Software Design Specification

for

Nardo's Delivery APP

Version 1.0

Prepared by

Adrian Green	*620107205*	*greenadrian123@gmail.com*
Marton Clarke	*620148076*	*martonclarke@gmail.com*
Laren Parkes	*620155536*	*larenparkes@gmail.com*

Course Instructor:	
Course:	*COMP2140 – Introduction to Software Engineering*
Studio Facilitator:	*Aldane Milton*
Date:	*November 16, 2023*

Github Repo Link

https://github.com/itsWoj/NardosFoodDeliverySystem.git

1.0 Project Overview

The client, Nardos, the owner, and manager of Nardos One Stop Shop, seeks a solution that expands beyond the physical location. Their aim is to create a delivery app that not only streamlines in-store operations but also reduces the number of people physically present at the store, especially during late-night hours. This software aims to provide a safe and convenient way for students to access food on campus by enabling them to place orders for delivery, thereby minimizing the need for crowding at the store during peak hours. The challenge lies in developing an app that not only manages inventory, transactions, and employee scheduling but also offers a user-friendly interface for ordering and tracking deliveries, ensuring timely and secure access to food, especially during late-night hours.

2.0 Architectural Design

2.1 General Constraints

Three primary limitations were identified: system development time, user device limitations, and user competency. It will take some time to finish the system in its entirety. A key deliverable for the project's successful conclusion is quality, even if not enough time is given to fully satisfy all needs and specifications.

User device limitations refer to whether the company will have a computer that can access and operate the software on it. If the gadget they have is able to run and show the web-based system properly, then there will be a serious problem. Furthermore, the operating systems needed to operate this software correctly include Windows, MacOS, and Linux, which can support the use of Firefox and Google Chrome. , Firefox or Internet Explorer web browsers.

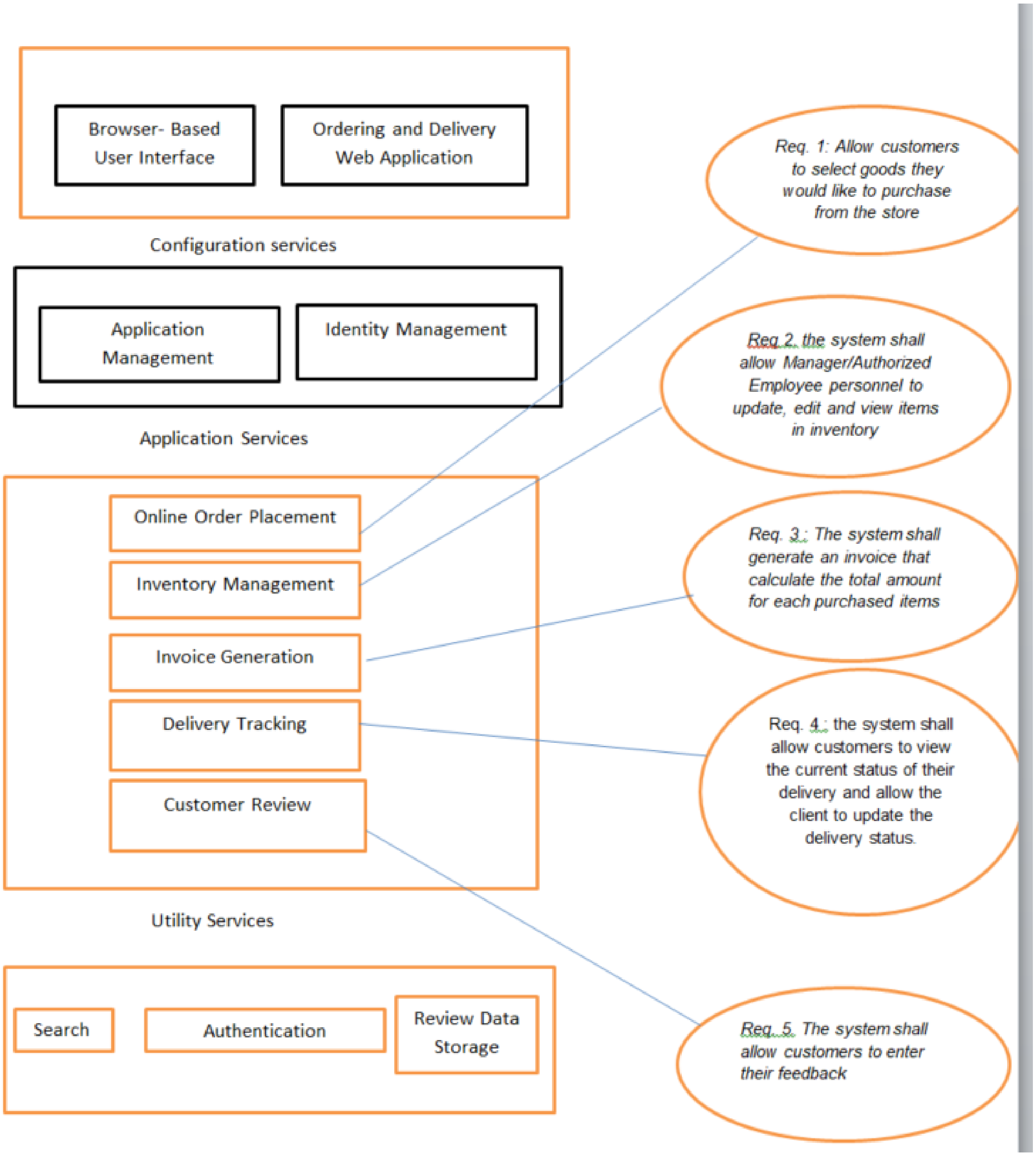
Competency: Since each developer has a different level of competency, it follows that some may find it difficult to design a particular feature of the software, and that users of the system are at least somewhat familiar with using a computer or smartphone.

