**Pizza Billing System Project**

**System Requirements Document**

**Submitted by**

**Group-1**

**Team:**

**Anurag Gajula**

**Komal Raj Nadella Venkata**

**Goutham Madarapu**

**Raj Kumar Sriramula**

**Yalavarthi, Sai Satwick**

**Under the Mentorship of**

**Prof: Shankari Ghosh. P**

**Submission Date: 03/26/2023**

***At***

*College of Business Administration*

*Business Information Systems Department*

***Central Michigan University***

***2023***

**Contents**

[**1.Introduction** 1](#_Toc130765379)

[**1.1 Purpose of Document** 1](#_Toc130765380)

[**1.2 Background** 1](#_Toc130765381)

[**2.Project Scope** 3](#_Toc130765382)

[**2.1 In-Scope Functionality:** 3](#_Toc130765383)

[**2.2 Out of Scope Functionality:** 3](#_Toc130765384)

[**3.System Purpose** 4](#_Toc130765385)

[**3.1 Users** 4](#_Toc130765386)

[**3.2 Responsibilities** 4](#_Toc130765387)

[**3.3 Need** 5](#_Toc130765388)

[**4.Business Requirements** 6](#_Toc130765389)

[**4.1 Functional requirements** 6](#_Toc130765390)

[**4.2 Non-Functional requirements** 6](#_Toc130765391)

[**4.3 System data requirements** 7](#_Toc130765392)

[**5.System Perspective** 9](#_Toc130765393)

[**5.1 Assumptions** 9](#_Toc130765394)

[**5.2 Constraints** 9](#_Toc130765395)

[**5.3 Risks** 9](#_Toc130765396)

[**6.Context Level Diagram** 10](#_Toc130765397)

[**Figure 1.1** 10](#_Toc130765398)

[**6.1 System Externals** 10](#_Toc130765399)

[**7.Use-Case Descriptions** 12](#_Toc130765400)

[**7.0 Use Case Diagram** 12](#_Toc130765401)

[**Figure 1.2** 12](#_Toc130765402)

[**7.1 User Registration** 13](#_Toc130765403)

[**Figure 1.3** 13](#_Toc130765404)

[**7.2 User-Login** 14](#_Toc130765405)

[**Figure 1.4** 14](#_Toc130765406)

[**7.3 Customer – Place order** 16](#_Toc130765407)

[**Figure 1.5** 16](#_Toc130765408)

[**7.4 Employee- Place Customer Order** 18](#_Toc130765409)

[**Figure 1.6** 18](#_Toc130765410)

[**7.5 Generate and Analyze Reports** 20](#_Toc130765411)

[**Figure 1.7** 20](#_Toc130765412)

[**8. Testing** 22](#_Toc130765413)

[**8.1Testing Strategies:** 22](#_Toc130765414)

[**8.2 Test Cases:** 22](#_Toc130765415)

[**8.3 Expected Results:** 22](#_Toc130765416)

[**9. Sign-Off** 23](#_Toc130765417)

[**10.Annexure** 24](#_Toc130765418)

[**10.1 ERD** 24](#_Toc130765419)

[**Figure 1.8** 24](#_Toc130765420)

[**10.1.1 ERD Business Rules** 24](#_Toc130765421)

[**Figure 1.9** 25](#_Toc130765422)

[**10.2 Level-1 DFD** 26](#_Toc130765423)

[**10.2.1 Level-1 Assumptions** 26](#_Toc130765424)

[**10.3 Gantt Chart** 27](#_Toc130765425)

[**Figure 1.10** 27](#_Toc130765426)

[**\*\*Acceptance Criteria** 28](#_Toc130765427)

# **1.Introduction**

This document contains an outline of the requirements that need to be met for the Pizza Billing System Project. In order to guarantee the success of the project, the purpose of this document is to offer a comprehensive understanding of the requirements, functionality, and constraints associated with the system.

## **1.1 Purpose of Document**

This document presents an overview of the requirements for a new system interface for pizza restaurants. Customers will have a better and more personalized ordering experience because of the new system, which will improve the current in-store ordering process and make it easier. This document outlines the scope, objectives, and goals of the new system, in addition to the functional requirements that have been framed with use cases and interaction diagrams. This document's purpose is to serve as a guide for the creation of an effective and user-friendly design for the new system as well as its implementation.

## **1.2 Background**

A pizza restaurant is currently working on the creation of a new in-store sales system to replace the one they have been using. The restaurant, which serves a selection of pizzas, is available for dining customers as well as take-out customers. The restaurant has recognized that there is room for improvement in the way that customers place orders as well as in the overall operations of the business. The new system will improve both the restaurant's productivity and the quality of the experience it provides for its customers.

