Software Design Document
CSC 131: Computer Software Engineering
Fall 2024
MVP TastyTreats

Project Overview:

Tasty Treats will provide a smooth and efficient operation for both customers and employees. Customers will be able to manage their preferences, personal information and order history, so that the customer may conveniently set up their orders to place, pay and pick up. The employees will also be able to view their performances, assigned schedules, request admin for a change in schedule and earnings. Lastly, the admins will manage and view the employees payroll information, performance and schedule. All providing an enhancing overall user experience.

Project Scope:

This web application's important features involve: Menu Display (modifying items in the menu), Shopping Cart, Payment Processing, Customer FeedBack, User Login and User Registration. For employees, the web application keeps track of their personal information, work hours (clocking in and clocking out times). The admin will be able oversee activities on the web app, including employee actions and user interactions.

Target Audience:

Consumers: Individuals looking for an easy way to place an order by using a user friendly interface.

Busy Professionals: Those who are looking for quick and convenient high quality meal solutions for their busy days.

Health-Conscious Individuals: Users interested in tasty and healthy food require a special category for them in the menu to choose food items from along with their nutritional information.

Use Cases:

Menu Display

- **User Case Name:** Display Menu Items
- **User Case Number:** 1
- **Authors:** Tim Yang

- **Actors:** Customer
- **Overview:** The customer wants to see the menu item list so that they would know on what options each restaurant has to offer
- **References:** User Interface, Database
- **Typical Flow Description:** Customer goes through the web application's menu page. System will obtain and display the list of menu items and details.

User Registration/Login

- **Use Case Name:** Login/Register User
- **Use Case Number:** 2
- **Authors:** Tim Yang
- **Actors:** Admin, Employee, Customer
- **Overview:** TastyTreat users want to have a profile for themselves to keep track of items, schedules and management based on their position.
- **References:** Database / UserInterface
- **Typical Flow Description:** User navigates to registration/login page. User log's in using credentials or put in registration details. System will validate the information. System will create a new profile or give access to existing profiles. Users will be directed back to the menu.

Employee Management

- **Use Case Name:** Employee Schedules Management
- **Use Case Number:** 3
- **Authors:** Tim Yang
- **Actors:** Admin
- **Overview:** As an admin, I want to be able to view and manage employee schedules so I can create and communicate work hours effectively.
- **References:** User Interface/ Database
- **Typical Flow Description:** The admin will log into their profile and will navigate to the employee management section. System will display the employee schedules where the admin will create/modify new schedules. Admin will save changes and the system will update the database.

Shopping Cart

- **Use Case Name:** Manage Shopping Cart
- **Use Case Number:** 4
- **Authors:** Tim Yang
- **Actors:** Customer

