PROJECT REPORT

On

Peaza - Kiosk Ordering App

Submitted by

Krish Patel (IU2041050078)
Kaksh Patel (IU2041050077)
Devarsh Patel (IU2041050074)

In fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

Computer Engineering

INSTITUTE OF TECHNOLOGY AND ENGINEERING

INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ

AMEDABAD-382115, GUJARAT, INDIA,

WEB: www.indusuni.ac.in

APRIL 2024

PROJECT REPORT

ON

Peaza - Kiosk Ordering App

AT

In the fulfilment of the requirement
for the degree of
Bachelor of Technology
in
Computer Engineering

PREPARED BY

Krish Patel (IU2041050078)
Kaksh Patel (IU2041050077)
Devarsh Patel (IU2041050074)

UNDER GUIDANCE OF

Internal Guide

Mrs. Divyani Tirthyani
Assistant Professor,
Department of Computer
Engineering,
I.I.T.E, Indus
University,
Ahmedabad

SUBMITTED TO

INSTITUTE OF TECHNOLOGY AND ENGINEERING

INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ

AHMEDABAD-382115, GUJARAT, INDIA,

WEB: www.indusuni.ac.in

APRIL 2024

CANDIDATE'S DECLARATION

I declare that final semester report entitled "**Peaza – Kiosk Ordering App**" is my own work conducted under the supervision of the guide **Mrs. Divyani Tirthyani.**

I further declare that to the best of my knowledge, the report for B. Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

Candidate's Signature
Krish Patel (IU2041050078)

Guide: Mrs. Divyani Tirthyani
Assistant Professor,
Department of Computer Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

CANDIDATE'S DECLARATION

I declare that the final semester report entitled "**Peaza – Kiosk Ordering App**" is my own work conducted under the supervision of the guide **Mrs. Divyani Tirthyani.**

I further declare that to the best of my knowledge, the report for B. Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

Candidate's Signature
Kaksh Patel (IU2041050077)

Guide: Mrs. Divyani Tirthyani
Assistant Professor,
Department of Computer Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

CANDIDATE'S DECLARATION

I declare that final semester report entitled "**Peaza - Kiosk Ordering App**" is my own work conducted under the supervision of the guide **Mrs. Divyani Tirthyani.**

I further declare that to the best of my knowledge, the report for B. Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

Candidate's Signature
Devarsh Patel (IU2041050074)

Guide: Mrs. Divyani Tirthyani
Assistant Professor,
Department of Computer Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING

INFORMATION TECHNOLOGY

2023 -2024

CERTIFICATE

Date:19/04/2024

This is to certify that the project work entitled "**Peaza - Kiosk Ordering App"** has been carried out by **Krish Patel, Kaksh Patel and Devarsh Patel** under my guidance in partial fulfillment of degree of Bachelor of Technology in **Computer Engineering (Final Year)** of Indus University, Ahmedabad during the academic year 2023 – 2024.

Mrs. Divyani Tirthyani Assistant Professor, Department of Computer Engineering, I.I.T.E, Indus University Ahmedabad	Dr. Seema Mahajan Head of the Department, Department of Computer Engineering, I.I.T.E, Indus University Ahmedabad

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of my Bachelor of Technology Project.

Firstly, I would like to thank my project guide, Mrs. Divyani Tirthyani, for their guidance and support throughout the project. Their valuable insights and suggestions have been instrumental in shaping my work.

I am also grateful to the faculty members of the department of information technology engineering for their support and encouragement. Their feedback and suggestions have been invaluable in improving the quality of my work.

I would like to express my gratitude to my family and friends for their unwavering support and encouragement throughout the project. Their constant motivation has been a source of inspiration for me.

Lastly, I would like to thank the university for providing me with the necessary resources and facilities to complete my project successfully.

Thank you all for your support and guidance.

Krish Patel, Kaksh Patel, Devarsh Patel

IU2041050078, IU2041050077, IU2041050074

Computer Engineering

TABLE OF CONTENT

ABSTRACT

Peaza is a game-changer in the realm of pizza ordering, offering a seamless and user-friendly web-based platform that redefines the customer-restaurant interaction. At the heart of Peaza lies its intuitive interface, which grants users easy access to an extensive menu brimming with a diverse array of culinary delights. This menu not only showcases the available food items but also furnishes detailed pricing information, empowering customers to make informed decisions effortlessly.

A standout feature of Peaza is its customization menu, which grants customers the power to tailor their orders precisely to their preferences and dietary needs. From selecting specific toppings to choosing the base, sauce, and cheese, Peaza ensures that each order is personalized to meet the unique tastes and requirements of every individual. This level of customization fosters a deeper sense of connection between customers and their dining experience.

Peaza's innovative approach doesn't stop there. Designed specifically for kiosk use, this platform streamlines the ordering process, significantly reducing wait times for customers. Moreover, by digitizing the entire process, Peaza has the potential to minimize the need for additional workforce, benefiting both restaurant owners and customers alike with enhanced efficiency and satisfaction.

In essence, Peaza stands out as a comprehensive and user-centric solution that not only simplifies the food ordering journey but also elevates customer satisfaction through its intuitive design and customizable features. By seamlessly integrating technology into the dining experience, Peaza sets a new standard for convenience, efficiency, and personalization in the realm of restaurant ordering.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

Abbreviations used throughout this whole document for Survey Application are:

HTML	Hypertext Markup Language
XML	Extensible Markup Language
HOD	Head of the Department
UML	Unified Modeling Language
CSS	Cascading Style Sheet
DBMS	Database management system
AJAX	Asynchronous JavaScript and XML

CHAPTER 1

INTRODUCTION

- **PROJECT SUMMARY**
- **PROJECT PURPOSE**
- **PROJECT SCOPE**
- **OBJECTIVES**
- **TECHNOLOGY AND TOOLS**
- **SYNOPSIS**

1.1 PROJECT SUMMARY

The Peaza project heralds a paradigm shift in the way customers engage with restaurants, particularly in the realm of pizza ordering. Designed as a user-friendly, web-based platform, Peaza offers a revolutionary interface that simplifies the entire ordering process. At its core lies an extensive menu, meticulously curated to showcase a diverse range of culinary offerings while providing transparent pricing information, empowering customers to make informed decisions effortlessly.

Peaza's standout feature is its customization menu, which grants users the freedom to tailor their orders according to their preferences and dietary requirements. From selecting toppings to choosing the base, sauce, and cheese, Peaza ensures that each order is personalized to meet the individual needs of every customer. This level of customization not only enhances the dining experience but also fosters a deeper sense of connection between customers and their food.

One of Peaza's key strengths lies in its adaptability for kiosk use. By streamlining the ordering process, Peaza significantly reduces wait times for customers, improving overall efficiency and satisfaction. Furthermore, the platform's digitized approach holds the potential to reduce the need for extra workforce, benefiting both restaurant owners and customers alike.

In summary, Peaza emerges as a comprehensive and user-centric solution that redefines the food ordering experience. Through its intuitive design and customizable features, Peaza not only simplifies the process but also elevates customer satisfaction to new heights. By seamlessly integrating technology into the dining experience, Peaza sets a new standard for convenience, efficiency, and personalization in the restaurant industry.

1.2 PROJECT PURPOSE

The purpose of this project is to implement an Online Ordering System for the pizzeria, aiming to address several key objectives. Firstly, the system seeks to eliminate the miscommunication often associated with phone orders by providing a visual menu for customers to make selections hassle-free. This not only enhances the ordering experience but also reduces errors in order processing.

Efficiency improvement is another crucial aspect targeted by the project. By offering an online platform, customers can significantly reduce their overall purchasing time, contributing to enhanced operational efficiency. Additionally, the system allows customers to place orders in advance via a mobile or web application, further streamlining the ordering process and reducing wait times.

Convenience enhancement is paramount in this project. The implementation of an online ordering system aims to improve the overall convenience for customers, ensuring a smooth and seamless ordering experience from start to finish.

Lastly, the project aims to provide the pizzeria with a competitive advantage in the market. By incorporating internet technology, the pizzeria can differentiate itself from competitors and deliver improved and efficient services to customers, ultimately strengthening its position in the industry.

1.3 PROJECT SCOPE

The project scope encompasses the development and implementation of an Online Ordering System for the pizzeria, focusing on addressing key objectives related to efficiency, convenience, and competitive advantage. The scope includes the design and creation of a user-friendly interface that offers a visual menu to facilitate hassle-free selection processes, thereby eliminating miscommunication commonly associated with phone orders. The system will be accessible via both mobile and web applications, allowing customers to place orders conveniently in advance. Efficiency improvement is a central aspect of the project scope, aiming to reduce overall purchasing time for customers by streamlining the ordering process. This entails optimizing the platform for quick and easy navigation, ensuring swift order placement and processing.

Convenience enhancement is also prioritized within the project scope. This includes features such as order tracking and customization options to cater to individual preferences and dietary requirements. Moreover, the project scope extends to leveraging internet technology to gain a competitive edge in the market. By embracing digital innovation, the pizzeria aims to enhance its services, differentiate itself from competitors, and strengthen its position in the industry.

Overall, the project scope encompasses the design, development, and implementation of an Online Ordering System aimed at enhancing efficiency, convenience, and competitiveness for the pizzeria and its customers.

1.4 OBJECTIVES

Peaza's primary objective revolves around the optimization of user activities within the realm of food ordering, aiming to reduce reliance on traditional paperwork methods while significantly enhancing efficiency. Through meticulous recording and management of details, Peaza endeavors to develop an intuitive and user-friendly online food ordering system. The core goal is to establish a platform that ensures reliability, convenience, and accuracy throughout the ordering process, ultimately delivering an unparalleled experience for customers.

At the heart of Peaza's objective lies the commitment to streamline the ordering experience. The platform's ordering interface is meticulously designed for seamless navigation and hassle-free transactions. Customers are empowered with the ability to effortlessly browse through the menu, select desired items, and proceed with the checkout process with minimal effort, all facilitated through an intuitive user interface. By simplifying the order placement journey, Peaza aims to minimize friction points, ensuring a smooth and enjoyable experience for every user. Moreover, Peaza strives to elevate the user experience by offering a range of convenient payment options. Whether it's traditional methods or modern digital wallets, the platform caters to diverse preferences, further enhancing the overall convenience and satisfaction of customers.

1.4.1 MAIN OBJECTIVES

Develop and implement Peaza, a user-friendly, web-based platform that revolutionizes the way customers interact with restaurants for pizza ordering, by creating an intuitive interface with an extensive menu and customization options, aiming to streamline the food ordering process and enhance customer satisfaction.

1.4.2 SECONDARY OBJECTIVES

Design and integrate an intuitive interface for Peaza that provides customers with access to an extensive menu showcasing a diverse range of culinary offerings, along with detailed pricing information, thereby enabling informed decision-making during the ordering process.

Develop a customization menu within Peaza that empowers customers to tailor their orders according to their preferences, including options for selecting specific toppings, base, sauce, and cheese, ensuring personalized and satisfying orders for each customer.

Optimize Peaza for kiosk use, with a focus on reducing wait times for customers by streamlining the ordering process and potentially reducing the need for additional workforce, thereby enhancing overall efficiency and satisfaction for both restaurant owners and customers alike.

1.5 TECHNOLOGY AND TOOLS

Front-end technologies used in this project include HTML5, CSS3, JavaScript, Bootstrap, React, Redux, MUI and back-end technologies include Node.JS and Express.JS. For the database, MongoDB is used.

1.5.1 FRONT-END TECHNOLOGY

- **HTML5:** HTML (HyperText Markup Language) is the basic language for development of primary web pages while HTML5 is a comprised of many elements, including the fifth revision of HTML, CSS3 and many JavaScript API's. It allows you to use the multimedia experience of the desktop on the web. Prior to this technology, experiences of this kind could only work on the desktop.

- **CSS3:** CSS (Cascading Style Sheets) consist of a group of formatting rules that you use to control the layout and appearance of the content on a web page. One really great feature of CSS is that one can store all the CSS rules in one document and keep that document separate from the HTML content and link the two together. Then, when a person make a change to the CSS that change is instantly and automatically updated in all the HTML files.

- **JavaScript:** JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more live and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

- **Bootstrap:** Bootstrap is a free and open-source front-end framework used for developing responsive and mobile-first websites and web applications. Bootstrap provides a collection of HTML, CSS, and JavaScript components and utilities that help developers quickly build user interfaces with consistent design and functionality across different devices and screen sizes.

- **ReactJS:** React is a popular JavaScript library for building user interfaces (UIs) for web applications. React is widely used by developers to create interactive and dynamic UI components. React is a powerful and flexible library for building modern web applications with dynamic and interactive user interfaces. Its component-based architecture, virtual DOM, and declarative programming model make it a popular choice among developers for building scalable and maintainable UIs.

- **MUI:** MUI, short for Material-UI, is a popular React UI framework that provides pre-designed components and styles following Google's Material Design guidelines. It offers a cohesive and customizable library for creating modern and responsive user interfaces, making it a preferred choice for developers building React-based applications.

- **Redux:** Redux is a state management library for JavaScript applications, commonly used with frameworks like React. It centralizes application state in a single store, making state management predictable and scalable. Redux facilitates predictable data flow and enables efficient handling of complex application states.

1.5.2 BACK-END TECHNOLOGY

- **Node.JS:** Node.js is a runtime environment that allows developers to run JavaScript code outside of a web browser. It is built on Chrome's V8 JavaScript engine and uses an event-driven, non-blocking I/O model, making it efficient and lightweight for building scalable network applications. Node.js is particularly popular for developing server-side applications, such as web servers and APIs, due to its

ability to handle a large number of concurrent connections with low overhead. It provides a vast ecosystem of libraries and modules through npm (Node Package Manager), which simplifies the development process by allowing developers to easily integrate third-party functionalities into their applications. Additionally, Node.js supports modern JavaScript features and asynchronous programming.

- **Express.JS:** Express.js, often referred to simply as Express, is a minimalist and flexible Node.js web application framework. It provides a robust set of features for building web and mobile applications, including routing, middleware support, and template engines. Express simplifies the process of creating server-side logic and handling HTTP requests, making it popular among developers for building APIs and web servers. Its modular and lightweight nature allows for quick development and easy integration of additional functionality through its vast ecosystem of plugins and middleware.