The diverse customer base at the pizza restaurant is comprised of individuals as well as groups who place orders for either takeout or dining in. Because there has been a rise in demand for the restaurant's services, there is now a pressing requirement for a billing system that is both dependable and easy to understand and use.

The current system has a number of flaws, the most notable of which are its inability to keep track of orders that customers have placed and its inability to generate accurate billing reports, the absence of a customer login system, and the tedious and error-prone nature of the manual process that is currently used to enter customer and order data. All of these problems can be corrected by updating the system.

The Pizza Billing System project was initiated with the purpose of enhancing the ordering and billing procedures in order to address these issues. The new system will allow customers to place orders online, track their orders, and view their billing history. Additionally, it will provide a user-friendly interface for restaurant staff to manage orders and generate reports. Customers will be able to view their billing history.

The restaurant will be able to provide a better experience for customers and increase the efficiency of its operations after the implementation of the new Pizza Billing System. This, in turn, will lead to increased customer satisfaction and ultimately the business.

**Problems with the current system:**

• Customers may face long wait times during peak hours

• The ordering process may not be user-friendly and cause confusion

• The restaurant staff may struggle with keeping track of orders and managing the overall operations

**Goals of the new system:**

• Provide customers with a seamless and efficient ordering experience

• Streamline the operations of the restaurant and make it easier for staff to manage orders

• Offer customers a user-friendly interface to place their orders

• Reduce wait times for customers during peak hours

• Ensure accuracy in the ordering process and improve the overall customer experience.

# **2.Project Scope**

The project scope includes the development of a software application to manage customer orders, billing, and reporting. Customers will be able to place orders for pick-up or dine-in through the system.

## **2.1 In-Scope Functionality:**

The following features and functionality will be included in the scope of this project:

* A customer login system to allow customers to place orders and manage their account information.
* A user login system to allow staff to manage customer orders, billing, and reporting.
* A database to store customer information, orders, and billing data
* A graphical user interface (GUI) using Python to provide a user-friendly interface for customers and staff.

## **2.2 Out of Scope Functionality:**

The following items are outside of the scope of this project:

* Development of mobile applications for the system.
* Advertising and marketing of the pizza menu and services.
* Physical pizza delivery, which will not be part of this project.

# **3.System Purpose**

## **3.1 Users**

The system will be used by two categories of people: customers and employees. The system can be used by staff to manage customer orders, billing, and reporting, and by customers to place orders and manage their account information.

**Customers:** The new pizza ordering system will make it easier for customers to find their desired menu items, place orders, and track their orders. Customers will be able to choose whether to place an order directly through the system or by dining in at the restaurant.

**Restaurant Staff & Manager:**

The new system will give the restaurant employees real-time information on customer orders, improving the accuracy of order tracking and decreasing the likelihood of human mistakes. Employees at the restaurant will also be able to view customer profiles, which should improve service. The marketing team will have access to information on customer ordering habits and the most popular items on the menu, allowing them to make more informed judgments about product offerings and advertising campaigns.

## **3.2 Responsibilities**

The primary responsibilities of the pizza billing system are as follows:

• Provide customers with access to accurate and up-to-date menu information.

• Allow customers to place orders and view their order history through the system.

• Allow customers to make payment for their orders online.

• Generate reports for store managers and owners with insights into the business performance, including sales trends, customer behavior, and order fulfillment efficiency.

• Provide store managers and owners with access to the database tables, including customers, orders, and order history, for analysis and reporting purposes.

Other desired features of the pizza billing system:

• A user-friendly and consistent interface throughout the system.

• Full-text search functionality for easier navigation and product discovery.

• Online help for customers to navigate the system and place orders.

• Password protection for sensitive information, such as customer details and payment information.

• Option to receive order confirmation and updates via email or SMS.

The system will not be responsible for delivery services, or physical inventory control. The system is designed to support the marketing and billing aspects of the business and provide valuable insights to store managers and owners to improve the overall customer experience.

## **3.3 Need**

The restaurant needs an up-to-date system that is also very effective for managing the orders that customers place and the bills that they get. The current method is outdated and inefficient, which leads to delays and errors in the process. The Pizza Billing System Project will deliver a solution that is user-friendly and efficient, which will improve the experience that customers have and boost the efficiency with which staff work.

# **4.Business Requirements**

## **4.1 Functional requirements**

**Customer login:** Customers should be able to create an account and log in to the system to place orders and manage their account information.

**Product information:** The system UI should display accurate and up-to-date information about all available products and prices.

**Order placement:** Customers should be able to select products and place orders through the new UI. The system should generate an order confirmation and provide customers with invoices.