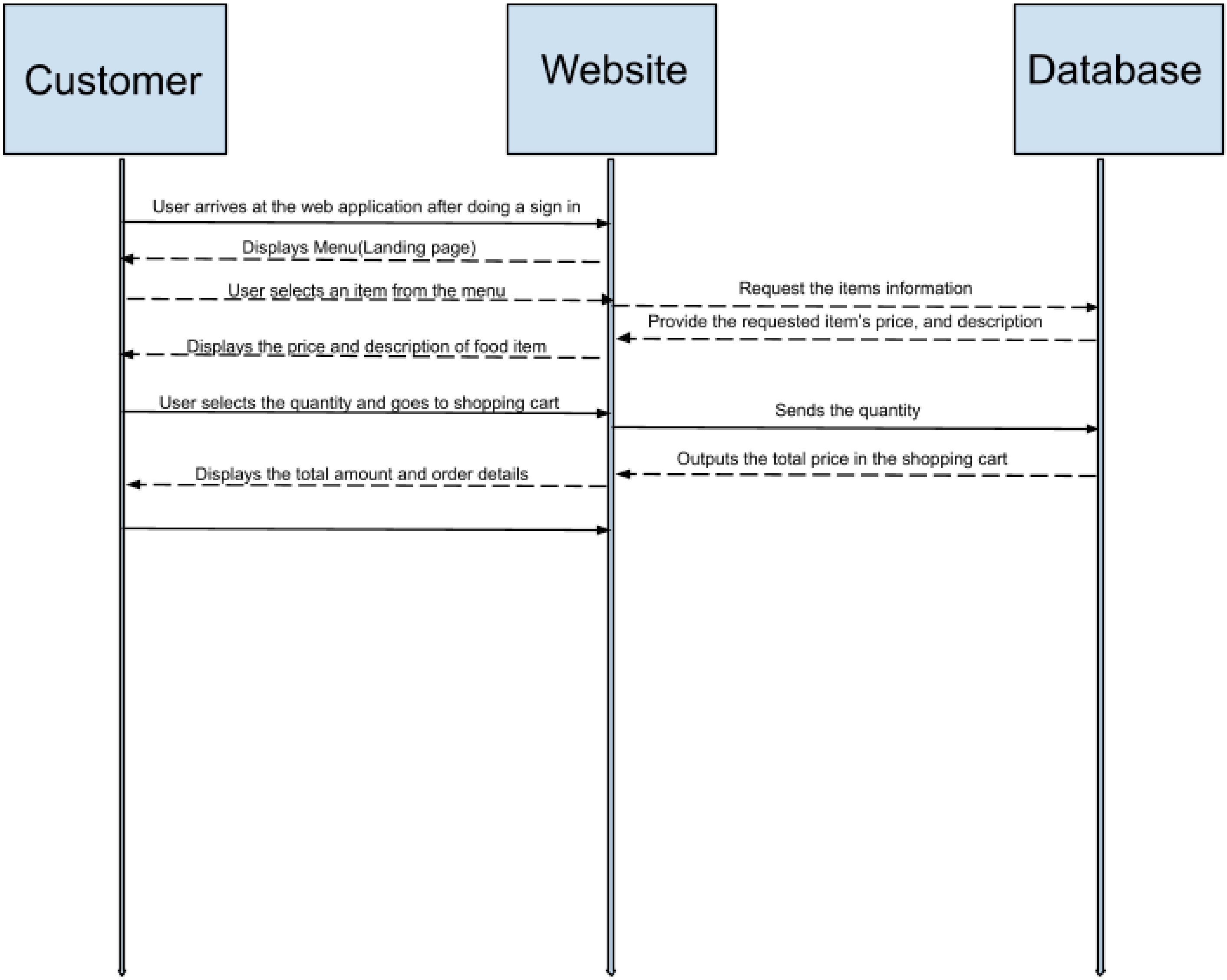
- **Overview:** As a customer, I want a feature that can keep track of items I have selected until I am ready to checkout.
- **References:** User Interface/ Database
- **Typical Flow Description:** The customer will add the item(s) they want to the shopping cart. System will add the items to the cart and update its content. Customers can view and change the shopping cart list at any time. Customers will checkout when they are ready.