1.5.3 DATABASE

- **MongoDB:** MongoDB is a type of database software, often referred to as a NoSQL database, designed to store and manage large volumes of data. Unlike traditional relational databases like MySQL or PostgreSQL, which organize data into tables with predefined schemas, MongoDB stores data in a flexible, JSON-like format called BSON (Binary JSON) within documents. These documents are stored in collections, which are a kin to tables in relational databases but are schema-less, meaning each document in a collection can have a different structure. This flexibility allows for easier and faster development, as well as the ability to handle unstructured or semi-structured data. MongoDB is known for its scalability, performance, and ease of use. It's widely used in modern web development, particularly in applications that require flexible data models, fast iteration, and the ability to handle large amounts of data.

1.6 SYNOPSIS

Peaza is a revolutionary web-based platform transforming the pizza ordering experience. With an intuitive interface, detailed menu, and customization options, Peaza empowers customers to personalize their orders. Designed for kiosk use, it streamlines the ordering process, reducing wait times and potentially decreasing the need for extra workforce. Peaza's user-centric approach enhances efficiency and customer satisfaction, setting a new standard in food service technology.

CHAPTER 2

PROJECT MANAGEMET

- **Project Planning Objectives**
- **Project Scheduling**
- **Risk management**

2.1 PROJECT PLANNING OBJECTIVES

The objectives of project planning encompass the selection of appropriate software, thorough examination of its scope, and the formulation of a project development strategy utilizing the Software Development Life Cycle (SDLC).

2.1.1 SOFTWARE SCOPE

Visual Studio Code (VS Code) is one of the most popular code editors used by developers working with JavaScript, HTML, CSS, ReactJS, Node.JS, Express.JS, and many other languages and frameworks. It offers a range of features that make it a favorite among developers. Here are some of the reasons why developers find Visual Studio Code comfortable to work with:

- **Intuitive Git Integration:** Visual Studio Code provides seamless Git integration directly within the editor. Developers can easily visualize changes, commit code, and manage branches without leaving the editor. The Git features include commit history, diff views, and branch management.

- **Live Server and Debugger:** VS Code comes with built-in support for live server functionality, enabling developers to see changes in their code instantly without manually refreshing the browser. Additionally, it offers a powerful debugger that allows for efficient debugging of JavaScript and Node.js applications.

- **Extension Marketplace:** Visual Studio Code boasts a vast extension marketplace where developers can find various extensions to enhance their development experience. This includes extensions for JavaScript linting, HTML/CSS autocompletion, React development tools, Node.js integration, and much more.

- **Intelligent Code Completion:** VS Code provides intelligent code completion, which helps developers write code faster and with fewer errors. It offers suggestions for variable names, function parameters, and even entire code snippets based on the context.

- **Package Management:** VS Code makes managing packages and dependencies easy through integrated package managers like npm and yarn. Developers can install, update, and remove packages directly from the editor's interface.

- **Customizable User Interface:** Visual Studio Code allows developers to customize the editor's UI according to their preferences. They can choose from a variety of themes, layouts, and settings to create a personalized coding environment.

- **Local History and Version Control:** Although VS Code does not have built-in local history, it offers robust version control integration with Git and other version control systems. Developers can track changes, compare revisions, and revert to previous versions with ease.

- **Support for Web Technologies:** VS Code provides excellent support for web development technologies such as HTML, CSS, JavaScript, ReactJS, and Node.js. It offers features like syntax highlighting, code formatting, and language-specific extensions to streamline web development workflows.

- **Community and Industry Adoption:** Visual Studio Code has gained widespread adoption in the developer community and is backed by Microsoft. It is actively maintained, regularly updated, and has a vibrant ecosystem of users and contributors.

  Overall, Visual Studio Code offers a powerful and versatile development environment for JavaScript, HTML, CSS, ReactJS, Node.js, Express, and other popular languages and frameworks, making it a top choice for developers worldwide.

2.1.2 PROJECT DEVELOPMENT APPROACH

Software Development Life Cycle: Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. The testing team follows Software Testing Life Cycle (STLC) which is similar to the development cycle followed by the developmentteam. There are following six phases in every Software development life cycle model:

2.1.2.1 Requirement Gathering and Planning

2.1.2.2 Analysis

2.1.2.3 Design

2.1.2.4 Implementation or Coding

2.1.2.5 Testing and Integration

2.1.2.6 Deployment and Maintenance

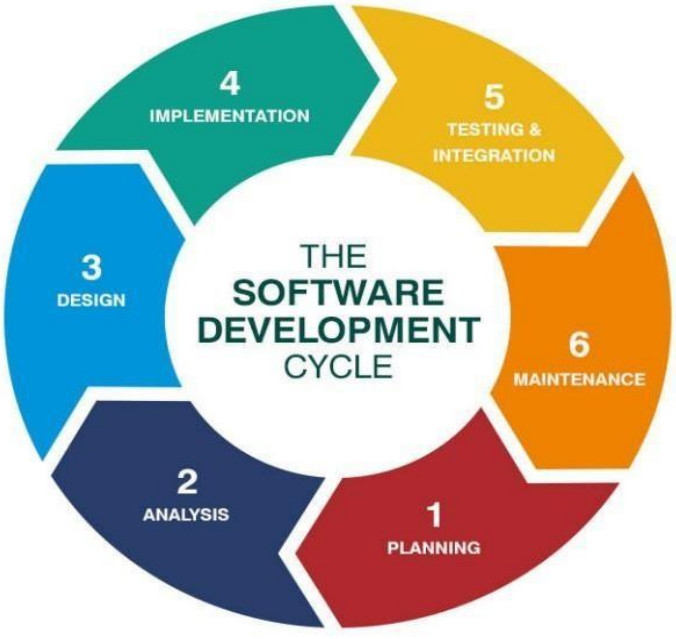

Fig 2.1.2.1: SDLC

2.2 PROJECT SCHEDULING

Project schedule means a mechanism that is used to communicate and know the tasks that are needed and must be done or performed. Effective project scheduling leads to success, reduced cost, and increased customer satisfaction. Scheduling in project management means listing out activities, deliverables, and milestones within a project that are delivered.

2.2.1 BASIC PRINCIPLE

The project started with choosing a project definition, and then all the required things were gathered, like which language to use and which framework to use. After that, the basic things started, which ended up with making a whole working software for an online food ordering system, including both admin and customer sides.

2.2.2 PROJECT ORGANIZATION

Project Planning: The problem is decomposed into smaller problems, softwaremanagers use historical project data (as well as personal experience andintuition) to determine estimates for each.

- The final estimates are adjusted by taking project complexity and risk intoaccount. The resulting work product is called project management plan.
- Project planning also includes description of the project task, activity and function, dependencies, resource requirements and detail schedules.
- Project planning involves estimating how much time, efforts, money and resources will be required build a specific software system. After the projectscope is determined and to define the best case and worst case scenarios. Sothat project outcome can be bounded.
- Project Scheduling: Software project scheduling is an activity that distributes estimated efforts across the planned duration by allocating the effort to specific software engineering tasks. Proper Scheduling requires:

1. All tasks appear in network.
2. Effort and timing are intelligently allocated to each task.
3. Interdependencies between tasks are properly indicated.
4. Resources are allocated for the work to be done.

2.2.3 TIMELINE CHART

2.2.3.1 Time Allocation

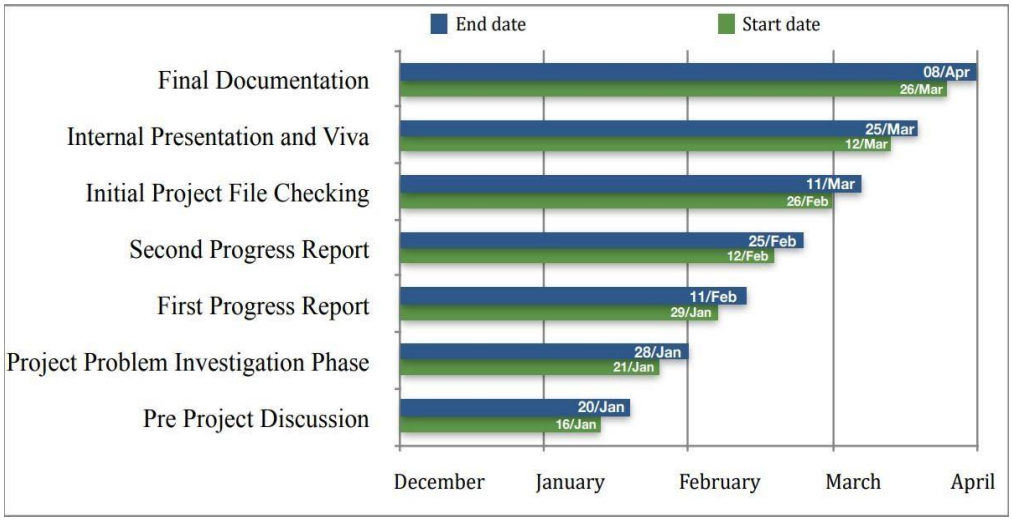

Fig 2.2.3.1: Time Allocation Chart

2.2.3.2 Task Sets

PROJECT CRITERIA	PRESENTATION
Pre Project Discussion	Project definition, Functionalities of project and Timeline chart of implementation.
Project Problem Investigation Phase	1. Explore the Project Problem 2. Analyze the Problem 3. Define the Problem 4. Problem Formulation/Objectives of the Project
First and Second Progress Report	Report to summarise the status of project.
Initial Project File Checking	Introduction about the project, Design phase, Working Methodology of the project, Results, Conclusion, and Future Directions.
Internal Presentation and Viva	Project execution, show skill set on the presented technology along with brief summary about all the phases and modules.
Final Documentation	Final hard copy of the file after the changes in the previous phases.

Fig 2.2.3.2: Task Sets

2.3 RISK MANAGEMENT

A risk is any unfavorable event or circumstance that can occur while a project is underway. Managing software is difficult. At any point in time, the code can start running in weird ways. Things can go wrong, and most of the time we are unaware of the problem. This can and will happen to all the developers out there. It is part of life. So, the main objective is to identify the risk, which can help us understand and manage uncertainty during the development of the project.

2.3.1 RISK IDENTIFICATION

The following are the possible risks associated with the project:

1. **Technical risks:** If the internet connection is poor, there may be some visual bugs. A future update on the website may lead to its shutdown. The website may be slow on some devices.
2. **Project Risks:** The scope may have been defined incorrectly, and the project may have gone in the wrong direction. The time limit might not have been properly calculated as per the scope. Therefore, it may be difficult to finish the project within the deadline and we may need to remove some functionalities.

2.3.1.1 Risk Identification artifacts

All the websites have different internal and external risks. Internal risk basically consists of hardware failure and power interruption, for which a solution is specified. External risks are associated with the website, like viruses, hacking, and the corruption of files. To solve this, we need to be extremely careful.

2.3.2 RISK PROJECTION

During the risk projection, each identified risk is considered in turn, and a judgement is made about the probability and seriousness of the risk.

Risk	Probability	Effects
Major upgrade in Node JS	Medium	Normal
Security issue	Low	Severe
Reusable code may not work in all places	High	Normal
May the system work slow	Low	Normal

2.3.2 Risk Projection

CHAPTER 3

SYSTEM REQUIREMENTS

- **User Characteristics**
- **Functional Requirement**
- **Non-Functional Requirement**
- **Hardware and Software Requirement**

3.1 USER CHARACTERISTICS

The primary users of the Peaza - Kiosk Ordering App will be customers who want to order food from pizzeria. These users are expected to have basic computer knowledge. Additionally, the system should be easily accessible and user-friendly to all users, including those with disabilities.

3.2 FUNCTIONAL REQUIREMENT

Functional requirements refer to the precise features, abilities, and attributes that a system must possess to fulfill the demands of its users. Within the context of the **Peaza - Kiosk Ordering App**, functional requirements may encompass aspects such as:

- **Admin login:** Admin will login into store account such that customer can order and staff can see pending order.
- **Interactive Menu:** An interactive menu showcasing various pizza options and additional products, accompanied by detailed ingredient lists with corresponding prices for transparency.
- **Personalized Pizza Creation:** Customers have the option to customize their pizzas by choosing specific ingredients, adjusting quantities, and selecting preferred toppings, sauces, and cheeses to suit their taste preferences.
- **Flexible Payment Methods:** Customers can choose between online payment methods or opt for cash-on-delivery to complete their orders, providing flexibility and convenience during the checkout process.
- **Order Confirmation:** Once the order is placed, customers receive confirmation on payment page, ensuring a seamless and informed ordering experience from start to finish.

This streamlined process guarantees a user-friendly and efficient ordering experience for customers whether they are accessing the pizzeria's website or mobile application.

3.2.1 ACTIVITY AND PROPOSED SYSTEM

The proposed system should allow customers to search for restaurants and view their menus. Customers should be able to place orders, make payments, and track the status of their orders. The system should also allow restaurants to receive and process orders and update their menus.

3.3 NON - FUNCTIONAL REQUIREMENT

Non-functional requirements encompass the attributes and limitations that a system must adhere to in order to effectively serve its intended purpose. For the **Peaza - Kiosk Ordering App**, non-functional requirements may include:

- **Performance:** Ensuring swift and efficient system response, especially during peak usage periods.
- **Scalability:** Ability to manage growing user numbers and transactions seamlessly.
- **Reliability:** High system availability, minimizing downtime for maintenance or upgrades.
- **Security:** Safeguarding sensitive data like customer details and payment information against unauthorized access.
- **Usability:** Offering a user-friendly interface that's intuitive and easy to navigate.
- **Compatibility:** Supporting various devices, operating systems, and web browsers for seamless access.
- **Accessibility:** Ensuring the system is accessible to users with disabilities.
- **Maintainability:** Maintaining a well-documented and easily manageable codebase for efficient upkeep.

3.4 HARDWARE AND SOFTWARE REQUIREMENT

3.4.1 HARDWARE REQUIREMENT

- **Memory:** 4 GB
- **CPU:** Intel Core i3-2340UE
- **Storage**: None

3.4.2 SOFTWARE REQUIREMENT

- **Operating System :-** Windows, Linux, Android, MAC
- **Software System :-** MongoDB
- **Windows System :-** Windows XP or higher

CHAPTER 4

SYSTEM ANALYSIS

- **Study of Current System**
- **Problems in Current System**
- **Requirement of New System**
- **Process Model**
- **Feasibility Study**

4.1 STUDY OF CURRENT SYSTEM

The prevailing method for food ordering in many restaurants primarily involves customers physically visiting the establishment to place their orders, engaging in direct interactions with restaurant staff. This process typically entails waiting in queues, verbally communicating order details, and relying on staff members to accurately capture preferences. However, this manual approach has inherent limitations. Customers may lack access to comprehensive menu information, hindering their ability to make well-informed decisions. Moreover, the verbal nature of order placement can lead to misunderstandings or errors in recording customer preferences, potentially resulting in order inaccuracies and customer dissatisfaction.