**Payment processing:** The system should be integrated with a secure payment gateway to process payments. Customers should be able to view and manage their payment information.

**Order tracking:** Customers should be able to view the status of their current and past orders, including order history.

**Reports and analysis:** Store managers should have access to a range of reports and analysis tools, including sales reports, customer behavior analysis.

## **4.2 Non-Functional requirements**

**Reliability:** The system must be dependable and available 24/7, with minimal downtime. The system must be able to handle multiple concurrent users and maintain consistency of data. Error messages should be clear and concise to help users resolve issues quickly. The system should also have robust backup and recovery mechanisms in place to ensure data protection.

**Usability:** The system must be easy to use for customers, sales agents, and product owners. The interface should be intuitive and user-friendly, with clear navigation and accessible features. The system should also support various devices and screen sizes for maximum accessibility.

**Performance:** The system must be able to respond quickly to user requests, even during peak usage periods. The system should load quickly, with minimal delay in processing transactions. The system must also be able to handle large volumes of data efficiently without slowing down or crashing.

**Security:** The system must protect sensitive information, such as passwords and customer data, from unauthorized access and theft. The system should also comply with industry standards for data protection and privacy. The system should have robust authentication and authorization mechanisms in place to ensure only authorized users can access restricted areas of the system.

**Supportability:** The system must be easy to maintain, with clear documentation and a well-defined process for updates and upgrades. The system should also have a help desk or customer support system in place to assist users with technical issues and questions. The system should be designed with scalability in mind, to accommodate future growth and changes in user needs.

## **4.3 System data requirements**

**Customer Information** : This section outlines the data requirements for customers using the Pizza ordering system. The system will collect the necessary data during the ordering process and store it in the customer table of the Pizza database. Additionally, the system will store encrypted login information for customers, which will also be saved in the customer table. The following information will be collected and stored:

* Customer\_ID
* First\_Name
* Middle\_Name
* Last Name
* Address (City,State,ZipCode)
* Phone number
* Email\_Address
* Username
* Password

**Order :** This section will outline the information that needs to be collected about each order placed on the system. This data will be stored in the order table in the pizza database. The order information that will be stored includes:

* Order ID
* Order\_ratings
* Customer\_ID
* Employee\_ID
* Order\_Date
* Order\_Total
* Order\_Qty

**Employee Information:** This section will provide details on the data that needs to be gathered about the employees, including login information that will be stored in the Pizza database . The data will be kept in the Employee table. The information that will be gathered about the employees includes:

* Employee\_ID
* First\_Name
* Middle\_Name
* Last\_Name
* Email\_Address
* Contact number.
* Address (City,State,ZipCode)
* Username
* Password

**Item Information**: The item information section specifies the details that must be collected about the menu items that are available for order. This data will be stored in the "menu item" table in the Pizza database. The item information that needs to be stored includes the following:

* Item ID
* Item Name
* Item Type (e.g., Pizza, sides, drinks)
* Item Price
* Item Size

**Payment Information**: The Payment Information section outlines the necessary data to be collected for each payment made by a customer, which will be stored in the payment table of the Pizza database. This includes the following payment details:

* Invoice Number
* Method of Payment (Cash or Card)
* Order ID
* Customer ID
* Payment Date
* Amount Paid

Ensuring the accuracy and completeness of this information is critical to proper processing and tracking of payments, and to maintaining proper records of financial transactions in the system.

**Order Status Information**: The Order Status Information section specifies the details that need to be collected about the status of each order placed, which will be stored in the Order Status table of the Pizza database. This information is critical as it will enable customers to track the status of their orders in real time.

The information to be collected includes:

* Order ID
* Order Status (e.g., Pending, Preparing, Ready)

**Order Item Information**: The Pizza ordering system will collect information about each item ordered within an order, which will be stored in the order item table in the Pizza database. This information includes:

* Order ID
* Item ID
* Item Quantity

\*Note that a visual representation of the Pizza ordering system's data requirements can be found in the [Entity Relationship Diagram (ERD)](#_7.1_ERD) located in the Annex. The ERD provides a detailed view of the relationships between the various data entities in the system and their attributes, including customers, orders, menu items, employees, and payments. The ERD serves as a critical tool for understanding the system's data structure and relationships and is a valuable reference for system developers and administrators.

# **5.System Perspective**

## **5.1 Assumptions**

* The pizza billing system will be used in a pizza restaurant or pickup service environment.
* The system will be used by employees who are trained to operate it.
* The system will be accessed through a web-based interface.
* The system will accurately calculate the cost of pizza orders and process payments.

## **5.2 Constraints**

* The system must comply with relevant regulations and laws related to financial transactions and customer data privacy.
* The system must be user-friendly and intuitive to reduce errors and improve efficiency.
* The system must be designed with security measures in place to protect customer information and financial data.
* The system must be able to handle a high volume of orders during peak hours.

