Team A

CSc 322 Final Project

Software Requirement Specification

For Bhai Brothers LLC

By: Abdul Andha, Baljinder Hothi, Jay Noppone Pornpitaksuk,

Abrar Habib and Meftahul Ahsan

Version 2.0

Revision History

Date	Version	Description	Author
03/24/24	1.0	Added actors and use case descriptions in section 2.1	Abdul Andha
3/25/24	1.0	Added use-case diagram in section 2.1	Abdul Andha
03/26/24	1.0	Added definitions in section 1.3	Abrar Habib
03/26/24	1.0	Added dependencies and assumptions in section 2.2	Abrar Habib
03/26/24	1.0	Added section 1 and section 1.1	Baljinder Hothi
03/25/24	1.0	Added and revised overview and references for section 1.4 and 1.5	Meftahul Ahsan
03/26/24	1.0	Revised supplementary specification for section 3.2	Meftahul Ahsan
03/26/24	1.0	Added the scope and use-case reports (section 3.1)	Jay Noppone
03/26/24	1.0	Reworked and added more information to sections 1 and 1.1	Baljinder Hothi
04/15/24	2.0	Added Pseudo Codes for most methods	Baljinder Hothi
04/19/24	2.0	Added use case, collaboration class diagrams	Jay Noppone
04/19/24	2.0	Revised Pseudo Codes for most methods	Baljinder Hothi
04/19/24	2.0	Created a Figma based GUI of the system	Abdul Andha
04/19/24	2.0	Implemented a navbar on the webpage	Abdul Andha
04/19/24	2.0	Added petrinets, collaboration class and ER diagram	Meftahul Ahsan
04/19/24	2.0	Revised psuedocode for design system	Meftahul Ahsan
04/19/24	2.0	Updated table of contents	Meftahul Ahsan and Baljinder Hothi
04/19/24	2.0	Relocated sections to appropriate places	Baljinder Hothi
04/19/24	2.0	Updated Team Concern and Meeting Notes	Meftahul Ahsan and Baljinder Hothi

Table of Contents

Software Requirements Specification

1. Introduction

This Software Requirements Specification (SRS) document is prepared for the BHAI Brothers project, which focuses on developing a Computerized Food Order and Delivery System as part of our academic endeavor. The document provides a comprehensive overview of the project's objectives, including the enhancement of the food ordering and delivery process for better efficiency and user experience. It describes the system's planned features and the roles of various stakeholders such as chefs, delivery personnel, and customers. The goal of this document is to ensure a clear understanding of the project scope and the solutions we propose to implement, making it a crucial reference for all project participants.

1.1 Purpose

This Software Requirements Specification (SRS) for the Computerized Food Order and Delivery System project outlines the development of a platform designed to streamline the food ordering and delivery process. It targets three main user groups: Store personnel (chefs, delivery staff, and food importers), Customers (Registered and VIP customers), and Surfers. The system will feature user-specific login capabilities to enhance personalized experiences and improve system management. Key functionalities include a graphical user interface for menu browsing, personalized dish recommendations, a feedback mechanism for service quality, and protocols for managing user interactions and performance evaluations. This SRS aims to provide a clear and structured overview of the project requirements, ensuring a comprehensive understanding of the proposed solutions to enhance the efficiency and user friendliness of the food ordering and delivery process.

1.2 Scope

The BHAI Brothers Project includes developing and implementing a computerized food & delivery system. This system is a way for customers to interact with food services, and enhance efficiency & convenience during the process. Key aspects of our project include a user-friendly interface that is visually appealing and brings clarity of our product to the user. Main actors are our surfers, customers, VIP customers, manager, delivery driver, food importer and the chef(s). Surfers will be able to scroll through our entire website, but not be able to order any food until account creation. At this point, they are now customers and allowed to order food, leave reviews for the food and the delivery, and write in discussions about the restaurant. VIP customers will have personal benefits after passing a threshold of account spending on our restaurant. Food importers will have access to their own page of the website where they will plan their food imports, organize inventory, etc. The chef(s) will also be able to view this page since they must also know food imports that are coming in to prepare for the day. Delivery drivers will be able to view the main website, but mainly be viewing active delivery orders to handle deliveries. The manager will also be able to view this to supervise orders to ensure customer satisfaction. Managers will also have access to their own webpage that has access to administrative privileges throughout the website, and handle staff pay and employee management.

1.3 Definitions, Acronyms, and Abbreviations

- **JavaScript -** JavaScript is a scripting language that enables you to dynamically update content, control multimedia, animate images, and more through the use of third party libraries.
- **Python -** Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility, widely used for web development, scientific computing, data analysis, artificial intelligence, and more.
- **Flask -** Flask is a lightweight and flexible Python web framework that provides tools and libraries to build web applications easily and quickly with minimal boilerplate code.
- **HTML -** HyperText Markup Language.
- **CSS -** Cascading Style Sheets; is a style sheet language used for describing the presentation of a document in a markup language such as HTML.
- **React -** React is a free and open-source front-end JavaScript library for building user interfaces based on components.
- **NextJS -** Next.js is a React framework that enables server-side rendering, static site generation, and client-side rendering with built-in features for efficient and scalable web development.
- **MongoDB -** MongoDB is a NoSQL database management system that uses a document-oriented data model.
- **API -** Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications.
- **GUI -** Graphical User Interface. The visual method of interacting with a website.
- Includes widgets, buttons, images, and things of a visual and tactile nature.
- **Customers -** Is a registered user who can browse, search, order, and vote on food delivered. They can also start or participate in discussions on cooks, dishes, and delivery drivers.
- **Chefs -** At least two individuals who decide the menus for the store.
- **Deliver Drivers -** At least two drivers who compete to deliver food to customers.

- **Food Importers -** At least two importers who handle food purchases and from outside food suppliers.
- **Manager -** An individual who processes customer registrations, complaints, and compliments. They will also be responsible for hiring, firing, giving raises and pay cuts to chefs and delivery drivers.
- **Surfers -** An individual who can only browse menus and ratings. They can apply to be a registered customer, but they will have a fixed amount of deposit money and has to be checked by the manager.

1.4 References

- Python Documentation
- JavaScript Documentation
- Python Flask Documentation
- NextJS Documentation
- React Documentation
- MongoDB Documentation

1.5 Overview

The remainder of this document provides a general description of the project, including a user case diagram that exhibits the users of this project. This document is divided into three sections, which is started by a discussion of the overall description of the project. Use cases and user case diagrams are discussed within the description. Section 3 provides specific and supplementary requirements needed for the Online Diner project. Following this, Section 3 delves into specific and supplementary requirements that are required for the Online Diner project. Section 4 includes supporting documentation which offers additional information on references, additional diagrams, or any supplementary materials essential for the implementation of the Online Diner system.

2. Overall Description

2.1 Use-Case Model Survey

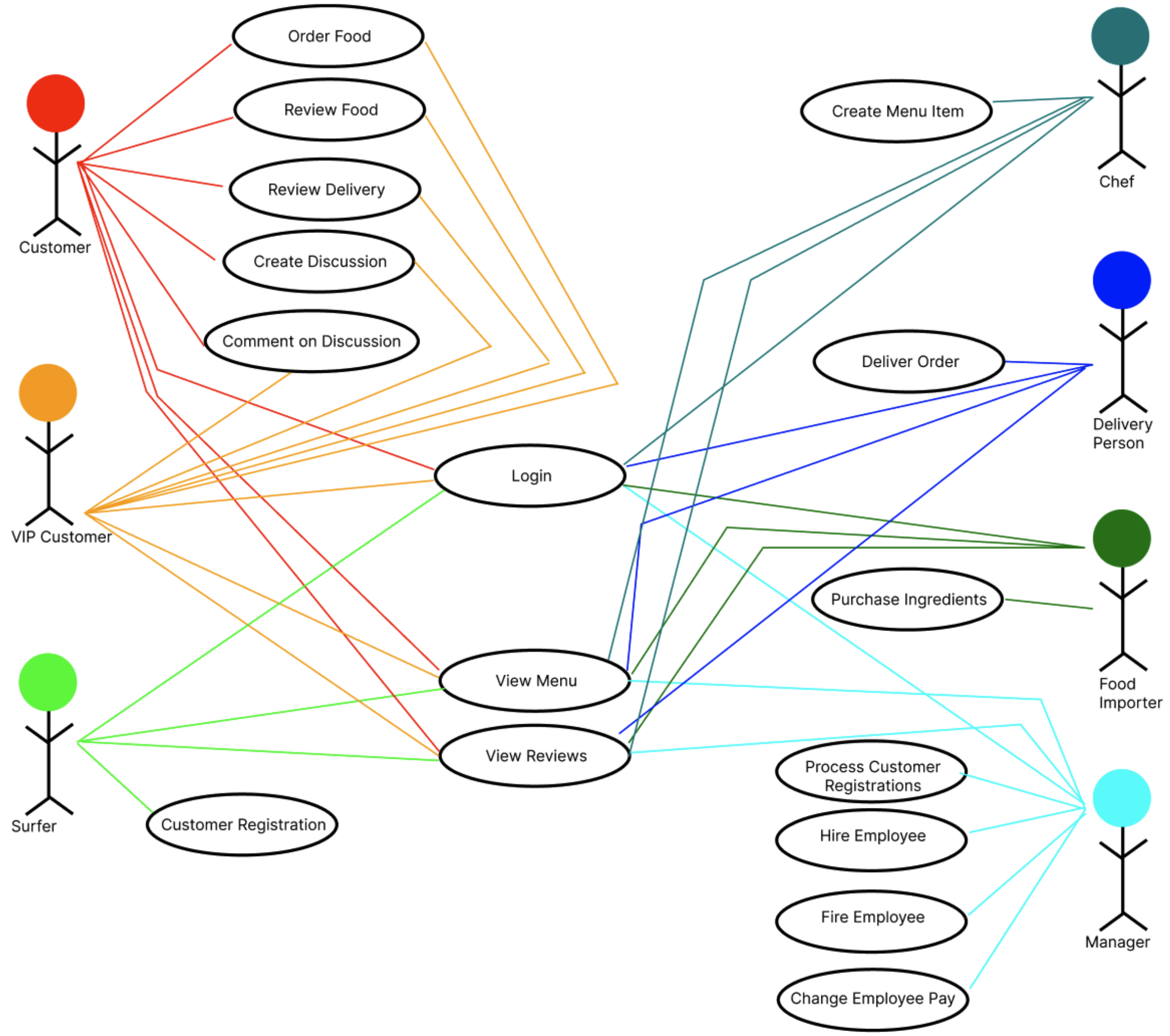
Actors

1. Chef: A user responsible for deciding menu items
2. Delivery Person: A user who competes with other delivery people for food delivery
3. Food Importers: A user that handles food purchases from outside suppliers for the restaurant
4. Manager: A higher level user who performs administrative tasks
5. Customer: A user who can order from the restaurant
6. VIP Customer: A customer with special benefits and privileges
7. Surfer: A user who can browse the restaurant

Use Cases

1. Login: Allows chefs, delivery people, food importers, managers, customers, and vip customers to log into the system
2. Create Menu Item: Allows the chef to add an item to the menu
3. Deliver Order: Allows delivery people to deliver orders to customers
4. Purchase Ingredients: Allows food importers to buy food for the restaurant
5. Process Customer Registration: Allows manager to accept or deny customer registrations
6. View Reviews: Allows all users to view reviews left by customers
7. Hire Employee: Allows manager to hire new employees
8. Fire Employee: Allows manager to fire existing employees
9. Change Employee Pay: Allows manager to give raises/cuts to employees
10. View Menu: Allows all users to view the menu
11. Order Food: Allows customers and vip customers to order food
12. Review Food: Allows customers and vip customers to review delivered food
13. Review Delivery: Allows customers and vip customers to review delivery quality
14. Create Discussion: Allows customers and vip customers to start discussion threads
15. Comment on Discussion: Allows customers and vip customers to participate in discussions
16. Customer Registration: Allows surfers to apply to be a registered customer

2.2 Use Case Diagram



2.3 Assumptions and Dependencies

- Flask REST framework will work as intended
  - The backend API/framework will be able to correctly handle all API requests from the frontend.
  - It will be able to handle requests concurrently as well as handle authentication properly and securely.
  - The backend API/framework will be able to successfully connect and query the database, returning the correct data that was requested.
- MongoDB is correctly designed.
  - There are no repetitive tables or user information.
  - Data is correctly inserted into tables and correctly displayed.
  - Foreign keys are set up correctly to relate two different tables together.
  - Tables are correctly formatted, with the right amount of data, correct data types, and are not corrupted.
- The application will be web based.