2.2 Alternatives Considered

The client-server and MVC architectures were the alternatives that were collectively examined. The dependencies and associations in our system are not well represented by the client-server architecture. Additionally, we had a really hard time understanding how to integrate the client-server architecture into our solution. The layered architecture is simpler to understand in terms of difficulty. We choose to employ this architecture for this reason.

The other option under consideration was the MVC, as previously mentioned. Despite being intended for web-based applications, the MVC paradigm did not align with the overall functionality of the system under creation. After some reading, I realized that the major purpose of designing with this architecture is to reuse existing frameworks in order to reduce code complexity. It was decided not to use it because MVC as it did not meet the standards for completing this project. The system we designed in our opinion fits well with the layered architecture.

2.3 System Architecture Diagram


Browser- Based User Interface
Ordering and Delivery Web Application
Configuration services
Application Management
Identity Management
Application Services
Online Order Placement
Inventory Management
Invoice Generation
Delivery Tracking
Customer Review
Utility Services
Search
Authentication
Review Data Storage
Req. 1: Allow customers to select goods they would like to purchase from the store
Req 2. the system shall allow Manager/Authorized Employee personnel to update, edit and view items in inventory
Req. 3.: The system shall generate an invoice that calculate the total amount for each purchased items
Req. 4.: the system shall allow customers to view the current status of their delivery and allow the client to update the delivery status.
Req. 5. The system shall allow customers to enter their feedback

2.3.1 Architectural Description

The system's key requirements are divided into three distinct "services" by the layered architecture diagram. These services are Configuration, Application, and Utility. Each of which is a crucial component of the system that needs to be created.

Configuration and its components speak to the system's architecture and precise operation. The system's fundamental components, application and identity management, will monitor and regulate how the system's interactive elements interact with users.

The majority of the system's features are visible through Application Services. These are the features that users will engage with and the ways in which the system will deliver the functions for which it was intended. These apps, which are among the developers' primary concerns, might be regarded as the most crucial elements of the system.

Additional features that will be integrated into the program to help with functionality, security, and general user interaction are referred to as utility services. When combined with the applications, these particular elements are crucial components that will improve the software for the users and offer a better, more effective system.

For each of the listed key functions, there are other criteria in addition to the breakdown. demonstrating the requirements that must be completed for them to be deemed successfully incorporated into the system.

2.3.2 Architecture Justification

The data pertaining to the stock items will be kept in a database within the Data Management and Security layer. This will allow proprietors to easily and safely access the data whenever updates are required. The layered architecture most accurately depicts the process of authentication that the owners will use to gain access to specific data.

- Maintenance and testing will be easily carried out in this layer.
- Roles are easier and more effectively developed by using this architecture.
- Failure of this system will have a smaller chance of happening when compared to the client server model.
- Authentication and Authorization fits perfectly for this layered model with regards to how we plan to organize data.
- Storage and handling of information is also well accounted for in this architectural model.