Furthermore, during peak hours or busy periods, the manual order-taking process may contribute to longer wait times, exacerbating customer frustrations and impacting overall service efficiency. The absence of digital platforms for ordering also restricts customization options, making it challenging for customers to tailor their orders according to specific dietary needs or preferences.

Overall, while the current system may suffice for some scenarios, its reliance on manual processes presents challenges in terms of efficiency, accuracy, and customer experience, highlighting the potential benefits of adopting digital solutions like **Peaza** to streamline and enhance the food ordering process.

4.2 PROBLEMS IN CURRENT SYSTEM

The current system of food ordering in many restaurants faces several challenges that impact the overall customer experience. Firstly, the reliance on manual processes, such as walk-in orders and in-person interactions, can lead to inefficiencies and errors in order taking. Customers may find themselves waiting in queues, and the verbal communication of order details may result in misunderstandings or missed additions. This can lead to order inaccuracies, where customers receive food items that do not align with their preferences or dietary requirements.

Another significant issue is the limited availability of detailed menu information. Customers often have limited access to comprehensive menu information, such as

ingredients, pricing, or nutritional content, which can make it challenging for them to make informed decisions about their orders. This lack of transparency can contribute to dissatisfaction, as customers may feel unsure about what they are ordering.

Furthermore, during peak hours or busy periods, the manual order-taking process can contribute to longer wait times for customers. The absence of digital platforms for ordering also restricts customization options, making it difficult for customers to tailor their orders according to their specific preferences or dietary needs.

Overall, these challenges cumulatively impact customer satisfaction and contribute to a less-than-optimal dining experience. Addressing these issues through the adoption of digital solutions, such as Peaza, can streamline the food ordering process, enhance efficiency, improve accuracy, and ultimately lead to a more satisfying experience for customers.

4.3 REQUIREMENT OF NEW SYSTEM

The new food ordering system, Peaza, designed specifically for kiosk usage, has distinct requirements tailored to this purpose. It necessitates a user-friendly interface optimized for kiosk screens, showcasing a comprehensive menu with detailed information on culinary offerings. Customization options for toppings, base, sauce, and cheese remain essential for personalized orders. Efficient order management and secure payment integration are crucial for smooth transactions. Additionally, compatibility with various kiosk hardware setups and reliability to minimize downtime are key considerations. Peaza's primary focus on kiosk usage ensures a streamlined and efficient food ordering experience within the physical restaurant environment.

4.4 PROCESS MODEL

The **Spiral model** stands as a cornerstone in software development methodologies, renowned for its adept handling of project risks. Visualized as a spiraling path with undefined loops, its adaptability shines through as each project's unique needs dictate the number of phases required. At the helm of this model sits the project manager, dynamically shaping the process to navigate the complexities of requirements, design, implementation, and testing.

Each loop in the spiral encapsulates a phase, serving as a structured journey through the software development cycle. The project manager's key responsibility lies in assessing the project risks and determining the appropriate number of phases needed to address them effectively. This flexibility ensures that the development process remains agile and responsive to evolving project dynamics.

The spiral's geometry offers a symbolic representation of project progress and expenditure. The radius of the spiral reflects the cumulative cost of the project, while the angular dimension denotes the advancement within the current phase. This visual metaphor provides stakeholders with a clear understanding of the project's trajectory.

Within each phase, the development process unfolds across four distinct quadrants, each with its specific objectives and activities. Firstly, requirements are meticulously gathered and analyzed, laying the groundwork for identifying alternative solutions. Subsequently, potential risks are identified, evaluated, and mitigated, often through the creation of prototypes to validate proposed solutions.

The third quadrant witnesses the transformation of conceptual solutions into tangible products, with features developed and rigorously tested to ensure functionality and quality. Finally, stakeholders review the progress made thus far, providing valuable feedback to inform planning for the next phase. This iterative approach fosters continuous improvement and ensures alignment with stakeholders' evolving needs and expectations.

In essence, the Spiral model embodies a dynamic and iterative approach to software

development, characterized by adaptability, risk management, and stakeholder collaboration. By embracing uncertainty and prioritizing flexibility, organizations can navigate the complexities of software development with confidence and efficiency.

Fig 4.4.1: Spiral Model

4.5 FEASIBILITY STUDY

4.5.1 TECHNICAL FEASIBILITY

The system will leverage cutting-edge technologies like Node.js and Express.js for the backend, ensuring robustness and efficiency. For the frontend, a modern stack comprising ReactJS, HTML5, CSS3, Bootstrap, and MUI will be utilized, guaranteeing a sleek and intuitive user interface. These technologies boast vibrant community support, streamlining the development process significantly.

Additionally, the system will be deployed on Vercel, a powerful deployment platform known for its scalability and reliability. By harnessing Vercel's capabilities, the system will be poised for seamless scaling and high availability, ensuring optimal performance even under heavy loads.

4.5.2 OPERATIONAL FEASIBILITY

The upcoming system prioritizes user-friendliness and intuitive navigation, enhancing overall user experience. With real-time updates on order status, customer inquiries and complaints are expected to decrease significantly. Additionally, a feedback mechanism will enable customers to rate both food quality and service, fostering continuous improvement and customer satisfaction.

4.5.3 ECONOMICAL FEASIBILITY

The investment in developing and deploying the system will be carefully weighed against the anticipated returns, including boosted revenue and heightened customer satisfaction. This evaluation ensures that resources are allocated judiciously, maximizing the system's impact on business performance.

4.5.4 SCHEDULE FESABILITY

During schedule feasibility assessment, an organization gauges the projected timeframe for project completion. This analysis also uncovers potential constraints, including internal factors like technical limitations, technological requirements, budgetary constraints, and resource availability. By scrutinizing these areas, the feasibility analysis ensures that any hindrances to the proposed project are identified and addressed proactively.

CHAPTER 5

DETAIL DESCRIPTION

- **Customer Module**
- **Store Module**

5.1 CUSTOMER MODULE

The Customer interface stands as a cornerstone in any kiosk food ordering system, serving as the primary gateway for users to interact with the platform. This pivotal module empowers customers to effortlessly place orders and complete payments within the platform's ecosystem.

Home Screen: The home screen of Peaza features a visually appealing display of various pizza options, enticing customers with a diverse range of culinary choices. Alongside these options, the cart prominently displays the total price of items added, providing customers with real-time visibility into their order's cost.

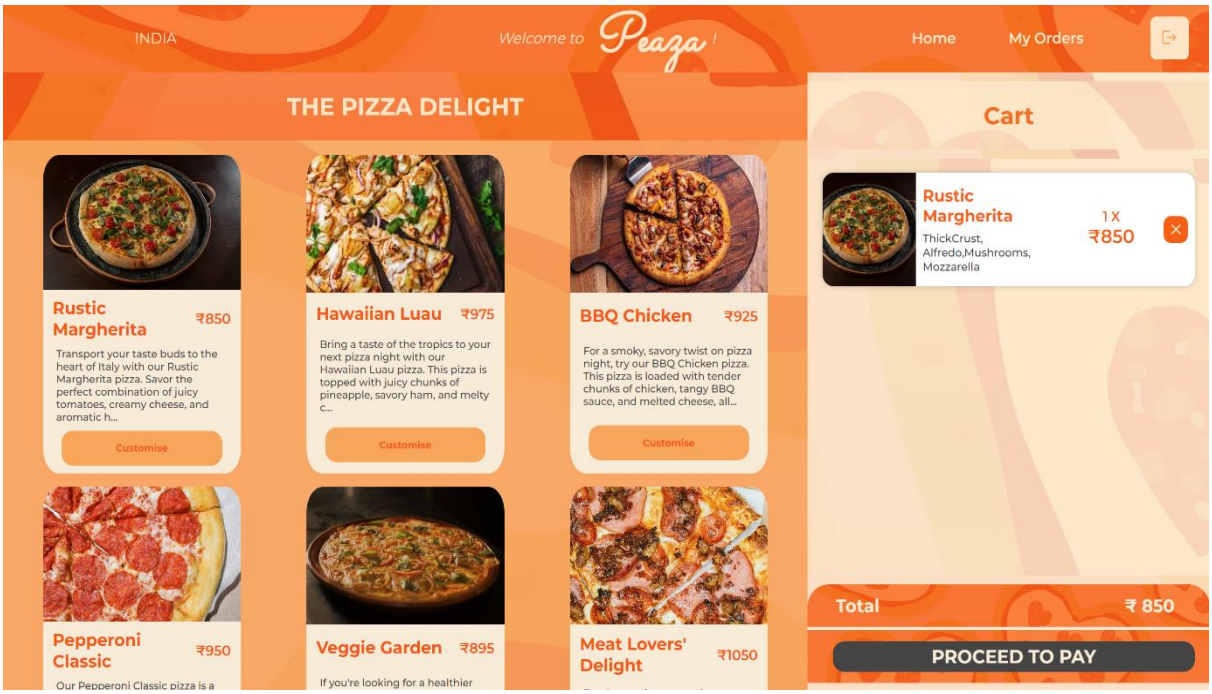

Fig 5.1.1: Home Screen

Menu Items : On Peaza's menu screen, customers are presented with a variety of pizza options, each accompanied by a "customize" button. This feature allows customers to tailor their orders to their preferences by selecting specific toppings, base types, sauces, and cheeses. The customization button streamlines the ordering process, empowering customers to create personalized pizzas that align with their taste preferences and dietary requirements.

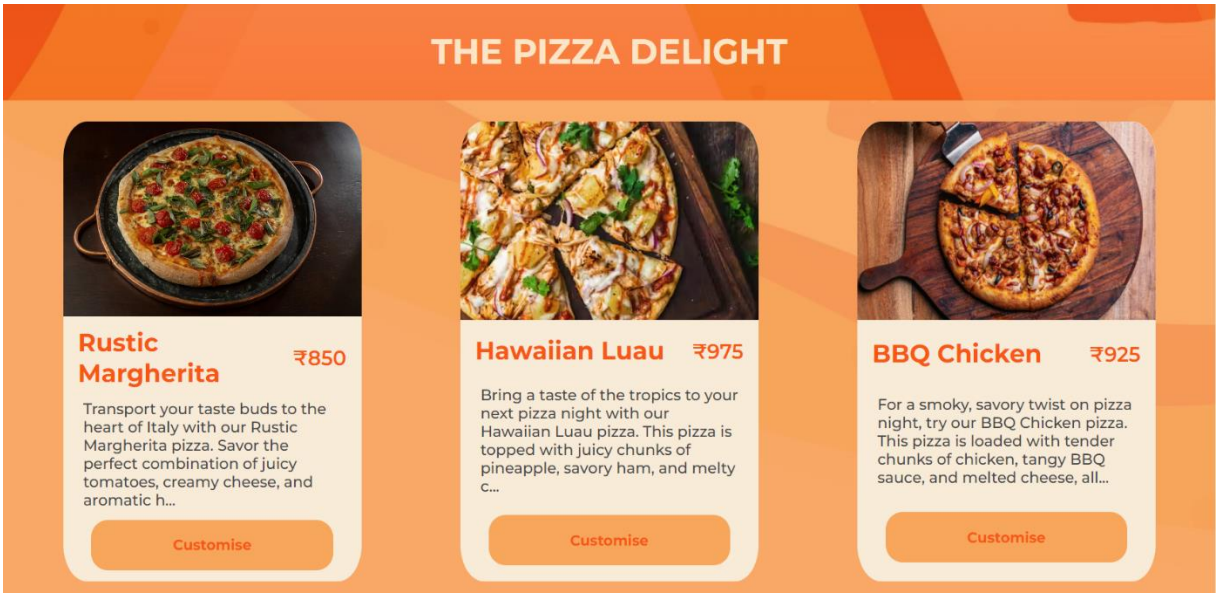

Fig 5.1.2: Menu Items

Customization Panel : The customization panel offers a dynamic interface for customers to tailor their pizzas to their exact desires. From crust thickness and toppings to sauce selections and cheese blends, every element is customizable. It empowers customers to create their perfect pizzas, ensuring a personalized and satisfying culinary experience.