- Users will be able to access the application through many devices (phone, laptop, desktop, etc.) on their web browser
- There will be multiple types of users as defined in section 1.3 and each type of user will have the proper permissions.

3. Specific Requirements

3.1 Use-Case Reports

Features/Actions	**Permitted Users**	**Pages**
Customer Registration	Surfer	Registration Page
Logging In/Out	All Users except Surfer	Login Page/Logout Button
View Menu & Reviews	Surfer, Customer, VIP Customer	Menu/Order Page (Includes reviews under each item)
Order Food	VIP Customer, Customer	Menu/Order Page
Process Customer Registrations	Manager	Manager-only management page
Create and write on discussions	Customer, VIP Customer	Discussion Page
Hire & Fire employees	Manager	Manager-only management page
Handle Employee Pay	Manager	Manager-only management page
Purchase Ingredients	Food Importer, Chef	Food Importer Planning Page
Food & Delivery Review	Customer, VIP Customer	Review/Rate Us page
Create Menu Items	Chef	Menu/Order Page with Editing Access
Deliver Orders	Delivery Driver	Delivery Order Page (Item #, Item Contents, Delivery Address, Name)

3.2 Supplementary Requirements

- Membership Program: A paid subscription that is optional to the customers. With this monthly subscription plan, customers will be able to redeem discounts at every purchase, along with a monthly $10 coupon valid at any of the family restaurants.
- Perks for Employees: Employees will receive benefits such as discounts not only at company-owned restaurants but also at franchises outside the family brand.

4. Group Meeting Memo

4.1 Meeting Notes

1. 04/12/2024
   - ➢ All members of the group are present
   - ➢ Frontend final decision: Next.JS for frontend design, prioritizing an MVP, building an accurate GUI with Figma to display functionality and user friendliness to the project
   - ➢ Backend: MongoDB + Flask backend, storing all user information, food imports, active orders, etc.
   - ➢ Distributed diagrams, GUI, and Pseudocode are assigned equally to all members based on workload.

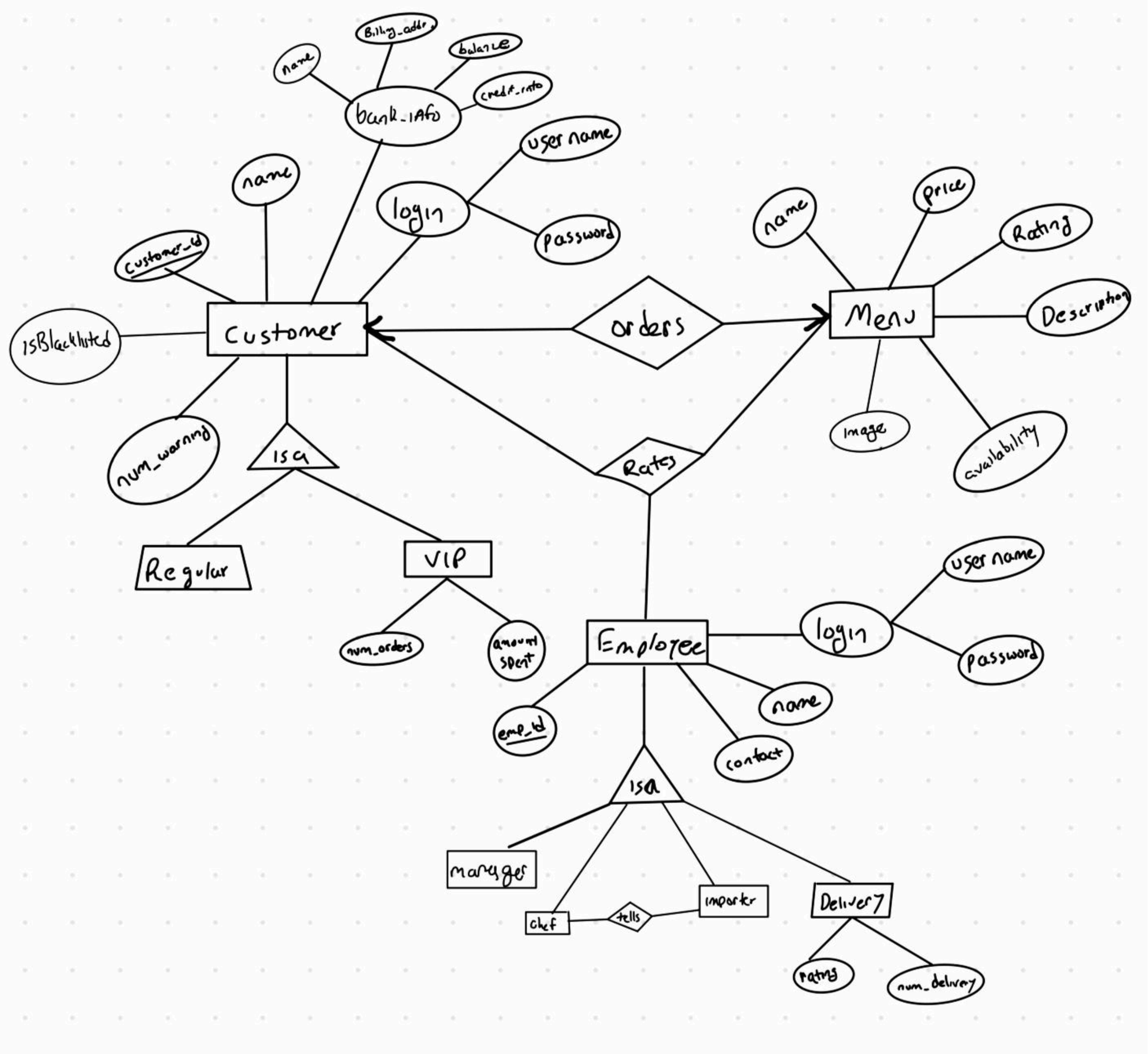
2. 04/17/2024
   - ➢ Distribution of tasks and diagrams that are required for the updated phase report
     - Tasks include psuedocodes of functions and function prototypes
     - Diagrams include collaboration class diagrams, use case diagrams, petrinets and an ER diagram
   - ➢ A Github repository has been created which includes the required files, two phase reports, a README.md file and a requirement file with all of the required tools for the project

4.2 Team Concerns

1. Project Implementation
   a. Regarding the implementation of different classes for different users, Team A is moving at a slower pace than usual. Reasons being mentioned are studying for midterms for other courses, being able to meet up at a time which works for everyone and majority of the time being spent on the phase II report.
   b. Since the team members are anticipating an upcoming break, the project implementation will start promptly. Resources will be used from the phase report as well as documentations listed above to help speed up the process of implementing the project.