3.0 Architecture Decomposition

3.1 Component Decomposition – Modules

Requirement ID	Architecture Component	Module Name	Description
	Nardos One Stop Shop Delivery User Interface	Nardos One Stop Shop UI	This will provide a minimal text-based interface that will allow system setup, to create the initial system users and modify administrative privileges
		Admin View	This will provide a User interface that of which only a person with administrator privileges can see
		Customer View	This will provide a User interface that the customers and regular viewers can see

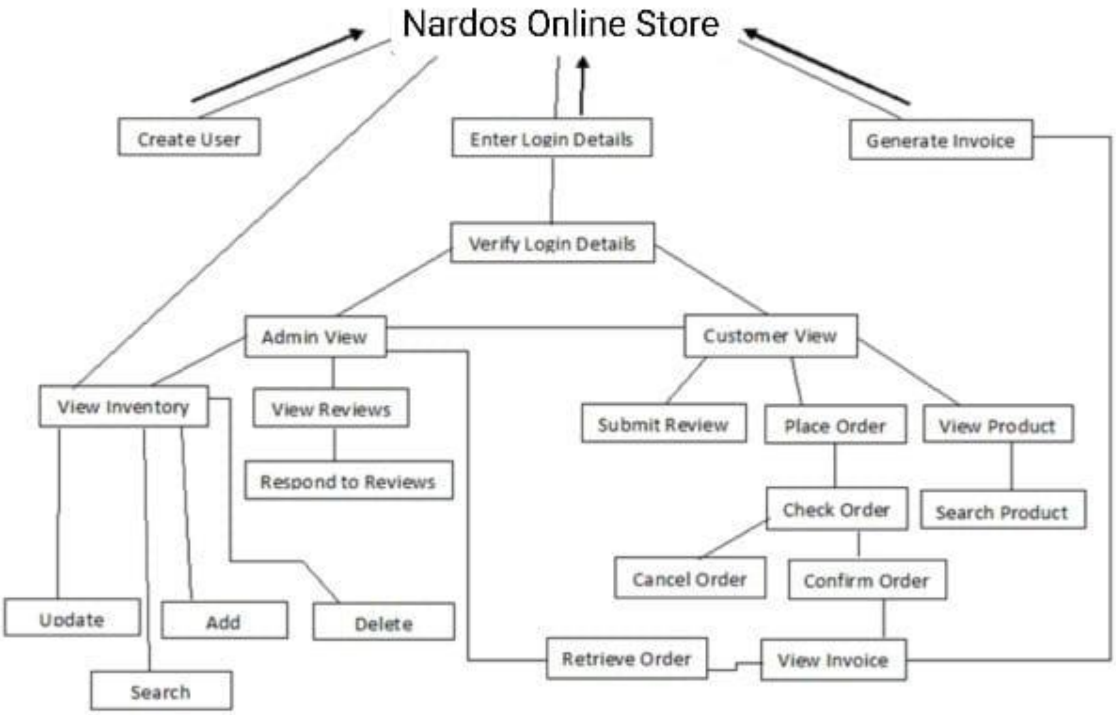
	Authorization and Authentication	Create user Login Detail Verify Log in Detail	Provides authority flags and revocation to different levels of users Interact with components seeking to verify a user’s identity
Req 1	Online Order Placement	View Products Search Products	this will provide an interface to both the potential customers and admin where they can view and interact with the products listed This will provide an interface to potential customers to allow them to be able to search for a specific product

		Place Order Check Order Cancel Order	this will allow the customer to interact with the system to add an item to the online stores cart for purchasing this will prompt the user to check the order details before finalizing their purchase this will allow the customers to cancel any order before it has been processed and delivered
Req 2	Inventory Management	View inventory	this interacts with the database server to accurately display information about the inventory an and provides an interface that holds components which can be used to modify the contents of the inventory database

		Update inventory Search Inventory Delete inventory	this interacts with the database server to alter the contents of an existing item stored in the database server This interacts with the database server to allow admin to search the Inventory for existing items this interacts with the database server to erase an existing item stored in the database server this interacts with the database server to add an item not stored in the database server

		Add inventory	
Req 3	Invoice generation	Generate invoices	This will create a digital invoice displaying the total costs for the products purchase and any other relevant information regarding the purchase
Req 4	Delivery Tracking	Track Delivery	Products will be tracked and the status of products will be updated based on delivery processes
Req 5	Customer Review	View Review Respond to Review	This module interacts with the database servers and displays the reviews of customers. The module has two separate views. a view by the customers which show reviews other customers have made and the admins response. the admin will be able to see the reviews of all customers and their responses. This also allow customer to see their review.

3.2 Structural Design – <Structure Chart >



3.2.1 Design Notes

Create User

- This module allows for account creation

Enter and Verify login details

- Both these modules authenticate and authorize admin users and customer users.

Retrieve Order

- Allows the admin to receive order information that a customer placed.

View and Respond to Reviews

- Allows the admin to communicate with customer reviews based on product and general company information.

Customer View- The customer will be able to access only these function within the system.

