

Pizza Restaurant Online Ordering System

Group 5

**Created by: Mark Legaspi, Jacob Lenz, Felicia Mach, Garret
Thompson, & Risa Luthor**

First Draft: October 8, 2023

Second Draft: October 15, 2023

Third Draft: October 22, 2023

Fourth Draft: October 29, 2023

Table of Contents

1. Project Vision and Description	3
Scope of Work.....	3
2. Team Roles.....	6
3. Collaboration Methodology	8
4. The Definition of “Done”	9
5. Product Design	10
Use Case Diagram:	13
6. Sprint 1 Retrospective Summary Report	21
7. Sprint 2 Retrospective Summary Report	22
8. References	23

1. Project Vision and Description

In a dynamic restaurant industry, the proprietors of a beloved pizzeria have hired our team to construct a more than just website because they want to create a digital entrance that captures the spirit of their physical establishment in a dynamic restaurant market. In this project, adapting to shifting client expectations, emphasis is placed on ease, accessibility, and digital interaction.

The goal of this project is to create an online presence that captures the coziness, flavor, and charm of the pizza rather than simply creating a website. Our objective is to establish a seamless connection between the restaurant and its customers by offering a user-friendly platform that enables customers to easily browse the menu, personalize orders, and complete transactions. By starting this initiative, we hope to reimagine the dining experience and make it more successful in the digital era while maintaining the distinct character of the pizzeria.

Scope of Work

Website Development

- *User-Friendly Design:* The website will have a simple, contemporary look and feel with a simple user interface. It will be device-specifically tuned to provide the best possible user experience on PC, tablets, and smartphones.
- *Repository of information:* The restaurant's website will act as a central location for information. It will contain thorough explanations of the background, atmosphere, and distinguished features of the restaurant. The restaurant's interior, food and employees will all be captured in high-resolution images to create an immersive

environment. A visual feast will be provided by the menu section, which will include enticing pictures of each meal. Descriptions and prices will be readily available.

Online Ordering

- *Ordering:* Customers will place orders without difficulty. They can look over the menu, pick what they want, and personalize their orders. They will be guided through the process by a user-friendly interface, which will make it simple for them to add or remove toppings, choose a crust type, and define quantities.
- *Order calculation:* As clients adjust their orders, real-time calculations will make sure they are updated with the final cost. A thorough itemized bill will be available for viewing prior to placing the transaction, and any active discounts or specials will be immediately added.
- *Payment processing:* The website will use secure payment channels to ensure the privacy of users' financial and personal information. There will be a variety of payment methods accessible to suit different preferences, including credit cards, digital wallets, and online payment systems.

Menu Management

- *Store Manager Access:* To efficiently maintain the menu, a dedicated administration portal will be made available to the store manager. They are able to changes, prices, add new things, and activate or disable particular items based on stock levels. Limited-time deals, and seasonal promotions might be planned in advance.

- *Inventory Control:* To make sure that the menu represents current availability, the system will interface with the restaurant's inventory management system. Out-of-stock items will be automatically marked as not available for ordering.

Customer Support

- *FAQ Section:* The Frequently Asked Questions (FAQ) sections will cover all frequently asked questions from customers about delivery schedules, payment choices, and special requests.

Conclusion

Modern online ordering technology that improves customer satisfaction and optimizes restaurant operations will be the result of this project. We will equip the pizza shop with the tools it needs to succeed in the digital age by creating a user-friendly website, establishing secure payment processing, and offering powerful menu management tools. This complete system will enhance sales and customer loyalty by not only meeting but exceeding client expectations.

2. Team Roles

Product Owner (Mark Legaspi)

The product owner is who defines the project, meaning the state the why, who, and what is the project retaining. Product owners are the ones that have the final say on how the project will flow and product decisions.

Product Owner Responsibilities:

- Set Product Goals
- Maintain a product backlog of items that are needed to reach the product goal.
- Collect feedback from users, customers, and stakeholders on the product increments.
- Regularly update and fine tune the backlog with the developers.

Scrum Master (Jacob Lenz)

The Scrum Master holds the team accountable for their working agreements, scrum values, and the scrum framework. The Scrum Master is responsible for keeping the team on track and avoiding any stray away from the goal of the project.

Scrum Master Responsibilities:

- Coach and mentor, the team by having meetings about the project.
- Making sure that we don't stray from the scope and goals of the project.
- Help the team and stakeholders understand what our plans are and how we plan to implement them.