5. Design Diagrams

5.1 ER Diagram



5.2 Use Case Diagram

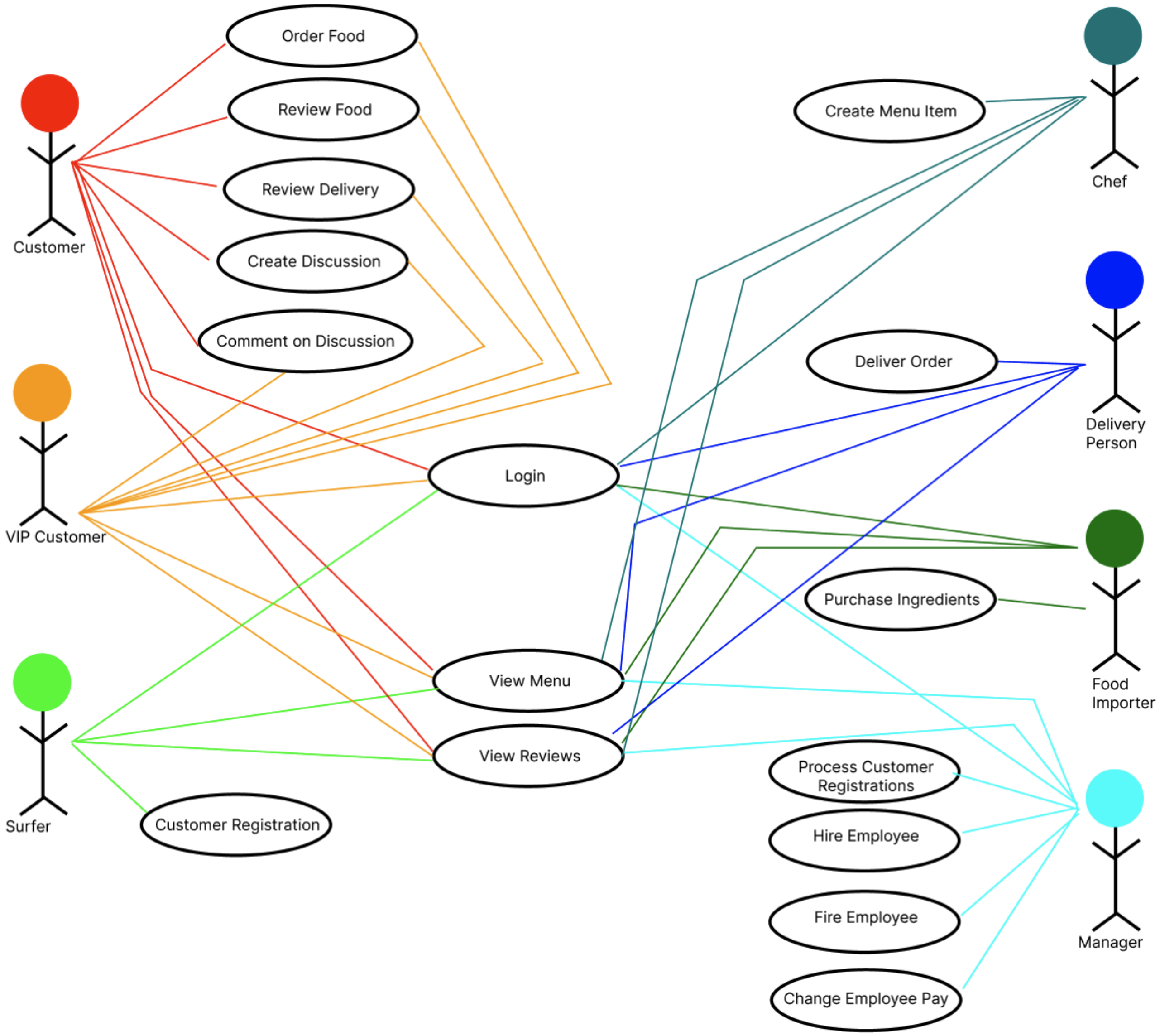


Figure 5.2.1: Use Case Diagram

5.3 Sequence Class Diagrams

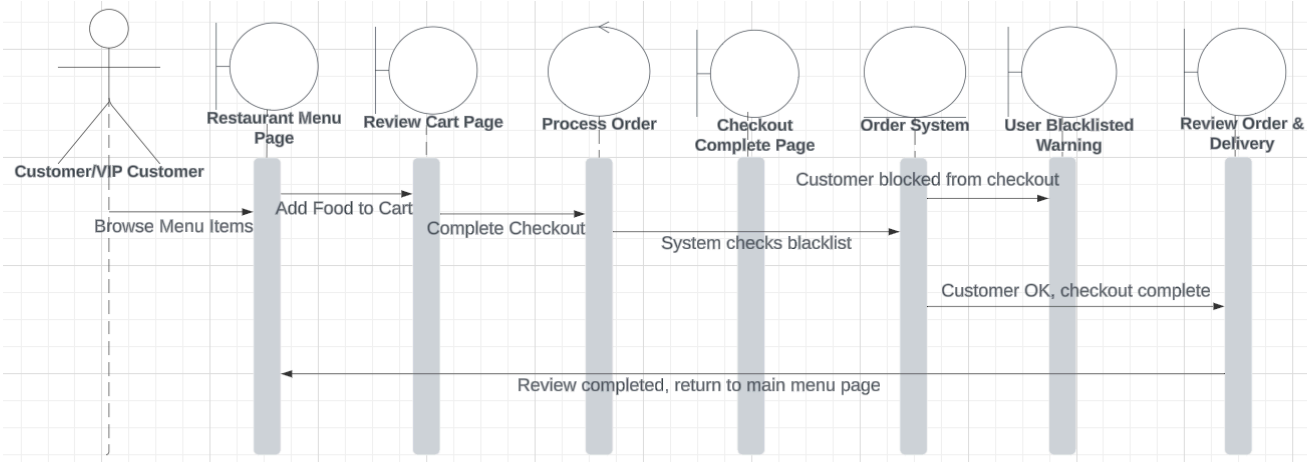


Figure 5.3.1: Order Food, Review Food & Delivery Sequence Class Diagram

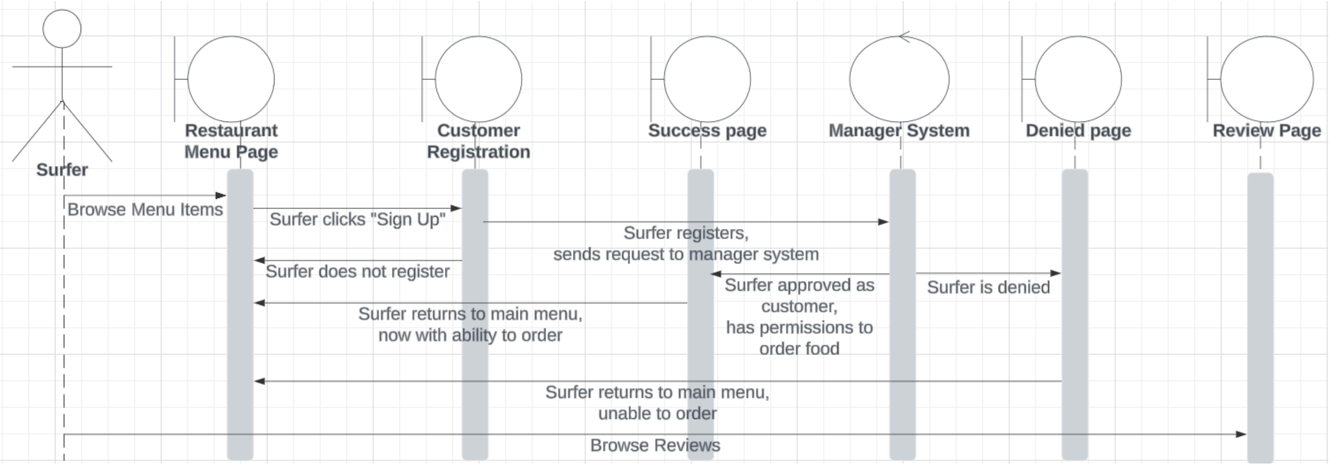

Surfer
Restaurant Menu Page
Customer Registration
Success page
Manager System
Denied page
Review Page
Browse Menu Items
Surfer clicks "Sign Up"
Surfer registers, sends request to manager system
Surfer does not register
Surfer approved as customer, has permissions to order food
Surfer is denied
Surfer returns to main menu, now with ability to order
Surfer returns to main menu, unable to order
Browse Reviews

Figure 5.3.2: Surfer Registration Sequence Class Diagram

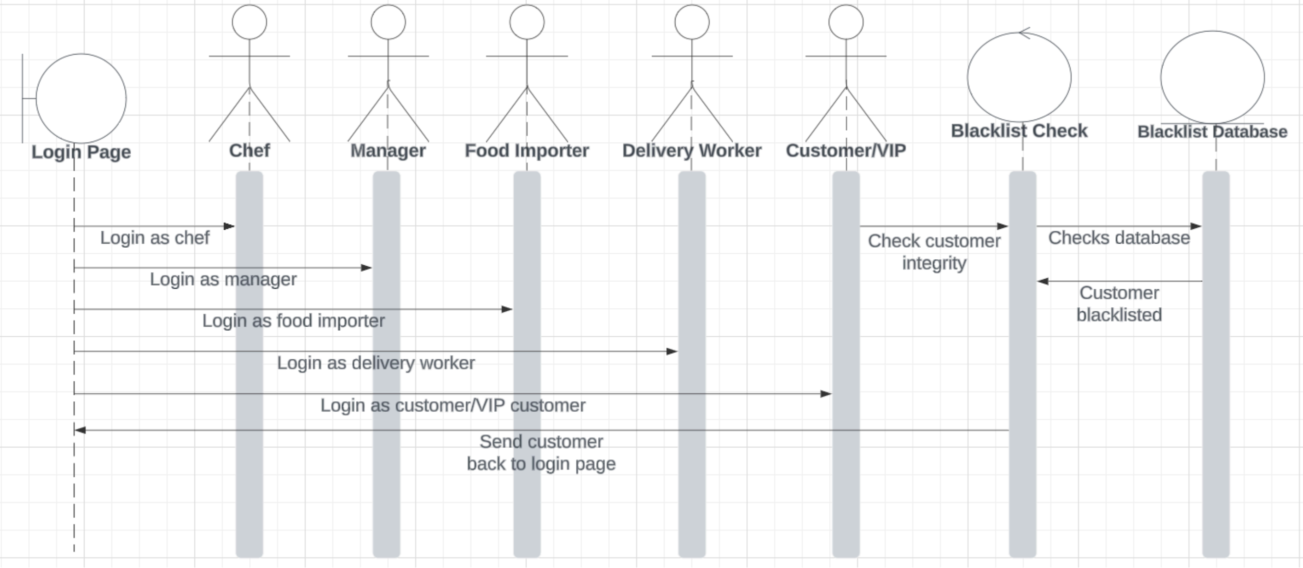
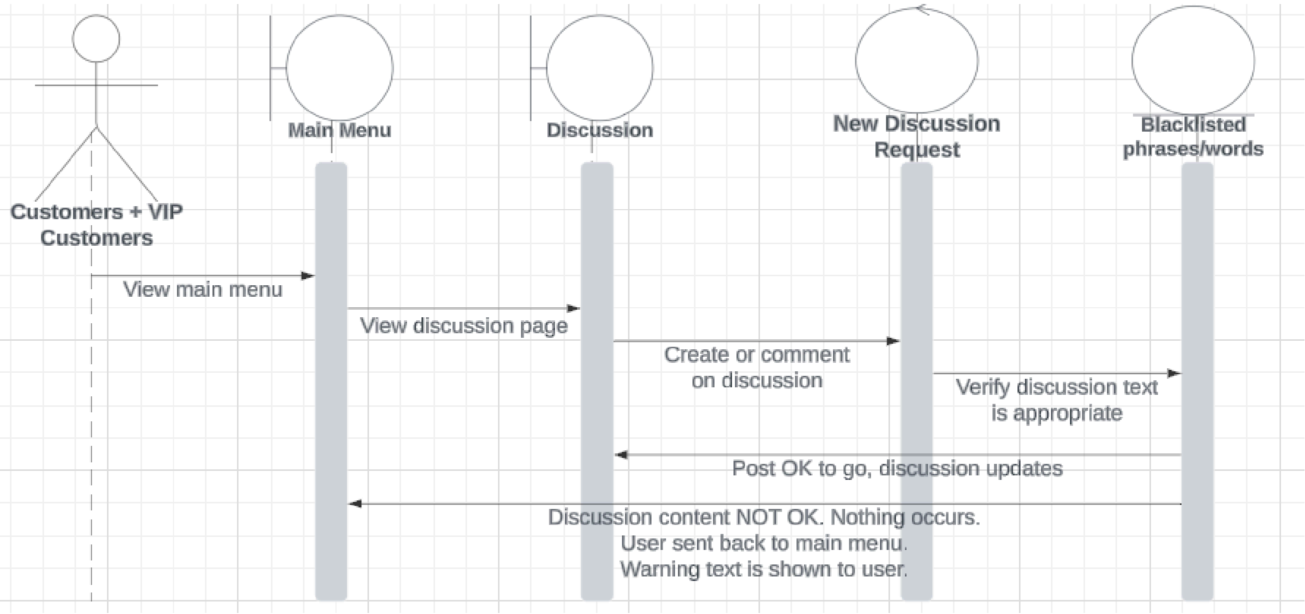
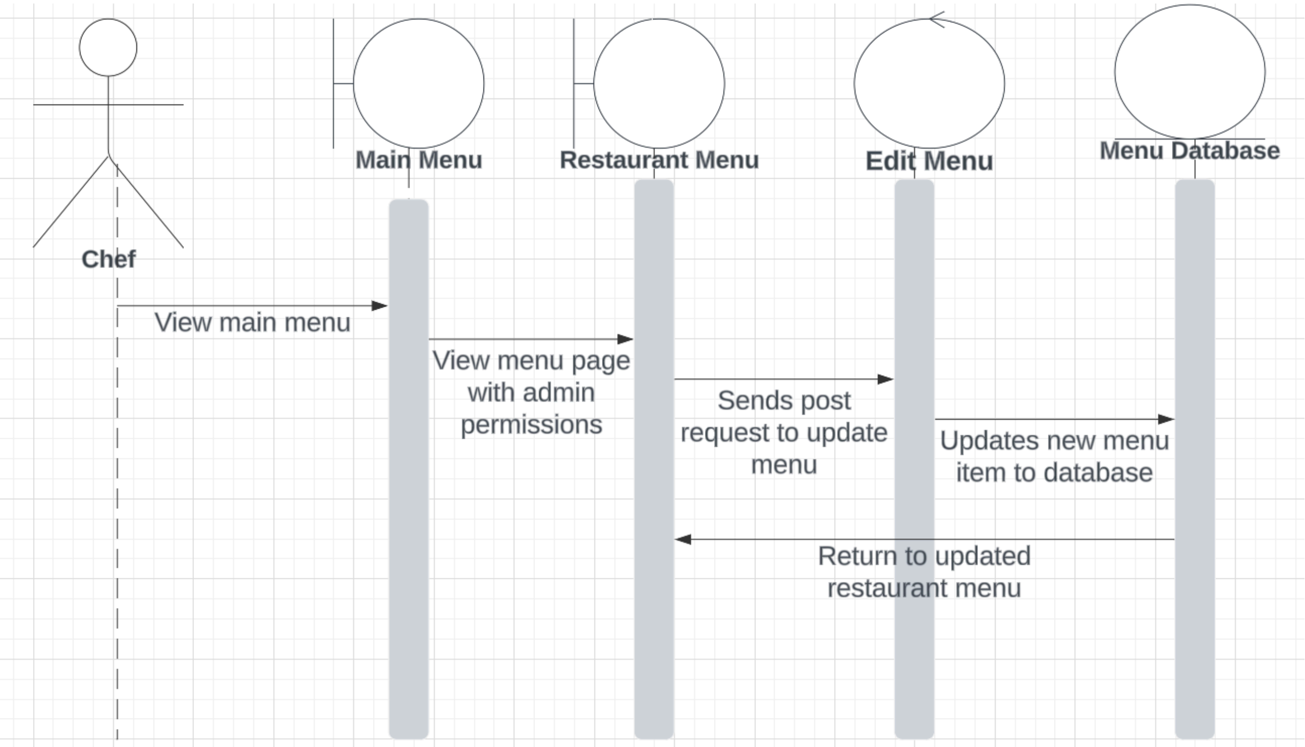


