

**Output**: The system will update the current account password to the new password

**Test Cases**:

**Description**: Change Password

**Test Inputs**: Three fields/ current password, new password, confirm password for account

**Expected Results**: a new password stored

**Dependencies**: none

**Initialization**: remember old password

**Test Steps**:

1. Old password confirmation
2. New password confirmation

**Requirement** #: 12 **Use Case** #: 12 **Name of the Use Case**: User changes email address

**Introduction**: Same as changing password just a little more difficult with it being an email because an email verifies the account

**Rationale**: This requirement gives the user access to their preferences

**Input**: User fills out two fields, one being the current password and one being the new email address before confirmation

**Requirement Description**: checks password to verify that they have the authority to perform this action

**Output**: System is updated to use the new email address and user is notified

**Test Cases**:

**Description**: User changes email address

**Test Inputs**: old password and new email address for account

**Expected Results**: new email address associated

**Dependencies**: none

**Initialization**: prefer address change

**Test Steps**:

1. Old password
2. New email address

**Requirement** #: 13 **Use Case** #: 13 **Name of the Use Case**: Add Product to Favorites

**Introduction**: Everyone has something that they like the most over others, so favorites can be stored

**Rationale**: This requirement is for users to gain a product faster that they are most interested in

**Input**: User wish to bookmark so they click star icon

**Requirement Description**: helps the user for future references when they visit the product page

**Output**: sends a message to the database to add that product to that user’s favorite list

**Test Cases**:

**Description**: Add Product to Favorites

**Test Inputs**: click of star icon

**Expected Results**: a list of favorites

**Dependencies**: none

**Initialization**: most liked product

**Test Steps**:

1. Star icon

**Requirement** #: 14 **Use Case** #: 14 **Name of the Use Case**: View Favorites

**Introduction**: User gets to see what they like most in one section

**Rationale**: This requirement is for users to have their favorite products in order

**Input**: on the account page user clicks on ‘favorites list’ link

**Requirement Description**: An option available on the page is to view your favorite list which contains product

**Output**: The user can view their favorite items

**Test Cases**:

**Description**: view favorites

**Test Inputs**: click of favorites list

**Expected Results**: see list of favorites

**Dependencies**: none

**Initialization**: favorites list made

**Test Steps**:

1. View favorites list

**Requirement** #: 15 **Use Case** #: 15 **Name of the Use Case**: Remove items from favorite

**Introduction**: A way to delete a product that is no longer valued

**Rationale**: This requirement is for users to keep their favorite list correct

Input: clicking on a small button to the right of the item in the list

**Requirement Description:** An option available next to the items in the list to delete if no longer wanted

**Output**: sends a message to the database to remove the item from the favorites list, when this is done it will inform the user of its success and then refresh the page to reflect the updated information.