Developer Team (Felicia Mach, Garret Thompson, Risa Luthor)

The developer decides how to accomplish the project based off what the product owner applied for the developers to begin working on.

Developer Responsibilities:

- Organize and plan how to accomplish the work at hand based off what was the agreed terms of quality or also known as the definition of done.
- Making sure that the target goal is being met.
- Having planned meetings to make sure the original plan is still good or if there were changes that needed to be made to the plan.

3. Collaboration Methodology

Since the team is utilizing the Scrum methodology it is paramount that it is easy to stay in contact with each other and it is easy to share files between team members. The team will make use of CTU Messenger, a Slack Workspace, Zoom, and GitHub.

- **CTU Messenger:** This form of communication will be used when something of high priority needs to be communicated. This is because all team members will get an alert on their phone that they can respond to quickly.
- **Slack Workspace:** This will allow an easy way for the team to communicate low priority messages. It also provides an easy way for file sharing and to review the recorded Zoom meetings.
- **Zoom:** This will allow for group meetings that are scheduled each Thursday following the class's live chat. This meeting will allow the team to touch base and during the second class will act as the weekly Scrum meeting. These will be recorded so they can be referenced later.
- **GitHub:** When it comes time to code the system the team will use GitHub to help with version control and make it easier to merge code together.

4. The Definition of “Done”

“In short, DoD (Definition of Done) is a shared understanding within the Scrum Team on what it takes to make your Product Increment releasable (Madan, 2019).” This means that the overall project meets all the functional requirements, non-functional requirements, and the quality standards put forth by the product owner(s). These will be specified later in the product design section. Below are some key elements of what will help define when the product is done.

- **Code:** The code is free of errors, any common vulnerabilities, and all team members have reviewed the code. The code will be commented properly to make future sprints easier.
- **Requirements:** All the functional and non-functional requirements are met, and all their acceptance criteria are covered.
- **Testing:** The code passes all static and dynamic testing.
- **Deployment:** The system is ready to be deployed and any dependencies are up to date. Also, any configuration files are ready.
- **User and Stakeholder Approval:** The users, any stakeholders involved, and the product owner have reviewed the system and find it acceptable.

5. Product Design

Main Components

User Interface (UI): This is the system's front end. It's the part that customers interact with. It can be a website or a mobile app.

Database: This stores all the data the system needs to operate. This includes details of the menu items, user accounts, orders, and other information.

Backend Server: This is the software that runs on a server. It connects the UI to the database, handles user requests, and implements business logic.

Payment Gateway: This is a third-party service that handles payment processing. When a customer places an order, the backend server sends a request to the payment gateway, which processes the transaction and returns a confirmation.

Notification System: This notifies customers when their order has been received, is being prepared, and is ready for pickup or delivery.

Security: This component ensures that all transactions are secure and user data is protected.

Uses Case 1: User Registration and Authentication

- **Description:** This use case allows customers to create accounts and log in securely.
- **Components:**
 - Registration Page
 - Authentication Logic
 - Email Verification Process
- **Security Concerns:** Password encryption, secure email verification.

Use Case 2: Menu Presentation

- **Description:** Customers can browse the menu with categories, descriptions, and prices.
- **Components:**
 - Menu Categories Display

- Menu Item Display
- **Security Concerns:** Ensure menu data is protected from tampering.

Use Case 3: Adding Items to Cart

- **Description:** Customers can add menu items to their cart and customize their orders.
- **Components:**
 - Cart Functionality
 - Customization Options
- **Security Concerns:** Validate input to prevent injection attacks.

Use Case 4: Cart Management

- **Description:** Users can review and modify items in their cart.
- **Components:**
 - Cart Display
 - Item Removal
 - Quantity adjustment
- **Security Concerns:** Ensure secure session management and user authentication.

1. *Payment and Checkout*

Use Case 5: Payment Options

- **Description:** Integrates secure payment methods, including credit cards and digital wallets.
- **Components:**
 - Payment Gateways Integration
 - Payment Processing Logic
- **Security Concerns:** Compliance with PCI DSS.

Use Case 6: Order Confirmation

- **Description:** Notifies customers of successful orders and provides estimated delivery/pickup times.
- **Components:**
 - Email Notifications
 - On-screen Confirmation
- **Security Concerns:** Protect customer email data, use secure communication.

2. Menu Management

Use Case 7: Store Manager Dashboard

- **Description:** Provides the store manager with a real-time dashboard for menu management.
- **Components:**
 - Dashboard UI
 - Menu Item management
 - Pricing Update Feature
- **Security Concerns:** Role-based access control for the dashboard.