Order Confirmation and Checkout

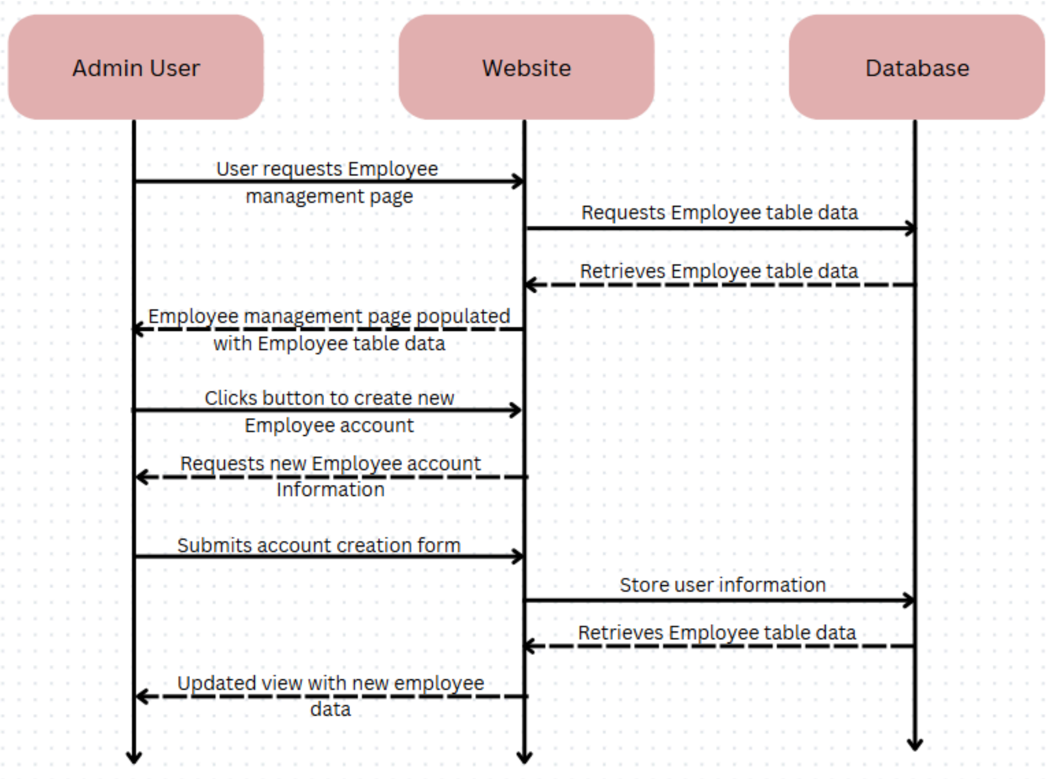
- **Use Case Name:** Order Confirmation and checkout process
- **Use Case Number:** 5
- **Authors:** Tim Yang
- **Actors:** Customer
- **Overview:** As a customer, I want to be able to view and manage employee schedules so I can create and communicate work hours effectively.
- **References:** User Interface/ Database
- **Typical Flow Description:** Customer examines the items in their shopping cart. Customer confirms the order. System will display the order details for payment. Customer proceeds to pay for the items.

Sequence Diagrams:

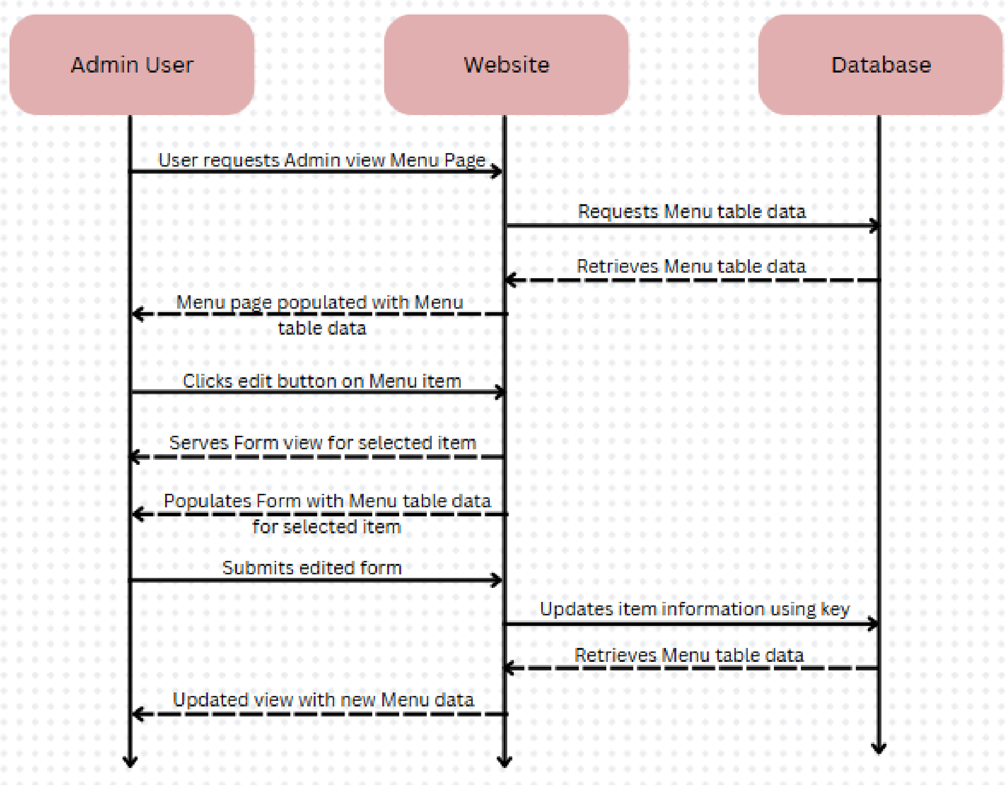
1. **Customer orders his/her meal from menu**


