

Donut Shop Website Information System Project

System Requirements Documentation

Jonah Reardon

Table of Contents

Customer Problem Statements and System Requirements	3
Functional Requirement Specification	11
System Sequence Diagrams	17
Activity Diagrams	20
User Interface Specification	26
Project Proposal	47
Project Plan	48
Reference	49

Customer Problem Statements and System Requirements

Customer Problem Statement

Problem Statement

Donut shop customers are trying to find donut shops near them while on the go but it takes a long time because there is no organized website or mobile version which makes them feel frustrated. Donut shop owners and managers find it difficult to keep track of all of the newly opened, existing, and closed stores. The donut shop managers find it hard to manage all of the store information because there is no centralized website which makes them feel overwhelmed. The donut shop customers expect that a website software system would help by allowing them to view store information digitally around the world and while on the go with a mobile device. The donut shop owners and managers expect that the website software system would help by providing a centralized and accessible website to keep store information up to date. Plus, the shop owners believe that the system provides an opportunity for the business to grow, expand, and possibly get more customers by being discoverable on the Internet.

Glossary of Terms

Donut shop: A donut shop is a physical building or place that sells donuts and possibly other food or beverages. The donut shop has a location which includes a street address, city, state, country, zip code. Donut shops also have store hours which includes when the store opens and closes. The store also has a store ID, name, and phone number. A graphic of a donut shop is below.



Donut shop customers: Donut shop customers are people who buy food or other goods from the donut shop.

Donut shop owners: Donut shop owners are people who own the donut shops.

Donut shop managers: Donut shop managers are people who manage the donut shops and its employees. They are people who are in charge of coordinating the work of other donut shop employees. Some managers can supervise or oversee other employees. Other managers can help manage the allocation of donut shop resources.

Donut shop store information: Donut shop store information is organized, meaningful data or a collection of facts about the donut shops. The store information contains information about the

locations such as street address, city, state, country, and zip code that are connected to each specific donut shop. Other donut shop information include the store hours, store ID, name, and phone number that is made meaningful by connecting the data to each donut shop.

System Requirements

Functional Requirements

No.	Priority Weight	Description
REQ-1	High	The system should have the capacity to contain at least 10 donut shop records.
REQ-2	High	Four different types of users (guests, donut shop customers, donut shop owners, and donut shop managers) should be able to view donut shop information.
REQ-3	High	The three different types of donut shops are gourmet, vegan, and variety.
REQ-4	Medium	Logged in donut shop customers should be able to favorite stores and view a list of their favorite stores on the home page.
REQ-5	High	Only 2 different types of users (donut shop owners and donut shop managers) should be allowed to add, edit, or delete donut shops in the system.
REQ-6	High	The system should not allow duplicates of stores. They cannot have the same store ID or store name.
REQ-7	High	A donut shop owner or manager must input donut shop data into the website system.
REQ-8	High	Response time for loading webpages must not exceed 5 seconds.
REQ-9	High	The system must be operational 24 hours, seven days a week (24/7).
REQ-10	Medium	The system must filter for shops based on the city or postal code in less than 4 seconds.
REQ-11	High	A donut shop record must be added, deleted, or changed only by a donut shop owner or manager.
REQ-12	High	The system must provide logon security for the application.

REQ-13	High	The system must allow users to log into their account by entering their username and password.
REQ-14	High	The system must allow users to log into their accounts on mobile by entering their username and password.
REQ-15	High	The system must allow users to reset their passwords by clicking on “I forgot my password”.
REQ-16	Low	The system should automatically put stores into viewable categories based on their city location.
REQ-17	High	The database records need to be updated in real time

Nonfunctional Requirements

Some of the requirements listed in the table above are non-functional requirements such as REQ-8, REQ-9, REQ-10, and REQ-17. These non-functional requirements are criteria about how the system should perform within categories such as the performance or speed of the system. Other categories include functionality, usability, reliability, and supportability.

User Interface Requirements

UI REQ No.	Priority Weight	Description
UIREQ-1	Medium	If the donut shop customer has favorited a store, the system and user interface should display a list of their favorite stores on the homepage.
UIREQ-2	Medium	If the donut shop customer has favorited a store, the system and user interface should display a favorite store tag in the store list page.
UIREQ-3	High	The user interface should allow a donut shop customer to favorite their store. This can be done by using a switch that the user can toggle.
UIREQ-4	High	If the user inputs the wrong login information, the user interface should display a notification and highlight the input boxes that need to be fixed.
UIREQ-5	High	If the user does not input information where they need to or inputs information that is in the wrong format, the user interface should display a notification and highlight the input boxes that need to be fixed.

UIREQ-6	High	The user interface should provide easy navigation and be quick to respond.
UIREQ-7	High	The user interface should have a homepage button (house icon) to quickly get back to the main page for faster navigation and for if the user ever gets lost.

Favorite stores on Homepage:



Favorite Stores Tag in List of Stores Page:



List of Stores
Search by City or Postal code

Search

Store 1 ☒ Favorite

Store 2

Store 3

Store 4

Store Details Page with Store Information and Favorite Switch:



Store 1 Details

Store ID

Contact Information

Store Number
Phone Number

Location

Address
City State
Country

Favorite: switch 2

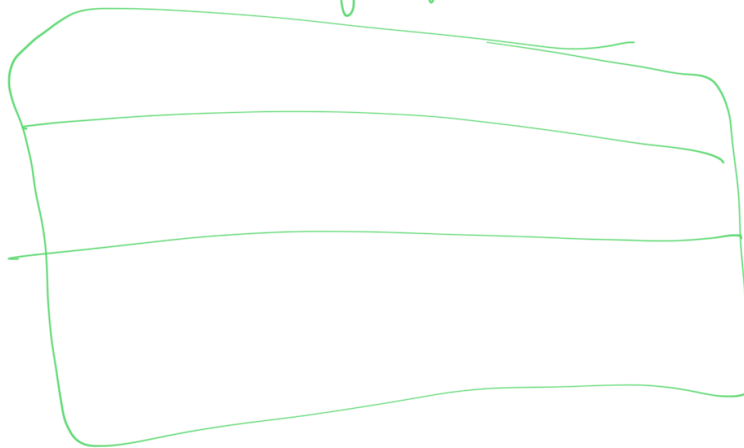


Store Picture

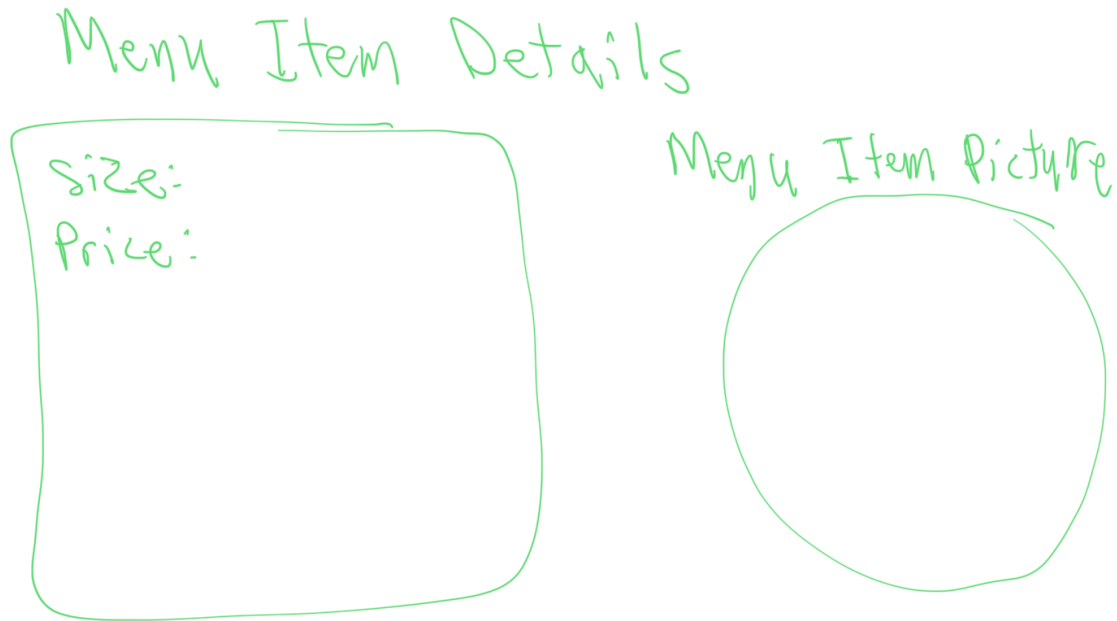


Donut Shop Menu Category List Page:

Menu Category List



Donut Shop Menu Items Details:



Plan of Work

- ✓ Week 1: Solidify what front-end, back-end, and database programs I am going to use.
- ✓ Week 2: Set up a project development environment by connecting front-end, back-end, and database software together, figure out the structure and framework for the system.
- ✓ Week 3: Create the login for the system so customers and shop owners can login.
- ✓ Week 4: Create the registration for new customers and shop owners to make a login or account.
- Week 5: Design the features for shop owners to add, modify, or remove stores, store details (store name, phone number, street address, city, state, country, zip code), food categories, menu items, and menu item details.
- Week 6: Implement the features for shop owners to add, modify, or remove stores, store details (store name, phone number, street address, city, state, country, zip code), food categories, menu items, and menu item details.
- Week 7: Continue to implement the features for shop owners to add, modify, or remove stores, store details (store name, phone number, street address, city, state, country, zip code), food categories, menu items, and menu item details.
- Week 8: Verify the features work, record, and turn in current progress for the midterm.
- Week 9: Iterate and improve current features for the shop owners.
- Week 10: Design the features for customers to be able to view the stores and filter for stores based on their city or postal code. Design the features for customers to view basic store, menu category, and menu item information
- Week 11: Implement the features for customers to be able to view the stores and filter for stores based on their city or postal code. Implement the features for customers to view basic store, menu category, and menu item information
- Week 12: Design the features for customers to be able to favorite stores and menu items.

Week 13: Implement the features for customers to be able to favorite stores and menu items.

Week 14: Debug and test to make sure all features work as intended.

Week 15: Test and record the demonstration and progress made for the final presentation.

Choosing and solidifying what front-end, back-end, and database programs I was going to use was harder than I thought. I had to do a lot of research to figure out the best technology stack I think I should use for this project. It took a while to learn the differences between all of the software programs.

Setting up the development environment took longer than I anticipated to try to connect the front-end, back-end, and database software together. Creating a login for the system and creating the different types of users so they have access to different parts of the system was more straightforward and similar to the way other software programs handle logging in. I started designing some of the features and figuring out the user interface. I will continue to work on designing, programming, and adding features to the system throughout the rest of the semester.