Use Case 8: Inventory Integration

- **Description:** Automatically updates item availability based on inventory status.
- **Components:**
 - Inventory System Integration
 - Item Status Update Logic
- **Security Concerns:** Protect inventory data and ensure secure API communication.

Non-Functional Concerns

Security

- Encrypt user data and conduct routine audits and security assessments.

Performance

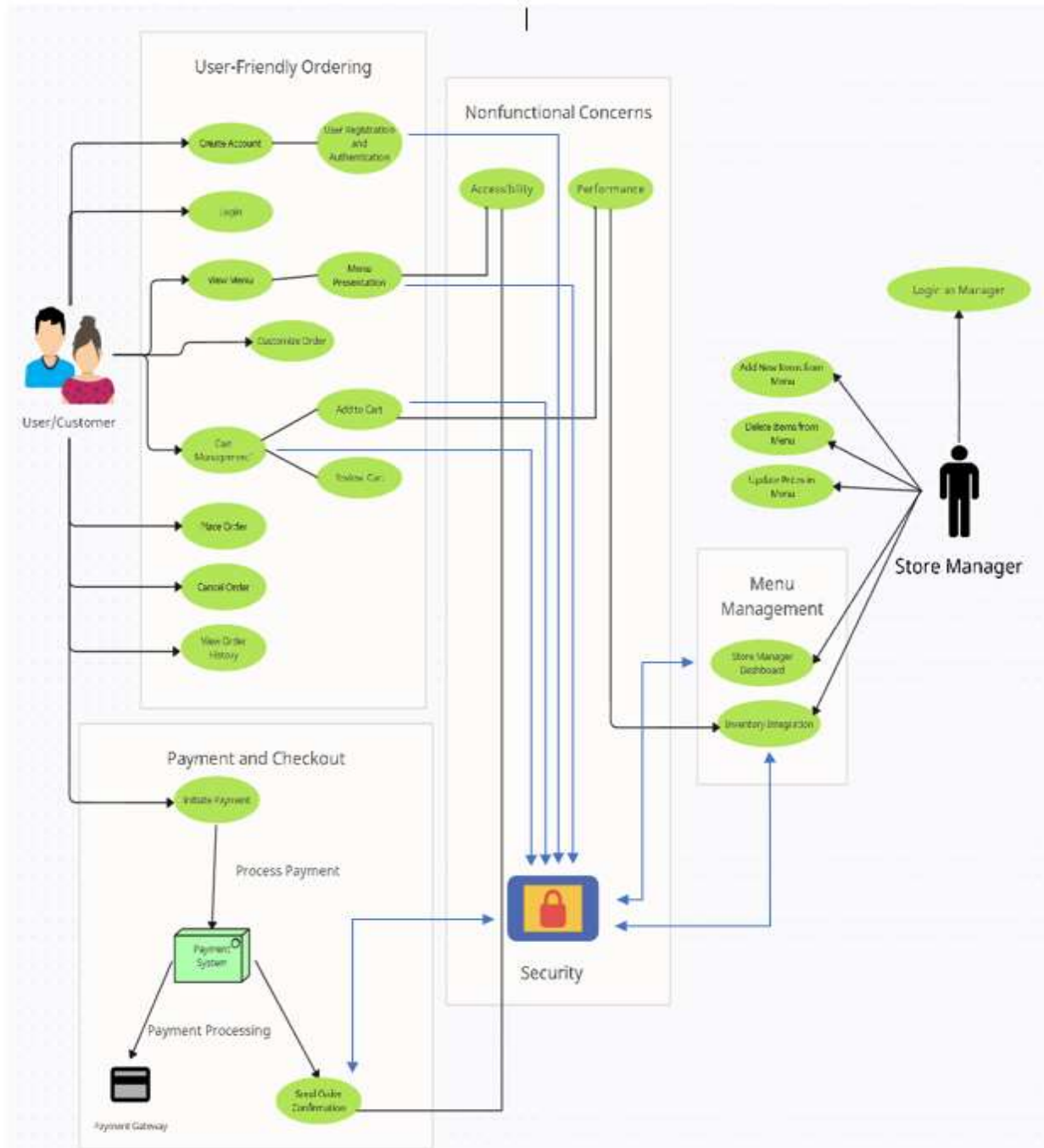
- To ensure system scalability, use load testing and database retrieval queries should be made more efficient.