Customer Order- The customer can place their order

- Check order - The customer will be able to check their order
- Confirm / cancel order- The customer should be able to confirm and cancel their order within a specific time frame
- view product - customers will be able to view the products there prices and quantity
- search product- customers will be able to search for a specific product
- View invoice- The customer will be able to view their generated invoice after confirming their order.

Invoice *Generation*

- The system will calculate the total amount for each item purchased and the customer will be able to see their invoice.

Software Requirements Specification

for

Nardo's Delivery APP

Version 1.0

Prepared by

Adrian Green	620107205	greenadrian123@gmail.com
Marton Clarke	620148076	martonclarke@gmail.com
Laren Parkes	620155536	larenparkes@gmail.com

Course Instructor:	
Course:	COMP2140 – Introduction to Software Engineering
Studio Facilitator:	Aldane Milton
Date:	November 04, 2023

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
First Draft Version 1	, Adrian Green, Marton Clarke, Laren Parkes	This is the first version. It contains all the primary information that the development team has collected.	25/10/23
Final Version	Adrian Green, Marton Clarke, Laren Parkes	This is the Final version. It contains all the Necessary information relating to the project	04/11/23

1 Overall Description

1.1 Product Context and Need

Nardos delivery app is a app that will allow persons on campus to order from Nardos at anytime. Nardos delivery app was made to combat the long lines and wait times that usually happens at Nardos by allowing students to order online they would not need to visit the location reducing the number of persons in the line at one time.

1.2 Product Functionality

- The system shall allow staff to manage information (add, update or delete) of stock for delivery
- The system shall allow order placement to be made by a customer
- The system shall allow a customer to track the delivery of their order to see whether it has been prepared, left for delivery or has been delivered.
- The system shall allow customer to make a feedback after order completion
- The system shall be able to generate an invoice for the order placed by a customer.

1.3 Stakeholders and Users Characteristics

- Manage/Owner (Most Important) - This stakeholder is responsible for providing all requirements for the system. They are concerned with the software's ability to make changes/updates through the system.
- Employees - These stakeholders are the persons whom a section of the software will be made for. They will be on the receiving end of the software, interacting with the information obtained.
- Customer - These are the persons that will make the order placement.
- Delivery personnel - These stakeholders are responsible for providing delivery status information that will be saved in the system.
- Development Team - These are the persons that will be developing the software. They are responsible for the functionality of the system on a whole.

1.4 Operating Environment

The operating environment of this web-based system is primarily for the manager/owner of Nardo’s . The software will operate on an online virtual environment that should be able to function and conform to any modern device with browsing capabilities. It should be able to display a web page on operating systems of persons on windows, android, iOS, Linux and macOS. The platform that will be accessing the software must be able to run a website using HTML 5, CSS and JavaScript, with a minimum display size of 5.4-inch. With regards to RAM (Random Access Memory) and free-ROM (Read Only Memory), the minimum required to operate the software comfortably is 2Gb and 1Gb respectively. The online system will be connected to a local inventory keep.

1.5 Design and Implementation Constraints

- Competency with web-design based languages - This constraint will affect the overall development of the software, as the competency of each developer varies which infers that there are aspects that some might struggle with when developing the software.
- Time to develop - This constraint will affect the overall development of the software as adequate time is needed by us (the development team) to successfully design and implement the specifications laid out by the owner and also ensure that good quality and proprietor’s needs are met.
- Network Connectivity - This constraint refers to the network connection needed in order for the system to be accessible. Since it is a web based system, it is needed to access the stock list and place an order so the information is sent to the manager.
- Operating System - This constraint speaks to the operating systems required to successfully have this software executed on. Windows 7, MacOS 10, Linux 10 or later. Previous version operating systems will not be able to run the software. Additionally, a secure internet browser such as Google Chrome, Firefox or Internet Explorer is needed.

1.6 Assumptions and Dependencies

- We assume that a significant number of users will be active on the site at once.

- We assume that over the course of its use, multiple different personnel users will use the system.
- We assume that the manager and the staff will have some level of preliminary computer experience.
- We assume that the company will have at least 1 computer for the manager and for the delivery staff.
- We assume that the company computer will have the required processing power required to run the web application.
- We assume that the company computer will have the required Operating System to run applications.
- We assume that the company has an internet connection with at least a speed of 8 mbps.

2 Specific Requirements

2.1 External Interface Requirements

2.1.1 Hardware Interfaces

The users shall be able to interact with this software via:

- Mouse – Options shall be displayed via the screen where the user can click on options desired click on option to interact with the software
- Keyboard- The keyboard is used to input data which is stored and processed by the software
- Internet enabled devices such as(Desktop, laptop, smartphone etc.) that support html5 upwards
- A good ethernet or Wi-Fi internet connection

2.1.2 Software Interface

The system will communicate with any browser that supports html5 and above and will run on any system that supports (windows 7 or later), the system will be able to request allocations for data in-order to store data given into a database, which can then be viewed by the user. Database tools and drivers will allow the maintenance of the database in real time and provide consistency.