Figure 5.3.3: Login Sequence Class Diagram



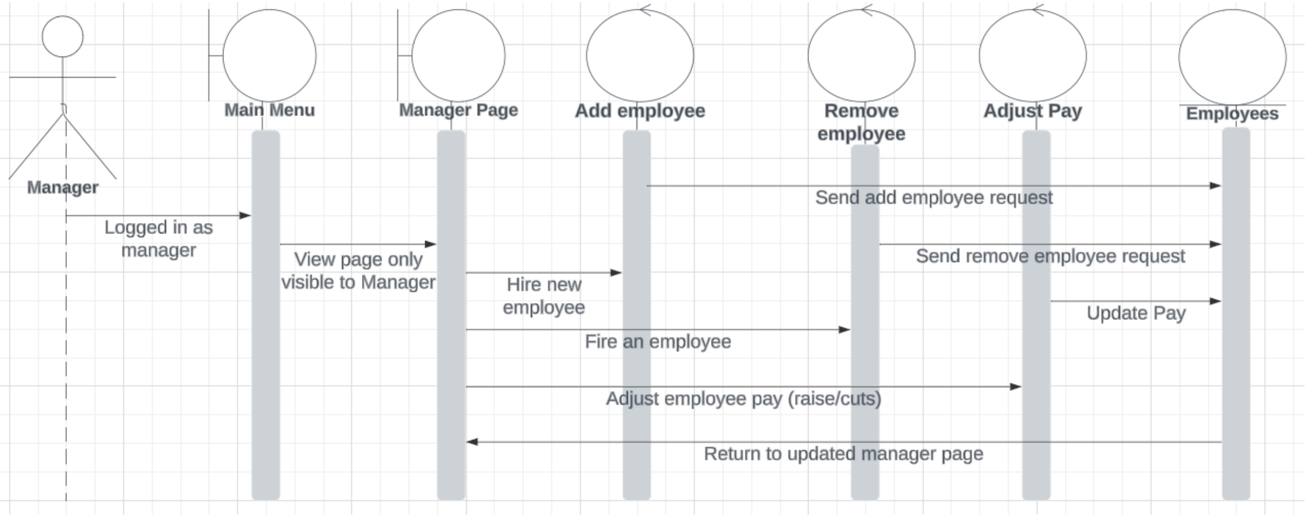
Customers + VIP Customers
Main Menu
Discussion
New Discussion Request
Blacklisted phrases/words
View main menu
View discussion page
Create or comment on discussion
Verify discussion text is appropriate
Post OK to go, discussion updates
Discussion content NOT OK. Nothing occurs. User sent back to main menu. Warning text is shown to user.

Figure 5.3.4: Discussion Sequence Class Diagram



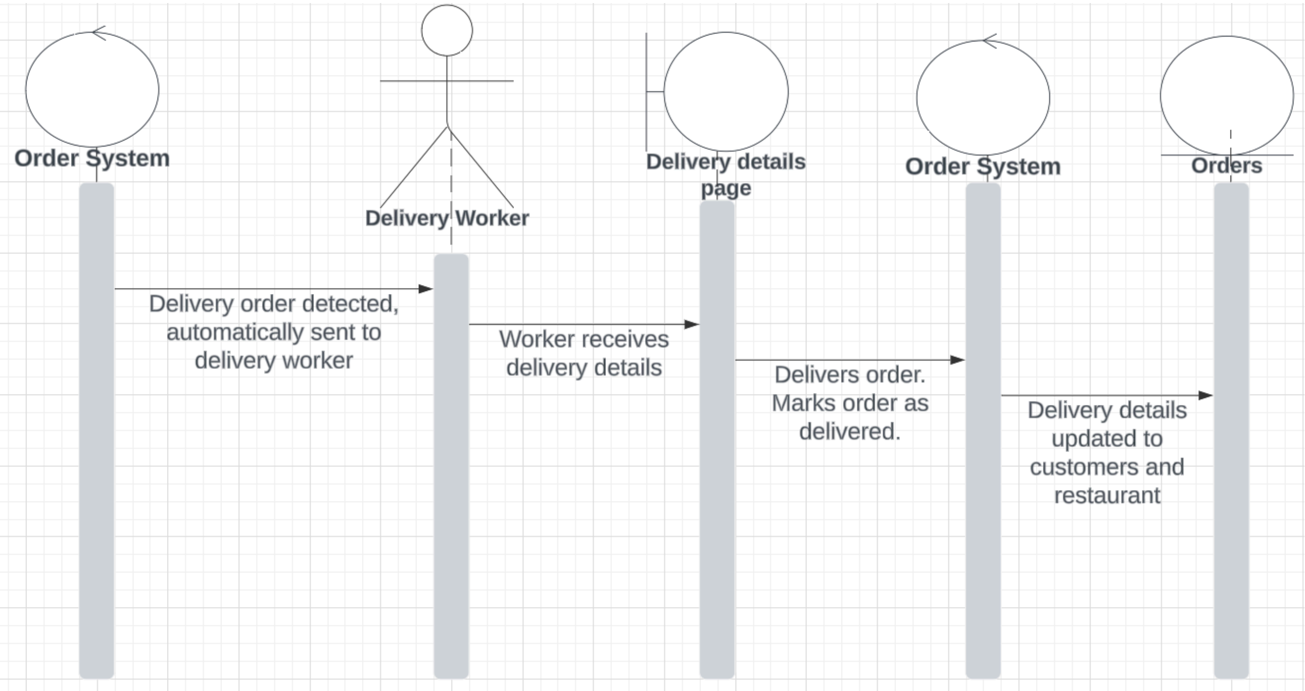
Chef
Main Menu
Restaurant Menu
Edit Menu
Menu Database
View main menu
View menu page with admin permissions
Sends post request to update menu
Updates new menu item to database
Return to updated restaurant menu

Figure 5.3.5: Chef Menu Update Sequence Class Diagram



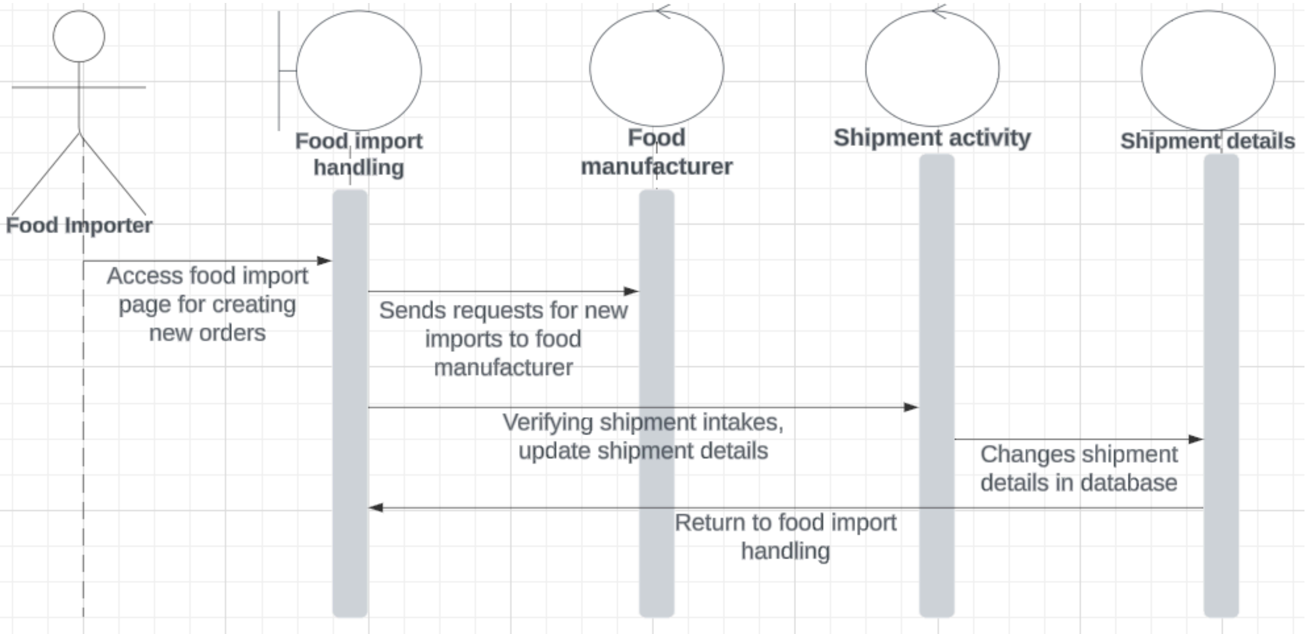
Manager
Main Menu
Manager Page
Add employee
Remove employee
Adjust Pay
Employees
Logged in as manager
View page only visible to Manager
Hire new employee
Fire an employee
Adjust employee pay (raise/cuts)
Send add employee request
Send remove employee request
Update Pay
Return to updated manager page

Figure 5.3.6: Manager Activity Sequence Class Diagram



Order System
Delivery Worker
Delivery details page
Order System
Orders
Delivery order detected, automatically sent to delivery worker
Worker receives delivery details
Delivers order. Marks order as delivered.
Delivery details updated to customers and restaurant

Figure 5.3.7: Delivery Worker Sequence Class Diagram



Food Importer
Food import handling
Food manufacturer
Shipment activity
Shipment details
Access food import page for creating new orders
Sends requests for new imports to food manufacturer
Verifying shipment intakes, update shipment details
Changes shipment details in database
Return to food import handling