Customer
Website
Database
User arrives at the web application after doing a sign in
Displays Menu(Landing page)
User selects an item from the menu
Request the items information
Provide the requested item's price, and description
Displays the price and description of food item
User selects the quantity and goes to shopping cart
Sends the quantity
Outputs the total price in the shopping cart
Displays the total amount and order details

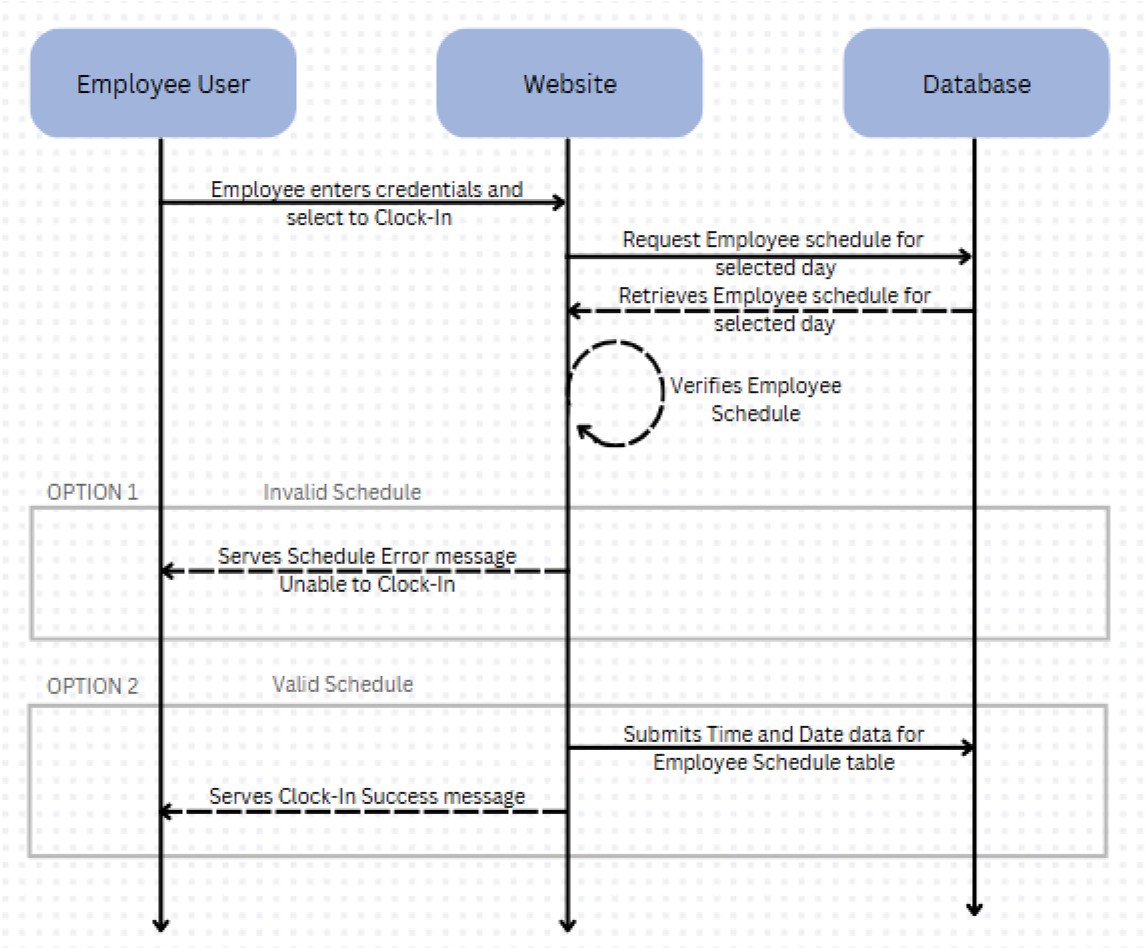
2. Admin User creates new Employee account


