Shopper Connect – Store at your Door

Prepared By

Aryan Sengar (2020 BTech CSE 013)

Vaibhav Tomar (2020BTech CSE 080)

Faculty Mentor

Dr. Amit Sinhal

Department of Computer Science Engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University Jaipur

April 2023

Content

Certificate

This is to certify that the project report entitled "**Shopper Connect – Store at your Door**" is submitted to the Department of Computer Science and Engineering, JK Lakshmipat University in partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a record of bona fide work carried out by Vaibhav Tomar, (2020BTechCSE080), Aryan Sengar, (2020BTechCSE013), under my supervision and guidance.

All help received by their team from various sources have been duly acknowledged. No part of this report has been submitted elsewhere for award of any other degree.

Dr. Amit Sinhal

Professor, Department of CSE, IET

JK Lakshmipat University, Jaipur

Acknowledgment

We have completed this project under the guidance and supervision of **Prof Amit Sinhal**, JK Lakshmipat University. I will be failed in my duty if I do not acknowledge the esteemed scholarly guidance, assistance, feedback and knowledge I have received from them towards the fruitful and timely completion of this work.

We express our deepest thanks to **Dr Dheeraj Sanghi,** Vice Chancellor, JK Lakshmipat University, and **Dr. Renu Jain**, Director, Institute of Engineering and Technology, JK Lakshmipat University for their constant support, encouragement, and g dance. We also acknowledge with a deep sense of reverence, our gratitude towards our parents for their direct or indirect support during the entire course of this project.

Sincerely,

Vaibhav Tomar

Aryan Sengar

Abstract

This project is a MERN stack web application that aims to connect users with nearby shops based on their location and category of interest. The application allows users to search for shops, view their details such as name, location, working hours, contact information, and user reviews. Users can also leave reviews and ratings for the shops they have visited. The project uses various technologies and libraries like React Leaflet for map integration, Redux for state management, and Mongoose for database management. The application is designed to be responsive and optimized for both desktop and mobile devices. Overall, this project aims to provide users with a convenient and user-friendly platform to discover and connect with nearby shops while also providing a useful tool for shop owners to manage their business. the MERN stack is a powerful and flexible solution for developing modern web applications. MongoDB provides a scalable and flexible database, Express.js provides a simple yet robust web framework, React.js allows for the creation of dynamic and efficient user interfaces, and Node.js allows for server-side JavaScript execution. The combination of these four technologies allows developers to build fast and scalable web applications with ease. With the MERN stack, developers can build web applications that are responsive, performant, and feature rich.

List of Figures

Introduction

1.1 Problem statement

The rise of e-commerce has made it difficult for brick-and-mortar shops to attract and retain customers, while consumers are seeking more efficient ways to connect with local shops. The challenge is to create a user-friendly platform that connects people with nearby shops and allows businesses to promote themselves to a wider audience. This problem is relevant to both small businesses and large corporations seeking to tap into new markets.

In an effort to address this problem, our team is proposing to create a platform that connects sellers and buyers, thereby reducing the time it takes for individuals to locate and purchase goods from nearby shops. Our idea is to develop a user-friendly platform where sellers can list their products for sale and customers can easily find and purchase goods from shops near them. This platform would facilitate transactions between sellers and buyers and provide accurate information about the location of shops, hours of operation, and the products available for purchase.

1.2 Objective

- The objective of the proposed platform is to make it easy and convenient for consumers to find and connect with shops in their local area.
- The platform aims to benefit both consumers and business owners by providing a seamless way to discover nearby shops and expand the customer base for small shop owners.
- The platform is intended to streamline the work process for both consumers and businesses, making it easier and more efficient to purchase and sell products.
- By facilitating connections between consumers and local businesses, the platform aims to promote community engagement and support local economies.
- Ultimately, the goal of the platform is to create a mutually beneficial ecosystem that fosters growth and prosperity for both consumers and businesses in the local area.

1.3 Model Adopted

Our software development team has adopted the Agile Model for the development of our location-based e-commerce platform, which is a common choice for startups due to its iterative development approach, early customer feedback, and intensive communication. Each iteration in the Agile Model involves several weeks of development and results in a complete working version of the software. The model focuses on delivering software functionality quickly, with less attention paid to detailed documentation like extensive requirement specifications and architecture descriptions. While this leads to fast product development, it can make maintenance more complex as it takes more time to identify and fix errors. Agile Model is all about close collaboration with the development team and customers. At the end of each iteration, stakeholders review the progress of the development and prioritize tasks that need to be completed in the near future. Frequent software releases are a characteristic of the Agile Model, as it involves continuous software improvement, bug fixes, updates, and the addition of new features. By breaking the project down into smaller, manageable pieces, the team can work efficiently and make continuous improvements to the platform throughout the development process.

Key Features of Agile Model

1. **Iterative development:** The Agile Model breaks down the project into smaller, manageable pieces and develops them in short development cycles, typically two to four weeks in length.

2. **Flexibility and adaptability:** The Agile Model allow for changes to be made to the project requirements and scope as needed, enabling the development team to adapt and adjust the project plan accordingly.
3. **Continuous testing and feedback:** The Agile Model involve regular testing and feedback from stakeholders to ensure that the software meets the needs of its users.
4. **Collaboration:** The Agile Model emphasizes close collaboration between the development team, stakeholders, and customers throughout the development process.
5. **Working software:** The Agile Model focuses on delivering software functionality quickly, with less attention paid to detailed documentation.
6. **Customer satisfaction:** The Agile Model prioritizes customer satisfaction by delivering software that meets the needs of its users.
7. **Team ownership:** The Agile Model encourages team ownership of the development process and enables team members to make decisions and take ownership of their work.