2.1.3 Communications Interfaces

The system shall run the HTTP protocol to display and interact with the system through a browser, The system shall use the TCP protocol to enable the safe delivery of data between the server and the users via a TCP socket.

2.2 Functional Requirements

Requirement ID#: 1 *Online Order Placement*
Use Case: *Order Placement*
Rationale: *Customers will be able to place an order for the goods they desire*
User Requirement: *Allow customers to select goods they would like to purchase from the store*
***System Requirements*:**
1. The system shall allow customer to view items
2. The user shall be allowed to select items
3. The system shall allow customer to confirm location of delivery
4. The user shall be allowed to cancel order before a certain time
Acceptance Criteria:
1. A confirmation message must be sent to the customer with a unique order id 100% of the time
Relates to/Dependencies: *Inventory Management*
Priority: *High*
Team Owner: *Marton Clarke*

Requirement ID#: 2 *Inventory Management*
Use Case: *Manage Inventory*
Rationale: ***:*** *Manager/Authorized Employee shall be able to access inventory to allow for more convenient inventory management.*
User Requirement: *the system shall allow Manager/Authorized Employee personnel to update, edit and view items in inventory.*
***System Requirements*:**
1. The system shall allow users to be Verified based on their roles
2. The system shall allow Manager/Authorized Employee personnel to update, edit and view items in the inventory.
3. The system shall be allowed to generate reorders
Acceptance Criteria:
1.The User is able to save and update the items in inventory 100% of the time.
Relates to/Dependencies: *None*
Priority: *High*
Team Owner: *Adrian Green*

Requirement ID#: 3 Invoice *Generation*
Use Case: *Generate invoices*
Rationale: The system shall calculate the total price of the item of each billing order

User Requirement: The system shall generate an invoice that calculate the total amount for each purchased items
***System Requirements*:**
1. The system shall generate the total amount for each customer's bill
2. The system shall display the name and address of the customer, cost per unit, quantity
3. The system shall assign bill to the correct unique customer order number
Acceptance Criteria:
*1.*The system shall generate an invoice with the correct calculated total100% of the time
Relates to/Dependencies: *Online Order Placement*
Priority: *High*
Team Owner: *Adrian Green*

Requirement ID: 4 *Delivery Tracking*
Use Case: *Track Delivery*
Rationale: *Owners also want customers to know that their delivery orders are being processed and that they will receive them in a timely manner.*
User Requirement: *the system shall allow customers to view the current status of their delivery and allow the client to update the delivery status.*
***System Requirements*:**
1. The system shall allow employee to update status on customer order
2. The system shall display updated order status to customer
3. The system will generate the date of the delivery
4. The system shall allow the delivery personnel to update the status of package delivery
Acceptance Criteria:
1.The system should be able to display the correct order status to customers 100% of the time.
Relates to/Dependencies: *Online Order Placement*
Priority: *High*
Team Owner: Marton Clarke

Requirement ID: 5 *Customer Review*
Use Case: *User Feedback*
Rationale: *User feedback is an important tool the Owner wishes to use to improve his business.*
User Requirement: *The system shall allow customers to enter their feedback*
***System Requirements*:**