Figure 5.3.8: Food Import Sequence Class Diagram

5.4 Petri Nets

Customers/VIP Customers

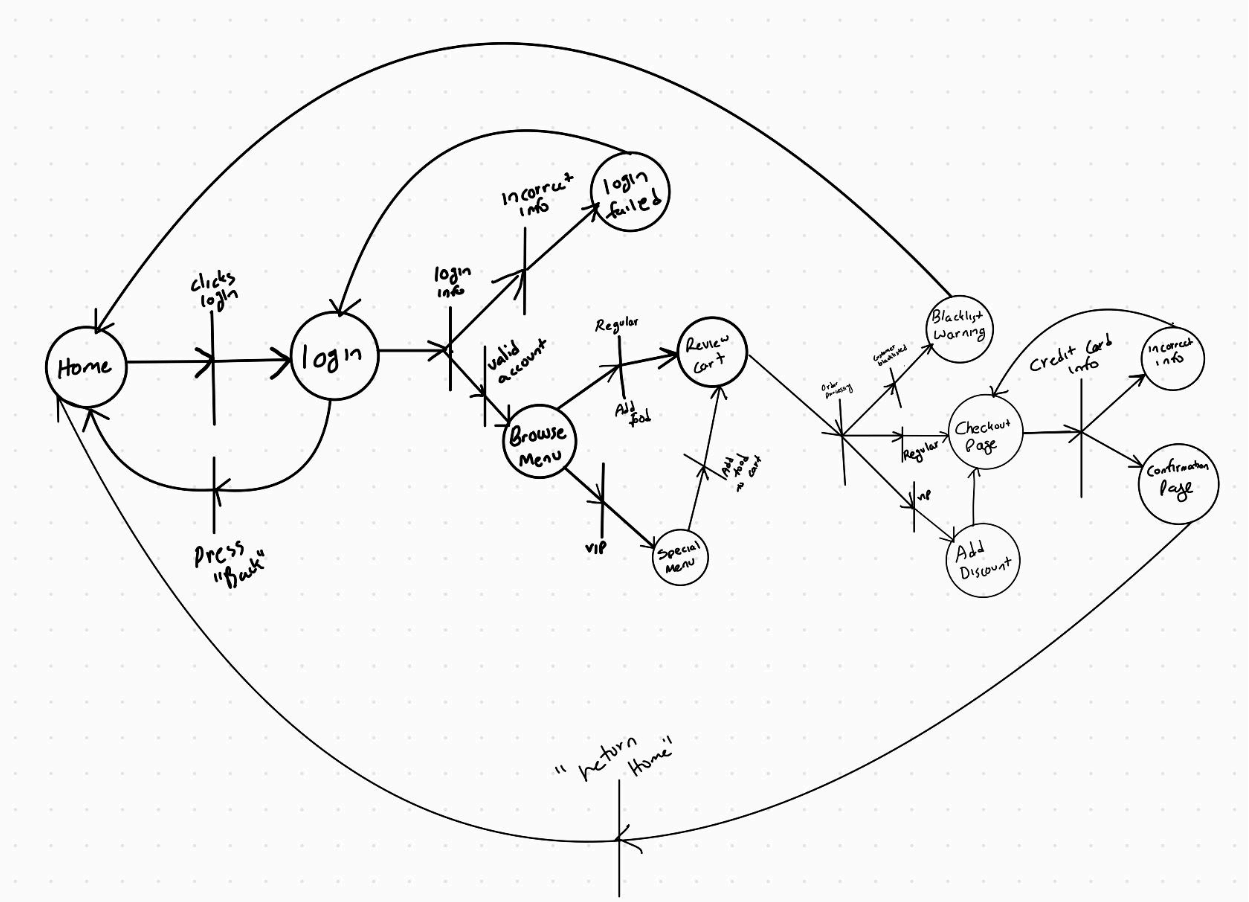


Figure 5.4.1: Customers Petri Net

Chef/Delivery/Importer

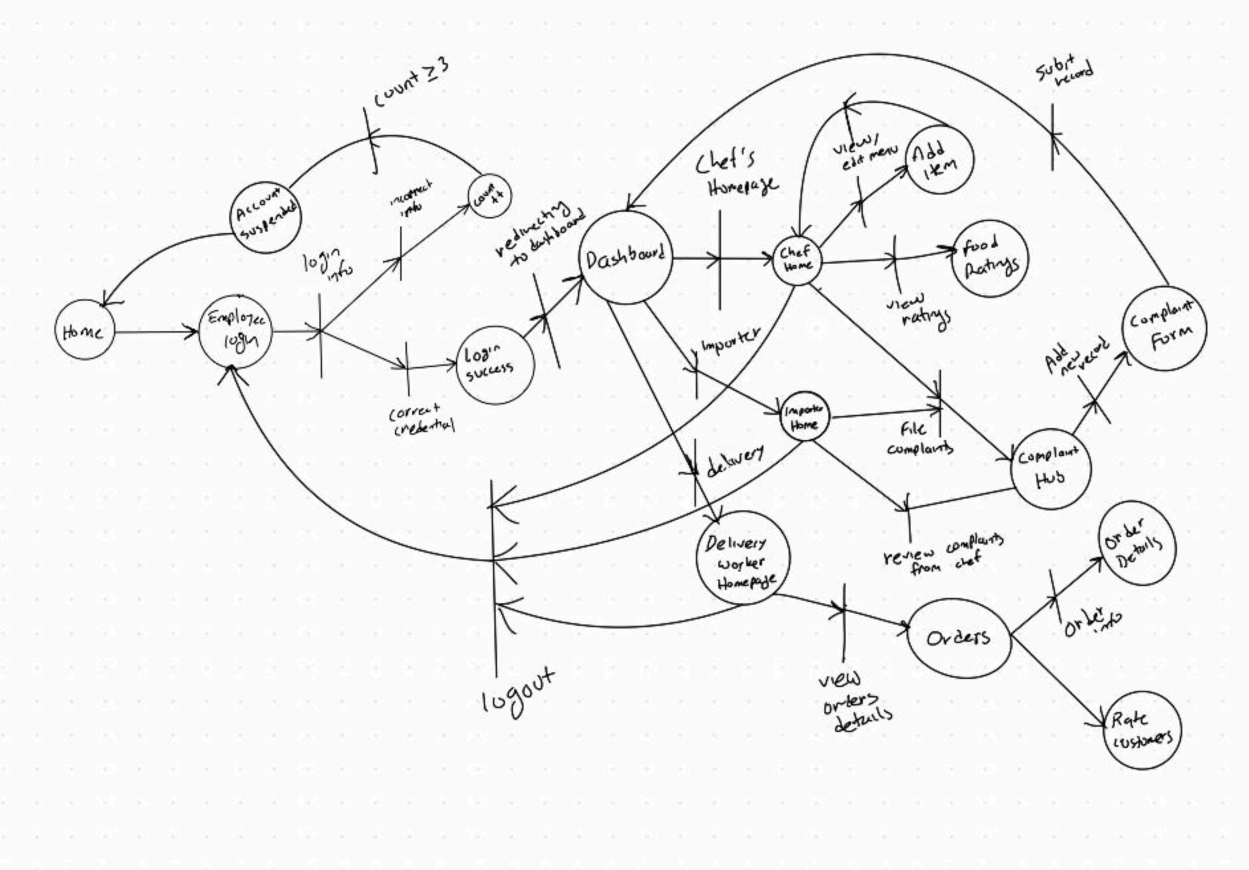


Figure 5.4.2: Employees Petri Net Diagram

Surfers

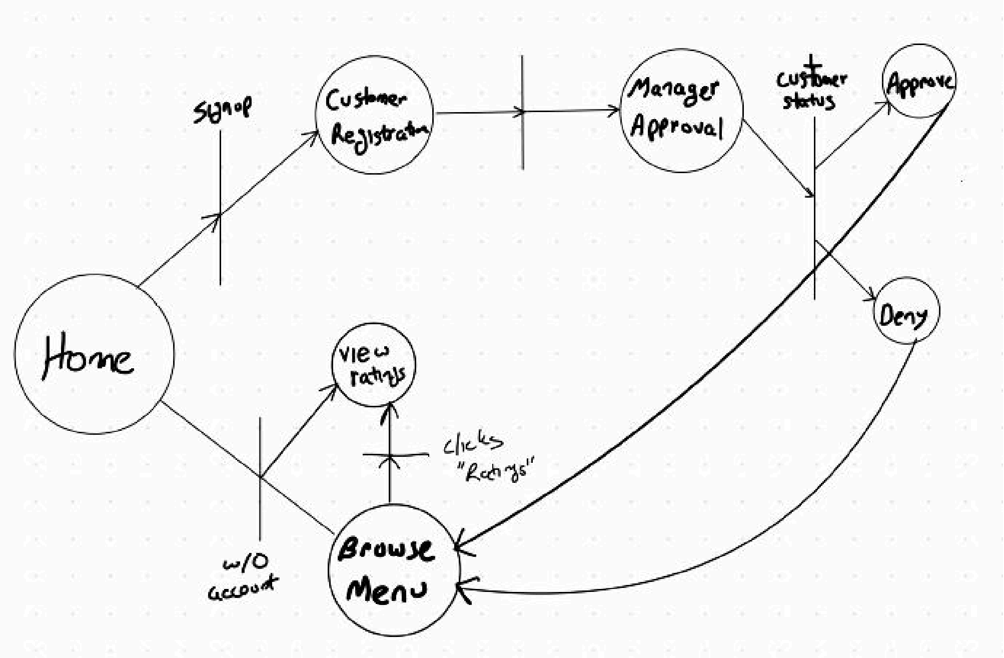


Figure 5.4.3: Surfers Petri Net Diagram

5.5: Collaboration Class Diagram

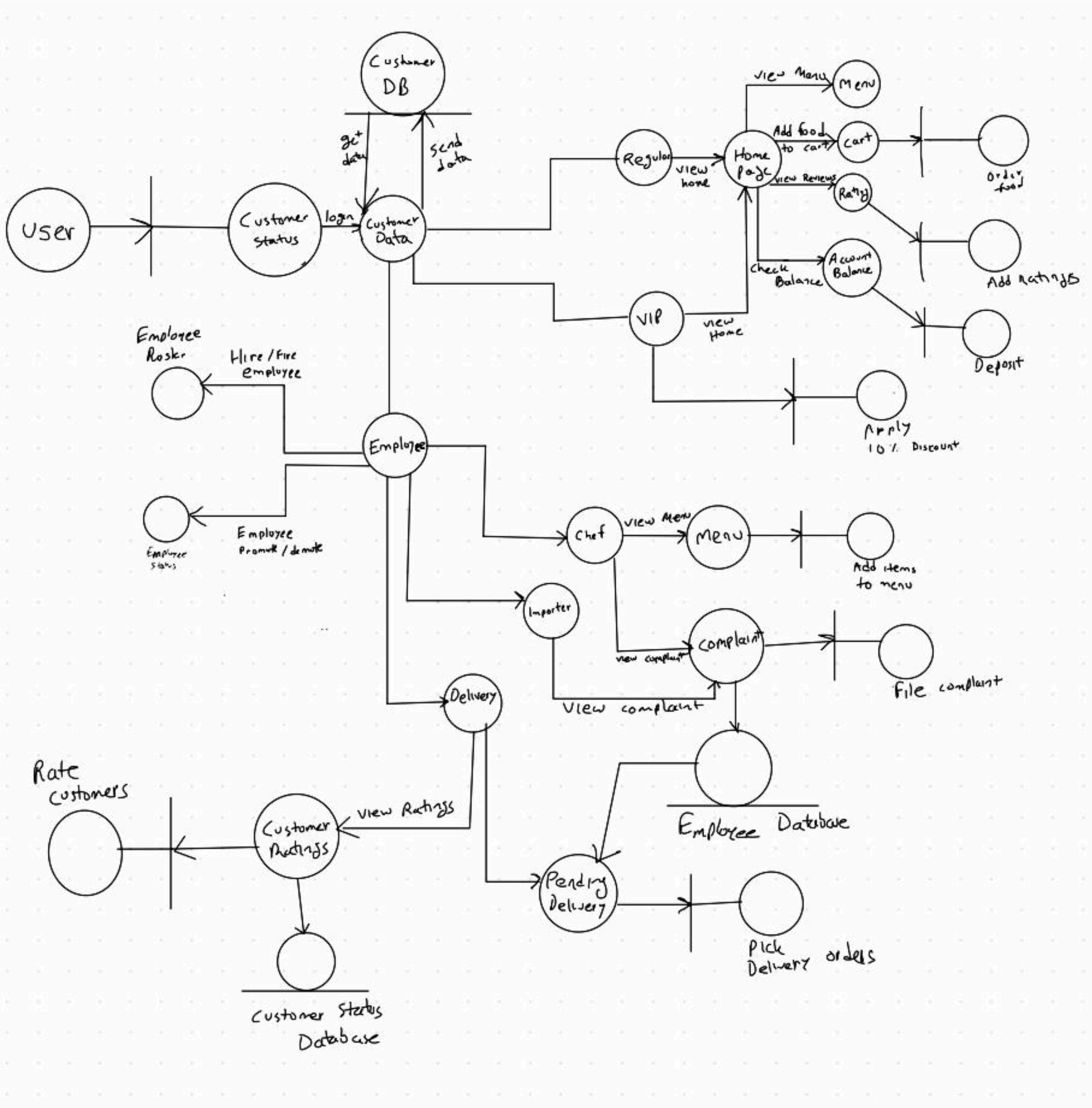


Figure 5.5.1: Collaboration Class Diagram for the System

6. Detailed Design

6.1 Project prototype

- **signIn**:
  - Precondition: The input is username and password
  - Postcondition: Logs the user into our website
  - function signIn(username, password):
  - if isValidCredentials(username, password):

  - logUserIn()

- **signUp**:
  - Precondition: The input is username and password.
  - Postcondition: The users information is stored in the database and the user is logged in
  ```
  function signUp(username, password):
     if !userExists(username):
        createUser(username, password)
        logUserIn()
  ```
- **getTypeOfUser**:
  - Precondition: The input is the users name
  - Postcondition: Returns the type of user, i.e. customer, chef, manager, driver, etc.
  ```
  function getTypeOfUser(username):
     userType = getUserType(username)
     return userType
  ```
- **addToMenu**:
  - Precondition: The input is an instance of a Menu item class.
  - Postcondition: WIll update the menu (global variable/class) with the menu item.
  ```
  function addToMenu(menuItem):
     menu.addItem(menuItem)
  ```
- **deleteFromMenu**:
  - Precondition: The input is menu item we want to remove.
  - Postcondition: Deletes the menu item from the menu.
  ```
  function deleteFromMenu(menuItem):
     menu.deleteItem(menuItem)
  ```
- **processOrder:**
  - Precondition: The input is an instance of the Order class.
  - Postcondition: The order is processed.
  ```
  function processOrder(order):
     order.process()
  ```
- **addReview:**
  - Precondition: The input is an instance of the review class.
  - Postcondition: The review is added to the website.
  ```
  function addReview(review):
     website.addReview(review)
  ```

- **approveCustomer:**
  - Precondition: Check whether a customer with email exists already. Check if the customer is blacklisted.
  - Postcondition: User is added to the database as a regular customer.
  ```
  function approveCustomer(email):
      if not userExists(email) and not isBlacklisted(email):
         createUser(email)
      else if isBlacklisted(email):
         // Handle blacklisted customer
  ```

```
    else:
        // Handle existing customer
```

- **isValidCredentials:**
    - Precondition: The inputs are a username and password.
    - Postcondition: Returns true if credentials match, false otherwise.

    ```
    function isValidCredentials(username, password):
        return database.checkCredentials(username, password)
    ```

- **userExists:**