Admin User
Website
Database
User requests Employee management page
Requests Employee table data
Retrieves Employee table data
Employee management page populated with Employee table data
Clicks button to create new Employee account
Requests new Employee account Information
Submits account creation form
Store user information
Retrieves Employee table data
Updated view with new employee data

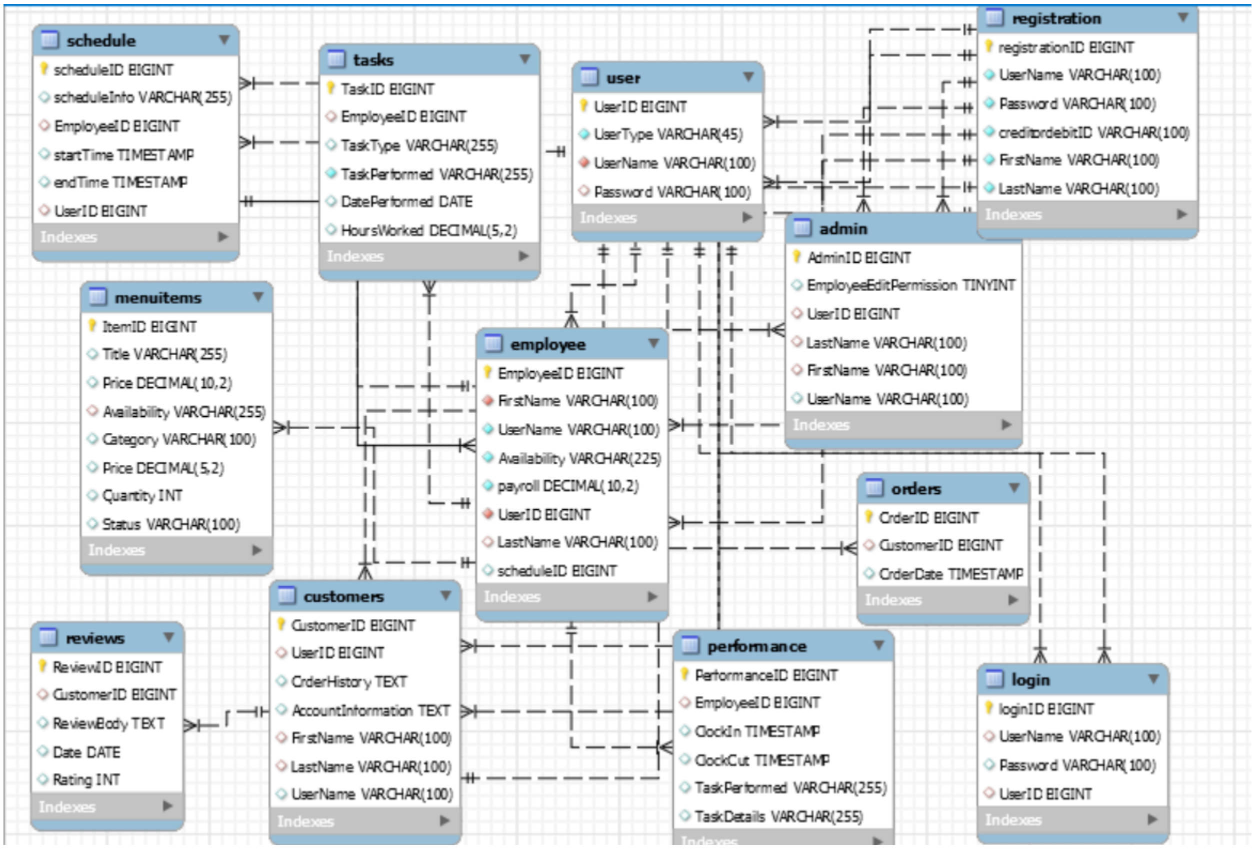
3. Admin User updates Menu Item


Admin User
Website
Database
User requests Admin view Menu Page
Requests Menu table data
Retrieves Menu table data
Menu page populated with Menu table data
Clicks edit button on Menu item
Serves Form view for selected item
Populates Form with Menu table data for selected item
Submits edited form
Updates item information using key
Retrieves Menu table data
Updated view with new Menu data