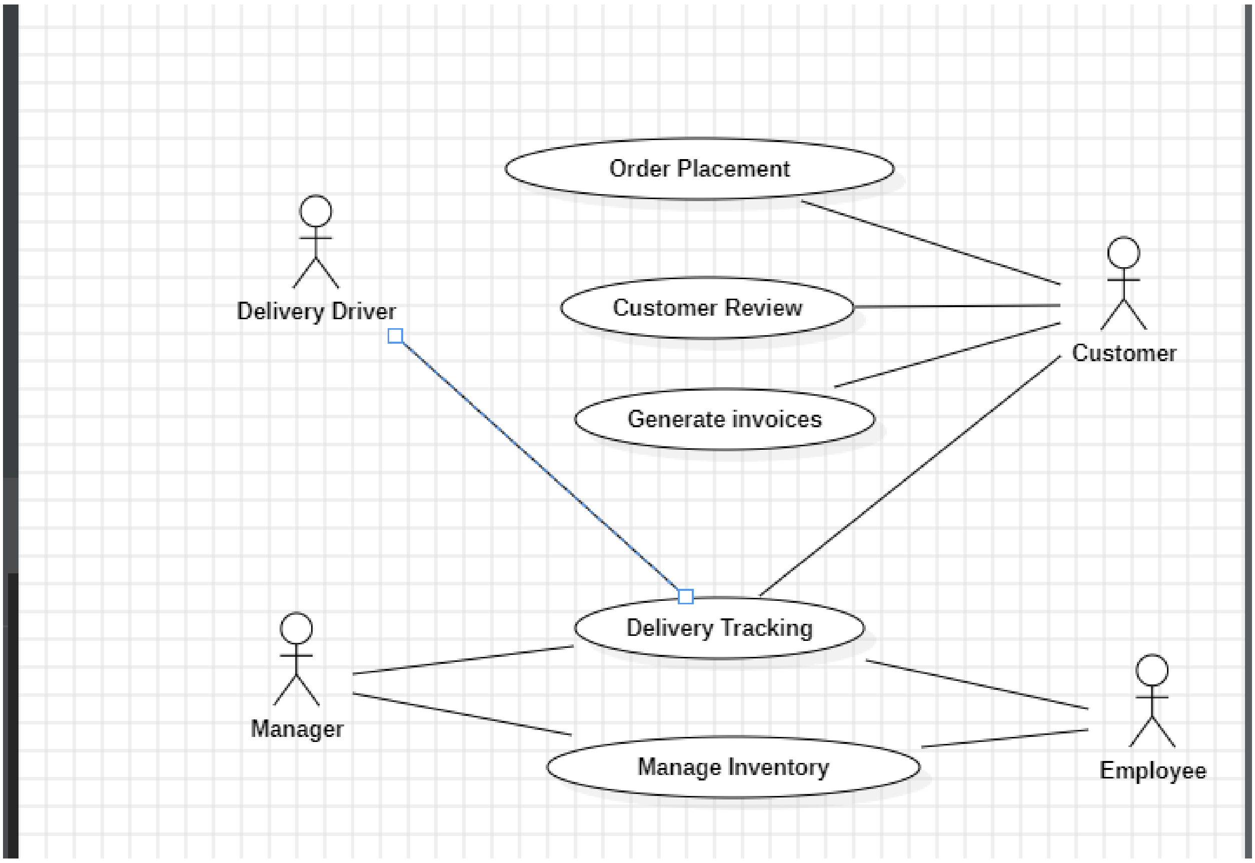
1. *The system shall allow users to enter their feedback*
2. *The system shall allow the user to validate their input before submission.*
3. *The system should store the user data by a reliable means*

Acceptance Criteria: *The system must be able to store user feedback 100% of the time*
Relates to/Dependencies: *Order Placement*
Priority: *Medium*
Team Owner: Laren Parkes

2.3 Behavioural Requirements

2.3.1 Use Case View

Use case Diagram



Use Case Narratives

Name	*Order Placement*
Description	*Allows for order placement by the customer based on the availability of the items from inventory*
Pre condition	*Item has to be currently in stock*

Basic Flow	1. User will browse for specific goods 2. Add item to cart 3. Enter required delivery information 4. Place order
Post condition	Order will be successfully placed.

Name	Manage Inventory
Description	Shall allow Manager/Authorized Employee to modify inventory
Pre condition	User has to be an Manager/Authorized Employee
Basic flow	1. Manager/Authorized Employee clicks on manage inventory feature 2. Manager/Authorized Employee add/delete or modify existing items 3. Manager/Authorized Employee saves the changes made to the inventory
Post condition	Changes successfully saved and inventory updated.

Name	Generate Invoices
Description	This will allow System to generate invoices for transactions that were done.
Pre condition	User has to be the Customer
Basic Flow	1. Invoices will be generated and displayed when an order is submitted
Post condition	Invoice with all goods bought will be generated.

Use Case Name	Track Delivery
Description	Customers should be able to see the status of the order. Employees and delivery personnel must update the status of order to reflect progress
Precondition	User must have place a valid order
Basic Flow	Employees will update the status of the order by confirming after all the items have been selected Customers will get display showing the status of their order Delivery personnel will update the status of the delivery.
Post-conditions	The system should allow customers to track the progress of their order

Use Case Name	User Feedback
Description	Providing feedback for each on the service received and the performance of delivery to make the process more effective working in meeting customer needs.
Precondition	They have an account on file.
Basic Flow	1. User enters the response on what they wish to save as feedback. 2. Prompt the user to validate the data entered.
Post-conditions	Save the comment/review or feedback to the system.

3 Other Non-functional Requirements

3.1 Performance Requirements

Order Placement Response Time: The system shall ensure that the response time for order placement, including the selection of items and confirmation of delivery location, does not exceed 10 seconds for 95% of the orders under normal load conditions. This ensures that customers can quickly and efficiently place their orders.