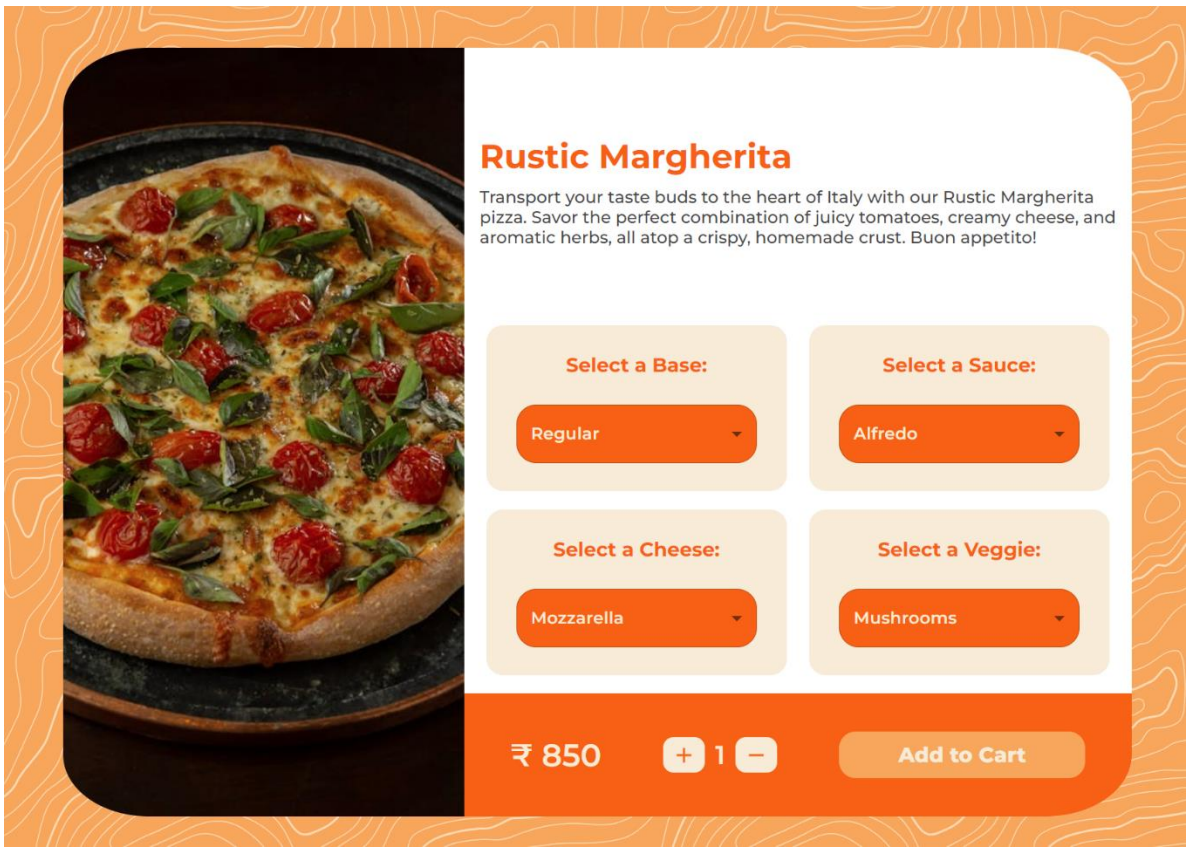

Fig 5.1.3: Customization Panel

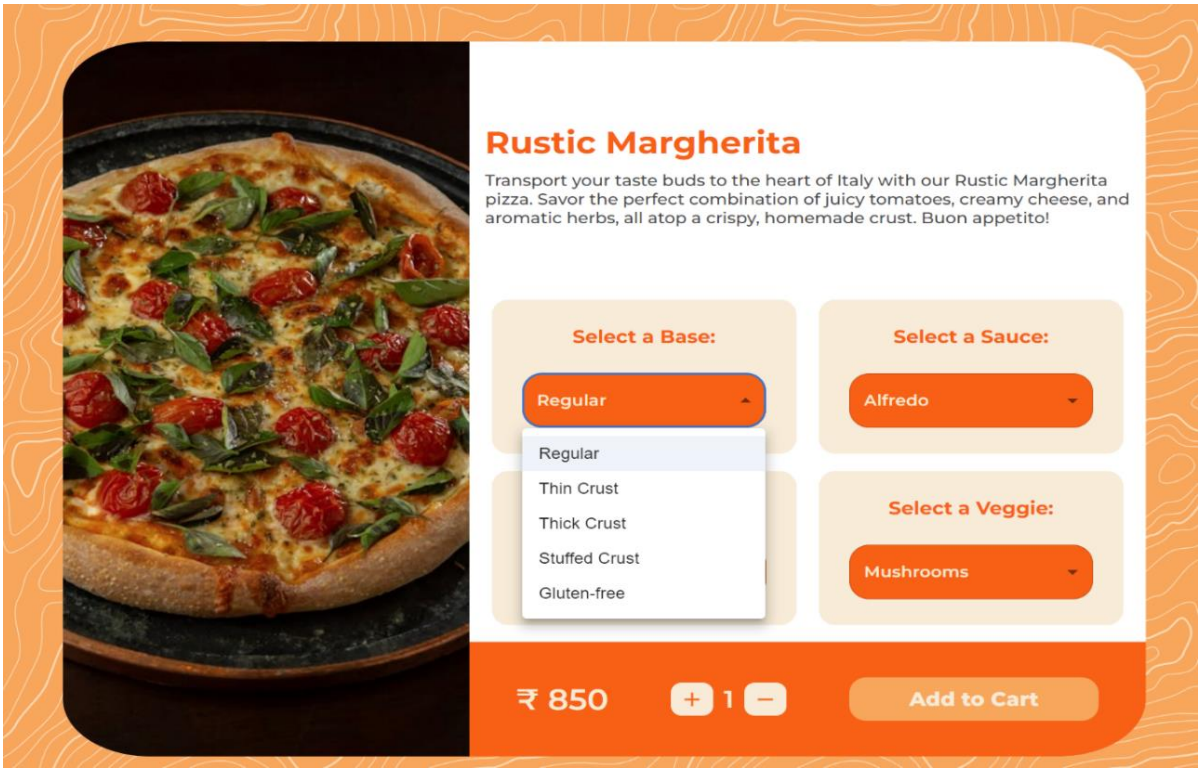

Fig 5.1.4: Base Customization

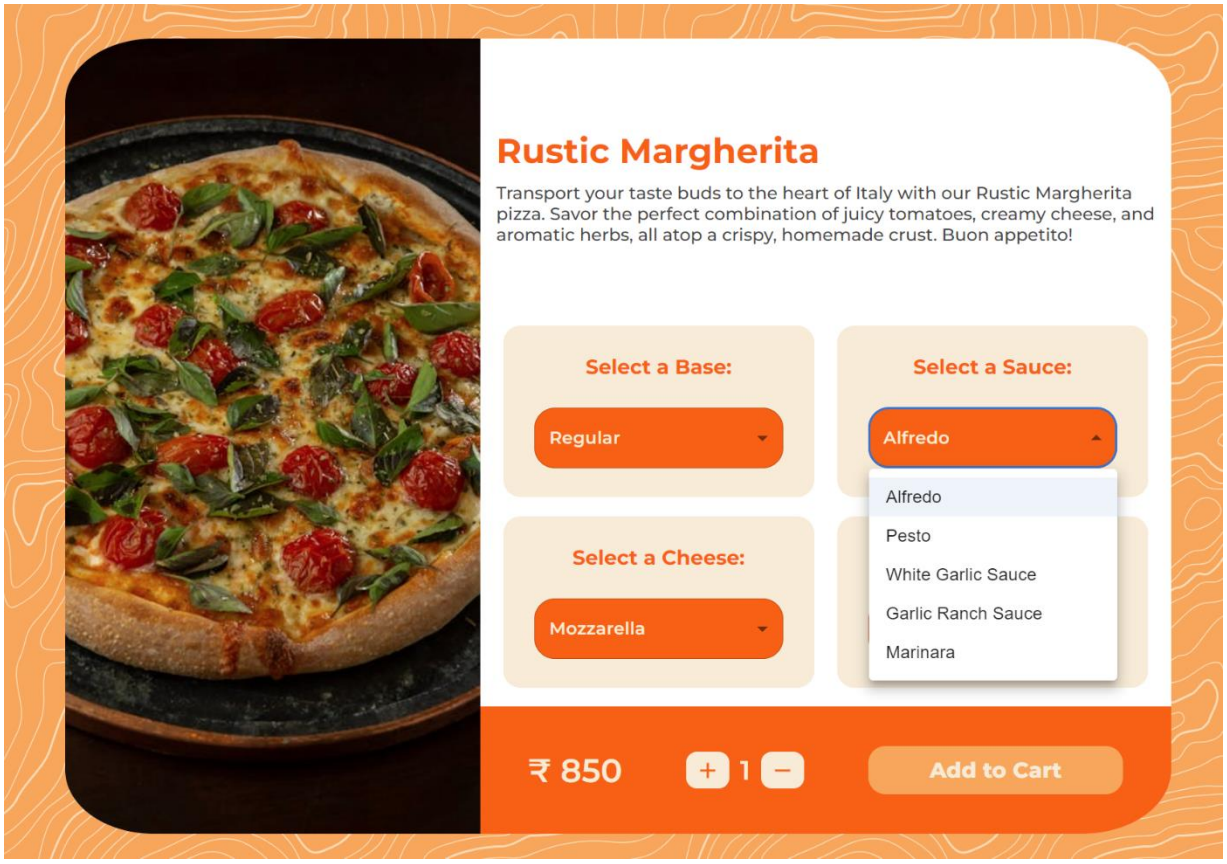

Fig 5.1.5: Sauce Customization

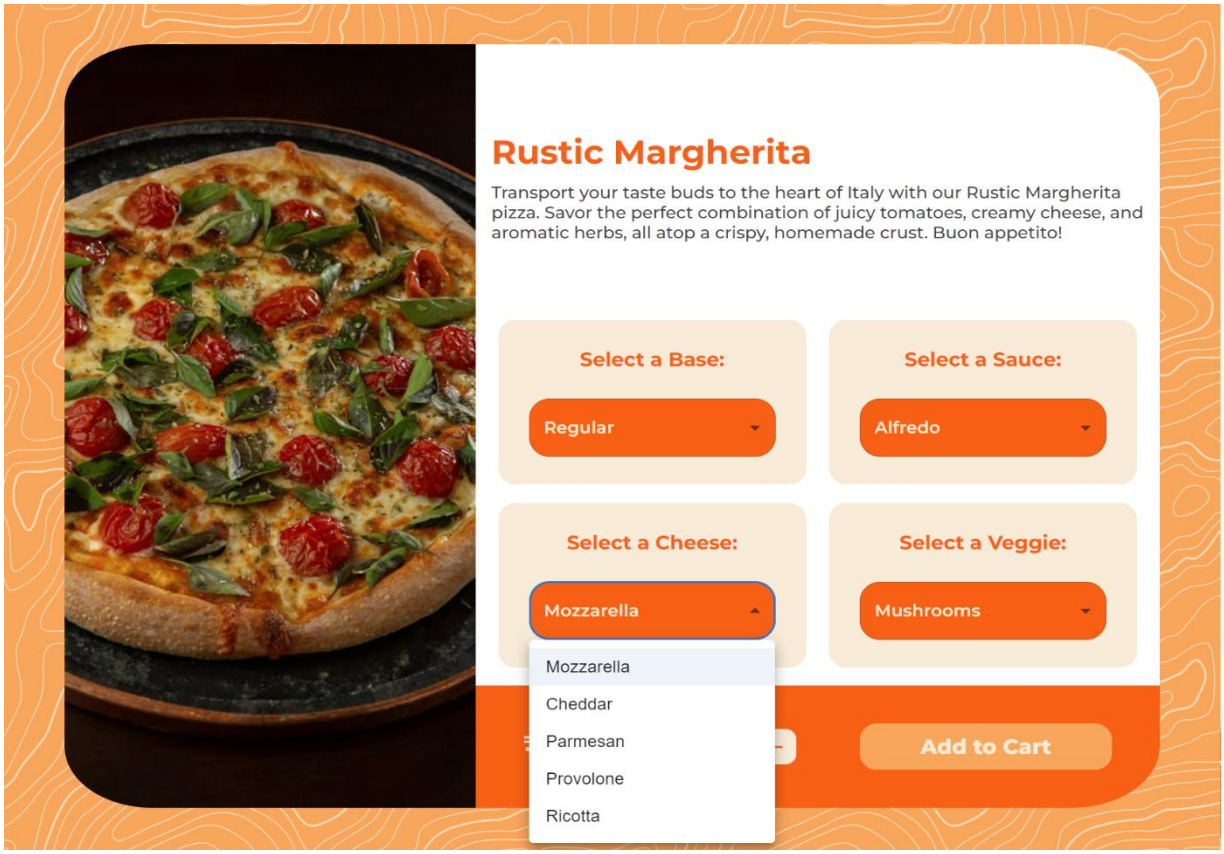

Fig 5.1.6: Cheese Customization

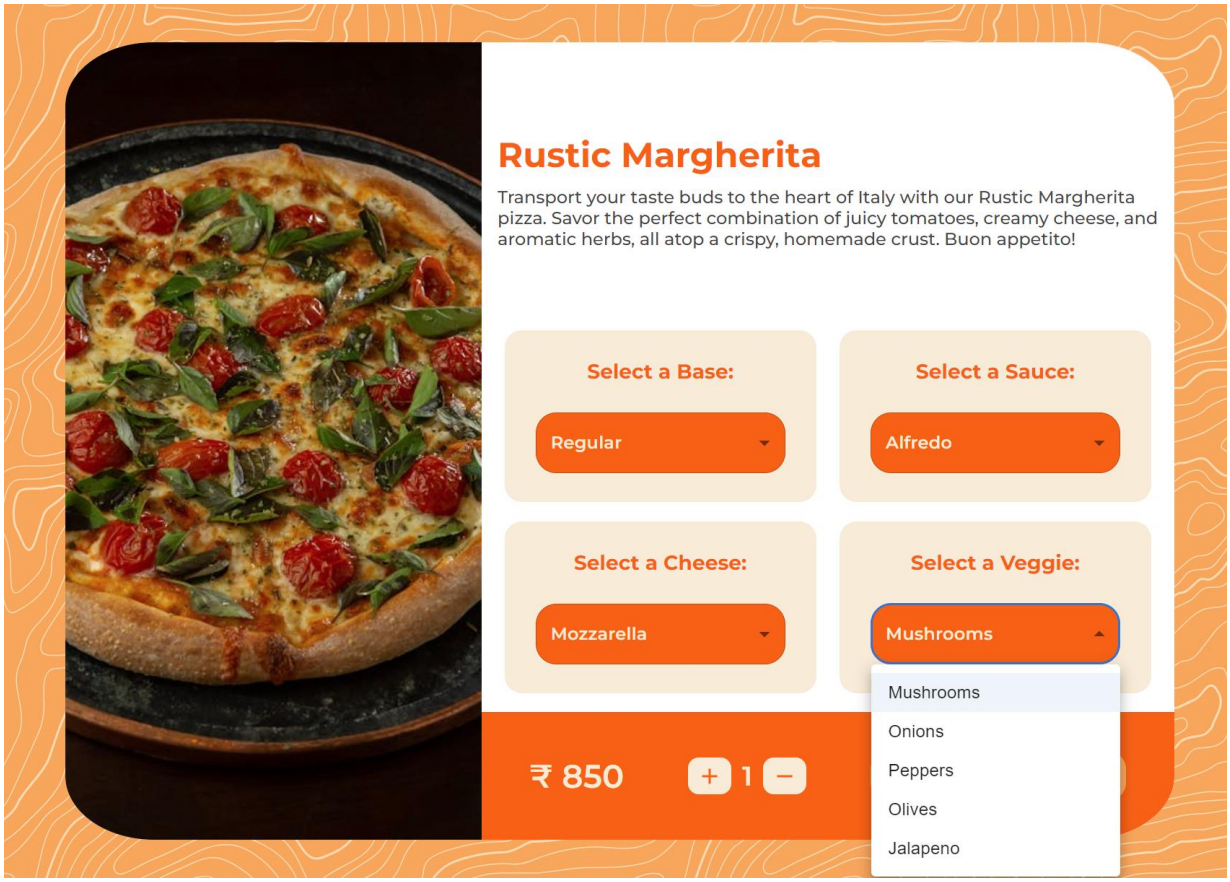

Fig 5.1.7: Veggie Customization

Cart Screen: Upon reaching the cart screen on Peaza, customers can view their customized pizza selections, complete with details such as quantity and individual prices. Additionally, the cart displays the total price of the entire order, providing customers with a clear overview of their purchases. The presence of a "proceed to pay" button streamlines the checkout process, allowing customers to easily move forward with payment and complete their orders hassle-free.