## **5.3 Risks**

* Data breaches or cyber-attacks could compromise the security of customer information and financial data.
* Technical glitches or system failures could result in errors in order processing or payments, leading to customer dissatisfaction.
* Employee errors or fraud could result in inaccurate billing or theft.
* Inadequate training or support could lead to inefficiencies or mistakes in system operation.

# **6.Context Level Diagram**

The context level diagram for a pizza billing system represents the high-level view of the system and its interactions with the customers, employees, and managers.

Diagram

Description automatically generated

## **Figure 1.1**

## **6.1 System Externals**

**Customers:** In a pizza billing system , customers are the main actors who interact with the system. They perform the following actions:

* Register: The customers have the option to register with the system, which can make the ordering process easier and faster in the future.
* View the menu: Customers can browse the menu to select their desired pizza and toppings.
* Place an order: Customers can place an order for a pizza by selecting the required toppings, size, and crust.
* Make a payment: Customers can make a payment for their order using cash, credit card, or any other accepted mode of payment.
* Receive order confirmation: Customers receive an order confirmation that contains the details of their order, the price, and the estimated time of Pick-up.
* Track order status: Customers can track the status of their order and know when it is ready for pickup.

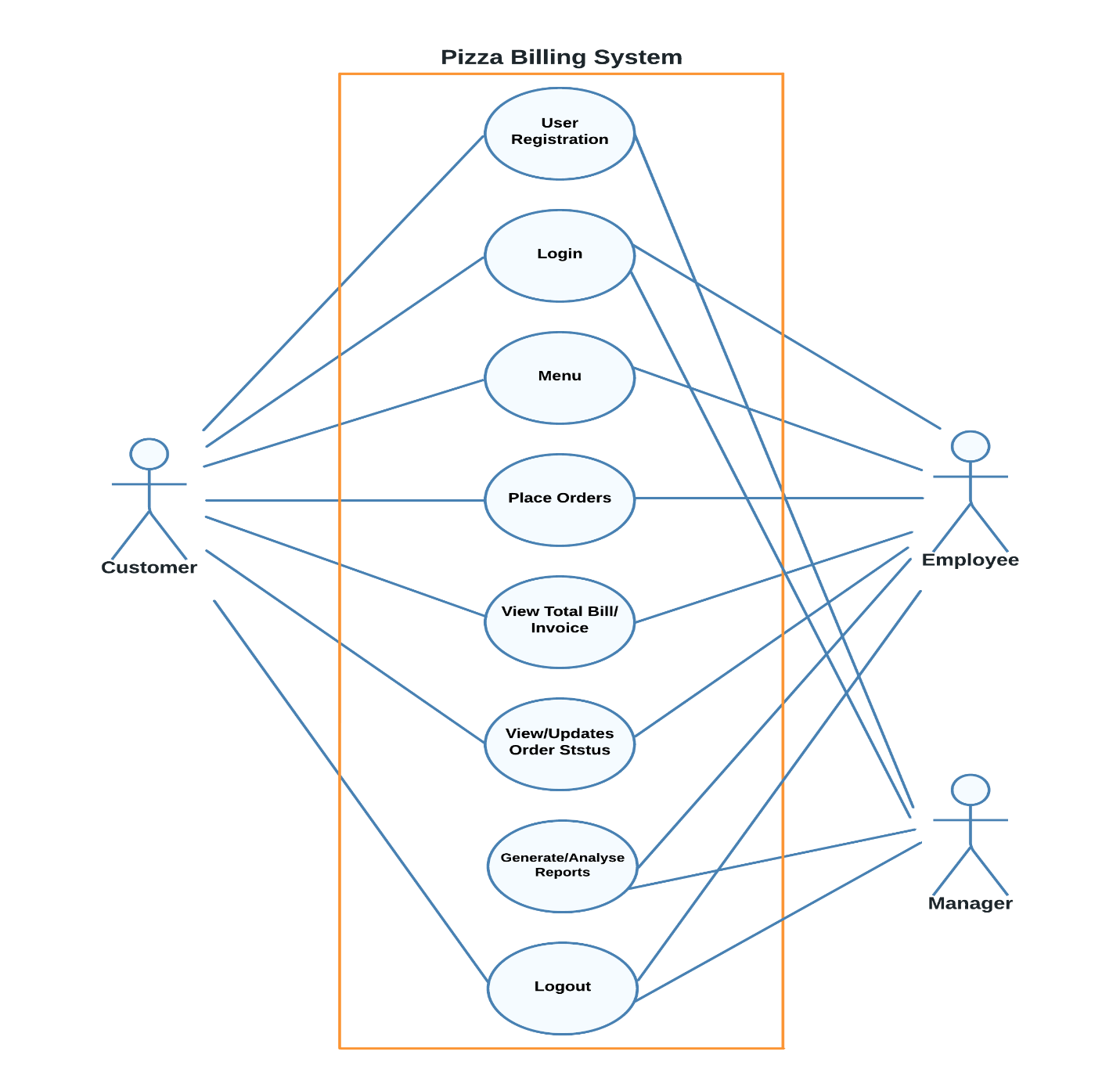
**Employee:** Employees play a crucial role in the pizza billing system as they handle customer orders. They are responsible for receiving and processing customer orders, either through the system or in-person at the store. To ensure a smooth and efficient process, employees will need to accurately enter customer information and order details into the system. Their actions directly impact the customer's experience and contribute to the overall success of the business.