Inventory Update Performance: The system shall allow Managers/Authorized Employees to update inventory, including adding, updating, or deleting items, within 5 seconds under normal load conditions. This ensures that inventory management tasks are efficient and timely.

Delivery Tracking Updates: The system shall update the delivery status in real-time and ensure that the status is visible to customers within 30 seconds of any status change, such as "prepared," "left for delivery," or "delivered." This ensures that customers can track the progress of their orders effectively.

Feedback Submission: The sys 3.1 Performance Requirements:

Order Placement Response Time: The system shall ensure that the response time for order placement, including the selection of items and confirmation of delivery location, does not exceed 10 seconds for 95% of the orders under normal load conditions. This ensures that customers can quickly and efficiently place their orders.

Inventory Update Performance: The system shall allow Managers/Authorized Employees to update inventory, including adding, updating, or deleting items, within 5 seconds under normal load conditions. This ensures that inventory management tasks are efficient and timely.

Delivery Tracking Updates: The system shall update the delivery status in real-time and ensure that the status is visible to customers within 30 seconds of any status change, such as "prepared," "left for delivery," or "delivered." This ensures that customers can track the progress of their orders effectively.

Feedback Submission: The system shall allow customers to submit feedback within 15 seconds after the completion of their order. This ensures that feedback is captured promptly while the customer's experience is still fresh in their memory.

Invoice Generation Time: The system shall generate invoices for completed orders within 5 seconds of order completion. This ensures that customers receive accurate invoices promptly.

Database Query Response Time: All database queries for item details, customer information, and order history shall return results within 2 seconds under normal load conditions. This ensures that users can access necessary data quickly.

System shall allow customers to submit feedback within 15 seconds after the completion of their order. This ensures that feedback is captured promptly while the customer's experience is still fresh in their memory.
Invoice Generation Time: The system shall generate invoices for completed orders within 5 seconds of order completion. This ensures that customers receive accurate invoices promptly.
Database Query Response Time: All database queries for item details, customer information, and order history shall return results within 2 seconds under normal load conditions. This ensures that users can access necessary data quickly.

3.2 Safety and Security Requirements

Safety Requirements:

1. Customer Data Protection: The system must ensure that customer data, including personal information and payment details, is securely stored and encrypted to protect against unauthorized access and data breaches.
2. Inventory Management Safety: The system must allow only authorized personnel (Managers/Authorized Employees) to update and manage inventory to prevent unintentional changes or errors in stock management.

Security Requirements:

1. User Authentication: The system must implement a robust user authentication mechanism, including strong password policies, to ensure that only authorized personnel can access and modify inventory data. It should also enforce password policies, such as minimum length and complexity requirements.
2. Access Control: The system should provide role-based access control to limit access to specific functionalities based on user roles. For example, delivery personnel should only have access to delivery status updates and not inventory management.
3. Data Encryption: Sensitive data, including customer payment information and user credentials, should be encrypted during transmission and storage. Industry-standard encryption protocols should be used to ensure data security.
4. Protection Against SQL Injection: The system must implement security measures to prevent SQL injection attacks by validating and sanitizing user inputs to protect the database from malicious queries.

5. Logging and Monitoring: Implement a robust logging and monitoring system to track and detect any suspicious or unauthorized activities within the application. Log files should be protected and accessible only to authorized personnel.

3.3 Software Quality Attributes:

1. Maintainability: The software shall be maintainable through the use of modular and well-documented code. All code changes and updates should be easy to implement, and new features should not disrupt existing functionality.
2. Availability: The system must ensure high availability, with downtime not exceeding 99.9% annually. This means the system should be operational and accessible to users almost all the time, minimizing service interruptions.
3. Usability: The software should be user-friendly, with an intuitive user interface to ensure ease of use. The system should provide clear instructions and user guidance, minimizing the learning curve for both customers and employees.
4. Reliability: The system should be reliable and responsive, ensuring that customer orders are accurately processed and delivered. It should not crash or experience critical failures during normal operation.
5. Scalability: The system should be designed to handle an increasing number of users and orders as the business grows. It should scale seamlessly to accommodate a higher load without performance degradation.
6. Data Integrity: Ensure the integrity of data stored in the system by implementing data validation checks to prevent errors and inconsistencies in inventory and order data.

Other Requirements

4.1 Database Management System:

Database Management System (DBMS): The system shall implement a reliable and scalable database management system (DBMS) to manage and store inventory.