Fig 5.1.8: Cart Screen

Payment Screen: On the payment screen of Razorpay, customers are presented with a range of payment options, including card payments, net banking, wallets, and the "pay later" option. This variety allows customers to choose their preferred payment method based on convenience, security, and personal preference. Whether they opt for direct card payments, use net banking for seamless transactions, prefer the convenience of digital wallets, or select the pay later option for flexibility, Razorpay ensures a versatile and user-friendly payment experience tailored to individual customer needs.

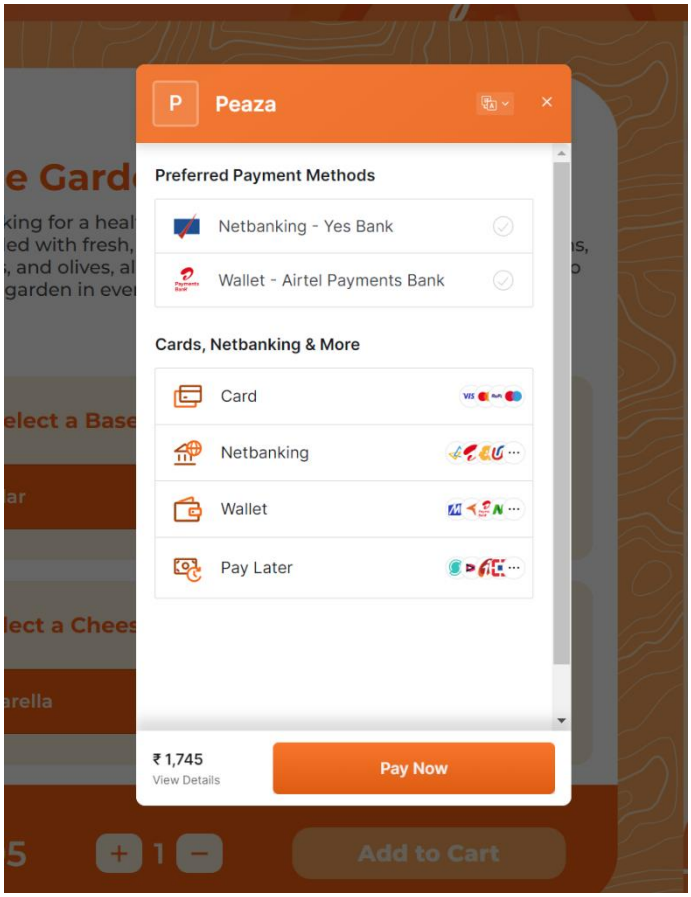

Fig 5.1.9: Payment Screen

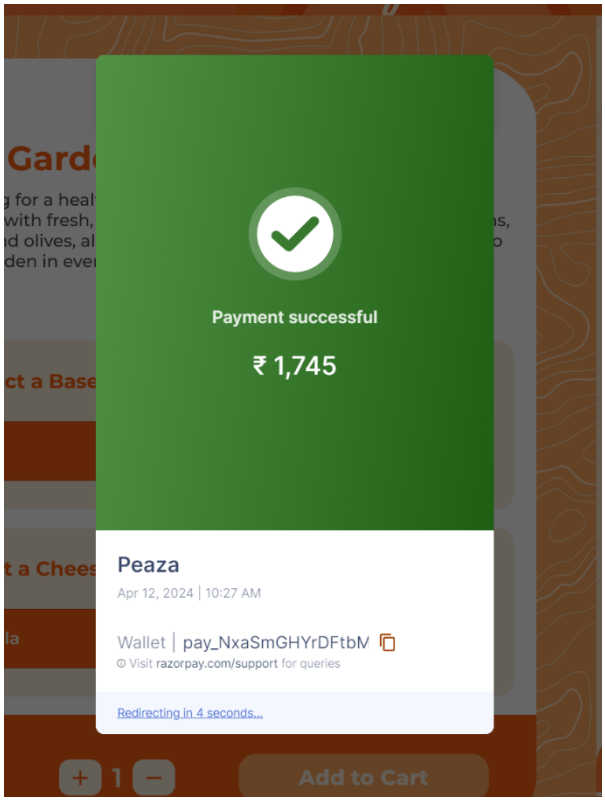

Fig 5.1.10: Payment Success Screen

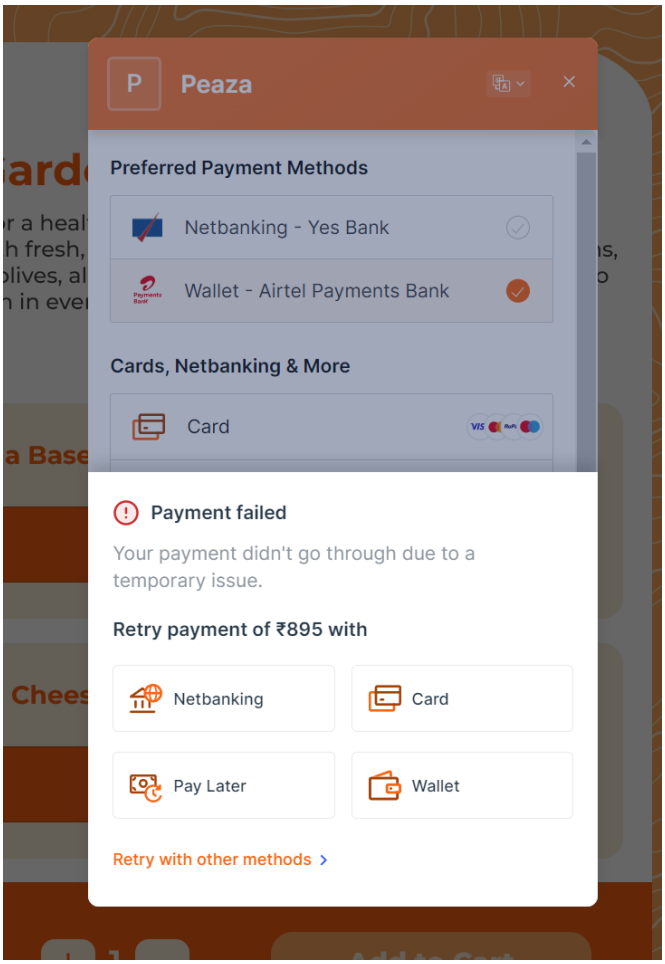

Fig 5.1.11: Payment Failure Screen

5.2 STORE MODULE

The Store Module is designed for store managers who manage the overall functionality and security of the platform.