Functional Requirement Specification

Stakeholders

- Donut Shop customers
- Donut Shop owners
- Donut Shop managers
- Donut Shop employees
- Donut Shop sponsor
- Donut Shop CEO
- Donut Shop president
- Donut Shop board of directors

Actors and Goals

Primary Actors

- Customer: This actor can filter to view stores and their details based on zip code. They can also favorite their stores and menu items to make it faster to find their donut shop and get the products they want.

Secondary Actors

- Donut Shop Manager: This actor can add, remove, or modify stores, their store details, store locations, food categories, menu items, and menu item details.
- System: This is responsible for providing details of favorited or filtered stores and menu items.

Use Cases

Donut Shop Manager (total: 30)

- Add store: to add a new store (2)
- Add food category: to add a new food category (2)
- Add menu item: to add a new menu item (2)
- Add menu item details: to add details about a menu item (2)
- Modify store: to modify information about a store (2)
- Modify food category: to modify a food category (2)
- Modify menu item: to modify a menu item (2)
- Modify menu item details: to modify details about a menu item (2)
- Remove store: to remove a store (2)
- Remove food category: to remove a food category (2)
- Remove menu item: to remove a menu item (2)
- Remove menu item details: to remove details about a menu item (2)
- View account: to view account details (2)
- Login/Logout: to login or logout from donut shop manager account (4)

Customer (total: 30)

- Favorite store: to take a store and designate it as a favorite (3)
- Unfavorite store: to take a store and designate it as not a favorite anymore (3)
- Favorite menu item: to take a menu item and designate it as a favorite (3)
- Unfavorite menu item: to take a store and designate it as not a favorite anymore (2)
- View stores: to view basic store information (2)
- View store details: to view the store detailed information (2)
- View categories: to view menu categories (2)
- View menu items: to view menu items (2)
- View menu item details: to view menu item details (2)
- View filter for stores: to view the filter for stores based on zip code (3)
- View account: to view account details (2)
- Login/Logout: to login or logout from donut shop customer account (4)

System (total: 7)

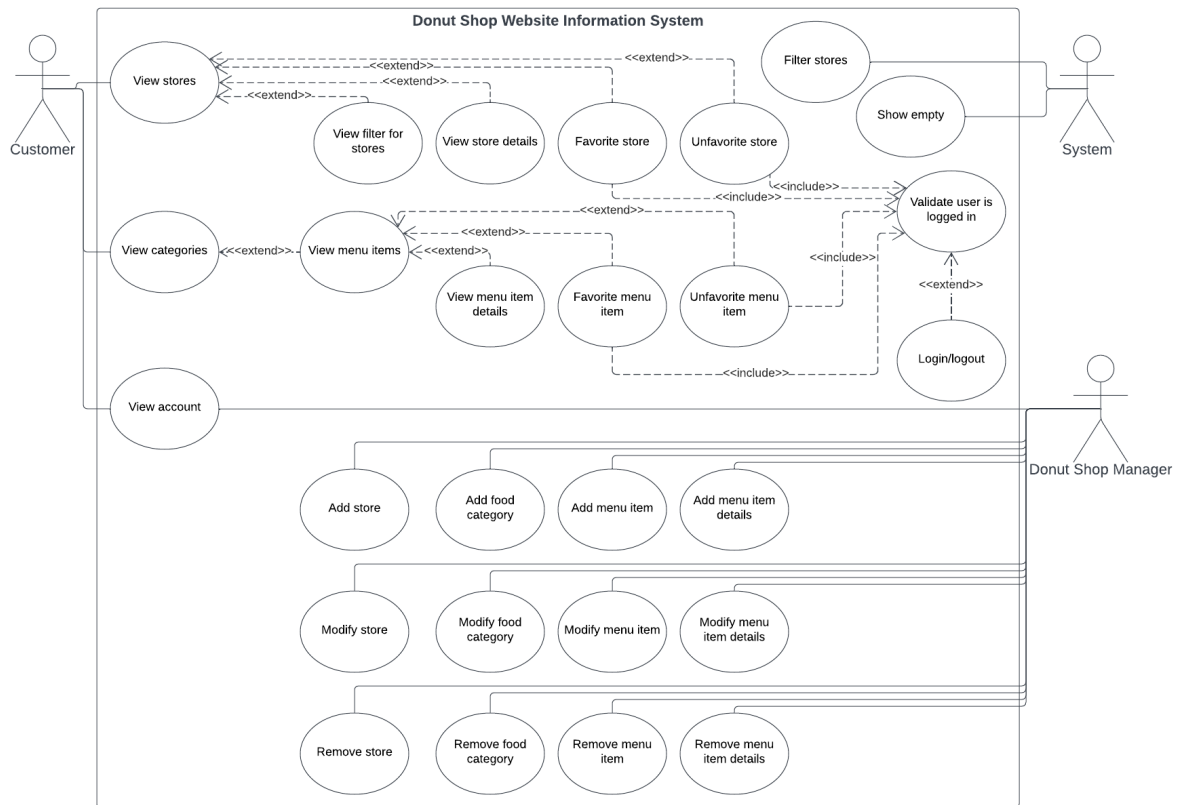
- Filter stores: to get the stores filtered (3)
- Show empty: to display the status that there are no shops (2)
- Validate user is logged in: to make sure the user is logged in to their account (2)

Use Case Diagram

The <<extend>> arrows indicate what the actor can do after another specific use case.

However, they do not have to do these use cases.

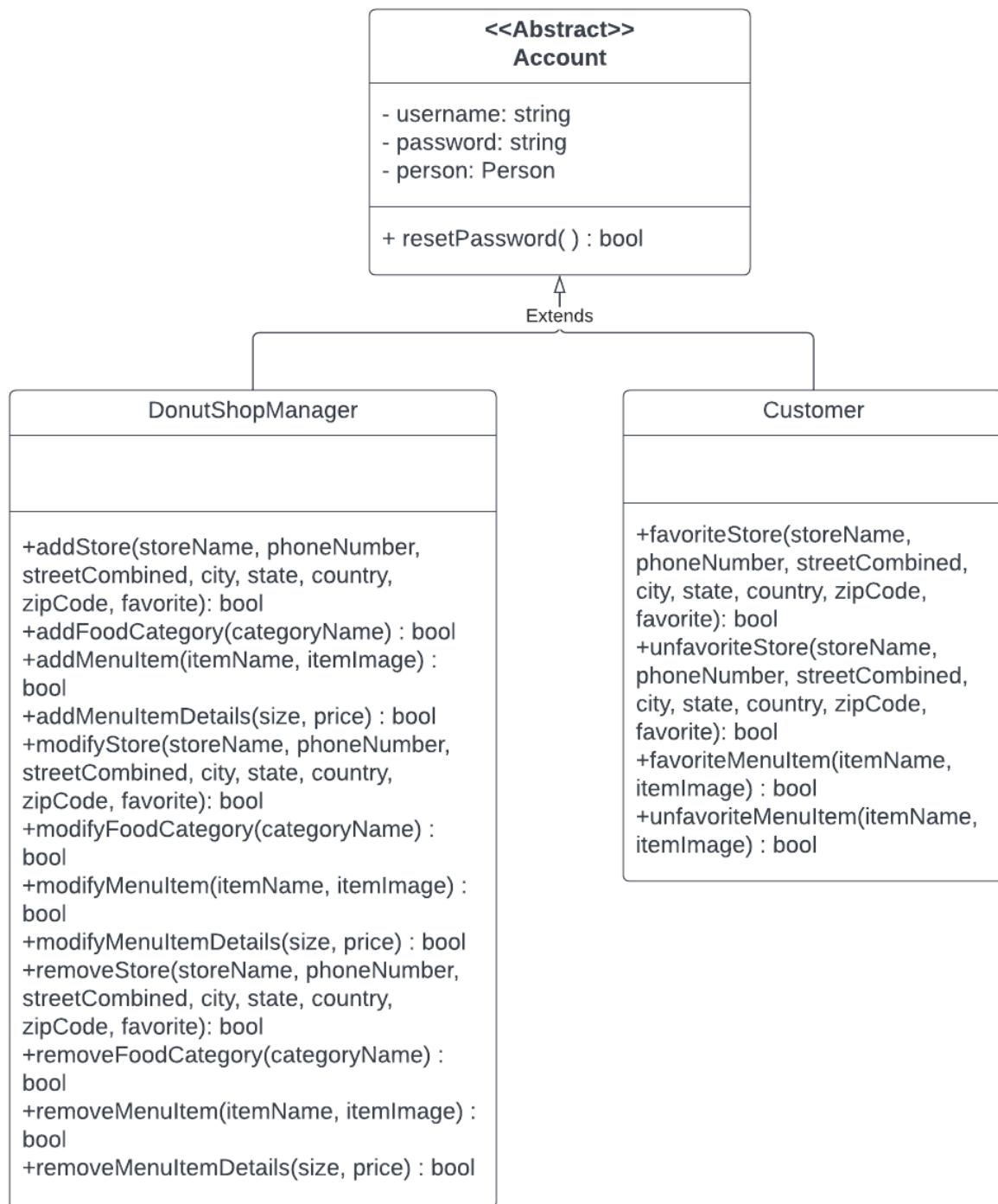
The <<include>> arrows indicate the use case that has to be done after another use case such as validating that a user is logged in to be able to favorite something.



Class Diagram

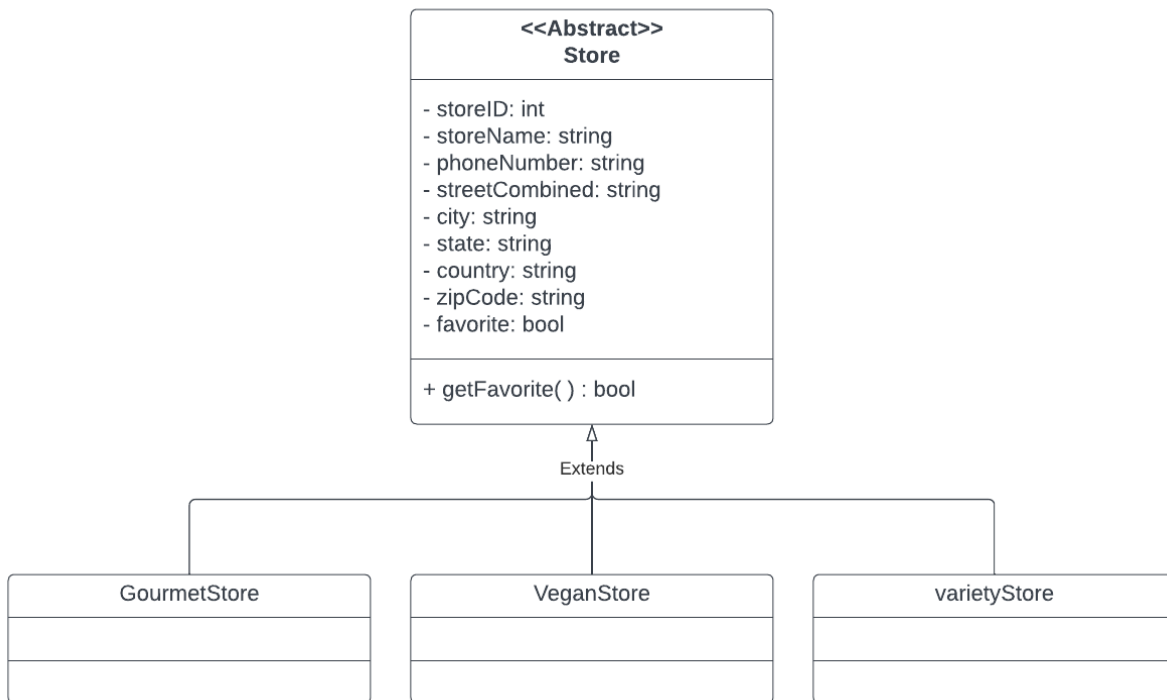
Account

Based on our requirements, there should be two types of accounts. One type of account is a Donut Shop Manager Account and another type of account is a Customer Account. We can define Account as an abstract class and both Donut Shop Manager and Customer as extended classes. Abstraction is more scalable and allows us to add more types of accounts later on.