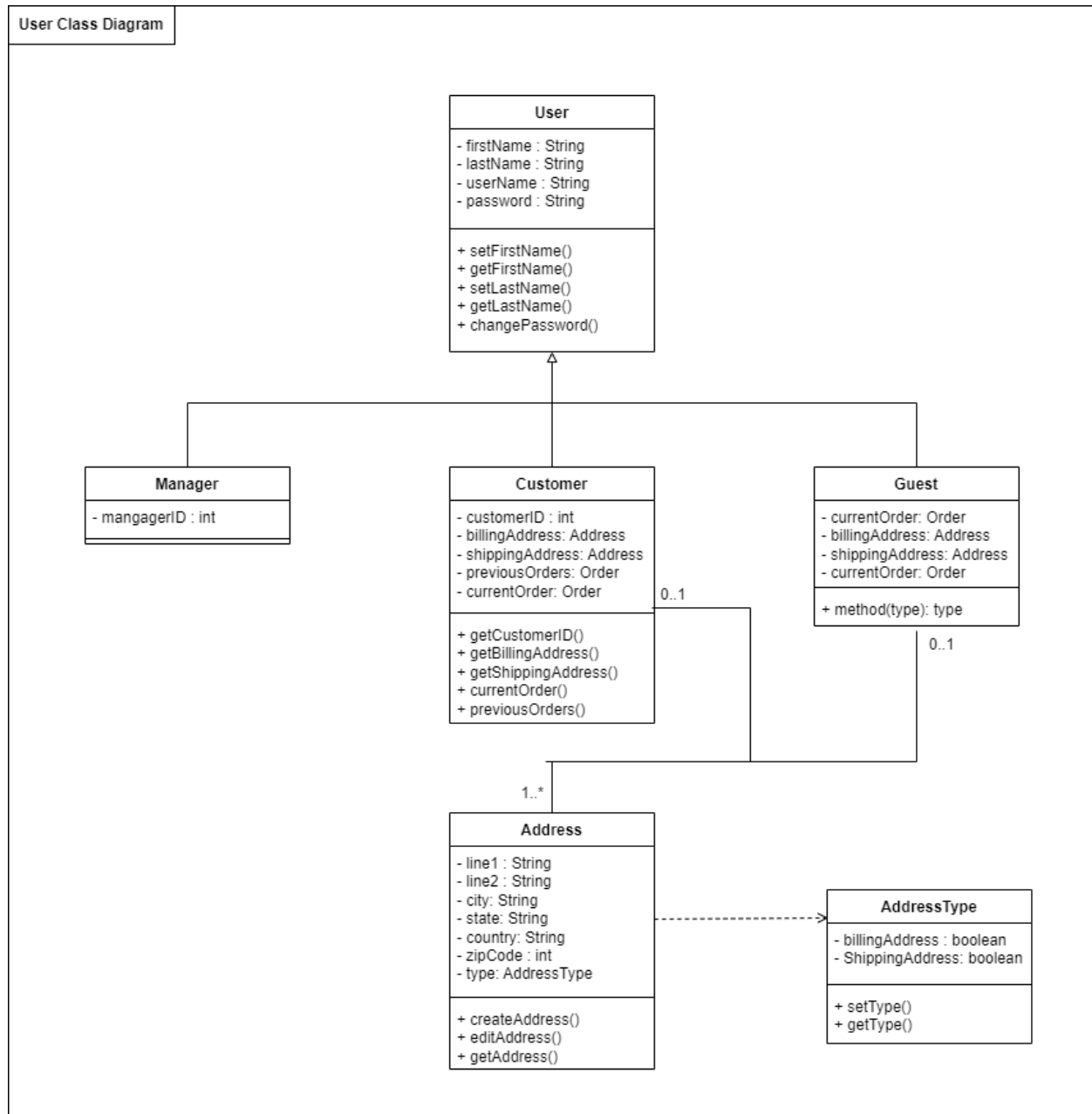
Accessibility

- WCAG 2.1 standards compliance is required and utilize assistive technologies to evaluate usability.