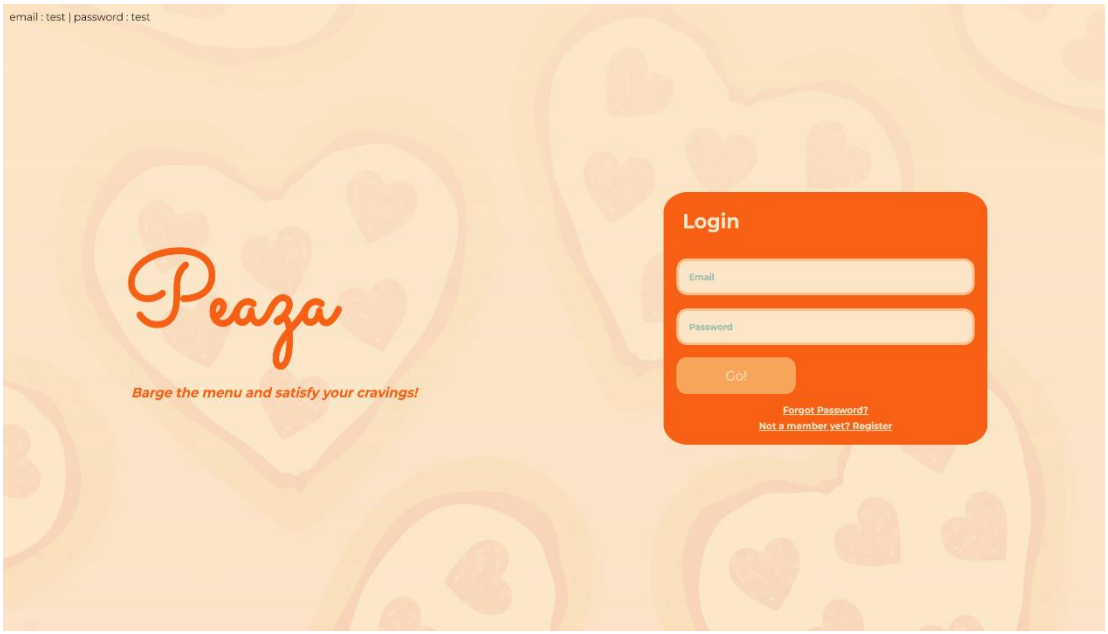

Fig 5.2.1: Store Login Screen

Fig 5.2.2: Orders Screen

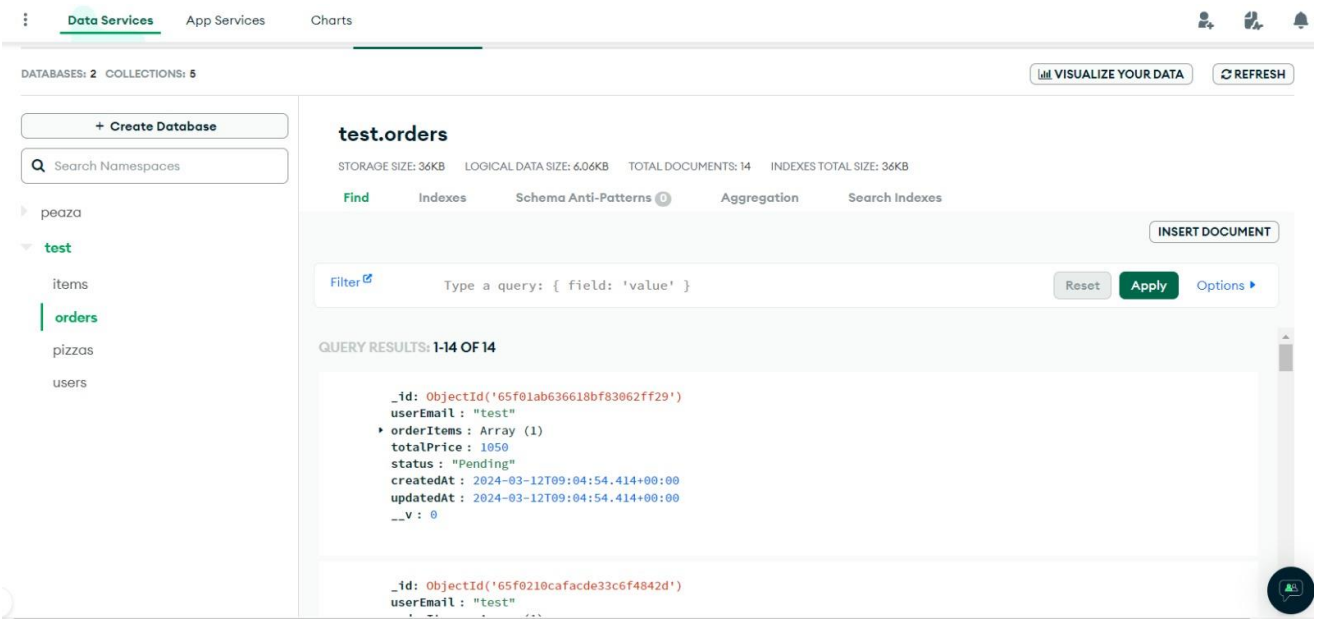

Fig 5.2.3: MongoDB orders database

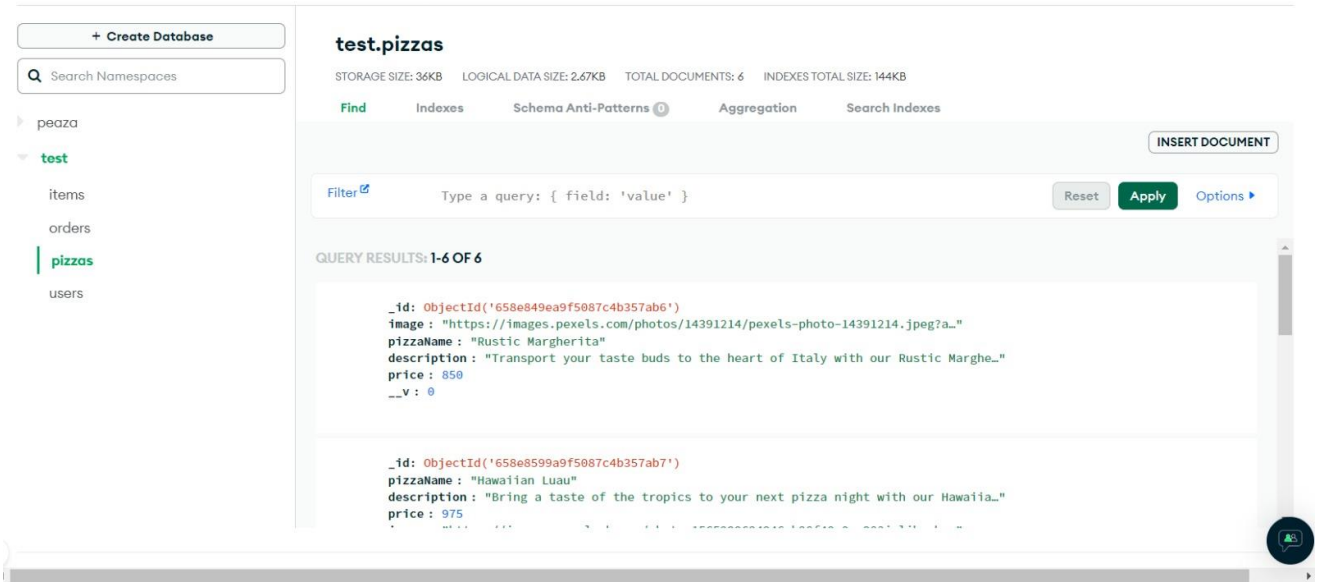

Fig 5.2.4: MongoDB Pizza's Database

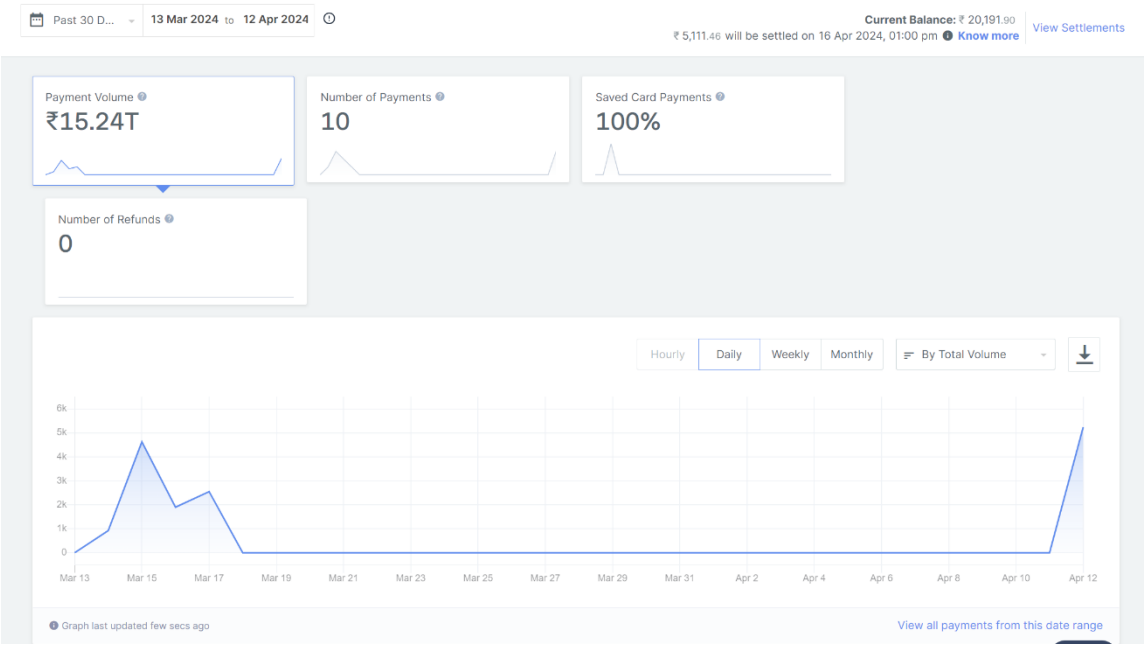

Fig 5.2.5: Payment Gateway's Interface-1

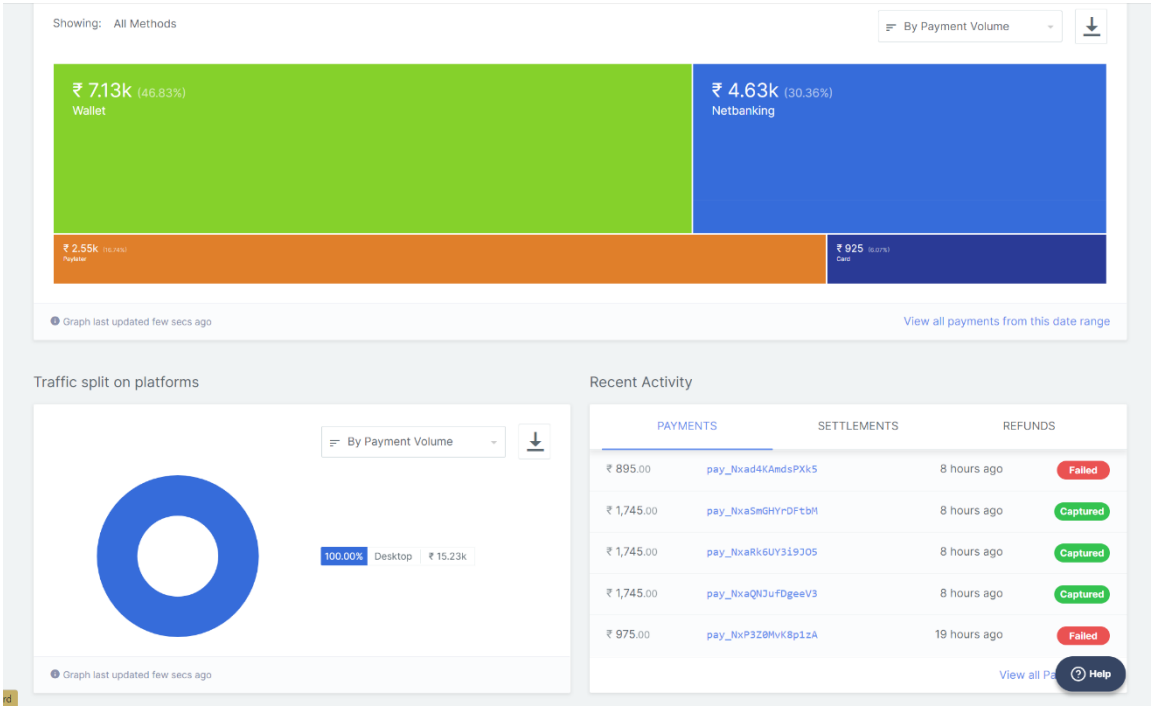

Fig 5.2.6: Payment Gateway's Interface-2

CHAPTER 6

TESTING

- **Black Box Testing**
- **White Box Testing**
- **Test Cases**

6.1 Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of testing can be applied to virtually every level of software testing: unit, integration, system, and acceptance. It is sometimes referred to as specification-based testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black- box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

Following are some techniques that can be used for designing black box tests:

- **Equivalence Partitioning*:*** It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.
- **Boundary Value Analysis*:*** It is a software test design technique that involves the determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.

Advantages: -

- Tests are done from a user's point of view and will help in exposing discrepancies in the specifications.
- Testers need not know programming languages or how the software has been implemented.
- Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer- bias.

Disadvantages: -

- Only a small number of possible inputs can be tested and many program paths will be left untested.
- Tests can be made redundant if the software designer/developer has already run a test case.

6.2 White Box Testing

White-box testing (also known as Glass Box Testing, Transparent Box Testing, Structural Testing) is a software testing method in which the internal structure / design / implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. White box testing is testing beyond the user interface and into the nitty-gritty of a system. This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.

Advantages: -

- Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
- It helps in optimizing the code.
- Extra lines of code can be removed which can bring in hidden defects.
- Testing is more thorough, with the possibility of covering most paths.

Disadvantages: -

- Since tests can be very complex, highly skilled resources are required, with a thorough knowledge of programming and implementation.
- Test script maintenance can be a burden if the implementation changes too frequently.
- Since this method of testing is closely tied to the application being tools to cater to every kind of implementation/platform may not be readily available.

6.3 Test Cases

Test cases are the specific scenarios that are used to test the system's functionality and behavior. In the context of kiosk food ordering system, the following are some of the test cases that can be used:

1. Test Case 01

Serial No.	Test Cases	Expected Result	Test Result
1.	Enter the valid store data.	Software should display the web - page.	Successful
2.	Invalid data	Software will give an error.	Successful

Table 6.3.1 Account Login

2. Test Case 02

Serial No.	Test Cases	Expected Result	Test Result
1.	Click on the customize button on pizza card.	Software will show the customization menu.	Successful
2.	Click on ‘select a sauce’ dropdown menu	Software will show various options menu.	Successful
3.	Click on ‘select a base’ dropdown menu	Software will show various options menu.	Successful
4.	Click on ‘select a veggie’ dropdown menu	Software will show various options menu.	Successful
5.	Click on ‘select a cheese’ dropdown menu	Software will show various options menu.	Successful

Table 6.3.2 Customization Menu

3. Test Case 03

Serial No.	Test Cases	Expected Result	Test Result
1.	Adds item to the cart by selecting the 'Add to cart' button.	It will be added into the cart.	Successful
2.	By clicking the plus sign	It will add the item one more time.	Successful
3.	By clicking the minus sign	It will remove quantities by one.	Successful
4.	By clicking the close button in cart.	It will remove the entire item from cart.	Successful

Table 6.3.3 Add to Cart

4. Test Case 04

Serial No.	Test Cases	Expected Result	Test Result
1.	Click on proceed to pay	Payment page pop-up will be shown. Customer must provide contact details.	Successful
2.	After entering contact detail, clicking the Proceed button.	Different payment options can be seen. Customer will select their preferred payment option.	Successful
3.	After clicking on preferred payment options.	Customer must put card, bank, wallet detail depending upon option chosen.	Successful
4.	On clicking pay now option	Payment will be shown as Success or Failure	Successful
5.	If payment fails	Payment can be done again.	Successful

Table 6.3.4 Bill Payments

CHAPTER 7

SYSTEM DESIGN

- **Class Diagram**
- **Use – Case Diagram**
- **Sequence Diagram**
- **Activity Diagram**
- **State Chart Diagram**
- **Data Flow Diagram**
- **ER Diagram**
- **Data Dictionary**

7.1 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

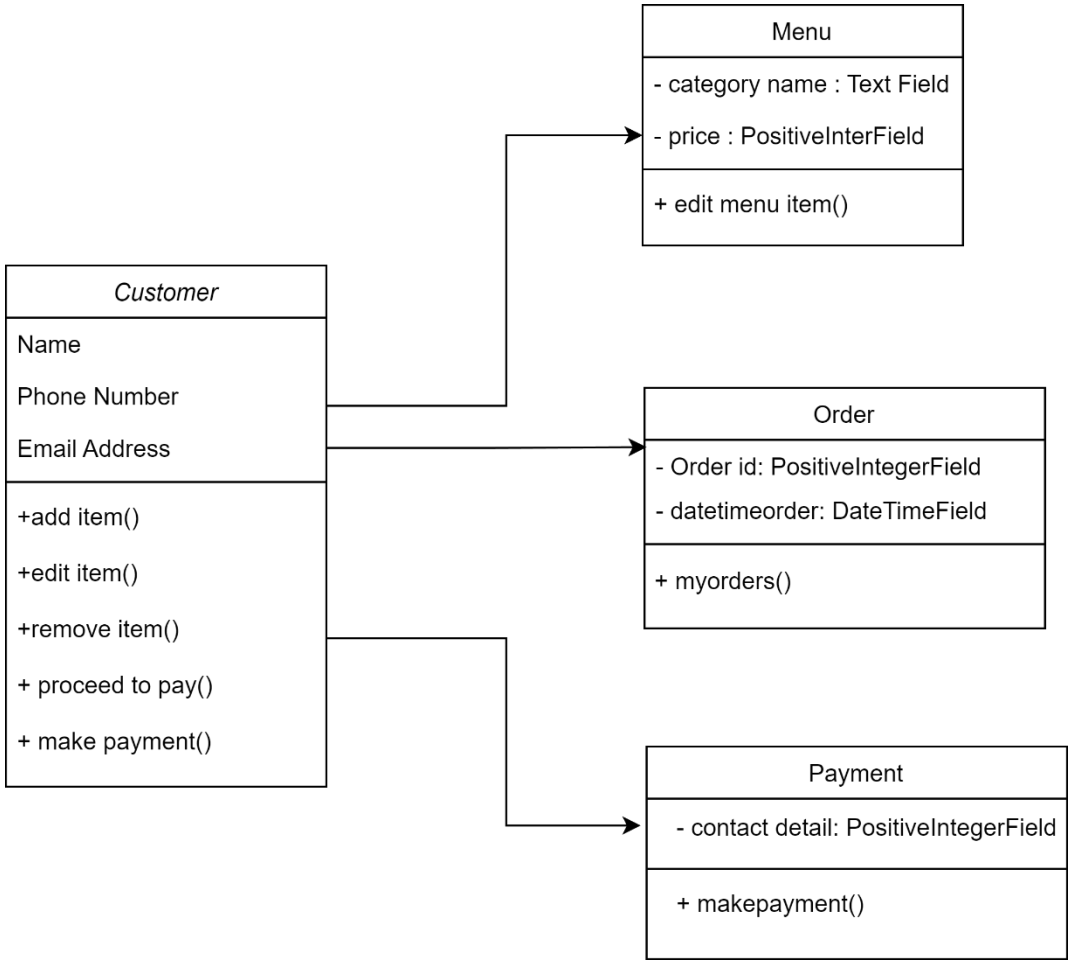

Fig 7.1.1: Class Diagram

7.2 USE – CASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships.

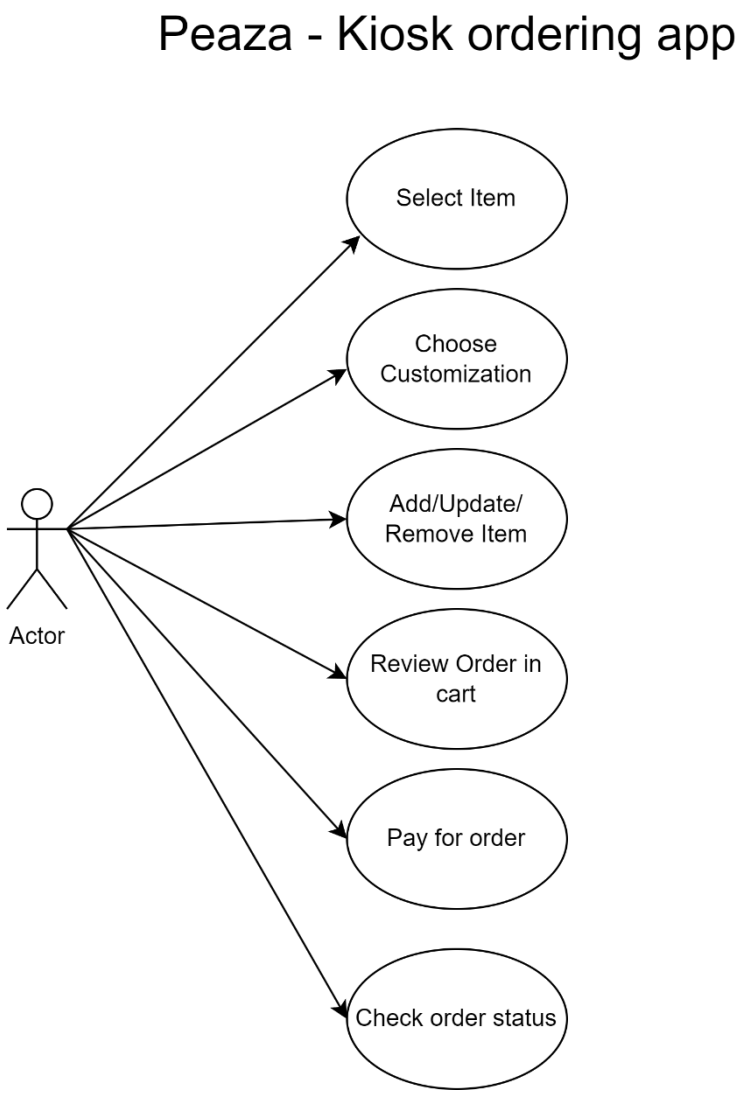

Fig 7.2.1: User's Use-Case Diagram

7.3 SEQUENCE DIAGRAM

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration.