Appendix

When it came to talking about and getting the specifications for the project's software. Interviews had a key role. Two (2) interviews in total were carried out. The first was an orientation meeting when the developers asked some questions to get a sense of the client's problems. The date of the initial interview was September 27, 2023. The purpose of the second interview was to gather details about the client's desired appearance for the suggested software solutions. The team offered some suggestions on how these problems should be resolved, and then they approved them. The date of this second meeting was October 8, 2023.

Interview questions and Answers from first interview:

1. What are some issues your business is currently facing?

“Well… currently people a ask if we can deliver. That a really it. Funny thing, you know, I actually have some of then texting and calling my phone, asking me to deliver to them”

2. Do you think a software that manages orders and delivery could fix this issue you have?

“Yes.”

3. What are some security measures you would want the system to have?

“What I would want is the system to not allow someone apart from the customer or an authorized person to change their order or see the order detail”

4. What are some key features you would like in your system?

“Well, it would be interesting if you could, like see the progress of the delivery, you know? Lots of people impatient and but mi nuh want them feel forgotten, so rather than having them call my phone, it would be nice if they could just look and see it.”

5. How would you like your employees to interact with the system being created?

“Well, I would want it to be easy for them to use. Just easy in general, so when new people come in, it isn’t too hard, you know? So just easy is what I would say.”

6. Do you think this would be beneficial to your customers?

“Yes.”

7. How do you think it would be beneficial?

“It would be easy. For them and me. I mean just being able to sit at home and do this would be fairly easy right? So I think it would be better for them, like how you can just do it like that.”

Test Plan

Test Plan

We tested the app to guarantee its functionality, user experience and that it performs how it should. It ensures that all features, from ordering and payment processing to user interactions, operate seamlessly. Additionally, thorough testing we aim to evaluate the app's ease of use, identifying and rectifying any user interface issues that could impede navigation or confuse users. This process also assesses the app's performance under varying conditions and across different devices to ensure smooth operation without crashes or slowdowns. Furthermore, during testing, we collect valuable feedback for continuous improvement and development, ensuring that when the code is implemented, it runs smoothly, creating a reliable and trustworthy platform for our food delivery service.

Scope of Testing:

Functionality Testing: Checking all features and functionalities such as account creation, menu browsing, order placement, payment processing and user-feedback mechanisms to ensure we meet the program requirements for the app.

Usability Testing: Evaluating the app's user interface, navigation flow, accessibility, and overall user experience to ensure that customers can easily maneuver the site and prevent confusion.

Performance Testing: Testing the app's responsiveness, load times, and overall performance to ensure it operates optimally before we implement it as a solution.

Objectives of Testing:

To Identify and resolve bugs, software defects, errors, and inconsistencies in the app's functionality, usability, and performance.

To ensure a reliable and Stable app that operates without crashes, freezes, or unexpected behaviors under normal scenarios.

To enhance and improve the user experience by addressing interface issues, ensuring intuitive navigation to make the website clearer and for easier accessibility.

Testing Approach

We used more of a black-box testing approach, examining the systems functionality without going into the code unless necessary to correct an issue, this was done to minimize changes due to time constraints, this approach also focusses on the customers perspective which would show the more important flaws of the system as the customers would be the main users of the app and hence need the greatest focus of efforts. It focuses on testing the system as a whole to ensure it provides the functionalities.

Test Case	Test Data	Associated Requirement	Expected Result	Actual Result	Pass/Fail	Comment
Case 1: check login to system Username and Password correct	Username Password	Req.0	Access granted	Access granted	Pass	
Case 2: check login to system Username or Password incorrect	Username Oassword	Req.0	Access failed	Access failed	Pass	Nothing to display that password is incorrect
Case 3: Order Placement	Multiple items Order Placed	Req1	All Orders places and records in the database	All Orders placed and record in the database	Pass	
Case 4: Inventory Management Adding inventory	Add Inventory.	Req 2	Add an Inventory	Added Inventory	Pass	
Case 5: Inventory Management Deleting Inventory	Delete Inventory	Req 2	Deletes inventory	Deleted inventory.	Pass	
Case 6: Invoice Generation	Generate Invoice	Req 3	Creates Invoice	Created Invoice	Pass	Invoice created along with order
Case 7: Delivery Tracking	Provides order status updates	Req 4	Driver and admin Updates Order Status	Admin can Updates Order Status	Failed	Delivery driver unable to update status
Case 8: Customer Review	View and respond to reviews	Req 5	View and respond to reviews	Admin can view and respond to reviews	Pass	Tickets allow reviews and other types of queries
Case 9: Order Details	Customer Address	Req 1	Check to see if address is valid	No checks	Failed	Failed to implement due to Time