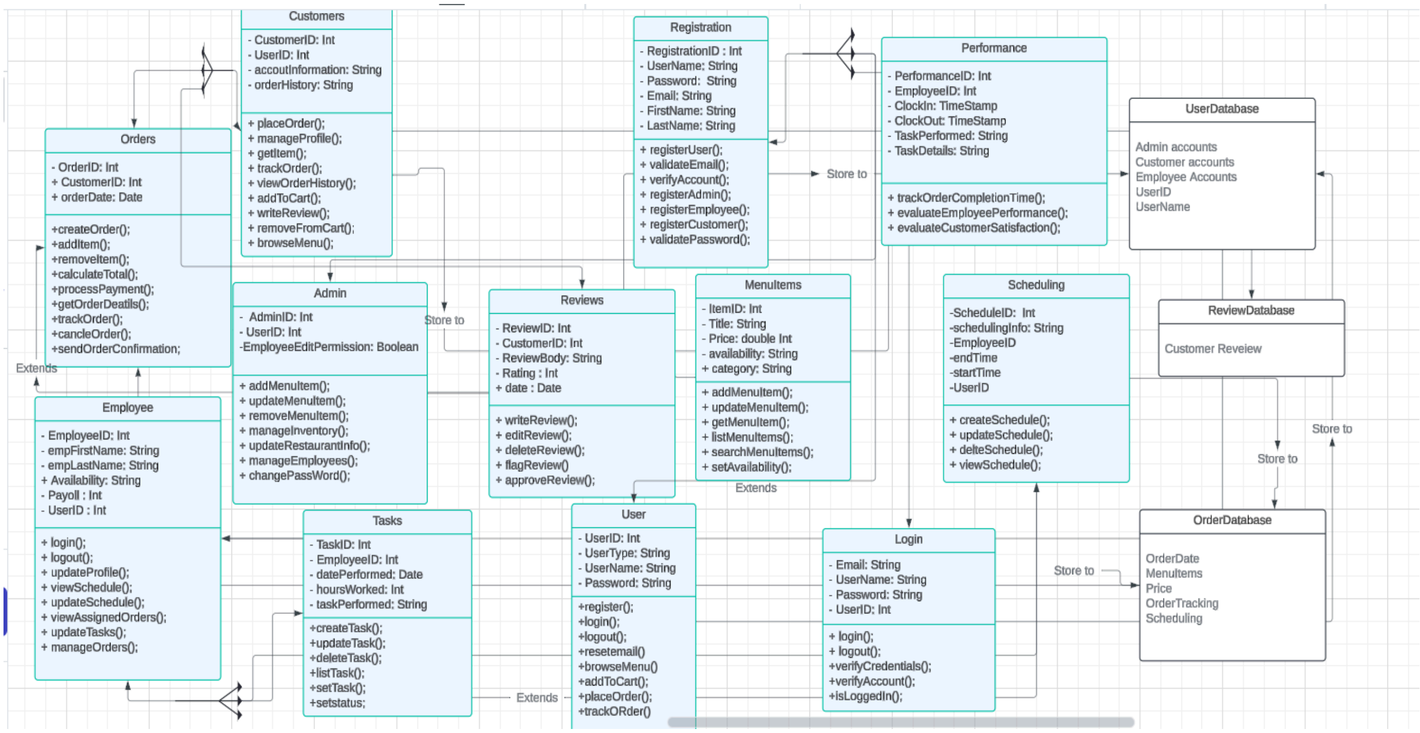
4. Employee User clocking in for shift


Employee User
Website
Database
Employee enters credentials and select to Clock-In
Request Employee schedule for selected day
Retrieves Employee schedule for selected day
Verifies Employee Schedule
OPTION 1
Invalid Schedule
Serves Schedule Error message Unable to Clock-In
OPTION 2
Valid Schedule
Submits Time and Date data for Employee Schedule table
Serves Clock-In Success message

Database Schema:



UML Class Diagrams:



Technologies and Tools:

Front End: JavaScript, React.js, HTML, CSS
Back End: Java, Spring boot
Database: MySQL
Cloud Service: Amazon Web Services
Software Testing: Jest, JUnit

Front End:

JavaScript:

- Chosen: A core scripting language for creating dynamic and interactive web pages.
- Reason: Enhances user experience with real-time interactions and updates in using JavaScript.