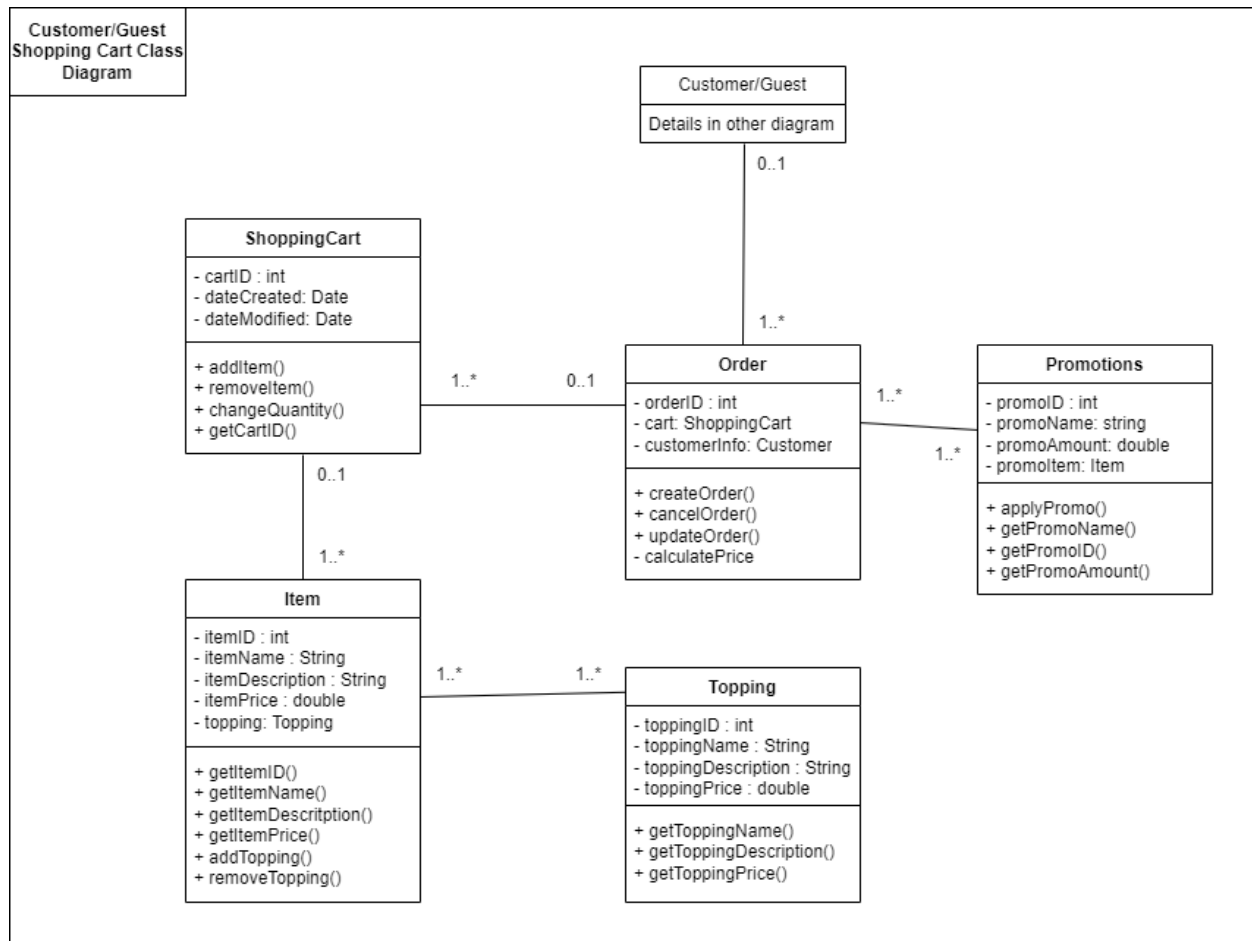
The key use of cases and components are outlined in this high-level design, with a focus on user-friendly ordering, safe payment processing, and menu administration. Additionally, it addresses important security and functional issues. Architectural selections, detailed technical design, and development task would be part actual implementation.

Use Case Diagram:

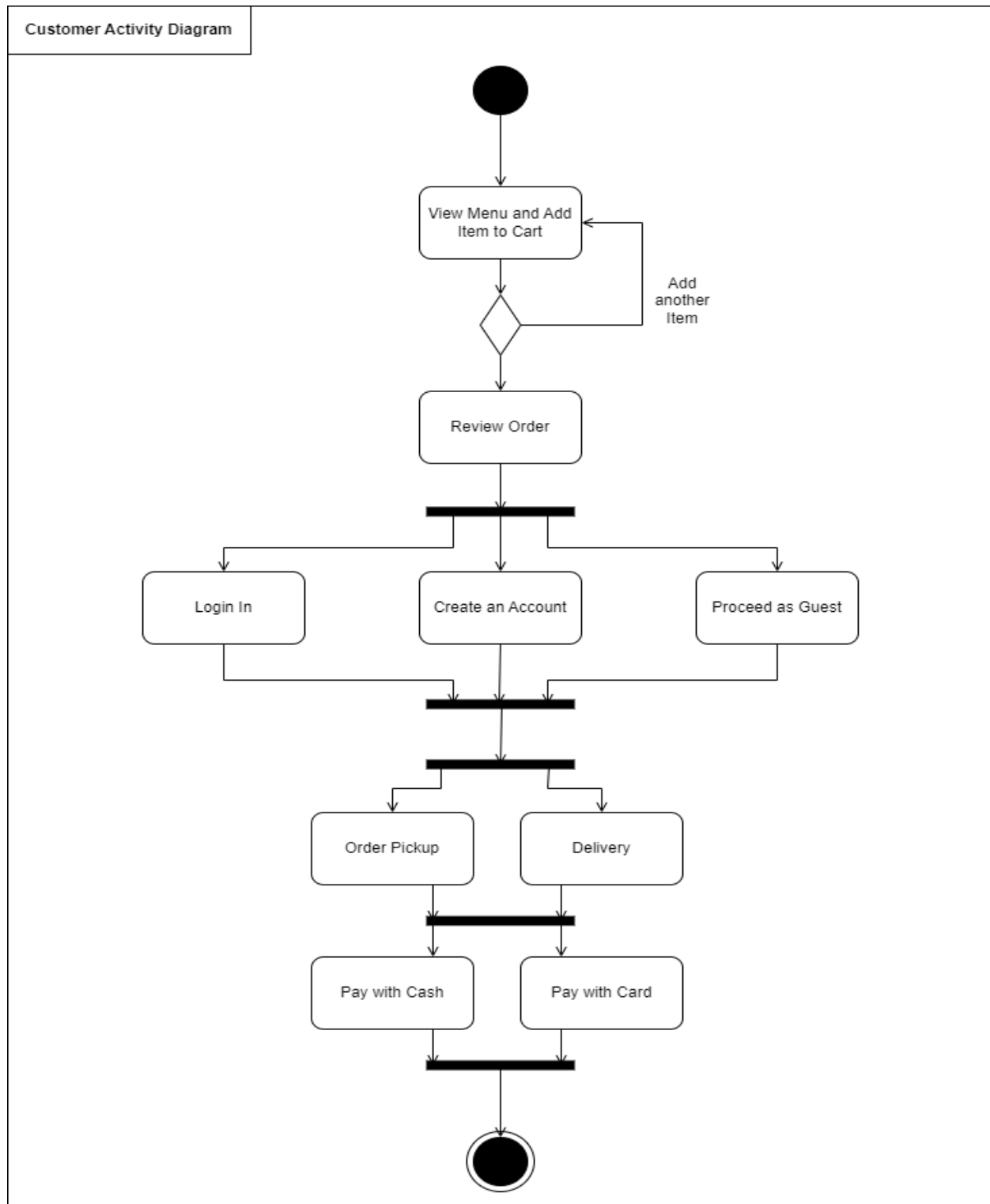




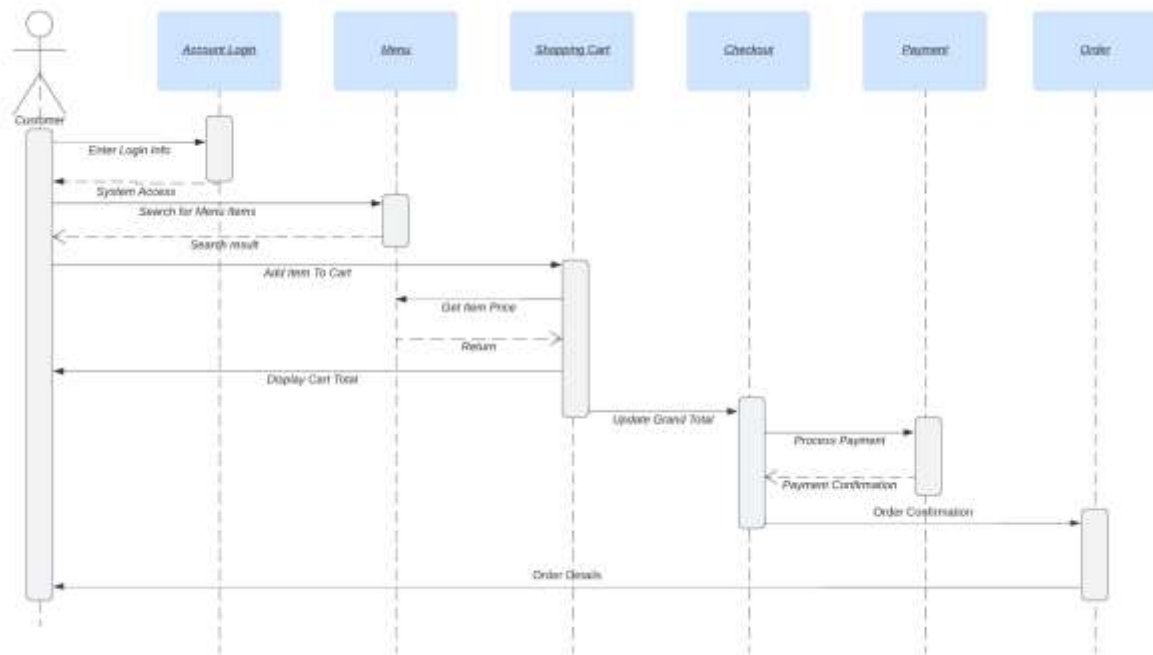
This class diagram shows the main user class and that the manager, customer, and guest classes inherit from it. The address class is removed from the customer and guest classes since a customer may have different billing and delivery addresses. This also comes into play later in the design of the database.