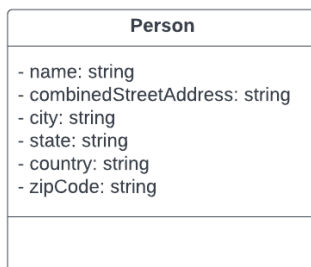


Store

Based on our requirements, there should be three types of stores. The three types of stores are Gourmet Store, Vegan Store, and Variety Store. We can define Store as an abstract class and have Gourmet Store, Vegan Store, and Variety Store as extended classes. Abstraction is more scalable and allows us to add more types of stores later on.



An example of the custom data type for Person is below:



The class Diagram is below:

The relationships between the classes consist of:

Association

- Account as a two way association with Store
- Store has a two way association with Category

Composition

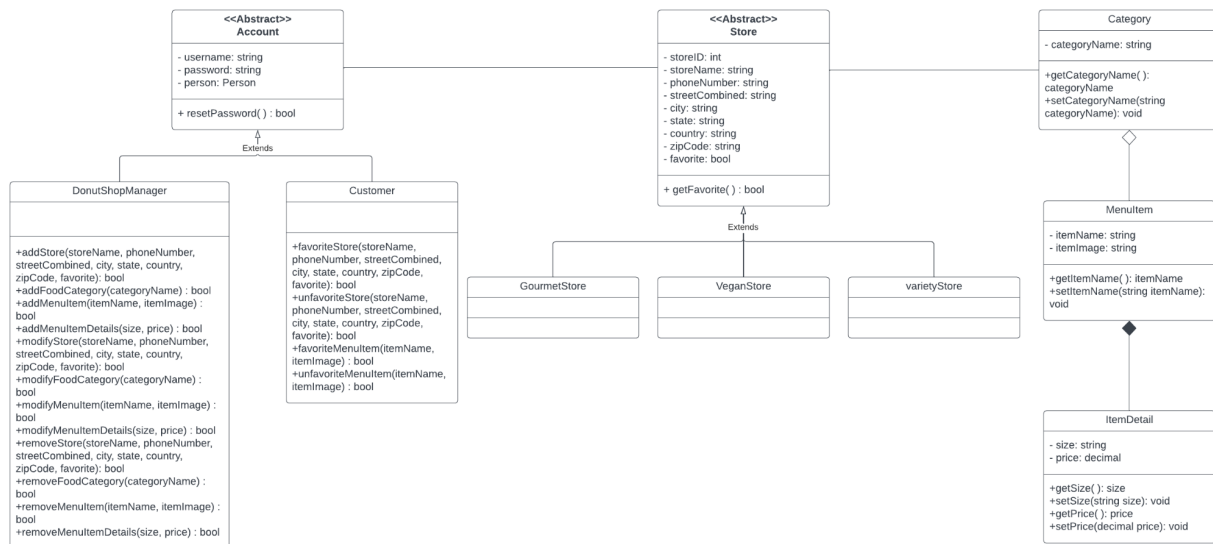
- The Menu Item includes or is composed of item Details

Aggregation

- This is a specific type of association where the Menu Items can exist independently from the Category

Inheritance

- As discussed earlier Donut Shop Manager and Customer inherit Accounts
- Gourmet Store, Vegan Store, and Variety Store inherit the Store class



System Sequence Diagrams

Sequence Diagram for Adding a Store

Actor: Donut Shop Manager (Bill)

Object: Store

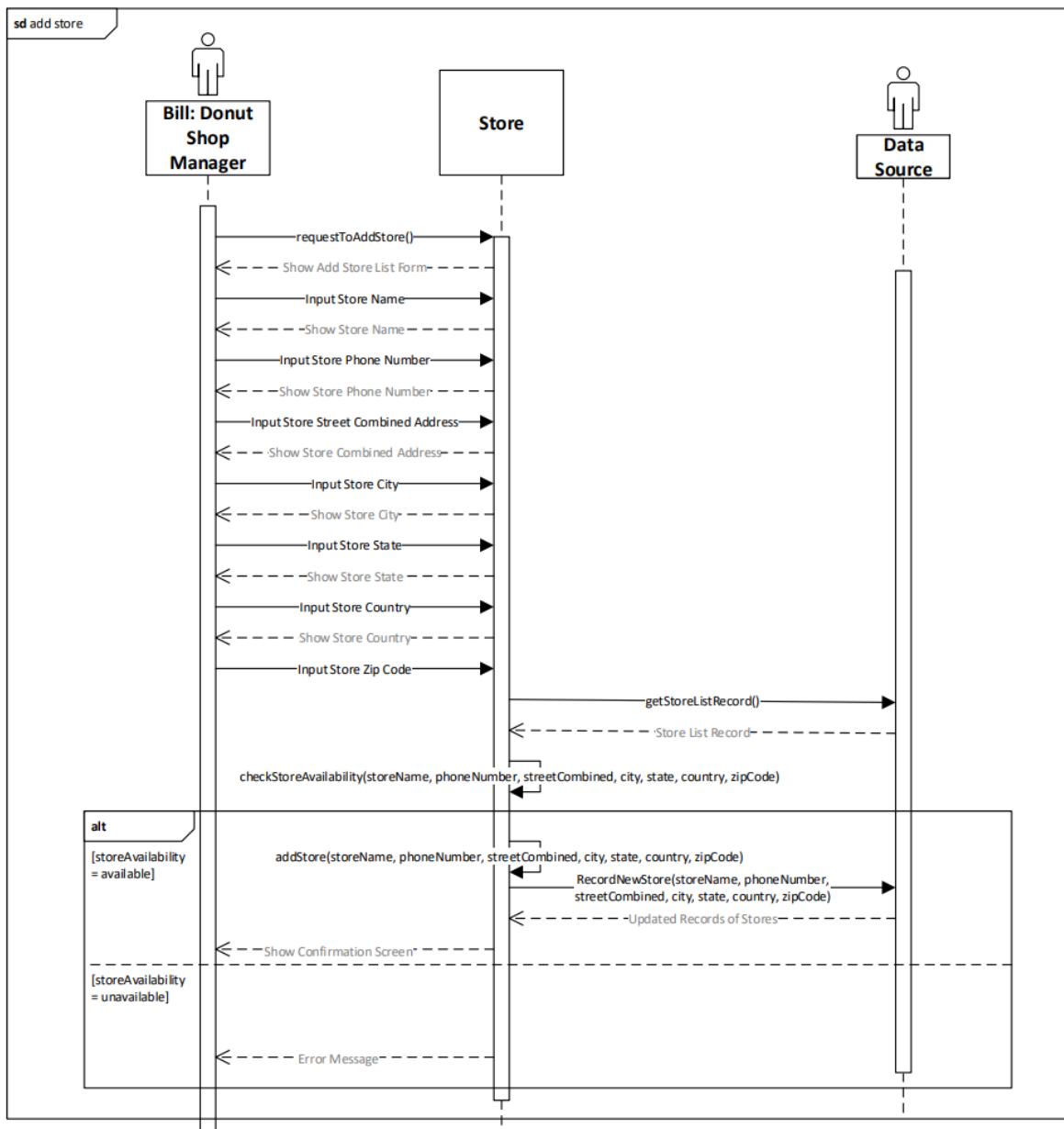
Interacts with the Data Source or Database

Steps of adding a store:

1. The Donut Shop Manager requests to add a store
2. The Donut Shop Manager Inputs Store Name, phone number, street combined address, city, state, country, and zip code information needed to make a store
3. The Store gets a list of stores from the datasource or database
4. The Store checks the store availability
5. If the store is available to add: 1. The store is added and recorded in the datasource or database 2. A confirmation screen is shown
6. If the store is not available to add and is unsuccessful: 1. The Donut Shop Manager sees an error message for the unsuccessful adding of the store and that the store is not available to be added.

The sequence diagram is shown below:

Sequence Diagram to Add a Store



Sequence Diagram for Adding a Food Category

Actor: Donut Shop Manager (Bill)