    - Precondition: The input is a username.
    - Postcondition: Returns true if the user exists, false otherwise.

    ```
    function userExists(username):
        return database.findUser(username) != null
    ```

- **createUser:**
    - Precondition: The inputs are a username and password.
    - Postcondition: A new user is created if the username does not already exist.

    ```
    function createUser(username, password):
        if not userExists(username):
            database.addUser(username, password)
            logUserIn(username)
            return true
        else:
            return false
    ```

- **getUserType:**
    - Precondition: The input is a username.
    - Postcondition: Returns the type of user.

    ```
    function getUserType(username):
        return database.getUserType(username)
    ```

- **menu.addItem:**
    - Precondition: The input is an instance of a Menu item class.
    - Postcondition: The menu item is added if it does not already exist.

    ```
    function menu.addItem(menuItem):
        if not menu.contains(menuItem):
            menu.add(menuItem)
    ```

- **menu.deleteItem:**
    - Precondition: The input is the menu item we want to remove.
    - Postcondition: The menu item is removed if it exists.

    ```
    function menu.deleteItem(menuItem):
        if menu.contains(menuItem):
            menu.remove(menuItem)
    ```

- **order.process:**
    - Precondition: The order is validated and ready to be processed.

- Postcondition: The order is processed, and inventory is updated.

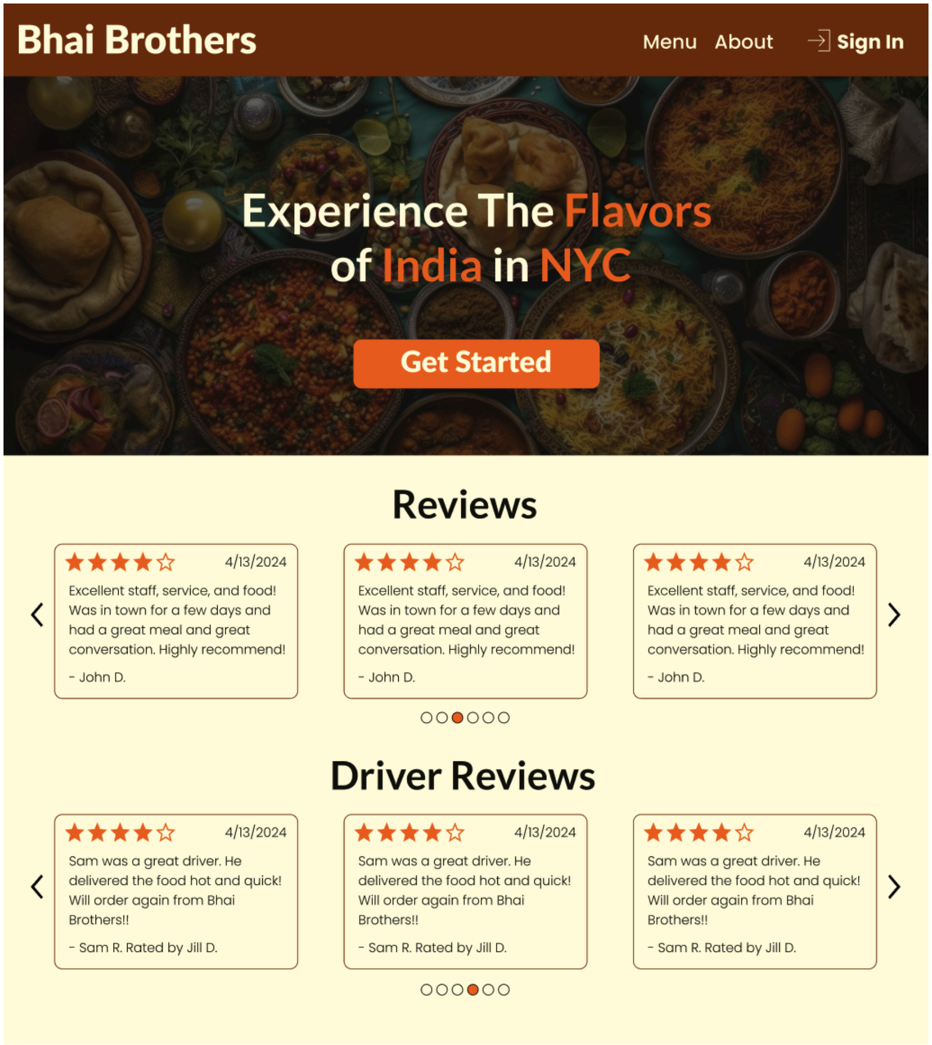
```
function order.process():
    if order.validate():
        payment.process(order.total)
        inventory.update(order.items)
        return 'Processed'
    else:
        return 'Failed'
```

- **website.addReview:**
  - Precondition: The input is a review to be added to the website.
  - Postcondition: The review is added if it is valid.

```
function website.addReview(review):
    if review.isValid():
        database.saveReview(review)
```

7. User Interface

7.1 Home Page



Reviews

★★★★☆ 4/13/2024
Excellent staff, service, and food! Was in town for a few days and had a great meal and great conversation. Highly recommend!
- John D.

★★★★☆ 4/13/2024
Excellent staff, service, and food! Was in town for a few days and had a great meal and great conversation. Highly recommend!
- John D.

★★★★☆ 4/13/2024
Excellent staff, service, and food! Was in town for a few days and had a great meal and great conversation. Highly recommend!
- John D.

Driver Reviews

★★★★☆ 4/13/2024
Sam was a great driver. He delivered the food hot and quick! Will order again from Bhai Brothers!!
- Sam R. Rated by Jill D.

★★★★☆ 4/13/2024
Sam was a great driver. He delivered the food hot and quick! Will order again from Bhai Brothers!!
- Sam R. Rated by Jill D.

★★★★☆ 4/13/2024
Sam was a great driver. He delivered the food hot and quick! Will order again from Bhai Brothers!!
- Sam R. Rated by Jill D.