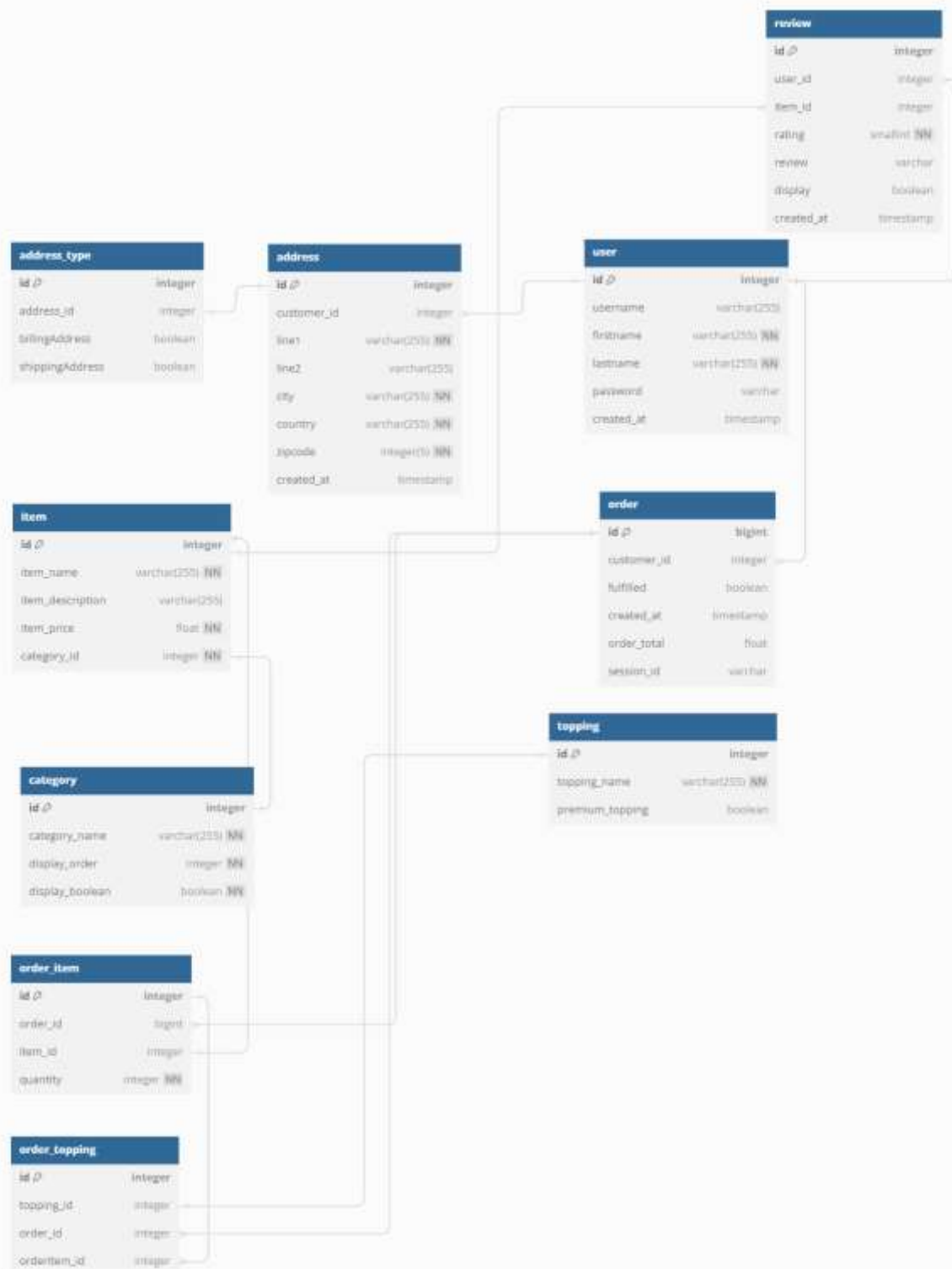


This class diagram interacts with the previous class diagram but was separated out for easier viewing. The diagram shows the different relationships between the classes that support the logic behind the shopping cart system that it is going to be implemented.



This activity diagram shows the process of a customer picking the items they want from the menu, adding them to their cart, the different account options, pickup or delivery, and what payment method they wish to use.





The above is a visual depiction of the database schema that will support the website. It is normalized to the 3rd level. The lines between tables show the relationships and where the foreign keys get their values from. Within each table the data type is declared and any restrictions such as not null. What isn't shown is that there will be user accounts created with restricted access that can only execute their needed queries. The front-end code will have safeguards in place to prevent the chance of an SQL injection attack.

6. Sprint 1 Retrospective Summary Report

Things That Went Well
<ul style="list-style-type: none">• Project Document, templates, and sprint planning gave the team members a strong starting point.• Team members' active involvement and contributions• Learning from each other during team meetings• Accomplished the sprint 1 objectives.• GitHub and Slack allowed the team to stay in contact easily and maintain one source of truth for the code.• Great communication within the team
Things That Could Have Gone Better
<ul style="list-style-type: none">• Difficult to get the whole team together at times.• Difficulty of working on the project on Mac OS.
Things That Surprised Us
<ul style="list-style-type: none">• Seamless integration of new tools or techniques by team members.• How well separate parts of the project came together with little difficulty.
Lessons Learned
<ul style="list-style-type: none">• The importance of maintaining stable version control between team members.• Knowledge gained from working with teammates during the sprint.• That communication is key between team members as tasks are completed.

7. Sprint 2 Retrospective Summary Report

Things That Went Well