Object: Food Category

Interacts with the Data Source or Database

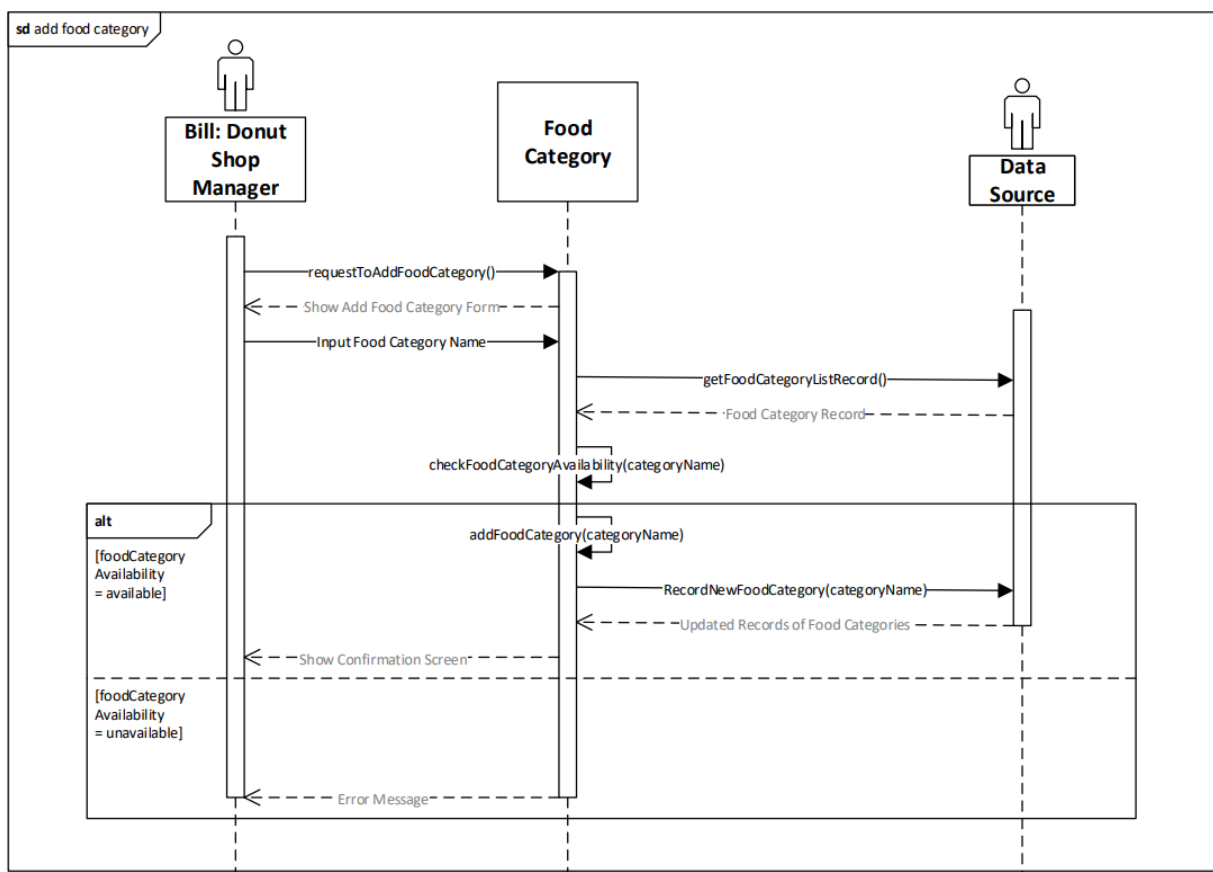
Steps of adding a Food Category:

1. The Donut Shop Manager requests to add a food category
2. The Donut Shop Manager Inputs the food category name

3. The Food Category gets a list of records of food categories from the datasource or database
4. The Food Category checks the food category availability
5. If the Food Category is available to add: 1. The food category is added and recorded in the datasource or database 2. A confirmation screen is shown
6. If the Food Category is not available to add and is unsuccessful: 1. The Donut Shop Manager sees an error message for the unsuccessful adding of the food category and that the food category is not available to be added.

The sequence diagram is shown below:

Sequence Diagram to Add a Food Category



Activity Diagrams

Activity Diagram for Favoriting a Store Use Case

States:

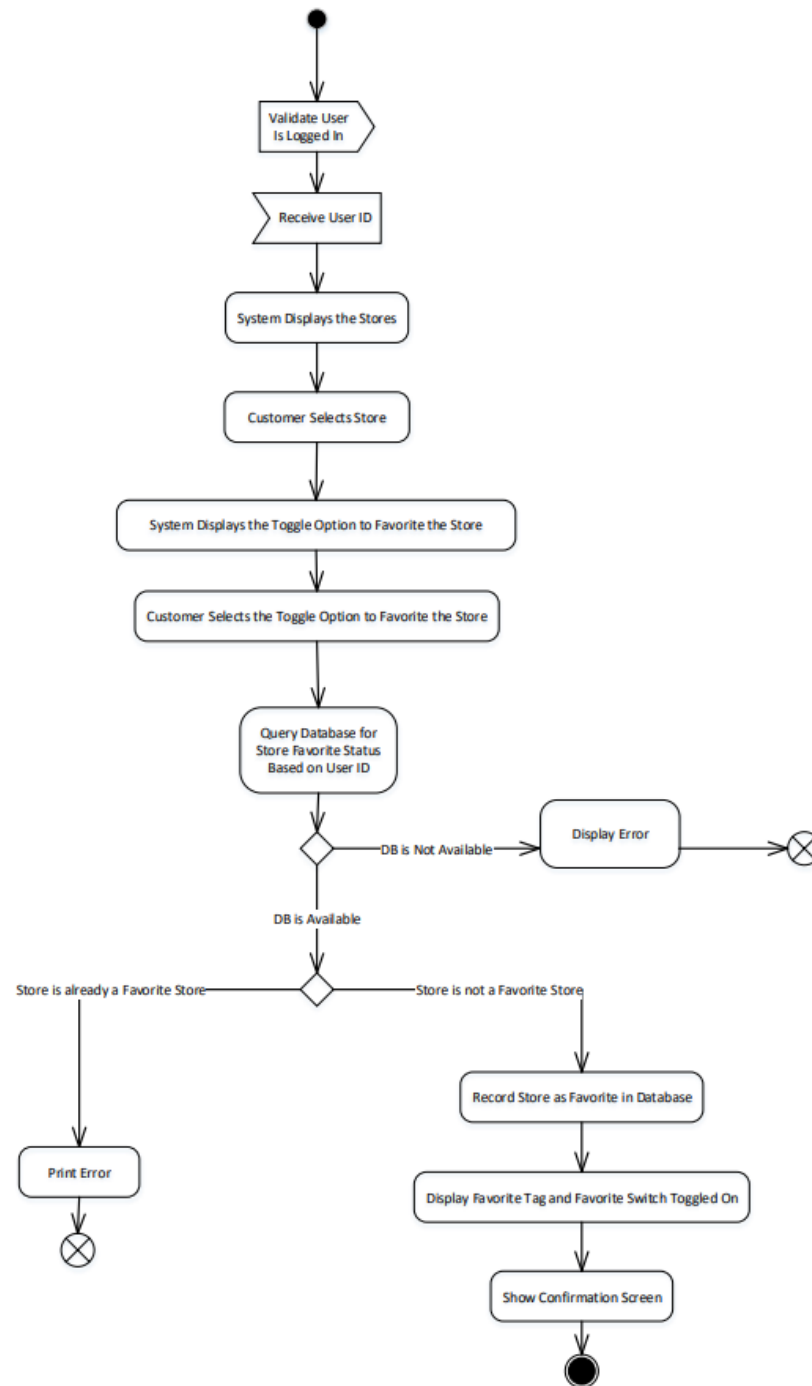
-Initial State: Make sure that the User is Logged In and then Show the Stores

-Final State: 1.The Customer sees a confirmation screen that the Store was favorited successfully and they can see the favorite tag and favorite switch toggled on. 2. The Store is already a favorite so the system prints an error for the customer to see.

Actions:

The Customer is logged in and views the stores that are displayed from the system. The customer selects the store they want to favorite and clicks the toggle option to favorite the store. The database is queried to figure out what stores the customer has favorited. The system then updates the store as a favorite if it is not already a favorite.

Activity Diagram for Favorite Store Use Case



Activity Diagram for Unfavoriting a Store Use Case

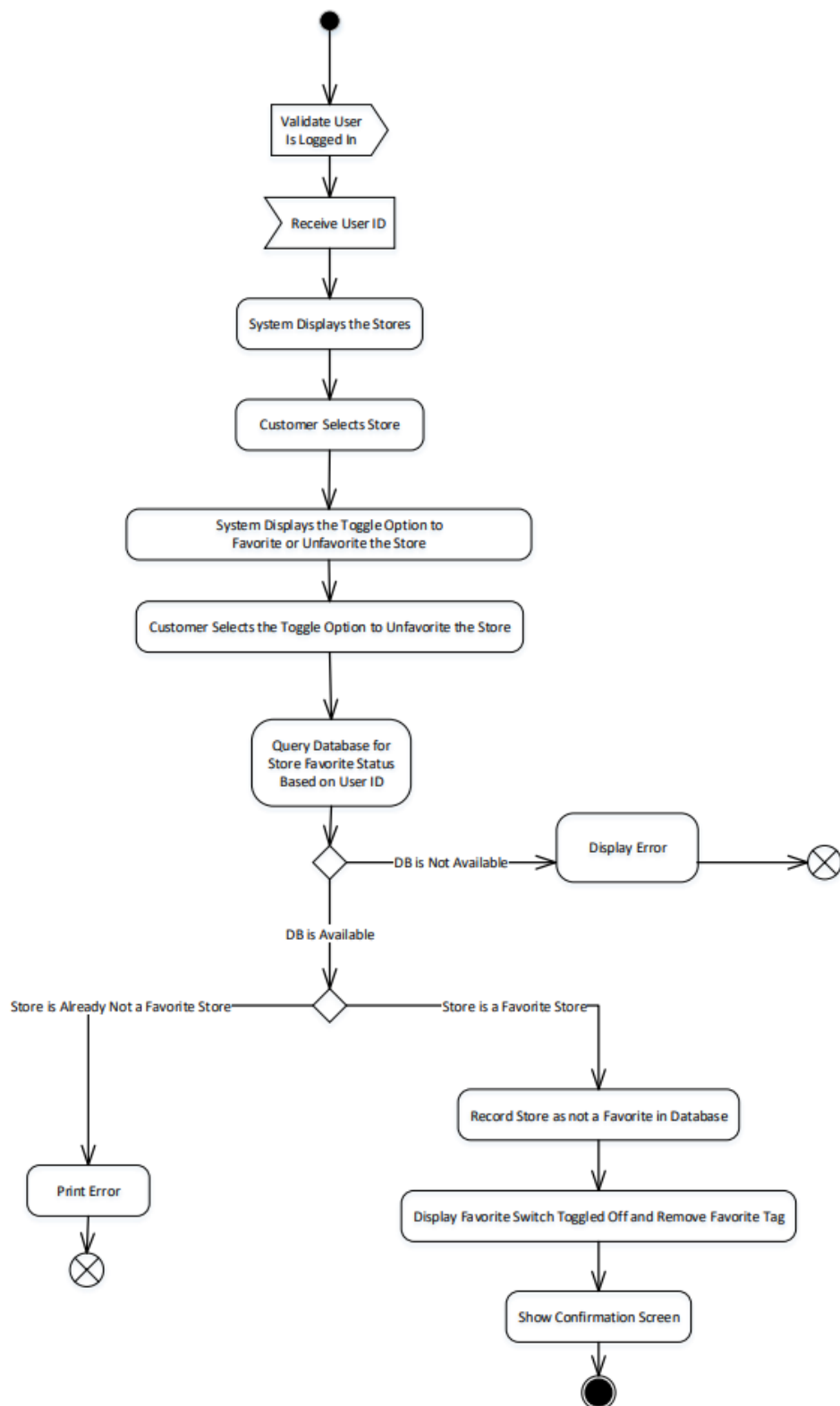
States:

- Initial State: Make sure that the User is Logged In and then Show the Stores
- Final State: 1. The Customer sees a confirmation screen that the Store was unfavorited successfully and they can see no favorite tag and the favorite switch toggled off. 2. The Store is already not a favorite so the system prints an error for the customer to see.