Figure 7.1.1: Home Page from the Web Application

7.2 Nav Bars - Surfer vs Logged In Users

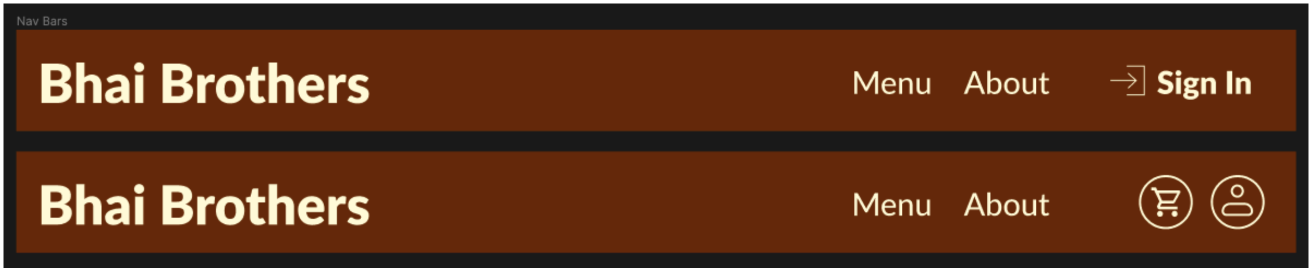


Figure 7.1.2: Nav Bars for both Users and Surfers

7.3 Sign up Page

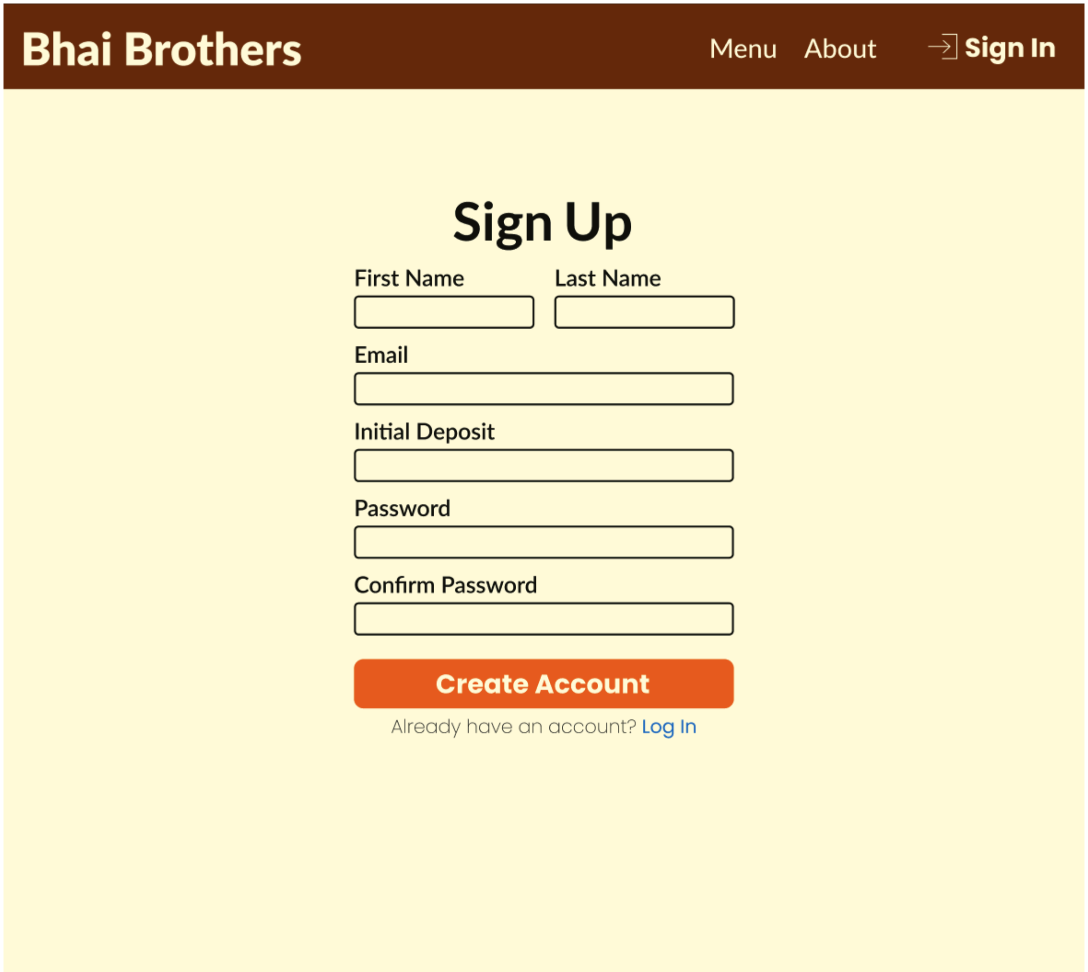


Figure 7.1.3: Sign-up Page for New Customers

7.4 Sign in Page

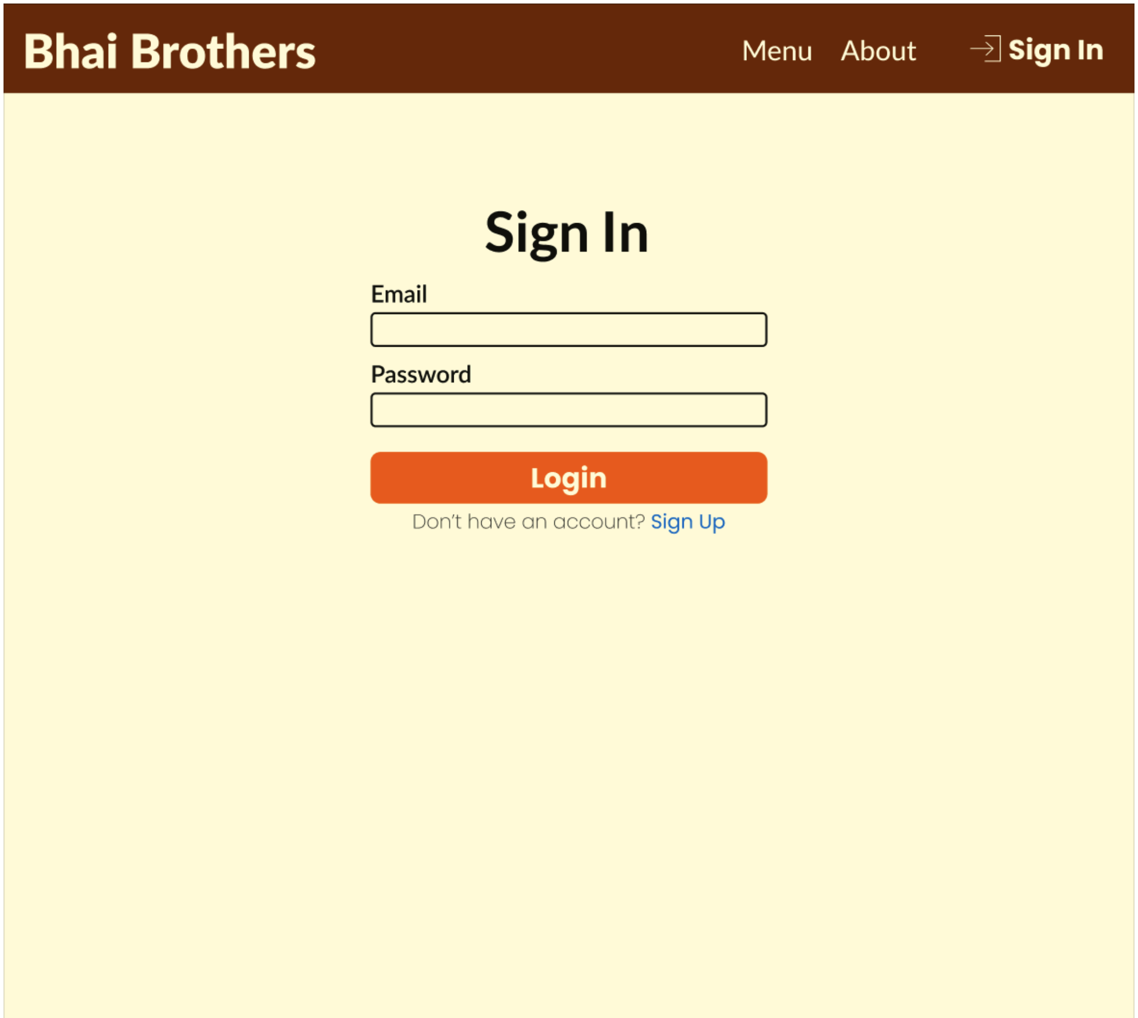
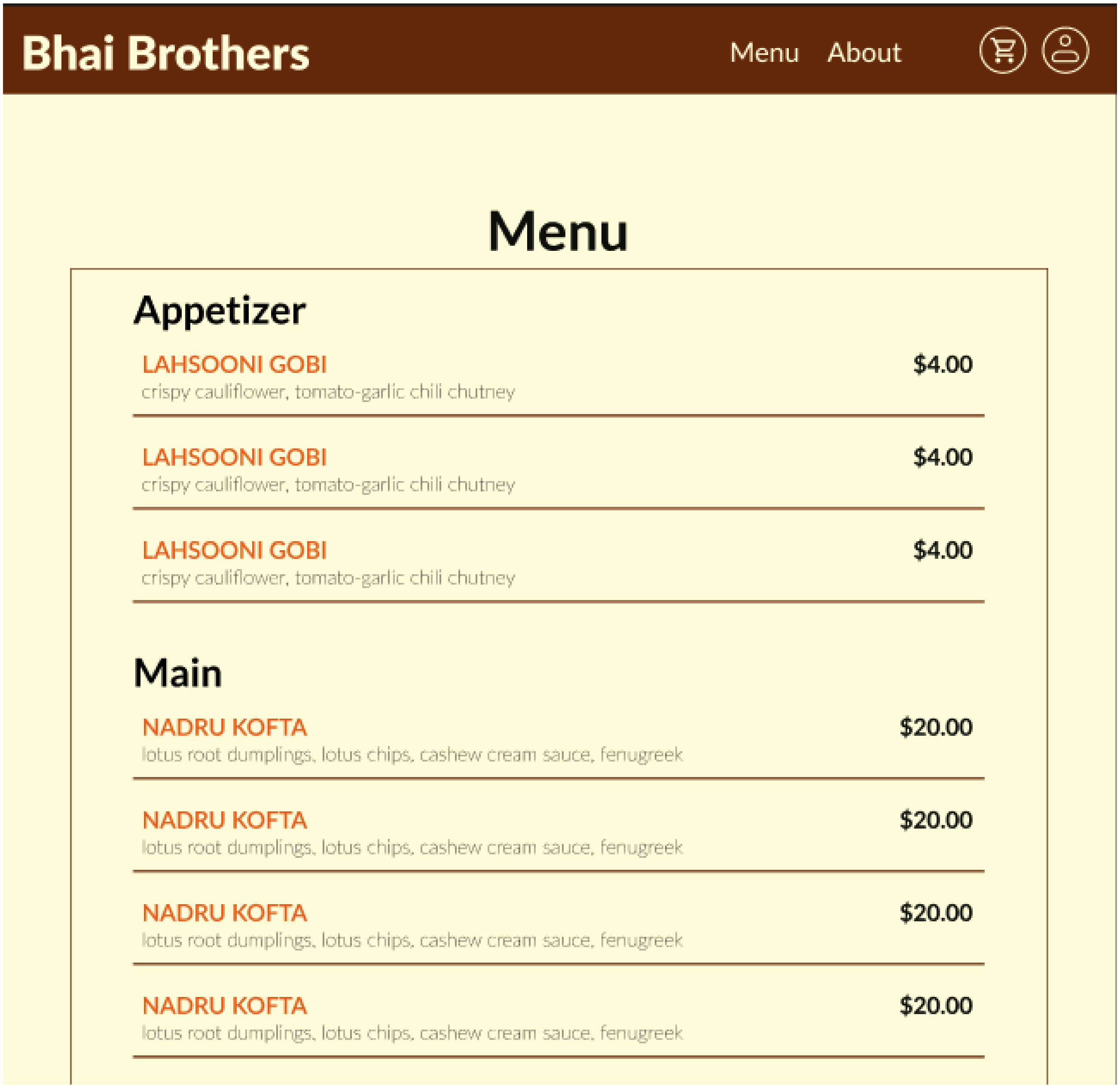


