

SW Engineering CSC648/848
Application Name: PizzaCraze
Section 04, Team 07

Monisha Mekala - Team Leader/co-Back-end Lead
Aishwarya Magar - Front-end Lead
Malieka Sutaria - Scrum Master
Nicholas Pagcanlungan - co-Back-end Lead
Joey Palanca - GitHub Master

Milestone 1
September 27, 2023

Revision History

Document Version	Date	Changes
1.0	27 th Sep, 2023	Initial Submission

Milestone 1

1. Executive summary

PizzaCraze is a well-established restaurant based in the heart of San Francisco. Our goal with our web application is to design an effortless way for our customers to get their perfect pizza fast. This web application is perfect for those who are avid pizza enjoyers, people in a time crunch, and even influencers looking to brag on social media. This web application will be important for us to be able to share our wonderful pizza with people across the area.

Our product, PizzaCraze, is a web application that will allow the user to browse a wide variety of customizable items on our menu for delivery. The user will be greeted with a wonderfully designed home page that is easy on the eyes, with a clear button that will direct them to our menu. From there the user can effortlessly click on the items they want to order, and will be prompted with a well-designed customization UI to craft their orders. After that, the item will be placed into their shopping cart. When the user is ready to make the order, they can click on the shopping cart at the top of the page where they can see the entirety of their order. From here the user has the option to make changes to their order, and when they are satisfied with their decisions, they can check out. If the user is not already logged in, then they will be prompted to log in/register an account. After that, the user will then be notified when the delivery has been made. Then they will have the option to share their post onto X with a cool poster. This is designed to be an easy effortless way for people to order amazing pizza right to their doorstep.

We are a team of dedicated students, here to design an easy way to get pizza to those who want it. With a multitude of experiences and backgrounds, we are confident in our skill to design

this amazing web application. From the front end to the back end, we have everything covered to develop this web application.

2. Personas and User stories:

Key Personas

1. John Smith

ID: 9413213

a. Age: 32

b. Profession: Food Critic

c. Hobbies: Kite flying, cooking, camping, and bird watching

User Story: John Smith is a world-famous food critic. Competing on Chopped in 2013 allowed him to fulfill his dream of showcasing his culinary skills to renowned chefs and gaining the opportunity to train under Michelin-star chefs. With his experience and knowledge, John has now set out to criticize new and upcoming restaurants in hopes of finding new hidden gems around the world. During his free time, he enjoys spending time with his two daughters, ages 4 and 12. Every Saturday John and his daughters go have a father-daughter day where they spend the day participating in various fun activities and end the day off with each other eating pizza and trying new pizza combinations.

Usage: John and his daughters can customize their pizzas as much as they desire, they can choose the level of spiciness and try new combinations. John can also post his reviews of our pizzas on his social media, allowing his followers to see his overall critiques on the new combinations of pizzas he and his daughters create each week.

2. Samantha Joy

ID: 2143642

- a. Age: 21
- b. Profession: Full-time University Student
- c. Hobbies: Art, painting, sculpting, dance, and swimming

User Story: Samantha is a full-time university student who is pursuing a degree in Ethics and a minor in Graphic Design. At a very young age, Samantha knew she had a passion for art as she won many awards for her painting and clay sculptures. Being the oldest of five children, she is often tasked to prepare dinner for her and her siblings. Often due to the fact Samantha is a full-time university student, she does not always have time to cook meals for all her siblings. Takeout is the best option for Samantha as it's quick and easy.

Usage: Samantha can order a whole bunch of pizzas without taking time away from her studies and pleasing each one of her siblings without worrying if they are going to eat or make a big deal. She is also able to track her order and see which phase of the process her pizzas are in.

3. Brad Gustafson

ID: 7865911

- a. Age: 78
- b. Profession: Retired
- c. Hobbies: Enjoys laying on the couch, spending time with his grandchildren, baking, picking flowers, and creating unique bouquets for his late wife.

User Story: Brad is now a retired Biochemical Engineer. He worked at Trey Laboratories for thirty years before he retired three years ago after his wife got sick. Brad and his late wife Mary were married for fifty-five years and met each other on the first day of Chemistry class. Brad spent the past three years taking care of his wife who recently passed away due to

medical conditions. Brad picked up some new hobbies after his wife's passing which include baking and making bouquets for his friends' and family's events. Brad's newest obsession has been eating pizza at least four days out of the week. Brad enjoys fusion pizzas; his favorite is Korean BBQ Pizza with crispy fried chicken skin.