Actions:

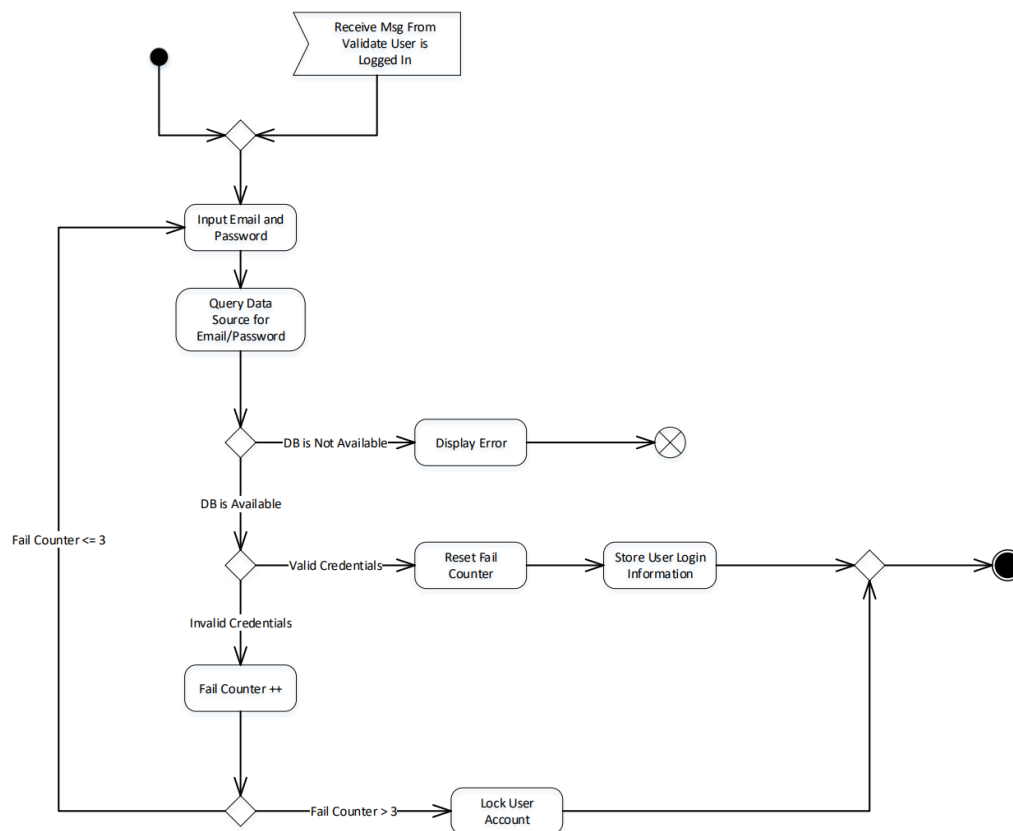
The Customer is logged in and views the stores that are displayed from the system. The customer selects the store they want to unfavorite and clicks the toggle option to unfavorite the store. The database is queried to figure out what stores the customer has favorited. The system then updates the store as not a favorite if it is a favorite.

The Activity diagram for unfavoriting a store is shown below:

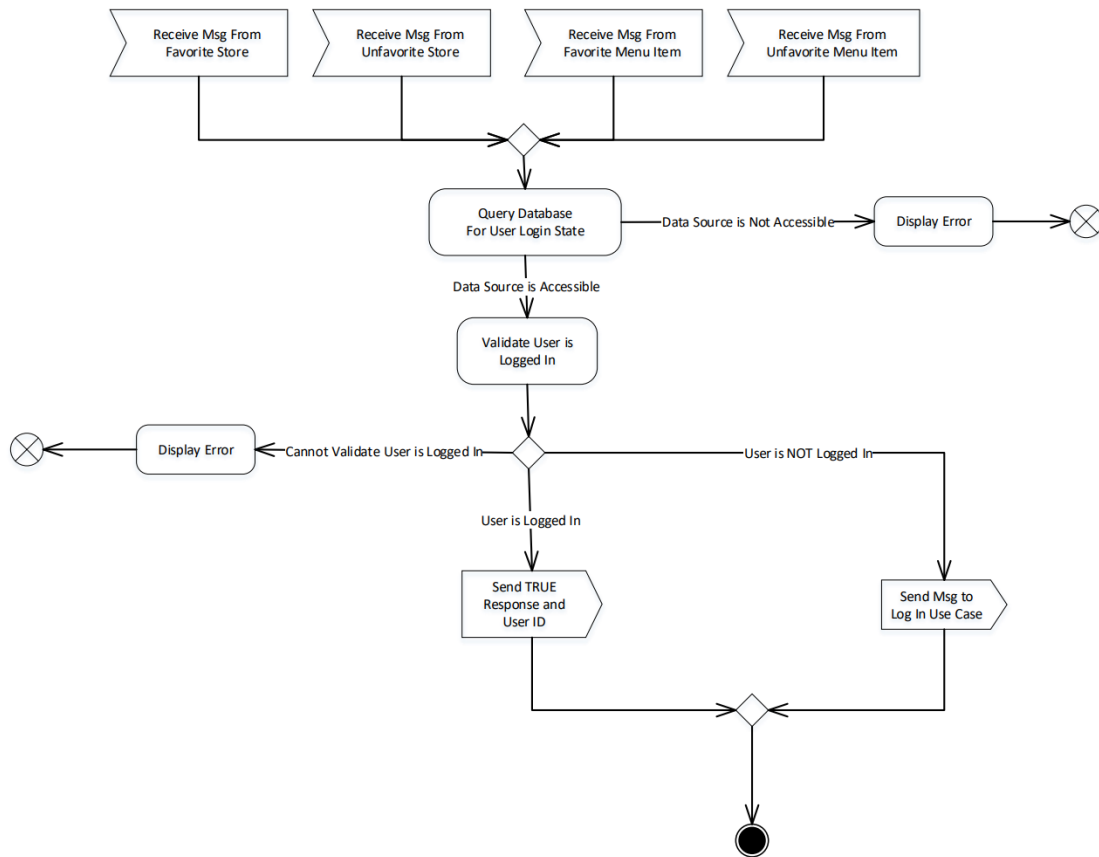
Activity Diagram for Unfavorite Store Use Case

The states and actions for the LogIn Use Case is displayed using symbols within the activity diagram below:

Activity Diagram for Log In Use Case



The states and actions for the Validate User is Logged In Use Case is displayed using symbols within the activity diagram below:

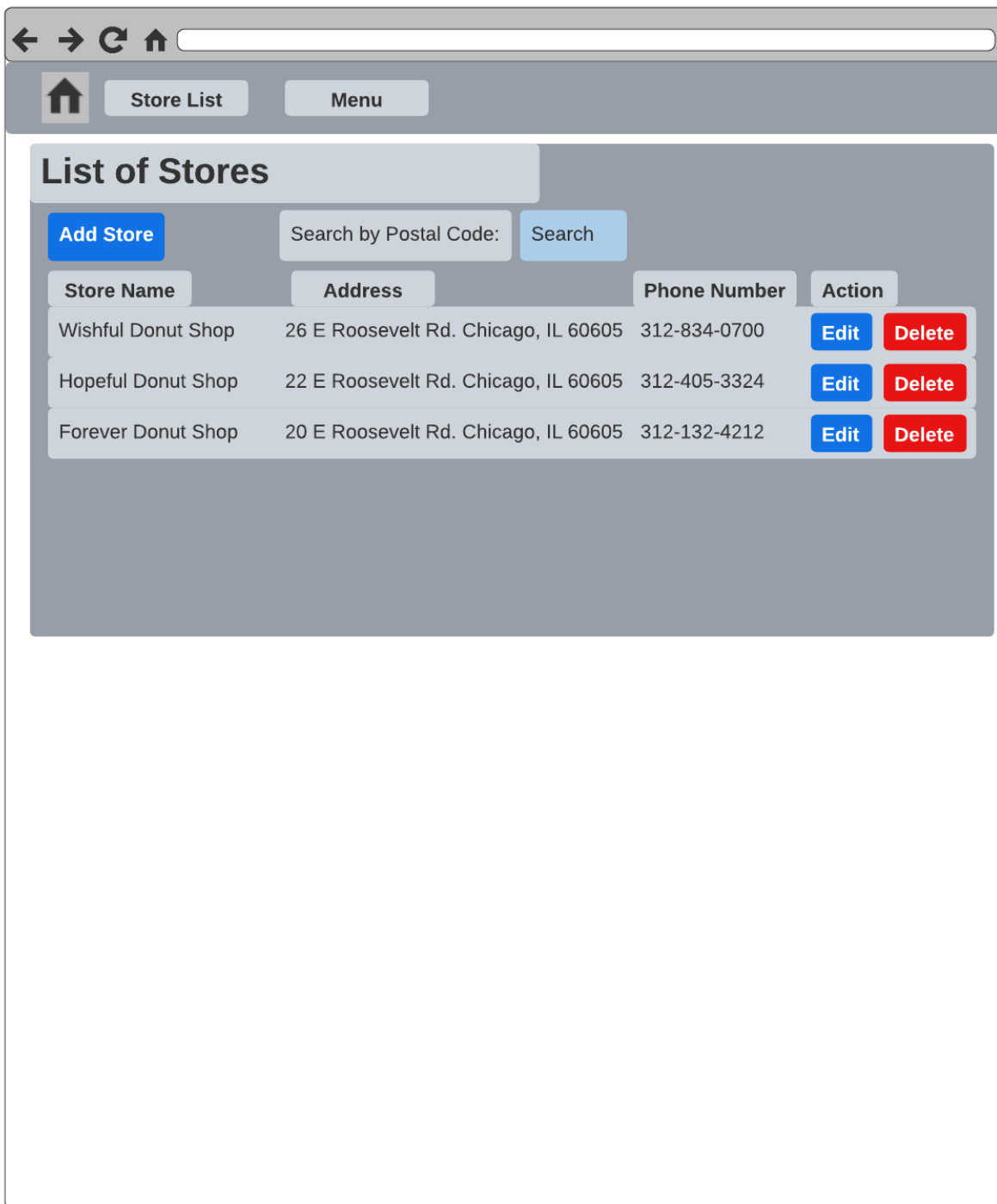
Activity Diagram for Validate User is Logged In Use Case

User Interface Specification

Preliminary Design

The User Interface for the Donut Shop Manager to view the list of stores, filter for stores use case, and click buttons to navigate to the add store use case, modify store use case, and delete store use case is shown below. The navigation bar on the top also helps navigate to the menu use cases such as the add food category use case.