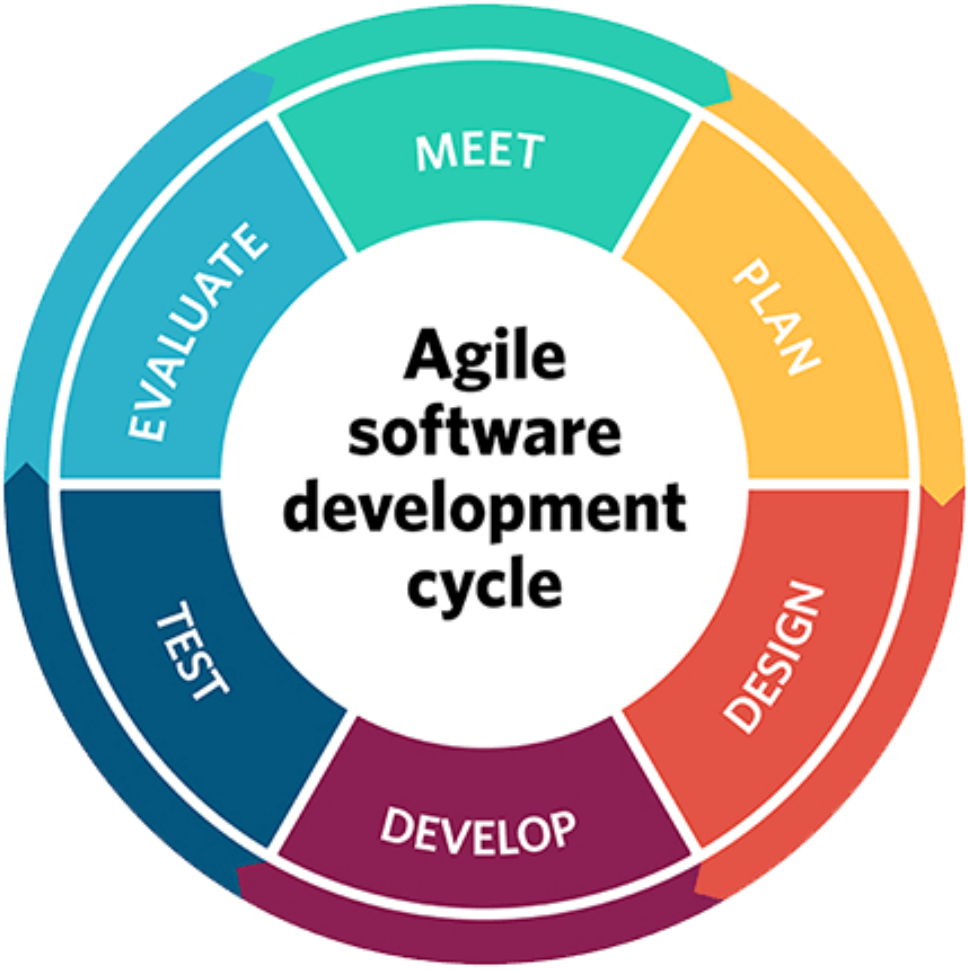

Figure 1 Agile Model

Requirement Engineering and Analysis

2.1 Feasibility Study

Feasibility Study is an assessment of the viability of a proposed project or solution, which helps to determine if it is worth pursuing or not. It is a systematic analysis of the potential benefits and drawbacks of a project, which helps to identify the best course of action to take the feasibility study typically involves a thorough analysis of the technical, economic, operational, and legal aspects of a proposed project. It considers factors such as market demand, competition, funding availability, resource availability, technological feasibility, and legal compliance. The results of the feasibility study are used to make an informed decision on whether to proceed with the project or not.

Figure 2 Context Diagram

The major feature discussed for the current scope of project were gathering the details of customer and store entry for the store information and more information about item, stock, feedback payment etc. The feasibility was further tested by analysing the requirement specified by seller and customer.

Benefits of our product and services

- Store Management
- Stock Management
- Order Report
- Customer List
- Feedback
- Login Credentials
- Payment Gateway
- Seller list

2.2 Information Modeling

Information modeling is a technique used to conceptualize and illustrate data and its relationships to gain a better understanding of complex information structures. It involves the creation of models that represent the entities, attributes, and connections between them that are crucial for a specific system or application. The purpose of information modeling is to identify the different types of data required for a system and how they relate to one another, which is essential for database design and software development.