React.js:

- Chosen: A JavaScript library to help build user interfaces.
- Reason: Helps make reusable user interface components, improving development efficiency.

HTML:

- Chosen: Used to help structure content on web pages.
- Reason: Provides a framework for the web application.

CSS:

- Chosen: Used for styling HTML elements.
- Reason: Allows the customization of appearances on a web application, improving visuals and user-friendliness.

Back End:

Java:

- Chosen: For backend logic and business processes.
- Reason: Provides a robust and secure platform. Its object oriented nature helps make cleaner code for easier maintainability and has scalability to handle large-scale operations.

Node.js:

- Chosen: Used for server-side scripting, microservices and asynchronous operations.
- Reason: Allows for the uses of JavaScript on the server side.

Spring Boot:

- Chosen: Creates spring-based applications, API’s and configuration.

- Reason: To simplify setup and development by providing default configurations to reduce extensive configuration uses. Can be used to create and deploy applications quickly, it also supports services in which classes in Java will use.

DataBase:

MySQL:

- Chosen: A relational database management system to store and manipulate data.
- Reason: Database system that has complex queries for transactions, offers security features, and is compatible with the tools and libraries listed.

Cloud Service:

Amazon Web Services:

- Chosen: Cloud computing platform service.
- Reason: Offers a scalable and secure solution for managing web applications with a range of services to support maintenance and deployment.

Testing Framework:

Jest:

- Chosen: Programming Toolkit specifically made for testing, is straightforward.
- Reason: This toolkit is specifically made for testing JavaScript and React.js that comes with built-in tools, making it easier to test codes.

JUnit:

- Chosen: A Testing tool for integration, tests for quality.
- Reason: This tool works well for testing Java applications and work seamlessly with backend code.

Development Environment:

Tool:

Git:

- Version control system that can be used to track change on code and different versions of the project.
- Repository Hosting: GitHub
- Setup:
  - Create a GitHub repository for the TastyTreat project.
  - Clone repository to local machines:

  - ○ Practice branching strategies for managing development workflows.

Collaboration Tools:

Jira:

- Create and assign tasks to users.
- Track progress using dashboards and reports.
- Team collaboration from notifications and messaging.

Scrum:

- Scrum is an agile methodology which encourages collaboration
- Breaks down projects into smaller, more manageable tasks named as sprints.

Figma:

- A collaborative web design application.
- Used for creating the prototype design for the TastyTreat web application.
- Build future references/insights for the final project.

Lucid.app:

- A collaborative tool to allow diagramming of different sorts.
- Used for creating UML diagrams to enhance implementation efficiency.

Canva:

- A graphic design platform to allow teams to work together on designs.
- Create user stories for a boost in visualizing design and code for the project.

Discord:

- A communication platform with real-time communication.
- Has channels, servers, role settings and file sharing, promoting an easy and effective collaboration.

Testing Framework:

Jest:

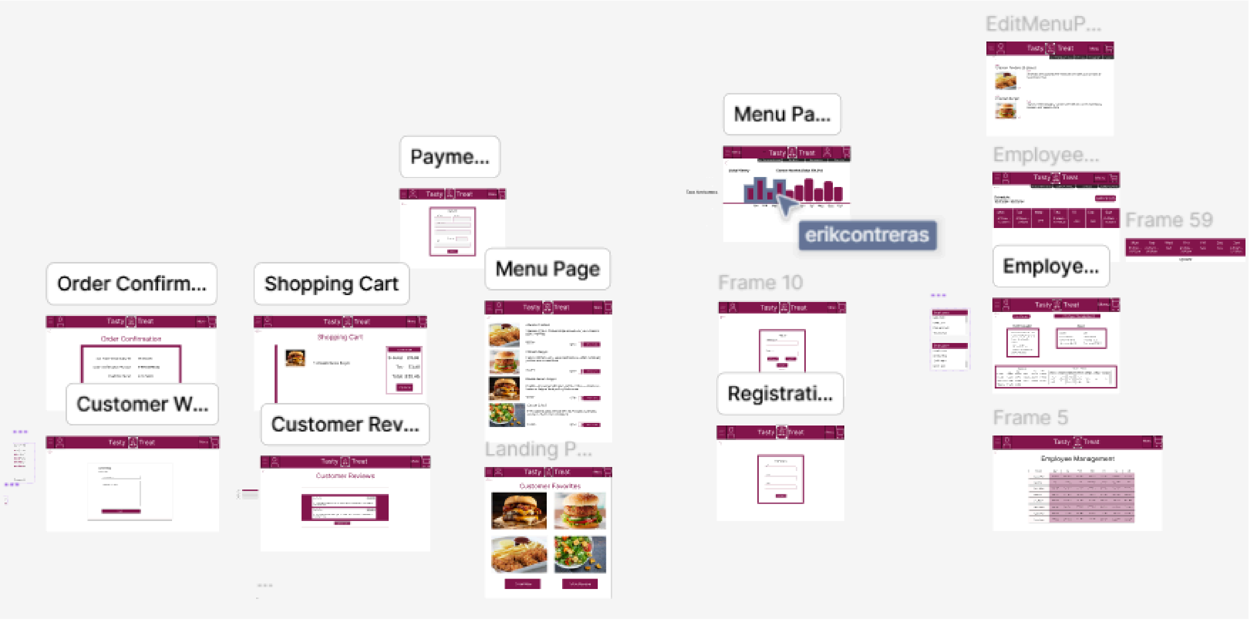
- Testing unit in front-end for JavaScript and React.js components.
- Setup: Install Jest and configure it for the TastyTreat project.

JUnit:

- Back-end testing unit for Java and Spring Boot.
- Setup: Add JUnit dependency to the following Maven/Gradle project on the Spring initializer web page.

Part 2: Design Prototype

Figma design prototype:



Risks and Mitigations:

Performance Issues:

Risk: Slow performance due to low scalability or inefficient queries.

Mitigation: Optimization of database indexes and queries. Montier application performance and scale resources if needed.

Deployment:

Risk: Deployment failures from unforeseen errors, making the application function inefficiently or not at all.

Mitigation: Go back to a previous version of the web application that is stable to begin again from there.

Team Collaboration Challenges:

Risk: Lack of communication and coordination among team members may hinder the project's success and time expectancy.

Mitigation: Use of collaboration tools as needed for clear communication and task management.

Conclusion:

Tasty Treats provides a smooth and efficient experience for customers wanting to order food, employees managing their schedule, and managers operating their restaurant by utilizing pathways catered to each user type.

The front end offers an easy to use and vibrant user interface by utilizing HTML and CSS paired with JavaScript and React.js for fast performance during client-side rendering. Functional and performance requirements will be tested using the Jest Testing Framework.

The back end provides reliable data transfers secured with credential checks and submission checks. Utilizing Java and Spring Boot to handle data sanitization, server security checks, and an object-oriented structure while Node.js provides efficient communication between the front end and the back end. Secure data transfers and performance requirements will be tested using the JUnity Testing Framework.

The data will be created and stored using SQL for structured storage and queries. Amazon Web Services will be utilized to host the website and the database.

Customers can view and customize their orders with a secured payment method for when they're ready to place an order. Personalized profiles and order confirmations ensure customers are able to get what they want.

Employees can view and manage their schedules while also being able to view the payroll information their accounts track for them.

Administrative users will be able to simplify their business operations by being able to track their customer satisfaction and employee performance. Admin users can also control Employee account creation and information to provide an extra layer of security on the business operations side.

Team members Contribution:

Tim Yang
Abhijit Singh Ubhi
Gurinderpreet Singh
Erik Contreras
Randy Kim
Zyed Azizi