List of Stores Page below:



← → ↻ ↑

Store List Menu

List of Stores

Add Store

Search by Postal Code: Search

Store Name	Address	Phone Number	Action
Wishful Donut Shop	26 E Roosevelt Rd. Chicago, IL 60605	312-834-0700	Edit Delete
Hopeful Donut Shop	22 E Roosevelt Rd. Chicago, IL 60605	312-405-3324	Edit Delete
Forever Donut Shop	20 E Roosevelt Rd. Chicago, IL 60605	312-132-4212	Edit Delete

User Interface for Add Store Use Case

How Users Enter Information: The Donut Shop Manager enters information by entering store information into the relevant text fields that are a light blue color in the add store user interface image below. The text fields have initial directions within the text fields to tell the Donut Shop Manager what content to put within the text fields such as “Enter Store Name”. This provides

directions so that the user is less likely to get confused. Once the Donut Shop Manager enters all of the store information into the relevant text fields, they can click the blue “Add Store” button to add the store to the list. If the user wants to cancel adding the store, they can click the white “Cancel” button at the bottom of the page. The buttons, text fields, and layout is shown in the mockup image below.

The mockup shows a web browser window with a navigation bar at the top containing a home icon, a 'Store List' button, and a 'Menu' button. The main content area is titled 'Add Store to List' and contains a form with the following sections and fields:

- Store Name:** A text input field with the placeholder 'Enter Store Name'.
- Location:** A section header for the following fields.
 - Store Street Address:** A text input field with the placeholder 'Enter Store Street Address'.
 - City:** A text input field with the placeholder 'Enter City'.
 - State:** A text input field with the placeholder 'Enter State'.
- Contact Information:** A section header for the following field.
 - Store Phone Number:** A text input field with the placeholder 'Enter Store Phone Number'.
- Buttons:** At the bottom of the form are two buttons: a blue 'Add Store' button and a white 'Cancel' button.

How the Results are Displayed for Users: The Results of adding the store are displayed to the Donut Shop Manager by showing success or error messages depending on if the store was added successfully or not. Image examples of stores successfully added and error messages are shown below. The results of the store being added to the list are also displayed in the Store List page as a new row.

Navigation Paths: The navigation path that the Donut Shop Manager follows to add a store includes that the user needs to log in and enter their username and password into the text fields. Then, the user clicks/navigates to the “Store List” button in the navigation bar at the top of the page. Then, they are taken to the “List of Stores” page where they are shown a list of stores. After that they click the blue “Add Store” button near the top of the screen which navigates them to the “Add Store to list” page. This is the page where they enter the store information and click the blue “Add Store” button to add the store to the list.

This is the navigation path: Login > Home Page > Store List > Add Store

←

→

↶

⬆

🏠

Store List

Menu

Add Store to List

Store Name:

Enter Store Name

Location

Store Street Address:

Enter Store Street Address

City:

Enter City

State:

Enter State

Contact Information

Store Phone Number:

Enter Store Phone Number

Add Store

Cancel

Store Added Successfully

←

→

↺

⬆

🏠

Store List

Menu

Add Store to List

Store Name:

Enter Store Name

Location

Store Street Address:

Enter Store Street Address

City:

Enter City

State:

Enter State

Contact Information

Store Phone Number:

Enter Store Phone Number

Missing Field: Please Enter 10 digits

Add Store

Cancel

Error: One or more fields have an error.
Please check and try again.

←

→

↻

↑

🏠

Store List

Menu

Add Store to List

Store Name:

Enter Store Name

Location

Store Street Address:

Enter Store Street Address

City:

Enter City

State:

Enter State

Contact Information

Store Phone Number:

Enter Store Phone Number

Add Store

Cancel

Error: Store Already Exists and is Unavailable to be Added.

← → ↻ ⬆

Store List Menu

Add Store to List

Store Name: Enter Store Name

Location

Store Street Address: Enter Store Street Address

City: Enter City

State: Enter State

Contact Information

Store Phone Number: Enter Store Phone Number

Add Store Cancel

Error: Unable to Add Store: Try Again Later

User Interface for Modify/Edit Store Use Case

How Users Enter Information: The Donut Shop Manager enters information by entering store information into the relevant text fields that are a light blue color in the Edit Store Information store user interface image below. The text fields are already filled in with the store information based on which store's edit button they clicked in the "List of Stores" page. The Donut Shop

Manager would then modify the text such as “Wishful Donut Shop” to change to a different store name. Next to the text fields there is text to indicate what store information to edit within the light blue text field. Once the Donut Shop Manager edits all of the store information in the relevant text fields, they can click the blue “Submit Edit” button to Edit the store in the list. If the user wants to cancel Editing the store, they can click the white “Cancel” button at the bottom of the page. The buttons, text fields, and layout is shown in the mockup image below.

How the Results are Displayed for Users: The Results of editing/modifying the store are displayed to the Donut Shop Manager by showing success or error messages depending on if the store was edited successfully or not. These success and error messages would be very similar to the images I showed previously except it would say “store edited successfully. The results of the store being edited/modified in the list are also displayed in the Store List page as an edited row.

Navigation Paths: The navigation path that the Donut Shop Manager follows to modify/edit a store includes that the user needs to log in and enter their username and password into the text fields. Then, the user clicks/navigates to the “Store List” button in the navigation bar at the top of the page. Then, they are taken to the “List of Stores” page where they are shown a list of stores. After that they click the blue “Edit” button in the row that corresponds to the store information that they want to edit which navigates them to the “Edit Store Information” page. This is the page where they edit the store information and click the blue “Submit Edit” button to edit the store in the list.

This is the navigation path: Login > Home Page > Store List > Edit

The screenshot shows a web browser window with a navigation bar at the top containing a home icon, a 'Store List' button, and a 'Menu' button. Below the navigation bar is a form titled 'Edit Store Information'. The form contains the following fields and sections:

- Store Name:** A text input field containing 'Wishful Donut Shop'.
- Location:** A section header followed by three text input fields:
 - Store Street Address:** '26 E. Roosevelt Rd.'
 - City:** 'Chicago'
 - State:** 'IL'
- Contact Information:** A section header followed by one text input field:
 - Store Phone Number:** '312-834-0700'
- Buttons:** At the bottom of the form are two buttons: 'Submit Edit' (blue) and 'Cancel' (white with blue text).

User Interface for Remove/Delete Store Use Case

How Users Enter Information: The Donut Shop Manager does not really need to enter information to delete stores. However, they do need to click on the red “Delete” button that corresponds and is in the same row as the store information that they want to delete in the “List of Stores” page. Then, the user is shown a confirmation screen to make sure that the user

wants to delete the store. This page is called "Delete Store". There is a red "Delete Store" button that the user needs to click to confirm the delete. If the user wants to cancel deleting the store, they can click the white "Cancel" button at the bottom of the page. The buttons, text fields, and layout is shown in the mockup image below.

How the Results are Displayed for Users: The Results of deleting the store are displayed to the Donut Shop Manager by showing success or error messages depending on if the store was deleted successfully or not. These success and error messages would be very similar to the images I showed previously except it would say "store deleted successfully". The results of the store being deleted in the list are also displayed in the Store List page as a row that is gone/deleted.

Navigation Paths: The navigation path that the Donut Shop Manager follows to delete a store includes that the user needs to log in and enter their username and password into the text fields. Then, the user clicks/navigates to the "Store List" button in the navigation bar at the top of the page. Then, they are taken to the "List of Stores" page where they are shown a list of stores. After that they click the red "Delete" button in the row that corresponds to the store information that they want to delete which navigates them to the "Delete Store" page. This is the page where they delete the store information and click the red "Delete Store" button to delete the store in the list.

This is the navigation path: Login > Home Page > Store List > Delete

←

→

↶

⬆

⬆

Store List

Menu

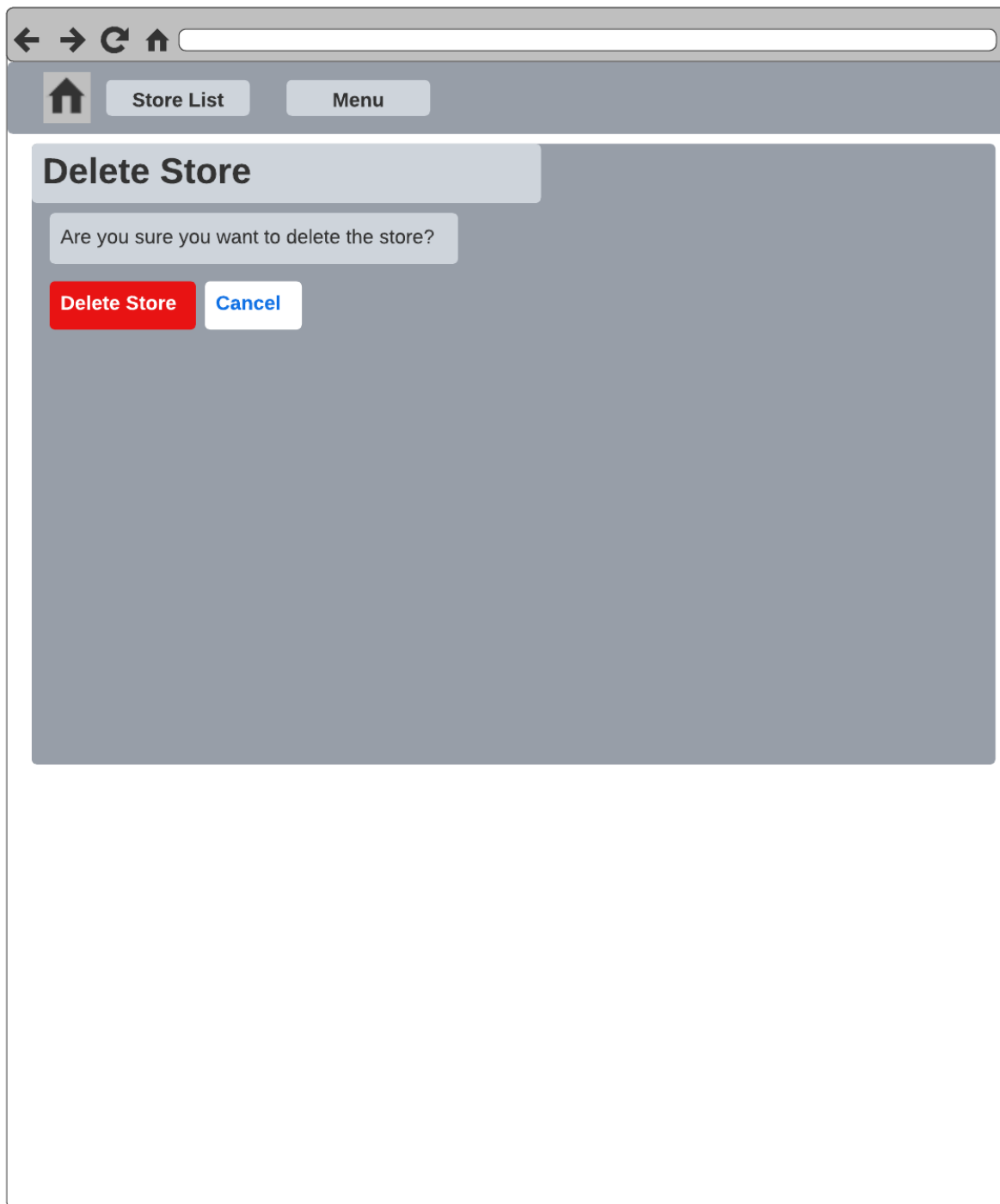
List of Stores

Add Store

Search by Postal Code:

Search

Store Name	Address	Phone Number	Action
Wishful Donut Shop	26 E Roosevelt Rd. Chicago, IL 60605	312-834-0700	<div>Edit</div> <div>Delete</div>
Hopeful Donut Shop	22 E Roosevelt Rd. Chicago, IL 60605	312-405-3324	<div>Edit</div> <div>Delete</div>
Forever Donut Shop	20 E Roosevelt Rd. Chicago, IL 60605	312-132-4212	<div>Edit</div> <div>Delete</div>



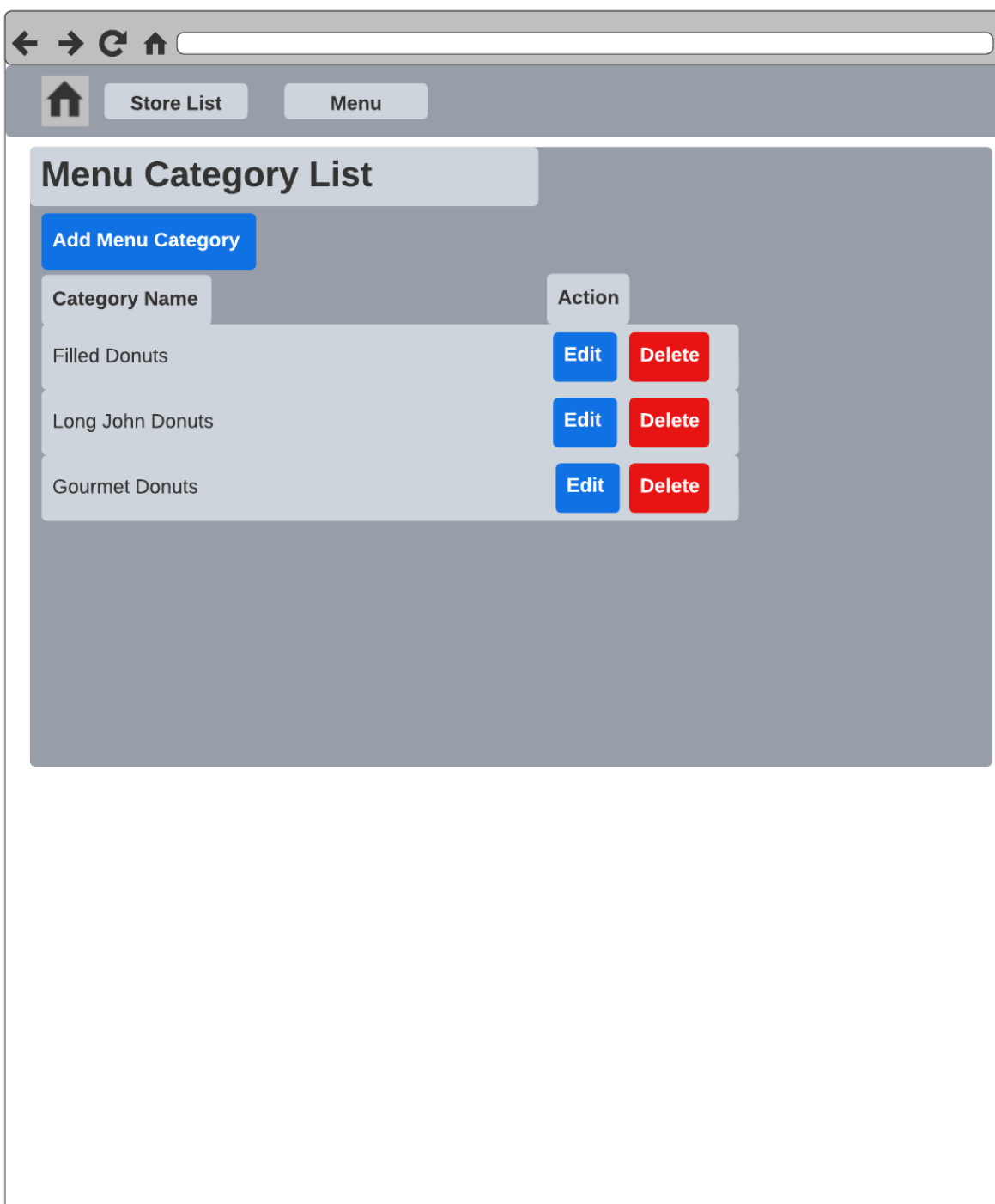
User Interface for Remove/Delete Food Category Use Case

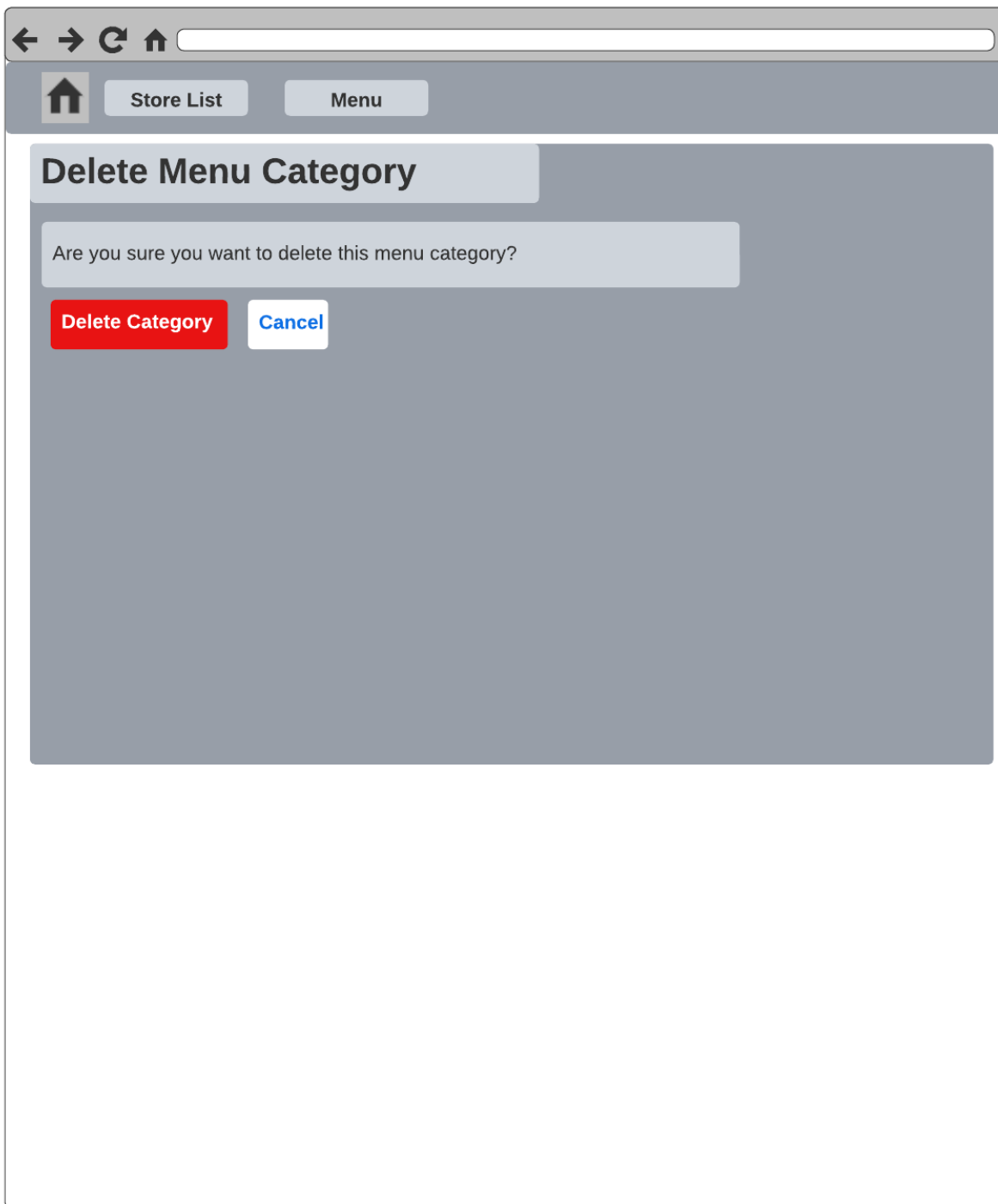
How Users Enter Information: The Donut Shop Manager does not really need to enter information to delete stores. However, they do need to click on the red “Delete” button that corresponds and is in the same row as the menu category information that they want to delete in the “Menu Category List” page. Then, the user is shown a confirmation screen to make sure

that the user wants to delete the menu category. This page is called “Delete Menu Category”. There is a red “Delete Category” button that the user needs to click to confirm the delete. If the user wants to cancel deleting the menu category, they can click the white “Cancel” button at the bottom of the page. The buttons, text fields, and layout is shown in the mockup image below.

How the Results are Displayed for Users: The Results of deleting the menu category are displayed to the Donut Shop Manager by showing success or error messages depending on if the store was deleted successfully or not. These success and error messages would be very similar to the images I showed previously except it would say “menu category deleted successfully”. The results of the menu category being deleted in the list are also displayed in the Menu Category List page as a row that is gone/deleted.

Navigation Paths: The navigation path that the Donut Shop Manager follows to delete a menu category includes that the user needs to log in and enter their username and password into the text fields. Then, the user clicks/navigates to the “Menu” button in the navigation bar at the top of the page. Then, they are taken to the “Menu Category List” page where they are shown a list of menu categories. After that they click the red “Delete” button in the row that corresponds to the menu category information that they want to delete which navigates them to the “Delete menu category” page. This is the page where they delete the menu category information and click the red “Delete menu category” button to delete the menu category in the list. This is the navigation path: Login > Home Page > Menu > Delete





User Interface for Add Menu Category Use Case

How Users Enter Information: The Donut Shop Manager enters information by entering menu category information (category name) into the relevant text field that is a light blue color in the add menu category user interface image below. The text field has initial directions within the text field to tell the Donut Shop Manager what content to put within the text fields such as “Enter

Menu Category Name”. This provides directions so that the user is less likely to get confused. Once the Donut Shop Manager enters all of the menu category information into the relevant text field, they can click the blue “Add Menu Category” button to add the menu category to the list. If the user wants to cancel adding the menu category, they can click the white “Cancel” button at the bottom of the page. The buttons, text fields, and layout is shown in the mockup image below.