Data Flow Diagram

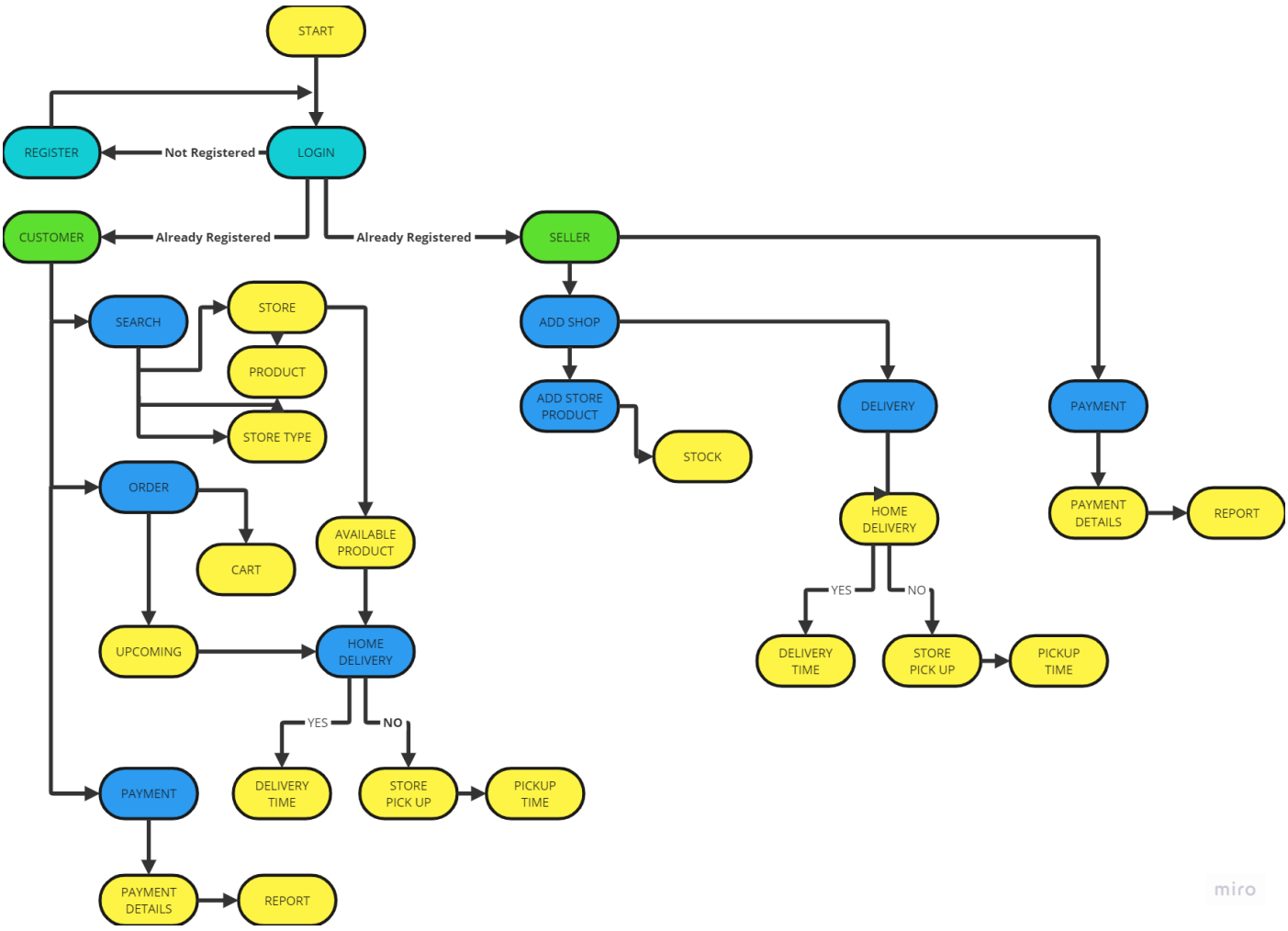

Figure 3 DFD LEVEL 1

2.2.1 Entity Relationship Diagram

ER diagram, also known as Entity-Relationship diagram, is a graphical representation of entities and their relationships to each other in a database. It is a tool used by developers, analysts, and designers to model the data in a system or application. The ER diagram consists of entities, attributes, and relationships.

Entities are objects or concepts that exist independently and can be distinguished from other objects. For example, in a hospital management system, entities could be patients, doctors, nurses, and wards. Attributes are characteristics of an entity that help describe it, such as the name, ID, gender, and address of a patient.

In this ER diagram, we have identified four entities: Shop, Owner, Customer, and Order.

Shop: This entity represents the physical shop that the owner owns. It has attributes like Shop_ID, Shop_Name, Shop_Location, and Shop_Category. Shop_ID is the primary key of this entity. Shop_Category attribute denotes the type of shop, such as clothing store, grocery store, or restaurant.

Owner: This entity represents the owner of the shop. It has attributes like Owner_ID, Owner_Name, Owner_Email, and Owner_Phone. Owner_ID is the primary key of this entity.

Customer: This entity represents the customer who wants to connect with nearby shops. It has attributes like Customer_ID, Customer_Name, Customer_Email, and Customer_Phone. Customer_ID is the primary key of this entity.

Order: This entity represents the order placed by the customer. It has attributes like Order_ID, Shop_ID, Customer_ID, Order_Date, and Order_Amount. Order_ID is the primary key of this entity. Shop_ID and Customer_ID are foreign keys, as they reference the primary keys of the Shop and Customer entities, respectively.

The relationships between these entities are as follows:

A Shop is owned by an Owner, and an Owner can own multiple Shops. This is a one-to-many relationship.

A Customer can place multiple Orders, and an Order can be placed by only one Customer. This is a one-to-many relationship.

A Shop can receive multiple Orders, and an Order can be placed for only one Shop. This is a one-to-many relationship.

Overall, this ER diagram provides a clear understanding of the entities and their relationships in the context of our problem statement.