Usage: Brad calls himself a pizza lover. He loves to eat pizza and is always looking for an easy pizza website to use that allows him to customize his pizza as well as post it on his Facebook to show his friends. One issue Brad has is posting pictures of his food, but due to our new feature, we can provide a picture of the finished pizza which makes it easy to post and share. Brad is also able to use the ChatBot when he has any issues with placing his order or asking for new pizza suggestions when he feels bored with his usual order.

3. Data Definitions:

We've defined important data components and entities relevant to our meal-ordering website that will serve as the foundation for the functioning of our application. All documentation, user interfaces, and code implementations will uniformly refer to these components.

- **LoginDetails:** This table will operate as a central location for user data, which includes crucial information like usernames, passwords, emails, phone numbers, and addresses. It is essential to the login process for authenticating and approving users.
- **AdminDetails:** This table is used to keep track of administrators' login information. It includes crucial data such as admin usernames and the hashed passwords that belong to them. Only authorized people are allowed access to this table, protecting the security of administrative powers.

- **MenuTable:** A crucial element, this table will have an extensive list of the menu items that are accessible for ordering. Each entry contains important information including the names of the items, their categories, prices, and descriptions. Users utilize it as a starting point when they browse and choose things for their orders.
- **OrderTable:** This table keeps track of customer orders in chronological order. It records crucial data such as order IDs, user IDs connected to them, item IDs, quantities, total prices, order dates, and order statuses (such as pending, completed, or canceled). For managing and tracking orders, this information is essential.
- **CartTable:** This table serves as a temporary storage area for goods that customers have put in their shopping carts. It has entries for amounts, item IDs, linked user IDs, and cart IDs. Users can examine and change their selections before submitting a purchase, facilitating a seamless purchasing experience.
- **FeedbackTable:** Improving the overall user experience is made possible through user feedback. User ratings, comments, linked user IDs, item IDs, and feedback IDs are all stored in this table specifically for user feedback. It gives consumers a way to express their ideas and aids in the platform's upkeep of quality standards.

The database for our meal-ordering website is built primarily from these tables. We will go further into the implementation specifics of each entity as we move through the following milestones to ensure a strong and effective system that effectively meets the needs of our users.

4. Initial list of functional requirements:

1. Login (Priority 1): Users should be able to log in to the system with their usernames and passwords which they created while signing. Only once they log in, they can place an order.
2. SignUp (Priority 1): Users should be able to create an account in the system by giving a username, email address, and password. Only after they sign up, they can log in to the system.
3. Browsing the Menu (Priority 1): Users should be able to check out all the food items in the menu list by browsing the menu page. The menu page should have all the available items. Every item should have a name, a short description, and a picture.
4. Filter Items (Priority 2): Users should be able to filter the food items they want to see in the menu list. Filters are Veg, Non-Veg, and Vegan.
5. Adding Items to the Cart (Priority 1): Users should be able to add the food items they want to the cart. He should also be able to remove the food items he no longer wants in the cart.
6. Customize the Food Items (Priority 2): Users should be able to specify additional changes that they want in their food, like the level of spice, and should be able to choose the toppings.
7. Quantity of Food Items (Priority 1): Users should be able to increase or decrease the quantity of the food items they want to add to the cart.
8. Place an Order (Priority 1): Users should be able to place an order for all the items in the cart.
9. Tracking the Delivery Status (Priority 3): Users should be able to track their orders.

10. Share on Social Media (Priority 2): Users should be able to share the food they ordered on their social media accounts.
11. Feedback Form (Priority 2): Users should be able to contact the admins to give any feedback they want.

5. List of non-functional requirements:

1. Compatibility:
 - a. The application should be compatible with Google Chrome, Firefox, Safari, and Edge on both mobile and desktop.
 - b. The application should load within seconds depending on the internet connection
2. Storage Space:
 - a. The application and its database should be hosted on AWS
3. Usability:
 - a. The application will have a well-designed UX design for ease of use
 - b. UX design should be optimized for different platforms
4. Fault Tolerance:
 - a. The application should be able to handle up to 1,000 users at a time on a standard server
5. Availability:
 - a. The application should be available 24/7 as long as you are connected to the internet except for maintenance
 - b. Display message when on maintenance
6. Security:

- a. The application should safely store the user’s account information in a database
7. Development:
- a. Application code should be well documented, and managed from the GitHub repository.
 - b. Changes will be documented
 - c. Unit tests to test functionality

6. Competitive analysis:

Planned VS Competitors

Planned	Competitors
Log In	Optional Log In
Level Intensity	Quantity Of Items
Live photos of ordered pizza	Social Media Sharing
Chatbot	Live representatives
Delivery status	Alerts