How the Results are Displayed for Users: The Results of adding the menu category are displayed to the Donut Shop Manager by showing success or error messages depending on if the menu category was added successfully or not. These success and error messages would be very similar to the images I showed previously except it would say “menu category added successfully”. The results of the menu category being added to the list are also displayed in the Menu Category List page as a new row.

Navigation Paths: The navigation path that the Donut Shop Manager follows to add a menu category includes that the user needs to log in and enter their username and password into the text fields. Then, the user clicks/navigates to the “Menu” button in the navigation bar at the top of the page. Then, they are taken to the “Menu Category List” page where they are shown a list of menu categories. After that they click the blue “Add Menu Category” button near the top of the screen which navigates them to the “Add Menu Category” page. This is the page where they enter the menu category information (category name) and click the blue “Add Menu Category” button to add the Menu Category to the list.

This is the navigation path: Login > Home Page > Menu > Add Menu Category

The screenshot shows a web browser window with a navigation bar at the top. The navigation bar contains a home icon, a 'Store List' button, and a 'Menu' button. Below the navigation bar, there is a modal dialog titled 'Add Menu Category to List'. Inside the modal, there is a form with a label 'Menu Category Name:' followed by a text input field containing the placeholder text 'Enter Menu Category Name'. Below the input field, there are two buttons: a blue 'Add Menu Category' button and a white 'Cancel' button with a blue border. The background of the modal is a solid gray color.

User Effort Estimation

Usage Scenario of Adding a new Store (called Wishful Donut Shop with an address of 26 E Roosevelt Rd. Chicago, IL 60605 and a phone number of 312-834-0700):

Sequence of Events:

The Donut Shop Manager will need to click to log in to their account. They will then need to enter their username and password into the text fields. Then, the user clicks the "Store List" button in the navigation bar at the top of the page. Then, they are taken to the "List of Stores" page where they are shown a list of stores. After that they click the blue "Add Store" button near the top of the screen which navigates them to the "Add Store to list" page. This is the page where they enter the store information. They will need to click each of the five text fields corresponding to each store information section and then input the store information by typing on the keyboard. In this scenario, they click on the "enter store name" text field and enter "Wishful Donut Shop". After that they click on the "enter store address" text field and enter "26 E Roosevelt Rd." After that they click on the "enter city" text field and enter "Chicago". After that they click on the "enter state" text field and enter "IL". After that they click on the "enter store phone number" text field and enter "312-834-0700". Then they click the blue "Add Store" button to add the store to the list.

This is the navigation path: Login > Home Page > Store List > Add Store

Number of Clicks and Keystrokes:

Click "LogIn" + click "enter Username" text field + 2 keystrokes for username of "hi" + click "enter Password" text field + 2 keystrokes for password of "hi" + click "LogIn Button" + Click "Store List" button in Navigation bar at the top of the screen + click "Add Store" + 5 clicks (one for clicking each store information text fields) + 60 keystrokes to type all of the store information into each text field (Wishful Donut Shop 26 E Roosevelt Rd. Chicago, IL 60605 312-834-0700 + click "Add Store"

$1 + 1 + 2 + 1 + 2 + 1 + 1 + 1 + 5 + 60 + 1 = 76$ clicks and keystrokes total

Number of Clicks:

Click "LogIn" + click "enter Username" text field + click "enter Password" text field + click "LogIn Button" + Click "Store List" button in Navigation bar at the top of the screen + click "Add Store" + 5 clicks (one for clicking each store information text fields) + click "Add Store"

$1 + 1 + 1 + 1 + 1 + 1 + 5 + 1 = 12$ clicks total

Number of Clicks if Already Logged In:

Click "Store List" button in Navigation bar at the top of the screen + click "Add Store" + 5 clicks (one for clicking each store information text fields) + click "Add Store"

$1 + 1 + 5 + 1 = 8$ clicks total

Number of Clicks if Already Logged In and just want to get to the Add Store Page:

Click "Store List" button in Navigation bar at the top of the screen + click "Add Store"

$1 + 1 = 2$ clicks total

Usage Scenario of Removing/Deleting a Store (called Wishful Donut Shop with an address of 26 E Roosevelt Rd. Chicago, IL 60605 and a phone number of 312-834-0700):

The Donut Shop Manager will need to click to log in to their account. They will then need to enter their username and password into the text fields. Then, the user clicks the “Store List” button in the navigation bar at the top of the page. Then, they are taken to the “List of Stores” page where they are shown a list of stores. After that they click the red “Delete” button in the row that corresponds to the store information (Wishful Donut Shop) that they want to delete which navigates them to the “Delete Store” page. This is the page where they delete the store information and click the red “Delete Store” button to delete the store in the list.

This is the navigation path: Login > Home Page > Store List > Delete

Number of Clicks:

Click “Login” + click “enter Username” text field + click “enter Password” text field + click “Login Button” + Click “Store List” button in Navigation bar at the top of the screen + click “Delete” button in the row with the store they want to delete + click “Delete Store” button

$1 + 1 + 1 + 1 + 1 + 1 + 1 = 7$ clicks total

Number of Clicks if Already Logged In:

Click “Store List” button in Navigation bar at the top of the screen + click “Delete” button in the row with the store they want to delete + click “Delete Store” button

$1 + 1 + 1 = 3$ clicks total

Number of Clicks if Already Logged In and just want to get to the Delete Store Page:

Click “Store List” button in Navigation bar at the top of the screen + click “Delete” button in the row with the store they want to delete

$1 + 1 = 2$ clicks total

Usage Scenario of Removing/Deleting a Menu/Food Category (called Filled Donuts):

The Donut Shop Manager will need to click to log in to their account. They will then need to enter their username and password into the text fields. Then, the user clicks the “Menu” button in the navigation bar at the top of the page. Then, they are taken to the “Menu Category List” page where they are shown a list of menu categories. After that they click the red “Delete” button in the row that corresponds to the menu category information (Filled Donuts) that they want to delete which navigates them to the “Delete Menu Category” page. This is the page where they delete the menu category information and click the red “Delete Category” button to delete the menu category in the list.

This is the navigation path: Login > Home Page > Menu > Delete

Number of Clicks:

Click “Login” + click “enter Username” text field + click “enter Password” text field + click “Login Button” + Click “Menu” button in Navigation bar at the top of the screen + click “Delete” button in the row with the menu category they want to delete + click “Delete Category” button

$1 + 1 + 1 + 1 + 1 + 1 + 1 = 7$ clicks total

Number of Clicks if Already Logged In:

Click "Menu" button in Navigation bar at the top of the screen + click "Delete" button in the row with the menu category they want to delete + click "Delete Category" button

$1 + 1 + 1 = 3$ clicks total

Number of Clicks if Already Logged In and just want to get to the Delete Store Page:

Click "Menu" button in Navigation bar at the top of the screen + click "Delete" button in the row with the menu category they want to delete

$1 + 1 = 2$ clicks total

Project Proposal

Problem Statement

The donut shop website information system is useful to help fix the issue of customers not having access to shop information in an organized, easy to use, fast, and useful manner. This website information system will allow customers to see store information digitally around the world and while on the go with a mobile device. This system will address the issues of customers not knowing the location of stores nearby them, the store hours, and details about menu items. This system provides an opportunity for the business to grow, expand, and possibly get more customers by being discoverable on the Internet. Also, the system will help the donut shop stay competitive with other businesses using websites. This website information system can be extended to be used for any restaurant or business.

Objectives of the System

The objectives of the system focus on trying to help manage and keep basic shop and menu information up to date. This system will help make the shop information easily accessible and organized. This system will help shop owners and managers easily be able to add new shops and remove shops that have closed down. This would increase transparency by making shop hours, locations, and recent shop openings/closings easily findable. Shop customers will be able to favorite stores and find stores nearby.

System Requirements

The requirements for the system are that the system must be able to let customers and shop owners do these activities:

What customers can do with the system:

- Favorite stores
- Favorite categorize menu items
- View stores nearby based on the city or postal code
- Keyword search and filter for stores based on the city or postal code
- View a list of all stores
- View categories, menu items, and their details to find items that the customer wants
- View basic store information

What Shop owners can do with the system:

- Add stores with store details and locations
- Add food categories, menu items, and details
- Modify stores with store details and locations
- Modify food categories, menu items, and details
- Remove stores with store details and locations
- Remove food categories, menu items, and details

Typical Customers

The system will have different access and features tailored to these different groups:

- Customers (people who buy food at the donut shop or like donuts)
- Donut shop owners or managers

Project Planning and Development Approach

The software, hardware, and network requirements for project planning are listed below.

The development approach and technical stack is in the software section.

1. Software
 - a. Front-end: **HTML/ CSS**
 - b. Back-end: **PHP**
 - c. Database: MySQL
2. Hardware
 - a. Desktop/PC
 - b. Mobile Devices (cellphone, tablet)
3. Network
 - a. Good Internet speed
 - b. As little website downtime as possible
 - c. Scheduled website maintenance once a week

Project Plan

Development Plan

Week 1: Solidify what front-end, back-end, and database programs I am going to use.

Week 2: Set up a project development environment by connecting front-end, back-end, and database software together, figure out the structure and framework for the system.

Week 3: Create the login for the system so customers and shop owners can login.

Week 4: Create the registration for new customers and shop owners to make a login or account.

Week 5: Design the features for shop owners to add, modify, or remove stores, store details (store name, phone number, street address, city, state, country, zip code), food categories, menu items, and menu item details.

Week 6: Implement the features for shop owners to add, modify, or remove stores, store details (store name, phone number, street address, city, state, country, zip code), food categories, menu items, and menu item details.

Week 7: Continue to implement the features for shop owners to add, modify, or remove stores, store details (store name, phone number, street address, city, state, country, zip code), food categories, menu items, and menu item details.

Week 8: Verify the features work, record, and turn in current progress for the midterm.

Week 9: Iterate and improve current features for the shop owners.

Week 10: Design the features for customers to be able to view the stores and filter for stores based on their city or postal code. Design the features for customers to view basic store, menu category, and menu item information

Week 11: Implement the features for customers to be able to view the stores and filter for stores based on their city or postal code. Implement the features for customers to view basic store, menu category, and menu item information

Week 12: Design the features for customers to be able to favorite stores and menu items.

Week 13: Implement the features for customers to be able to favorite stores and menu items.

Week 14: Debug and test to make sure all features work as intended.

Week 15: Test and record the demonstration and progress made for the final presentation.

Reference

Submitted Documents in Canvas

1. Proposal
2. Customer Problem Statements and System Requirements
3. Functional Requirements Specification
4. System Sequence Diagram and Activity Diagram
5. User Interface Specification