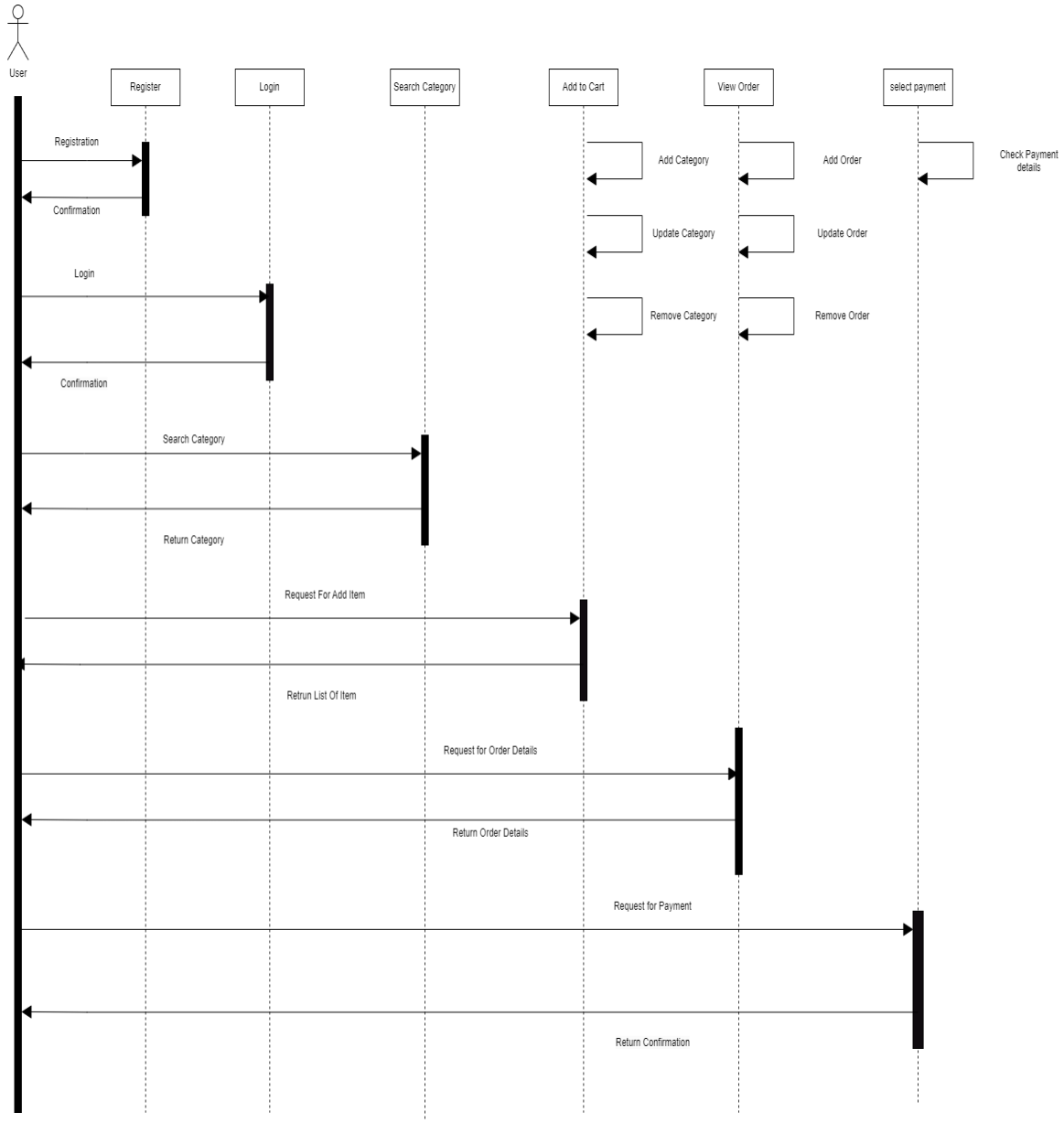

Fig 7.3.1: Sequence Diagram of User

7.4 ACTIVITY DIAGRAM

Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

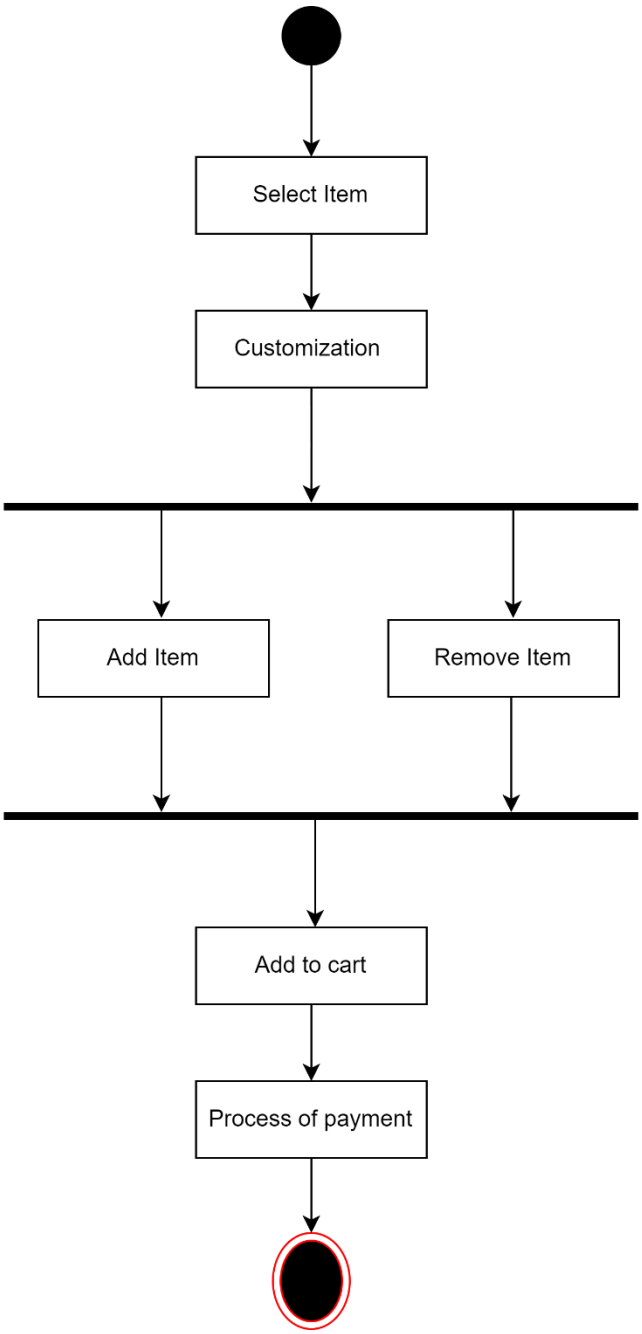

Fig 7.4.1: Activity Diagram of User

7.5 STATE CHART DIAGRAM

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

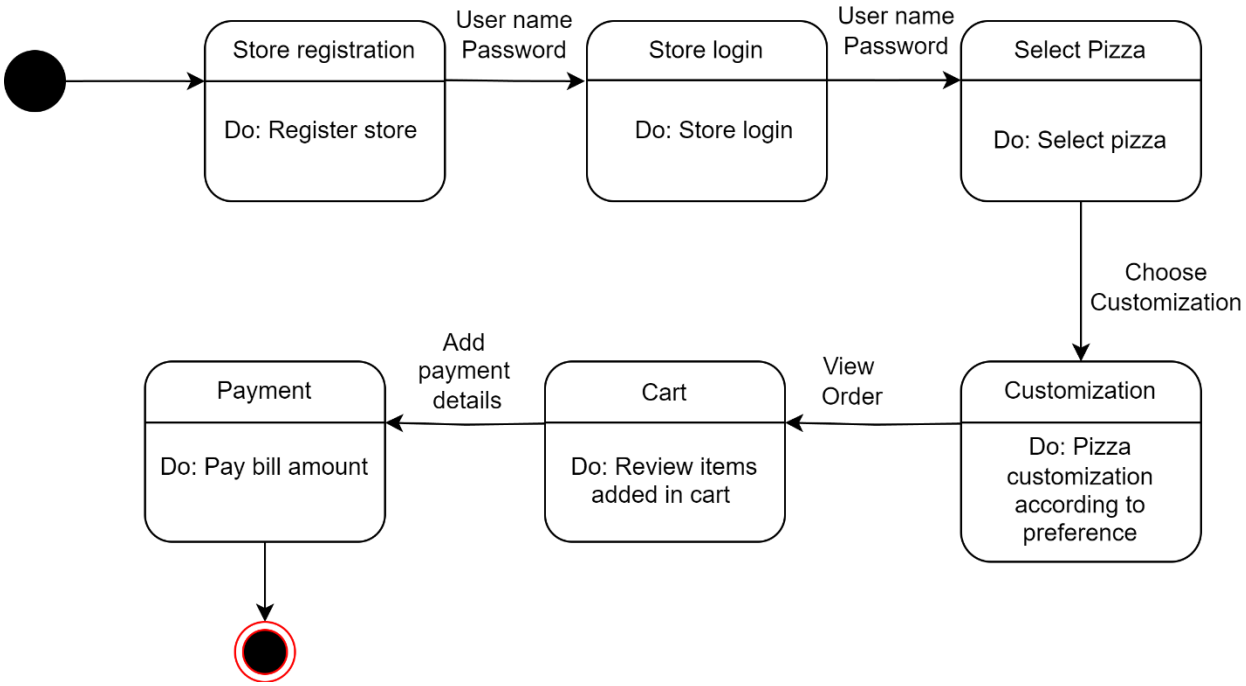

Fig 7.5.1: State Chart Diagram

7.6 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) illustrates the flow of data within a system, depicting how information moves from one process to another, through data stores, and between external entities. It provides a visual representation of the system's processes, data sources, data destinations, and data flows, aiding in understanding system functionality and data interactions.