<ul style="list-style-type: none"> • Strong communication and collaboration • Team members' active involvement and contribution • Accomplishment of sprint objectives • Continuous improvement of the project • Helped each other with PHP coding and checking each other's work.
Things That Could Have Gone Better
<ul style="list-style-type: none"> • The IDE's being more universal between windows and mac. • Better communication to make sure that team members weren't working on tasks that were already being worked on or completed. • Prioritizing features from user stories vs optional features
Things That Surprised Us
<ul style="list-style-type: none"> • The seamless integration of new tools or techniques by team members. Ensuring that we can all access materials and learn new languages. • How well the code went together at the end. • The team was very flexible when we ran into unforeseen difficulties. • The speed at which certain tasks were completed when compared to Sprint 1.
Lessons Learned
<ul style="list-style-type: none"> • Knowledge shared by team members during meetings allowed each other to benefit from their unique talents and skills. • Tolerance and patience are key to making the team successful in the long haul. • We should have picked what version of PHP the site was going to run on. Ran into issues on deployment with deprecated functions that were used. • Use the team's capacity of adaptation as a strength to overcome unforeseen obstacles. • The importance of regular team meetings.

8. References

Learning guides. FREE Learning Resources: UML, Agile, TOGAF, PMBOK, BPMN. (n.d.).

<https://www.visual-paradigm.com/guide/>

Madan, S. (2019, December 19). *Done understanding of the definition of “done.”* Scrum.org.

<https://www.scrum.org/resources/blog/done-understanding-definition-done>

The Scrum Team | Scrum Alliance. (n.d.). [https://resources.scrumalliance.org/Article/scrum-](https://resources.scrumalliance.org/Article/scrum-team#:~:text=%22The%20product%20owner%20maximizes%20the,owner%2C%20developers%2C%20and%20stakeholders.)

[team#:~:text=%22The%20product%20owner%20maximizes%20the,owner%2C%20developers%2C%20and%20stakeholders.](https://resources.scrumalliance.org/Article/scrum-team#:~:text=%22The%20product%20owner%20maximizes%20the,owner%2C%20developers%2C%20and%20stakeholders.)