**Manager:** In the new system, the manager is responsible for pulling reports and using them to make strategic business decisions. They use reports from the system to make smart decisions and keep the business running smoothly. These reports give them a look at important information like how much money the restaurant is making, like sales reports and customers information. With this information, the manager can make changes to improve the business and make sure customers are happy.

# **7.Use-Case Descriptions**

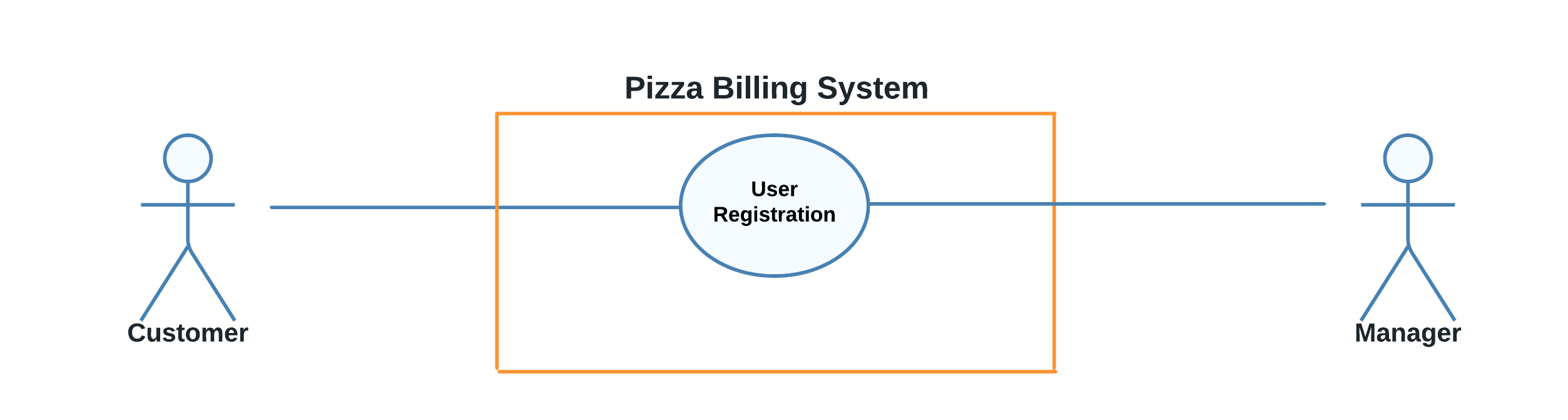
A use case description for a pizza billing system is a comprehensive and organized method of outlining the functionalities and interactions between the user and the system. It details the steps taken by the user to complete a specific task, such as ordering a pizza or managing their account information. The use case description is written in a way that is easy to understand for both technical and non-technical users, and it outlines the inputs, outputs, and conditions involved in each task. This description helps to ensure that the system functions as expected and that the user can complete tasks efficiently and effectively.