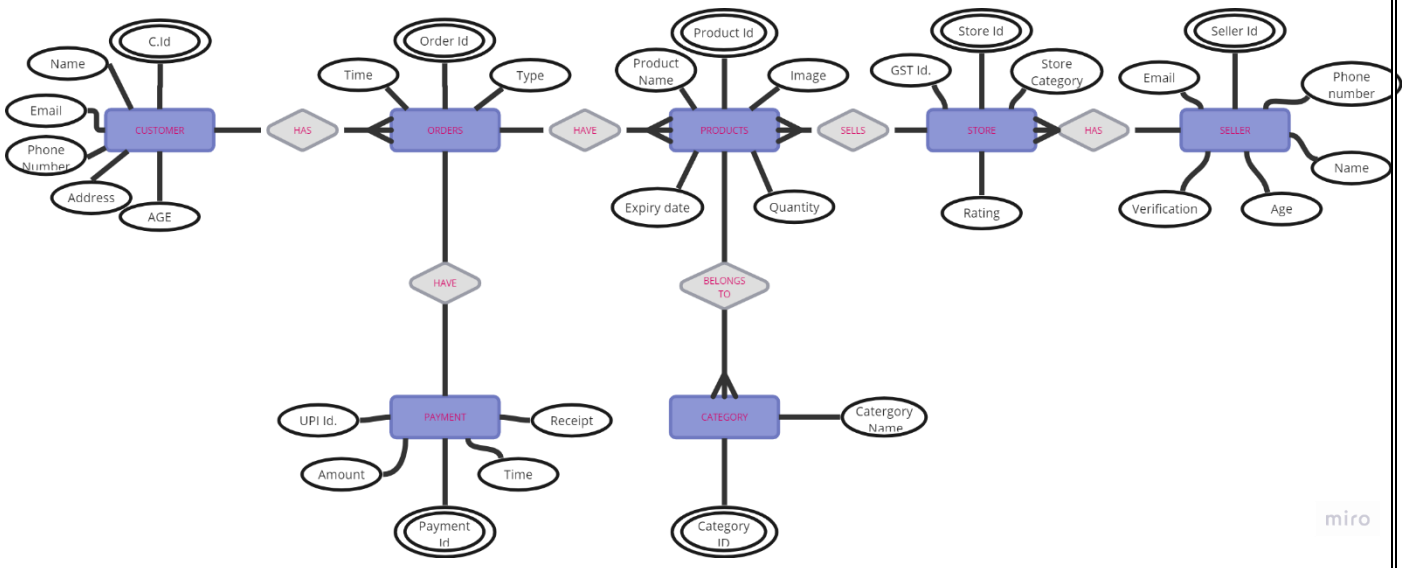

Figure 4 ER Diagram

2.3 Software requirements

2.3.1 Introduction

Technology refers to the tools, techniques, and systems that are created and used to solve problems, achieve goals, or perform tasks more efficiently. It encompasses a wide range of fields, including information technology, engineering, biotechnology, and more.
Technology has had a profound impact on society, transforming the way we communicate, work, learn, and even think. It has enabled us to accomplish things that were once thought impossible and has brought people closer together, breaking down barriers of distance and culture.
The history of technology can be traced back to prehistoric times, with the invention of simple tools and weapons. Over time, technology has evolved and become increasingly complex, with new innovations and advancements being made every day. Today, technology is an essential part of modern life and plays a critical role in shaping the future of our world.

MongoDB: MongoDB is a popular NoSQL document-based database that stores data in flexible, JSON-like documents. It is an open-source, scalable, and high-performance database that is used for modern web applications. MongoDB provides a flexible data model that can easily handle unstructured data and provides powerful querying and aggregation capabilities.

Express.js: Express.js is a web application framework for Node.js that provides a simple, yet powerful set of features for building web applications and APIs. It provides a robust set of HTTP utilities and middleware for creating HTTP servers and handling requests and responses. Express.js is designed to be lightweight and flexible, allowing developers to easily build scalable and modular applications.

React: React is a popular frontend JavaScript library for building user interfaces. It allows developers to create reusable UI components that can be easily composed together to create complex interfaces. React uses a virtual DOM to efficiently update the user interface in response to changes in data. It is widely used for building single-page applications, progressive web apps, and mobile applications.

Node.js: Node.js is a server-side JavaScript runtime environment that allows developers to build scalable and high-performance applications. It is built on top of the V8 JavaScript engine and provides a non-blocking I/O model that makes it ideal for building real-time applications. Node.js provides a rich set of built-in modules and packages that make it easy to build web servers, network applications, and command-line tools.

2.3.2 Functional requirements

Functional requirements for the software could include user registration and login, shop registration, shop search, and product/service listings.

2.3.3 Non-Functional requirements

Non-functional requirements for the software could include scalability, reliability, security, user-friendliness, and performance.

2.3.4 Interface requirement & Design constraint

Interface requirements for the software could include a user-friendly and responsive interface for both desktop and mobile devices. Design constraints could include the need for the software to be developed using specific technologies such as the MERN stack, as well as the need for the software to be scalable, maintainable, and secure.

Software Design

3.1 Architectural Design

Architectural design is the process of defining the overall structure and organization of a software system. It involves identifying the components of the system, their relationships, and the interactions between them. The goal of architectural design is to create a high-level view of the system that is understandable, maintainable, and scalable. This design typically includes decisions on technologies to be used, system interfaces, and major structural components. The architectural design serves as a blueprint for the development of the software system.

The architectural design and planning for our product is explained with help of various diagrams as listed below.