Figure 7.1.4: Sign in Page for Regular and VIP Customers

7.5 Menu Page - Surfer View



Menu

Appetizer

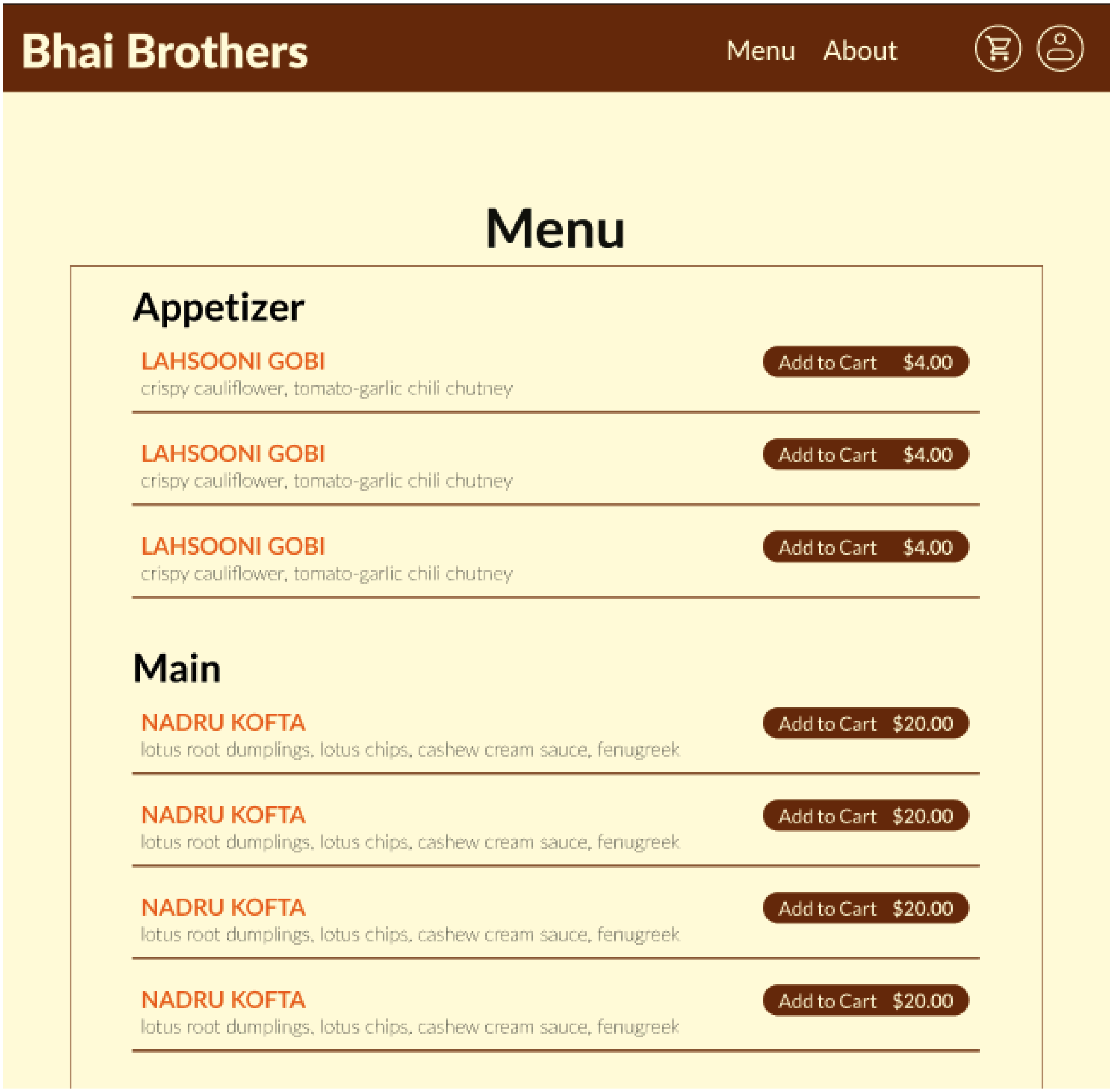
LAHSOONI GOBI crispy cauliflower, tomato-garlic chili chutney	$4.00
LAHSOONI GOBI crispy cauliflower, tomato-garlic chili chutney	$4.00
LAHSOONI GOBI crispy cauliflower, tomato-garlic chili chutney	$4.00

Main

NADRU KOFTA lotus root dumplings, lotus chips, cashew cream sauce, fenugreek	$20.00
NADRU KOFTA lotus root dumplings, lotus chips, cashew cream sauce, fenugreek	$20.00
NADRU KOFTA lotus root dumplings, lotus chips, cashew cream sauce, fenugreek	$20.00
NADRU KOFTA lotus root dumplings, lotus chips, cashew cream sauce, fenugreek	$20.00

Figure 7.1.5: Menu Item page for Surfers

7.6 Menu Page - Customer View



Menu

Appetizer

LAHSOONI GOBI
crispy cauliflower, tomato-garlic chili chutney
Add to Cart $4.00

LAHSOONI GOBI
crispy cauliflower, tomato-garlic chili chutney
Add to Cart $4.00

LAHSOONI GOBI
crispy cauliflower, tomato-garlic chili chutney
Add to Cart $4.00

Main

NADRU KOFTA
lotus root dumplings, lotus chips, cashew cream sauce, fenugreek
Add to Cart $20.00

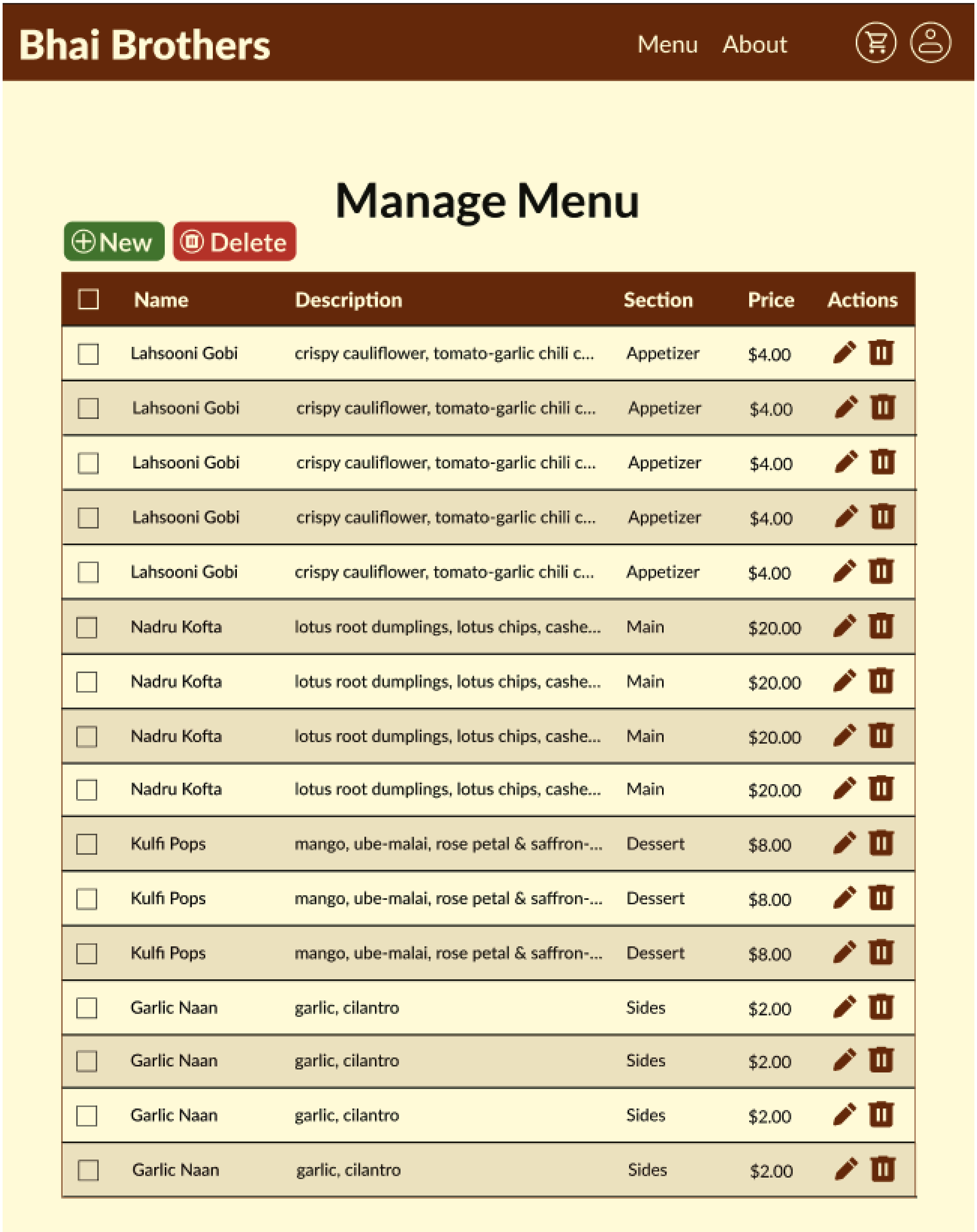
NADRU KOFTA
lotus root dumplings, lotus chips, cashew cream sauce, fenugreek
Add to Cart $20.00

NADRU KOFTA
lotus root dumplings, lotus chips, cashew cream sauce, fenugreek
Add to Cart $20.00

NADRU KOFTA
lotus root dumplings, lotus chips, cashew cream sauce, fenugreek
Add to Cart $20.00

Figure 7.1.6: Item Menu View for Customers (Regular and VIP)

7.7 Menu Page - Chef View



□	Name	Description	Section	Price	Actions
□	Lahsooni Gobi	crispy cauliflower, tomato-garlic chili c...	Appetizer	$4.00	✎ 🗑
□	Lahsooni Gobi	crispy cauliflower, tomato-garlic chili c...	Appetizer	$4.00	✎ 🗑
□	Lahsooni Gobi	crispy cauliflower, tomato-garlic chili c...	Appetizer	$4.00	✎ 🗑
□	Lahsooni Gobi	crispy cauliflower, tomato-garlic chili c...	Appetizer	$4.00	✎ 🗑
□	Lahsooni Gobi	crispy cauliflower, tomato-garlic chili c...	Appetizer	$4.00	✎ 🗑
□	Nadru Kofta	lotus root dumplings, lotus chips, cashe...	Main	$20.00	✎ 🗑
□	Nadru Kofta	lotus root dumplings, lotus chips, cashe...	Main	$20.00	✎ 🗑
□	Nadru Kofta	lotus root dumplings, lotus chips, cashe...	Main	$20.00	✎ 🗑
□	Nadru Kofta	lotus root dumplings, lotus chips, cashe...	Main	$20.00	✎ 🗑
□	Kulfi Pops	mango, ube-malai, rose petal & saffron-...	Dessert	$8.00	✎ 🗑
□	Kulfi Pops	mango, ube-malai, rose petal & saffron-...	Dessert	$8.00	✎ 🗑
□	Kulfi Pops	mango, ube-malai, rose petal & saffron-...	Dessert	$8.00	✎ 🗑
□	Garlic Naan	garlic, cilantro	Sides	$2.00	✎ 🗑
□	Garlic Naan	garlic, cilantro	Sides	$2.00	✎ 🗑
□	Garlic Naan	garlic, cilantro	Sides	$2.00	✎ 🗑
□	Garlic Naan	garlic, cilantro	Sides	$2.00	✎ 🗑

Figure 7.1.7: Item Menu for Chefs

8. Supporting Links

8.1 Github

- https://github.com/dddictionary/bhai-brothers-csc322