## **7.0 Use Case Diagram**



## **Figure 1.2**

## **7.1 User Registration**



## **Figure 1.3**

|  |  |
| --- | --- |
| Use Case Name: | User Registration |
| Summary: | To access personalized information, make orders or complete specialized transactions, it's necessary to create a new account by registering a unique username and password. This will allow you to securely access all the features and benefits that are tailored to customer needs. |
| Basic Flow: | 1. The registration process begins when a user is new to the system. 2. The system then requests the user's personal information, including their name, address, phone number, and email. 3. The user inputs the requested information. 4. The system will create a new username and id after the above information is entered by the user. 5. The system uses this information to determine the user's location and what level of access they will have and stores this information for future use. 6. The system will start a new login session and display a personalized welcome message, based on the user's preferences. |
| Alternative Flows: | In Step 2, if the mobile number and email-id provided by the user are identical to those of an existing user, the system will display an error message informing the user that they cannot use those details to register. The use case will then revert to Step 2 to allow the user to enter a different mobile number or email-id. |
| Extension Points: | None |
| Pre-conditions: | None |
| Post-conditions: | The system has successfully stored the user's data, which includes their specific username and password, location, and access level. The user has been logged in and the system has confirmed that the username does not already belong to another registered user. The user's data is now prepared to be used in upcoming orders and transactions. |
| Business Rules: | The user will have access according to the user type:   * Customers: Customers will be able to create accounts, log in, and place orders. They can also make payments, view their order history. * Employees: Employees can log in to view order details, modify order status updates, and manage payments. They will not have access to a customer's private data unless it's required to complete an order. * Manager: Managers will have complete access to the system, including all customer, order, and employee data. They will be able to view sales reports, make strategic business decisions based on the data gathered in the reports, and create and manage employee accounts. |

## **7.2 User-Login**

Graphical user interface

Description automatically generated

## **Figure 1.4**

|  |  |
| --- | --- |
| Use Case Name: | User Login |
| Summary: | To access personalized or restricted information, place orders, or perform other specialized transactions, a user must first log in to the system. This allows the system to determine their access level and provide the appropriate level of information and functionality. |
| Basic Flow: | 1. When a user accesses the login page, the use case begins. 2. The user is prompted to enter their username and password by the system and the user enters their login information. 3. The system compares the username and password to its database to verify them. 4. The system launches a new session for the user and shows the main menu based on their access level if the username and password are accurate. 5. The system displays an error message and sends the user back to the login page so they can try again if the username and password are entered incorrectly. 6. When a user logs out of the system or the session expires, the use case is over. |
| Alternative Flows: | If the user enters an incorrect username or password during the login process, the system will ask them to try again. The use case goes back to step 2 where the user will need to enter their correct username and password to continue. |
| Extension Points: | None |
| Pre-conditions: | The user must be Registered |
| Post-conditions: | After the user has logged into the system, they will have access to specific features and information based on their determined access level. They will be able to view previous transactions and orders for future reference. |
| Business Rules: | * To use the pizza billing system, a user must first have a registered account and use the correct username and password combination to log in. * The system will keep their login information secure to protect their privacy, and their access level will determine what information and features they can access. |

## **7.3 Customer – Place order**

Diagram

Description automatically generated

## **Figure 1.5**

|  |  |
| --- | --- |
| Use Case Name: | Customer Places his own order |
| Summary: | To access personalized or restricted information, place orders, or perform other specialized transactions, a user must first log in to the system. This allows the system to determine their access level and provide the appropriate level of information and functionality. |
| Basic Flow: | 1. If the customer is a first-time user, they log into the system or create a new account. 2. For their order, the customer selects the desired menu items and customization options. 3. The customer goes over the order details and confirms them, as well as any special instructions or delivery information. 4. The customer reviews and confirms the order's total cost and chooses a payment method. 5. The order is processed by the system, and the customer is given a confirmation number. 6. The system updates the customer's order history and issues a receipt to the customer. 7. The customer may log out of the system or place another order. |
| Alternative Flows: | * If the customer's payment method fails, the use case returns to step 4, and the system prompts the customer to enter a new payment method or to try again with the original payment method. * The customer may stop at any time before making the payment if he decides to cancel the order. |
| Extension Points: | Bill Customer |
| Pre-conditions: | A customer must have a registered account in the system, be logged into their account, have access to the "Place Order" feature, and have sufficient funds or credit available if necessary before they can place an order through the pizza billing system. Furthermore, the pizza ordering system must be operational and capable of accepting orders. |
| Post-conditions: | Following the placement of an order, the system will have stored the order details, the customer's account information, the items ordered, and the payment method. The customer will receive an order confirmation with a pickup time. The customer's account information will be updated to reflect the most recent transaction, as well as their order history and payment information. |
| Business Rules: | * Customers must have a registered account and provide valid payment information in order to place an order using the pizza billing system. * Before the order can be processed, the customer must confirm it, and their account information must be updated to reflect the order and payment. * The restaurant must be able to prepare the order in a timely manner. |