3.1.1 Flow Chart

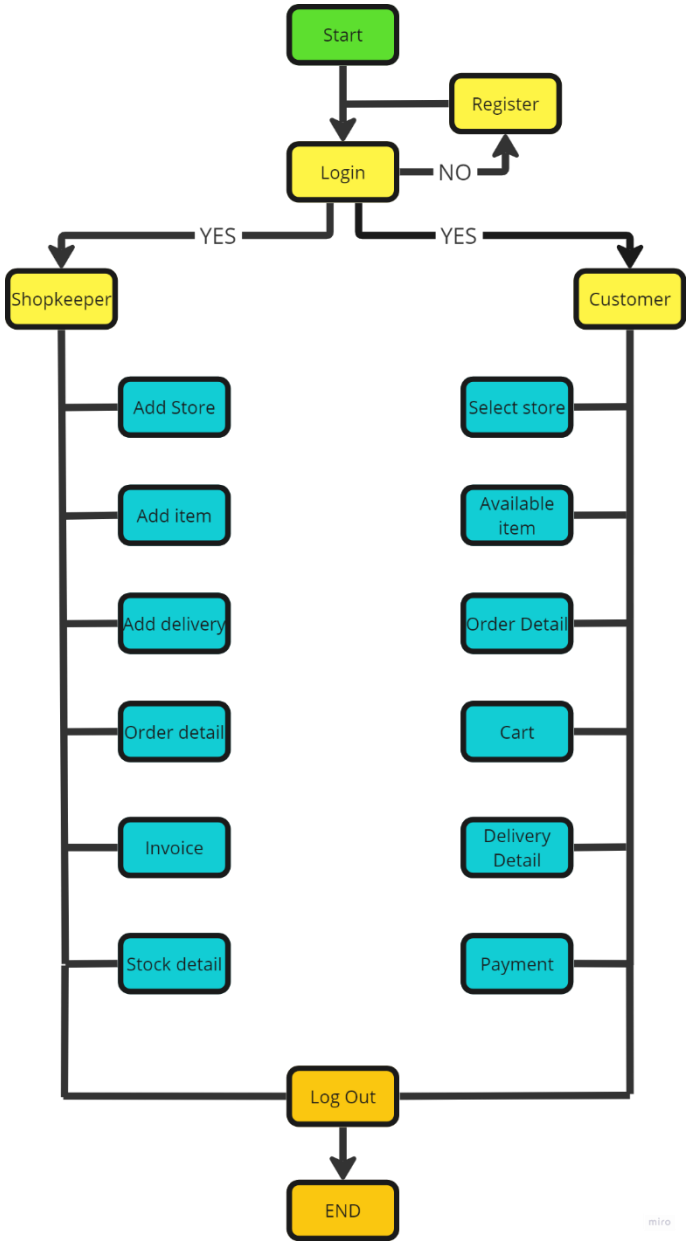

Figure 5 Flow Chart

3.1.2 Data flow Diagram Level - 2

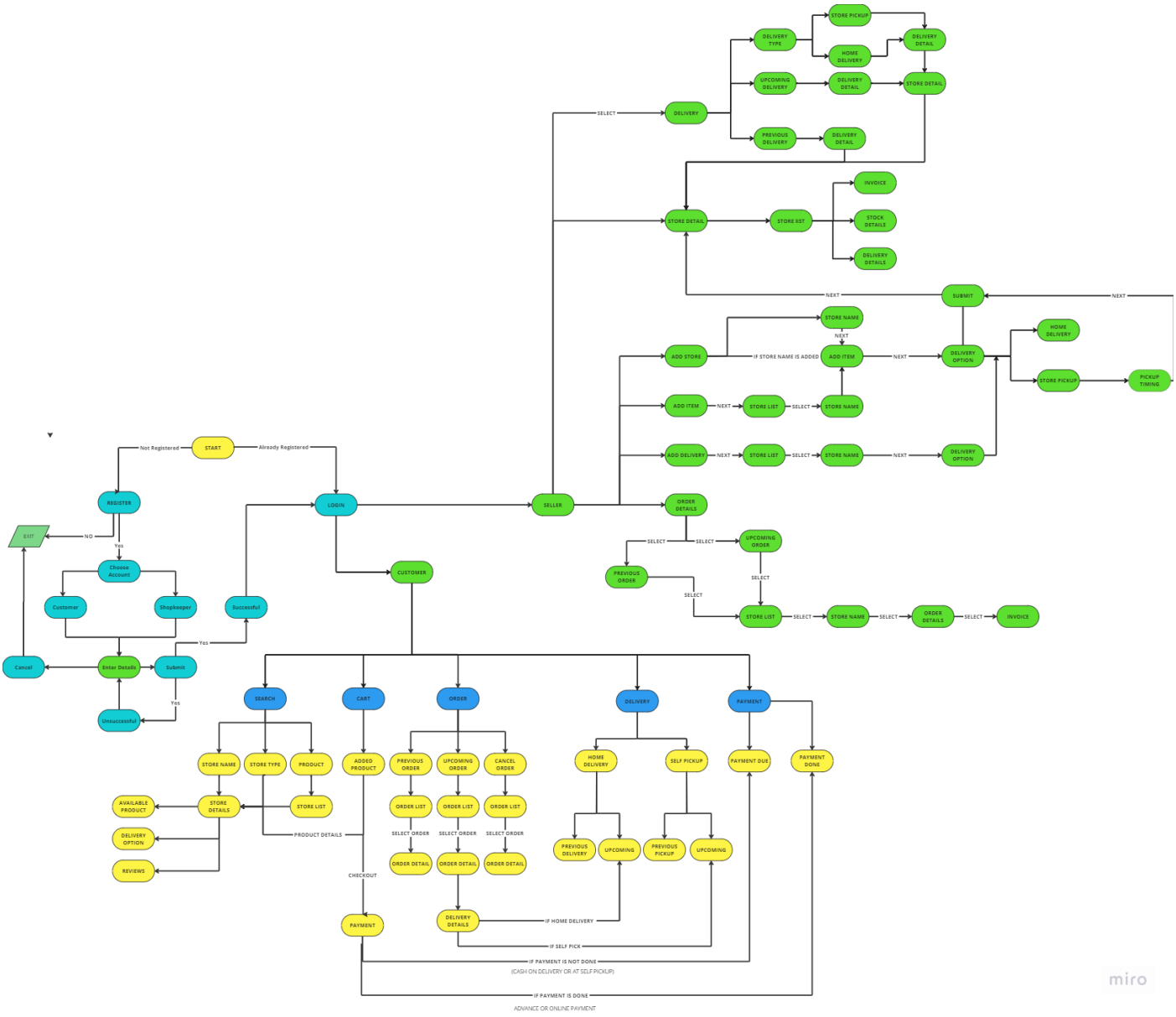

Figure 6 Data flow Diagram Level 2

3.1.3 Module Structure

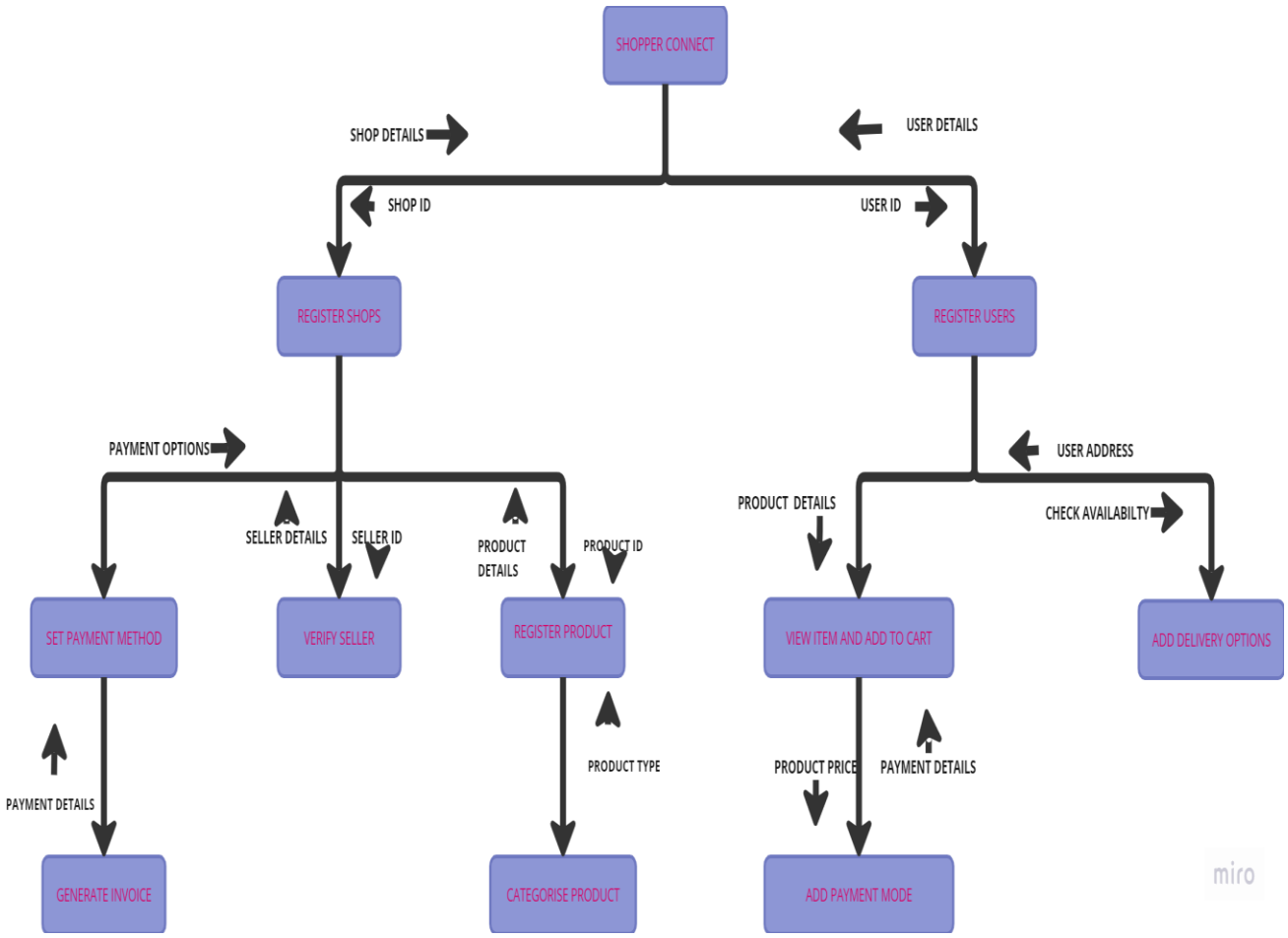

Figure 7 Module Structure

3.2 UML Diagram

Unified Modeling Language (UML) is a visual modeling language used for software development. It is used to specify, visualize, construct, and document the artifacts of a software system. UML diagrams are a set of graphical notations used to represent different aspects of a system, such as its structure, behavior, and interactions. UML diagrams include class diagrams, use case diagrams, sequence diagrams, activity diagrams, state machine diagrams, and more. These diagrams are used to communicate and understand the system design among team members, stakeholders, and developers.