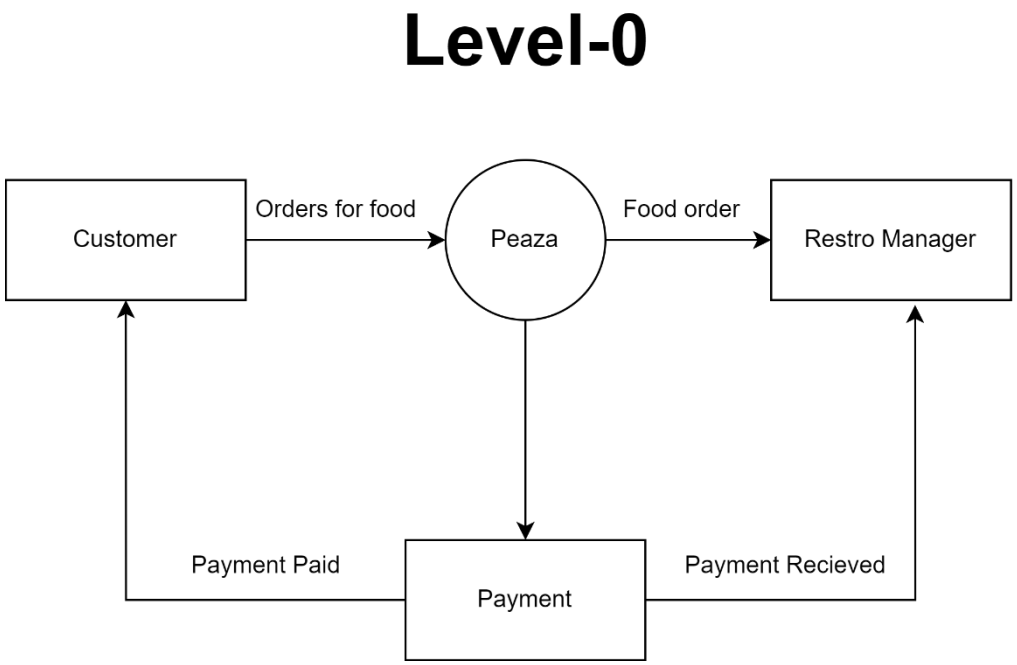

Fig 7.6.1: Data Flow Diagram LEVEL-0

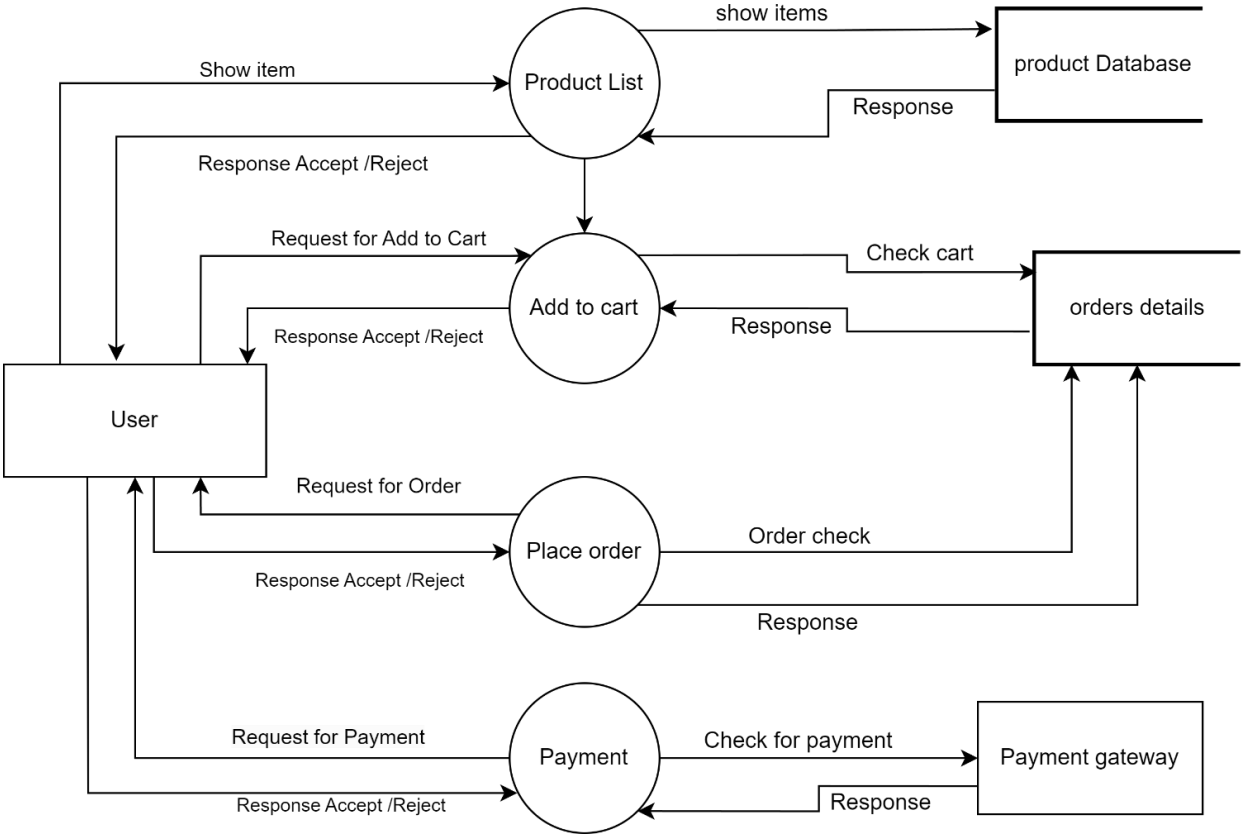

Fig 7.6.2: Data Flow Diagram LEVEL-1 for User

7.7 Entity Relationship Diagram

This model is used to define the data elements and relationship for a specified system. It develops a conceptual design for the database also it develops a very simple and easy to design view of data.

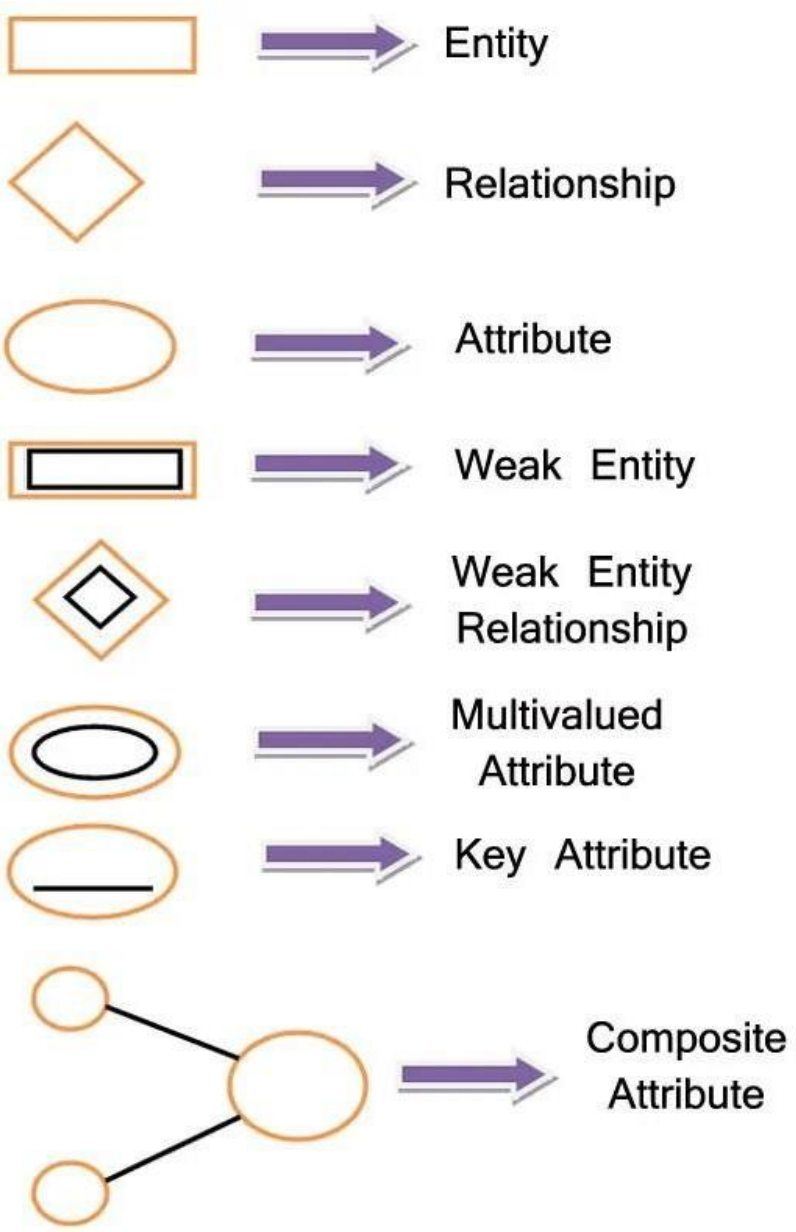

Fig: 7.7.1: ER Diagram Symbols

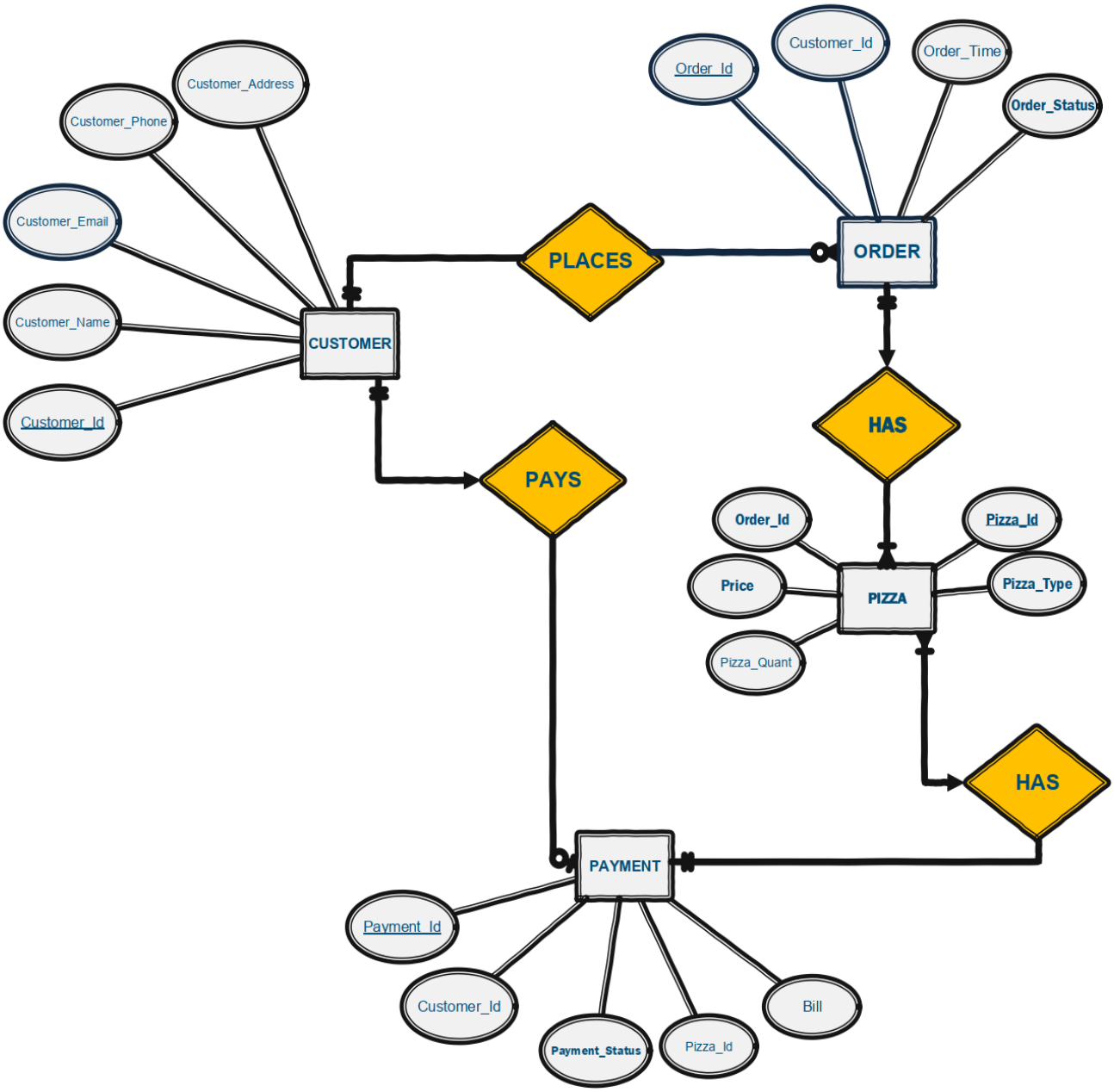

Fig: 7.7.2: ER Diagram

CHAPTER 8

LIMITATIONS AND FUTURE SCOPE

- **Limitation**
- **Future Enhancement**

8.1 LIMITATION

While considerable effort has been invested in ensuring the software's flexibility and user-friendliness, it's essential to acknowledge inherent limitations. While the software offers a wide array of options, certain intricate functionalities couldn't be incorporated due to logistical and technical constraints. Time constraints also played a significant role, limiting the software's dynamism and resulting in the omission of certain features, such as expanding the menu items.

Despite efforts to streamline usability, particularly for non-technical users, it's recognized that individuals unfamiliar with computer technology may encounter initial challenges. To mitigate this, the software includes step-by-step guidance to assist users throughout their interactions. Below is a list of limitations present in the Peaza – Kiosk Ordering App:

- **Device Restriction:** Since Peaza is designed primarily for kiosk use, it may not be accessible through desktop and mobile devices, limiting its reach to customers who prefer ordering remotely.
- **Location Dependency:** Customers can only use Peaza when physically present at the restaurant with access to the kiosk, which may not cater to customers looking for online or delivery options.
- **Scalability Challenges:** Peaza's effectiveness may be limited in handling a large volume of orders during peak times, especially if multiple kiosks are not available or if the system experiences technical issues.
- **Maintenance and Updates:** Managing and updating the kiosk-based system may require additional resources and expertise, impacting the overall cost and efficiency of the project.
- **Learning Curve:** Customers and staff may require time to familiarize themselves with the kiosk interface, potentially leading to initial usability challenges or slower adoption rates.
- **Accessibility Concerns:** The kiosk interface may not fully meet accessibility standards for users with disabilities, requiring additional efforts to ensure inclusivity.

8.2 FUTURE ENHANCEMENT

In a nutshell, it can be said that the future scope of the project revolves around maintaining information regarding:

- We can provide more advanced software for the online food ordering system, including more facilities.
- We will host the platform on online servers to make it accessible worldwide.
- We will integrate multiple load balancers to distribute the loads throughout the system.
- We will implement the backup mechanism for taking regular backups of the codebase and database on different servers.
- We will give options for discounts and coupons at the check-out page.
- We will also provide an option for people who book reservations to select the table number at which they will sit.

The outlined enhancements serve to expand the project's utility and user engagement by establishing comprehensive record-keeping mechanisms. These encompass various facets such as food items, categories, customer details, orders, and confirmations. By meticulously organizing this data, the system not only facilitates smoother operations but also enables insightful analysis for informed decision-making.

Moreover, acknowledging the evolving needs of today's dynamic consumers, there's potential to introduce novel methods within the kiosk food ordering framework. This adaptability ensures the system remains relevant and responsive to emerging trends and user preferences.

By keeping avenues open for future enhancements, the project maintains flexibility to incorporate additional functionalities or features as per user requirements. This proactive approach not only fosters continual improvement but also underscores a commitment to delivering a robust and adaptable solution that meets the evolving needs of its users.

CHAPTER 9

CONCLUSION

- **Conclusion**

9.1 CONCLUSION

The conclusion of the Peaza project report emphasizes its transformative impact on the restaurant industry's food ordering experience. Peaza's introduction has significantly improved the efficiency and customer satisfaction levels within restaurants. Its user-friendly interface has made ordering food a seamless process for customers, reducing the time spent waiting in lines or for staff assistance. The extensive menu options displayed on the kiosk provide customers with a wide range of culinary choices, enhancing their dining experience.

One of Peaza's standout features is its customization menu, allowing customers to personalize their orders according to their preferences and dietary requirements. This level of customization ensures that each order is tailored to meet the individual needs and tastes of the customer, contributing to higher levels of customer satisfaction and loyalty.

Although Peaza is designed for kiosk usage, limiting its accessibility to desktop and mobile devices, its impact on in-store operations cannot be understated. By streamlining the ordering process and digitizing it, Peaza has reduced errors, minimized wait times, and potentially lowered the need for additional workforce during peak hours. This not only benefits restaurant owners in terms of operational efficiency but also enhances the overall dining experience for customers.

Looking ahead, continuous improvements and adaptations will be vital to address Peaza's limitations, such as scalability challenges and accessibility concerns for customers who prefer online or delivery options. However, the positive outcomes achieved through Peaza's implementation demonstrate its potential to further revolutionize the food ordering landscape, making it an invaluable asset for restaurants striving to deliver exceptional service and customer satisfaction.

BIBLIOGRAPHY

REFERENCES

- List of Web References

[1] https://react.dev/learn

[2] https://www.lovelycoding.org/online-pizza-ordering-system-project-for-final-year/

[3] https://getbootstrap.com/docs/5.3/getting-started/introduction/

[4] https://www.w3schools.com/nodejs/nodejs_intro.asp

[5] https://www.mongodb.com/

[6] https://mui.com/core/

[7] https://razorpay.com/