## **7.4 Employee- Place Customer Order**

Diagram

Description automatically generated

## **Figure 1.6**

|  |  |
| --- | --- |
| Use Case Name: | For a customer, an employee can order pizza. |
| Summary: | The employee places an order for a customer by entering their information and payment details, processing the order, and updating the customer's account information. The aim is to ensure accurate and efficient processing of the order. |
| Basic Flow: | 1. The employee enters the system and selects the "Place Order" option. 2. The employee enters the customer's information, which includes their name, address, phone number, and any special-order instructions. 3. The employee enters the payment information, including the method of payment, after selecting the items to be included in the order. 4. The system verifies the payment information and, if everything is in order, processes the order and adds it to the restaurant's list of pending orders. 5. The system generates an order confirmation and displays it to the employee. 6. The system updates the customer's account information to reflect the most recent transaction, including order and payment details. 7. Before submitting the order, the employee verifies that the information is correct and, if necessary, updates or corrects it. |
| Alternative Flows: | * If the payment information provided by the customer is invalid, the use case goes back to step 3 , the employee may need to contact the customer to obtain a valid payment method or to inform the customer that their order cannot be processed. * Use case is cancelled if the customer has insufficient balance. |
| Extension Points: | Bill Customer |
| Pre-conditions: | * Before an employee can place an order on behalf of a customer, they must have a valid login to the system, have access to the "Place Order" feature, and the pizza ordering system must be up and running. * Additionally, the employee must have the customer's information, including their contact and payment details, at hand. |
| Post-conditions: | * The system will have recorded the order information, including the customer's account information, the items ordered, and the payment method, after the employee successfully placed an order on behalf of a customer. * The order will be processed and added to the pizza restaurant's list of pending orders. * The system will update the customer's account information, including their order history and payment information, to reflect the recent transaction. * The employee will receive an order placement confirmation. |
| Business Rules: | * The employee has the ability to bill the customer. They may also expand the customer's access to product information. * The employee must have a valid login to the system and access the system's "Place Order" feature. * The employee must have access to the customer's information, including their contact and payment information, and the order must be processed and added to the pizza restaurant's list of pending orders. |

## **7.5 Generate and Analyze Reports**

Diagram, schematic

Description automatically generated

## **Figure 1.7**

|  |  |
| --- | --- |
| Use Case Name: | Generate and Analyze Reports |
| Summary: | The employee generates reports to analyze data from the system, and the manager analyzes these reports to make informed decisions. |
| Actors: | * Employee * Manager |
| Basic Flow: | 1. The employee logs into the system and navigates to the "Reports" section. 2. The system displays a list of available reports. 3. The employee selects the desired report and specifies the required parameters (e.g., date range, data filters, report type). 4. The system generates the report and displays it on the screen. 5. The employee analyzes the report to draw insights and make recommendations based on the data. 6. The manager logs into the system and navigates to the "Reports" section. 7. The system displays a list of available reports. 8. The manager selects the same report previously generated by the employee and reviews it. 9. The manager analyzes the report to make informed decisions and takes appropriate actions based on the data. |
| Alternative Flows: | If the employee selects an invalid report or specifies incorrect parameters, the system displays an error message and allows the employee to try again. |
| Extension Points: | The manager may request additional reports from the employee based on the insights gained from the current report. |
| Pre-conditions: | * The employee and manager have valid user accounts and login credentials for the system. * The system contains relevant data to generate the reports. |
| Post-conditions: | * The employee has generated a report based on the specified parameters. * The manager has reviewed the report and made informed decisions based on the data. * The system has recorded the generation and review of the report for future reference. |
| Business Rules: | * Only employees with the appropriate access level can generate reports. * Only managers with the appropriate access level can review and make decisions based on the reports. * Reports can only be generated based on the data available in the system. |

# **8. Testing**

A testing plan for a pizza billing system includes the following:

## **8.1Testing Strategies:**

* Black-box testing: This testing strategy involves testing the system without any knowledge of its internal workings. It is used to ensure that the system meets the requirements specified in the acceptance criteria.
* White-box testing: This testing strategy involves testing the system with knowledge of its internal workings. It is used to ensure that the system is properly integrated and functioning as expected.

## **8.2 Test Cases:**

* Unit tests: These tests focus on testing individual components or modules of the system, such as the order processing module, the payment processing module, and the receipt generation module.
* Integration tests: These tests focus on testing how different components or modules of the system interact with each other, such as how the order processing module interacts with the payment processing module.
* System tests: These tests focus on testing the system as a whole, including all of its components and modules, to ensure that it is working as expected.

## **8.3 Expected Results:**

* The system accurately calculates the cost of pizza orders, including any discounts.
* The system processes payments securely and without errors, including credit card payments and cash transactions.
* The system generates accurate and detailed receipts for customers, including the order total, payment method, and any additional notes.
* The system allows for the customization of pizza orders by size and quantity, and accurately reflects any changes in the final price.
* The system can handle a high volume of orders during peak hours without experiencing significant lag or downtime.
* The system includes a user-friendly and intuitive interface for employees, reducing the likelihood of errors and improving efficiency.
* The system complies with relevant regulations and laws related to financial transactions and customer data privacy.
* The system can generate detailed reports on sales, revenue, and customer data, allowing for effective business analysis and decision-making.
* The system provides adequate training and support for employees to ensure proper use and operation.
* The system is fully tested and free of bugs or glitches that could impact functionality.