3.2.1 Class Diagram

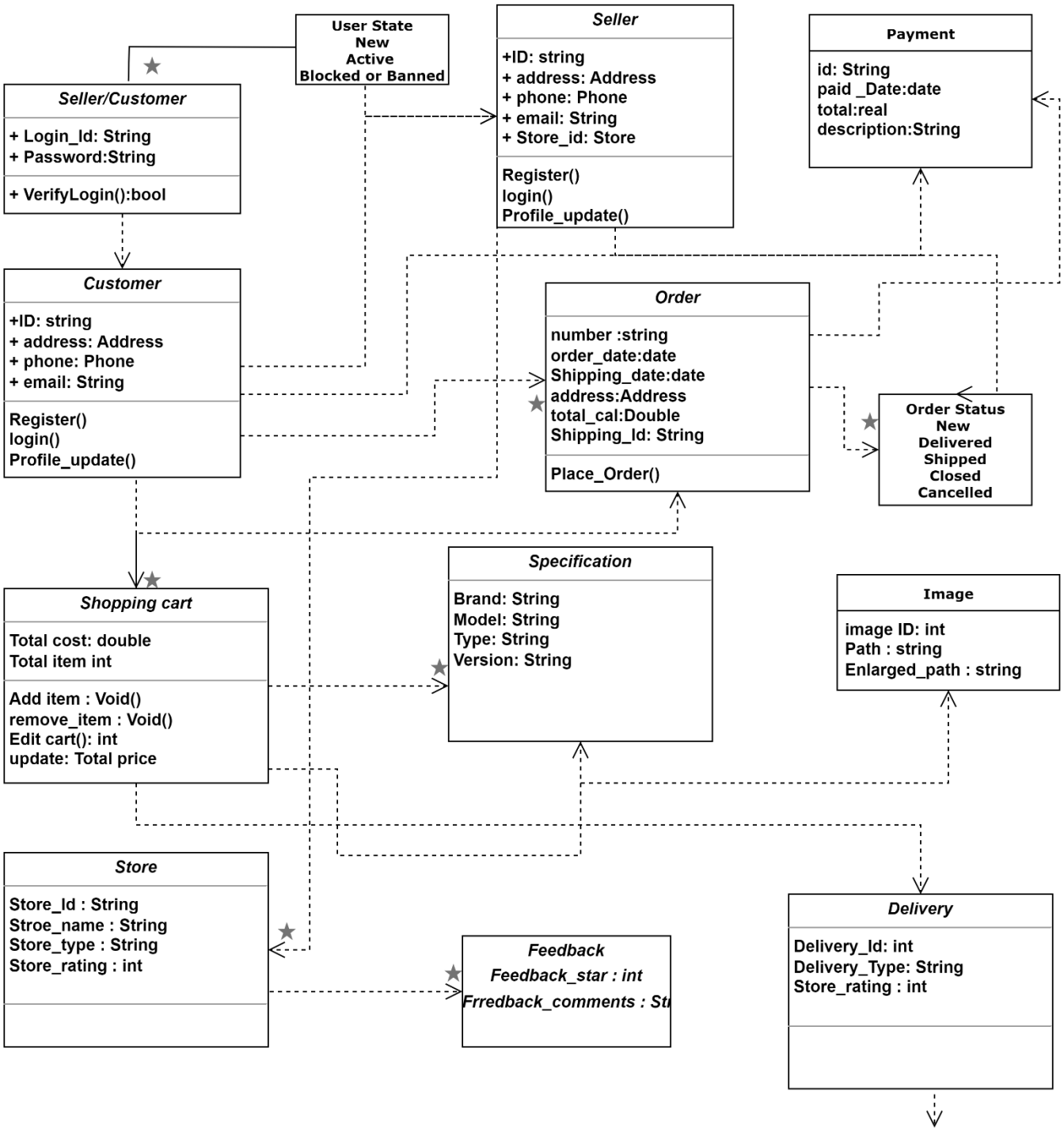

Figure 8 Class Diagram

3.2.2 Activity Diagram

a) User Side

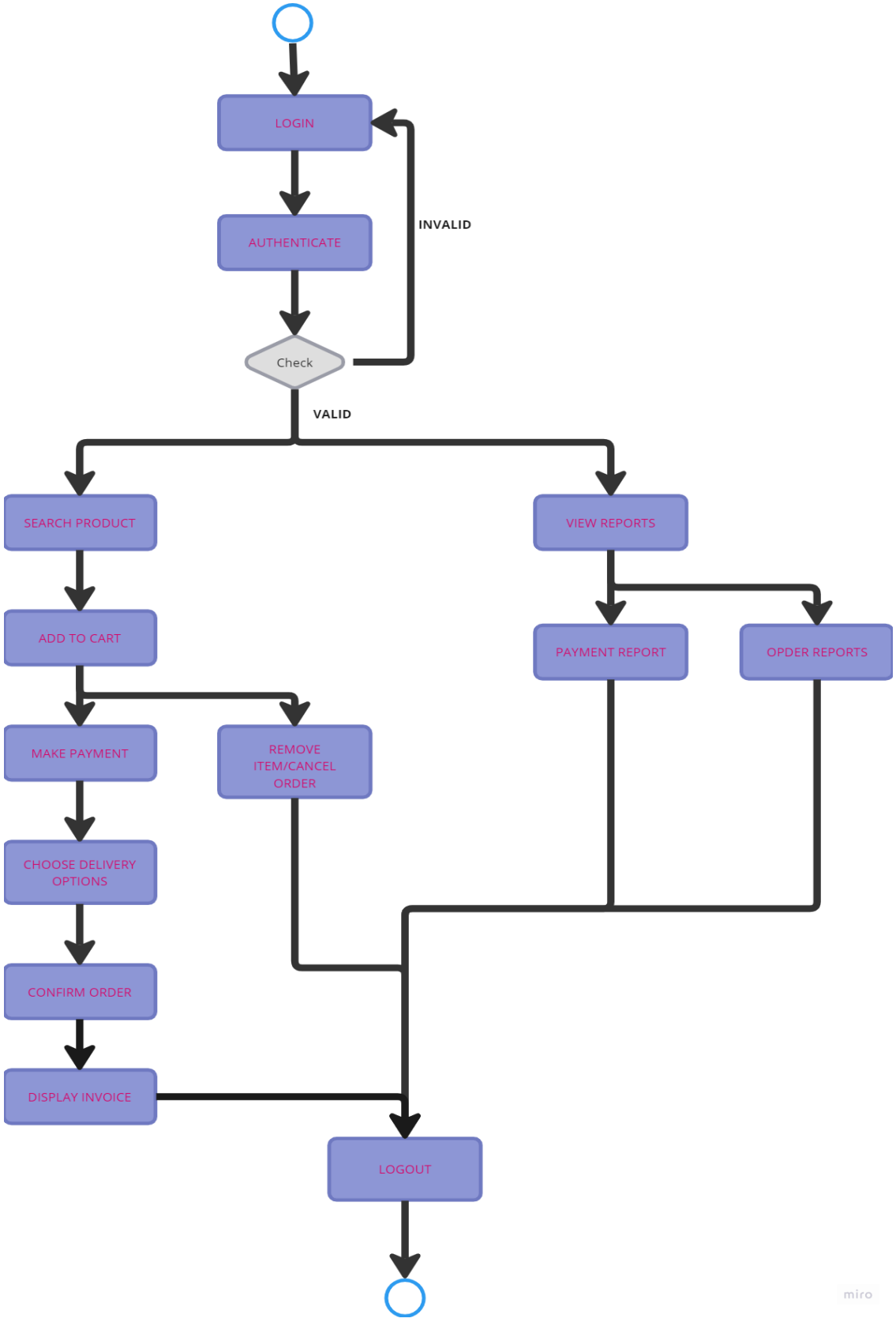

Figure 9 Activity Diagram (User Side)