A food ordering website offers numerous advantages by providing a convenient and efficient platform for users to browse, select, and order food online. Users can explore a diverse range of restaurants and cuisines from the comfort of their homes, saving time and effort. The website often includes detailed menus, prices, and user reviews, aiding customers in making informed choices. Additionally, online ordering reduces the likelihood of miscommunications compared to traditional phone orders. With features like order tracking and secure payment options, it enhances the overall user experience. For restaurants, a food ordering website expands

their reach, increases visibility, and streamlines order management, contributing to improved customer satisfaction and business growth. With that being said, our food ordering system has a new competitive edge that will set us apart from the rest. Many ordering systems will give you the option to check out as a guest, but for our application rather than having an option to check out using guests, we have been implementing a login requirement. This means our customers will need to log in or create an account to proceed to check out. When selecting a menu item with a specified "spicy intensity," you're choosing a dish based on the level of heat or pungency it possesses. This allows you to tailor your culinary experience to your preferred spice tolerance, ensuring that the selected dish aligns with your desired level of spiciness. Whether opting for mild, medium, or hot options, the spice intensity designation on the menu helps you customize your meal to suit your taste preferences. We also have an option that allows our customers to receive a picture of their finished pizza which they can then use to share on social media by downloading the picture that is provided. Rather than having live representatives, we have a chatbot that will help users recommend pizzas as well as any issues regarding their order. Finally, we have a delivery status system which is an online ordering system that provides real-time information about the current location and progress of your order. It serves as a valuable update, indicating whether the order is being prepared, out for delivery, or has reached its destination, offering customers transparency and assurance about the status of their delivery.

7. High-level system requirements:

- Frameworks: Bootstrap 5, ReactJs V18, ExpressJs V4.18.2
- Server Environment: NodeJs V18.17.1
- Web Server: Nginx V1.24.0

- Languages: HTML 5, CSS, Javascript.
- Database: MySQL V8.0.33
- Deployment Platform: AWS EC2, RDS
- Services: Github, Discord
- Supported Browsers: Google Chrome, Safari, Microsoft Edge, Firefox

8. Team:

Group Members:

- **Monisha** - Team Lead, Back End Lead
- **Aishwarya** - Front End Lead
- **Malieka** - Scrum Master
- **Nicholas** - Back End Lead
- **Joey** - GitHub Master

Detailed study plan:

- Aishwarya:
 - Technologies: React framework.
 - Specific Goal: Dive deep into React, understanding its components, state management, and routing.
 - 1st week: Learn the basics of React like JavaScriptXML, components, props, and event handling.
- Nicholas:
 - Technologies: JavaScript and Express framework.

- Specific Goal: Master JavaScript, particularly focusing on its usage in backend development. Also, become proficient in using the Express framework for building web applications.
- 2nd week: Learn the basics of JavaScript like arrays, loops, DOM manipulation, and event handling
- Monisha:
 - Technologies: AWS, Database Connection, and Express framework.
 - Specific Goal: Master the Express framework. This might involve tasks like establishing connections using JavaScript, and building a basic server with Express.
 - 3rd week: Learn the basics of Express framework like introduction to Node.js, routing in Express, HTTP requests and responses, middleware, and working with databases.

Key Milestones:

Monisha:

- Set up an AWS environment.
- Establish a connection between the front end and the database.
- Implement basic API endpoints for data retrieval and storage.

Nicholas:

- Gain a strong understanding of JavaScript fundamentals.
- Work on backend logic, data handling, and server-side operations.

Aishwarya:

- Dive deep into React, focusing on component structure and state management.
- Begin building out the front-end user interface.

Malieka:

- Facilitate team communication and progress tracking.
- Ensure that the Scrum process is followed effectively.

Joey:

- Manage the project repository on GitHub.
- Ensure version control, code review processes, and branching strategies are followed.

By following this structured approach, our team will have a clear path to learning and development, making the project progress smoothly and efficiently. We will have regular check-ins and updates to track the progress and address any challenges that may arise.

9. Checklist:

Task	Status	Additional notes
The team found a time slot to meet outside of class	DONE	
The Scrum Master shares meeting minutes with everyone after each meeting	DONE	
Everyone sets up their local development environment from the team's git repo	ON TRACK	

GitHub master chosen	DONE	
The team decided and agreed together on using the listed SW tools and deployment server	DONE	
The team is ready and able to use the chosen back/front-end frameworks	ON TRACK	
The team lead ensured that all team members read the final M1 and agreed/understood it before submission	DONE	