# **9. Sign-Off**

Project Title: Pizza Billing System

Project Group: Group 1

Professor: Ghosh

We, the undersigned stakeholders of the Pizza Billing System project, confirm that we have reviewed and agree to the requirements outlined in the Requirements Document, as well as the proposed solution described in the Design Document.

We acknowledge that the Acceptance Criteria and Testing Plan have been developed to ensure that the software system meets our requirements and functions as expected.

Furthermore, we confirm that we are satisfied with the proposed solution and agree that the software system is ready for implementation.

Signed:

[Stakeholder Name] Date

[Stakeholder Name] Date

[Project Group Member] Date

[Professor] Date

# **10.Annexure**

## A screenshot of a computer Description automatically generated with medium confidence**10.1 ERD**

## **Figure 1.8**

### **10.1.1 ERD Business Rules**

The database schema includes the following tables and relationships:

* **CustomersInformation:**
  + One-to-many relationship with Orders (one customer can have many orders)
  + One-to-many relationship with Payment (one customer can make many payments)
* **EmployeeInformation:**
  + One-to-many relationship with Orders (one employee can handle many orders)
* **Orders:**
  + One-to-many relationship with Order\_Item (one order can have many items)
  + One-to-one relationship with Order\_Status (one order can have one status)
  + One-to-one relationship with Payment (one order can have one payment)
  + Many-to-one relationship with customersInformation (many orders can belong to one customer)
  + Many-to-one relationship with employeeInformation (many orders can be handled by one employee)
* **Order\_Item:**
  + Many-to-one relationship with Orders (many items can belong to one order)
  + Many-to-one relationship with Items (many order items can be the same item)
* **Order\_Status:**
  + One-to-one relationship with Orders (one status can belong to one order)
* **Payment:**
  + One-to-one relationship with Orders (one payment can belong to one order)
  + Many-to-one relationship with customersInformation (many payments can belong to one customer)
* **Items:**
  + Graphical user interface

    Description automatically generatedOne-to-many relationship with Order\_Item (one item can be ordered many times in different orders)

## **Figure 1.9**

## **10.2 Level-1 DFD**

### **10.2.1 Level-1 Assumptions**

In this DFD, we sub divide the process to know the details of how the actions are performed. In the level 0 DFD we have sub divided the single process from context level diagram into six processes.

* The Manager is responsible for updating the menu items available for customers to select upon registration in the system. The relevant details for menu items are stored in the "Items Information" datastore.
* To register as a customer, the user must complete the registration process, and the information provided is stored in the "Customer Information" data store. The registered information can be accessed by authorized employees.
* Customers can select items from the menu, which are retrieved from the "Items Information" data store. After the customer completes the payment, the payment details are stored in the "Payment Information" data store. Once the transaction is successful, the customer receives an order confirmation, which is stored in the "Order Information" data store.
* The Pizza Billing system keeps track of the items, sales, and order information to generate reports that can improve the store's business. The reports are then analyzed by the Manager for marketing purposes and to better understand customer behavior.
* The Manager create an employee account and the information is stored in “Employee Information” data store.

## **10.3 Gantt Chart**

Timeline

Description automatically generated

## **Figure 1.10**

# **\*\*Acceptance Criteria**

The acceptance criteria for a pizza billing system. The following criteria should be met to ensure the system operates at an optimal level:

* Accurate Calculation of Pizza Orders: The system should accurately calculate the cost of pizza orders, including any discounts and taxes.
* Secure and Error-free Payment Processing: The system should process payments securely and without errors, including credit card payments and cash transactions.
* Generation of Accurate and Detailed Receipts: The system should generate accurate and detailed receipts for customers, including the order total, payment method, and any additional notes.
* Customization of Pizza Orders: The system should allow for the customization of pizza orders, including crust size and quantity, and accurately reflect any changes in the final price.
* High Volume Handling: The system should be able to handle a high volume of orders during peak hours without experiencing significant lag or downtime.
* User-friendly and Intuitive Interface: The system should include a user-friendly and intuitive interface for employees, reducing the likelihood of errors and improving efficiency.
* Detailed Sales and Revenue Reporting: The system should be able to generate detailed reports on sales, revenue, and customer data, allowing for effective business analysis and decision-making.
* Error-free and Fully Tested System: The system should be fully tested and free of bugs or glitches that could impact functionality.
* By meeting these acceptance criteria, the pizza billing system can be considered functional and reliable, providing a high-quality experience for both employees and customers.