b) Seller Side

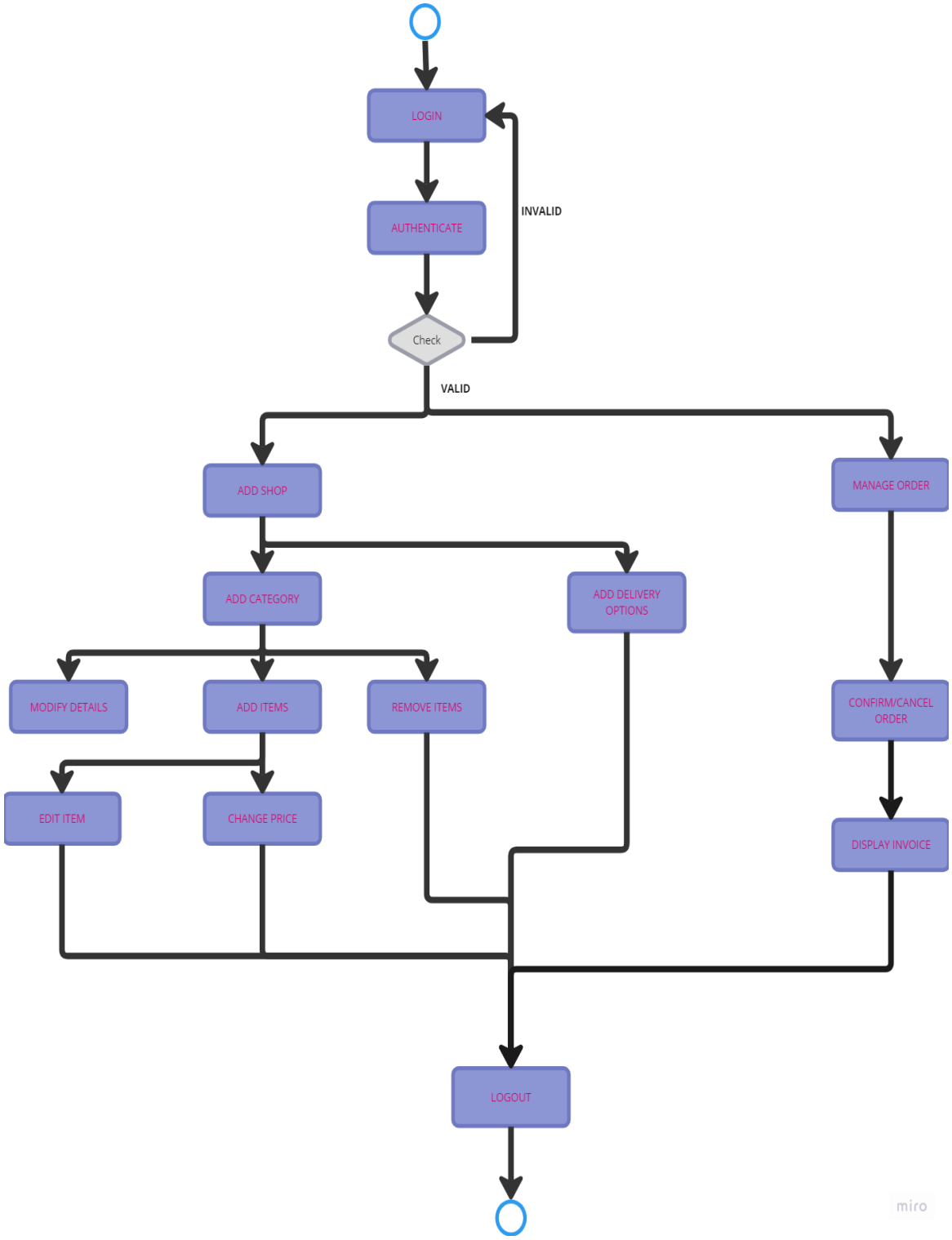

Figure 10 Activity Diagram Seller Side

3.2.3 Use case Diagram.

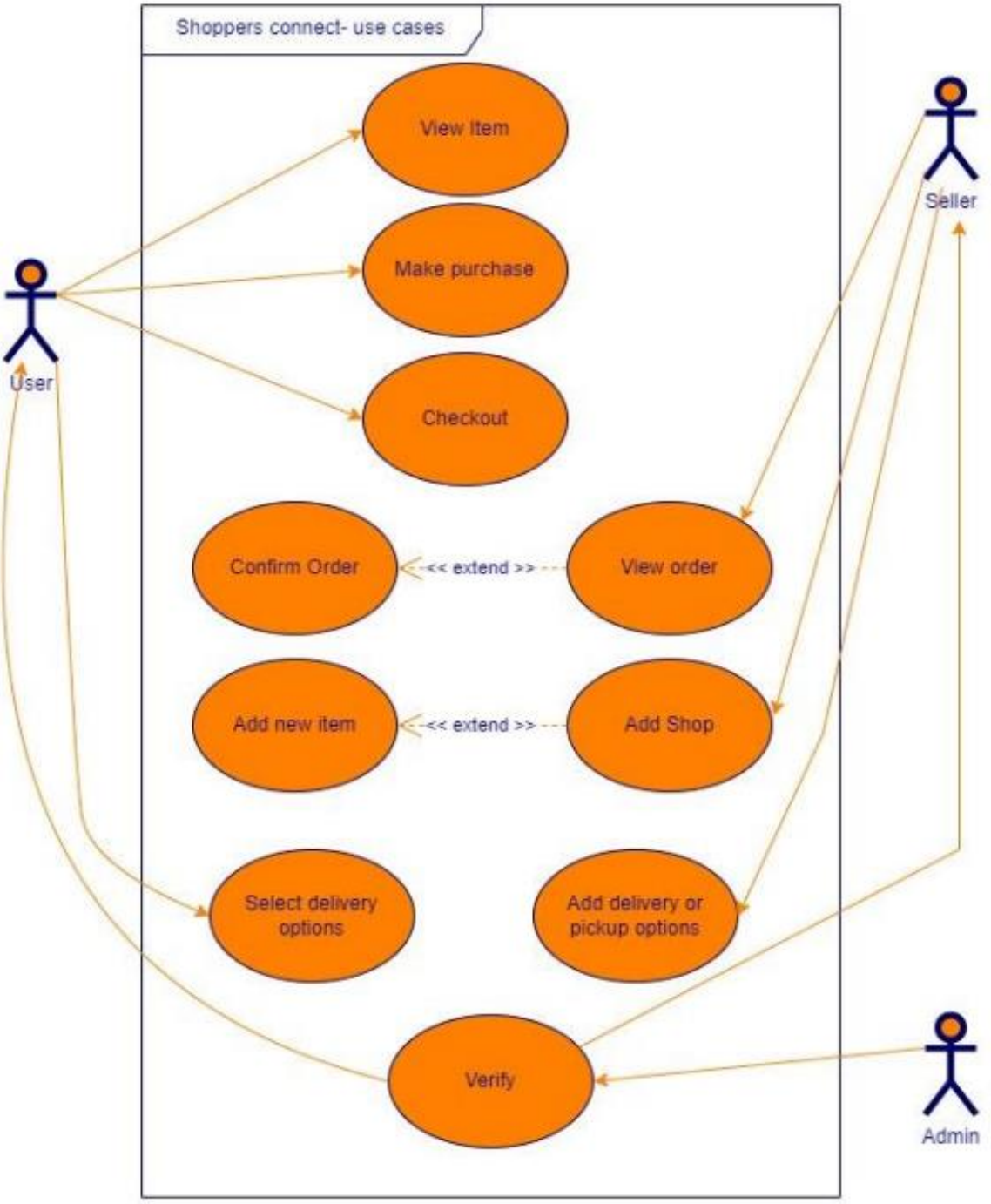

Figure 11 Use Case Diagram

3.2.4 Sequence Diagram

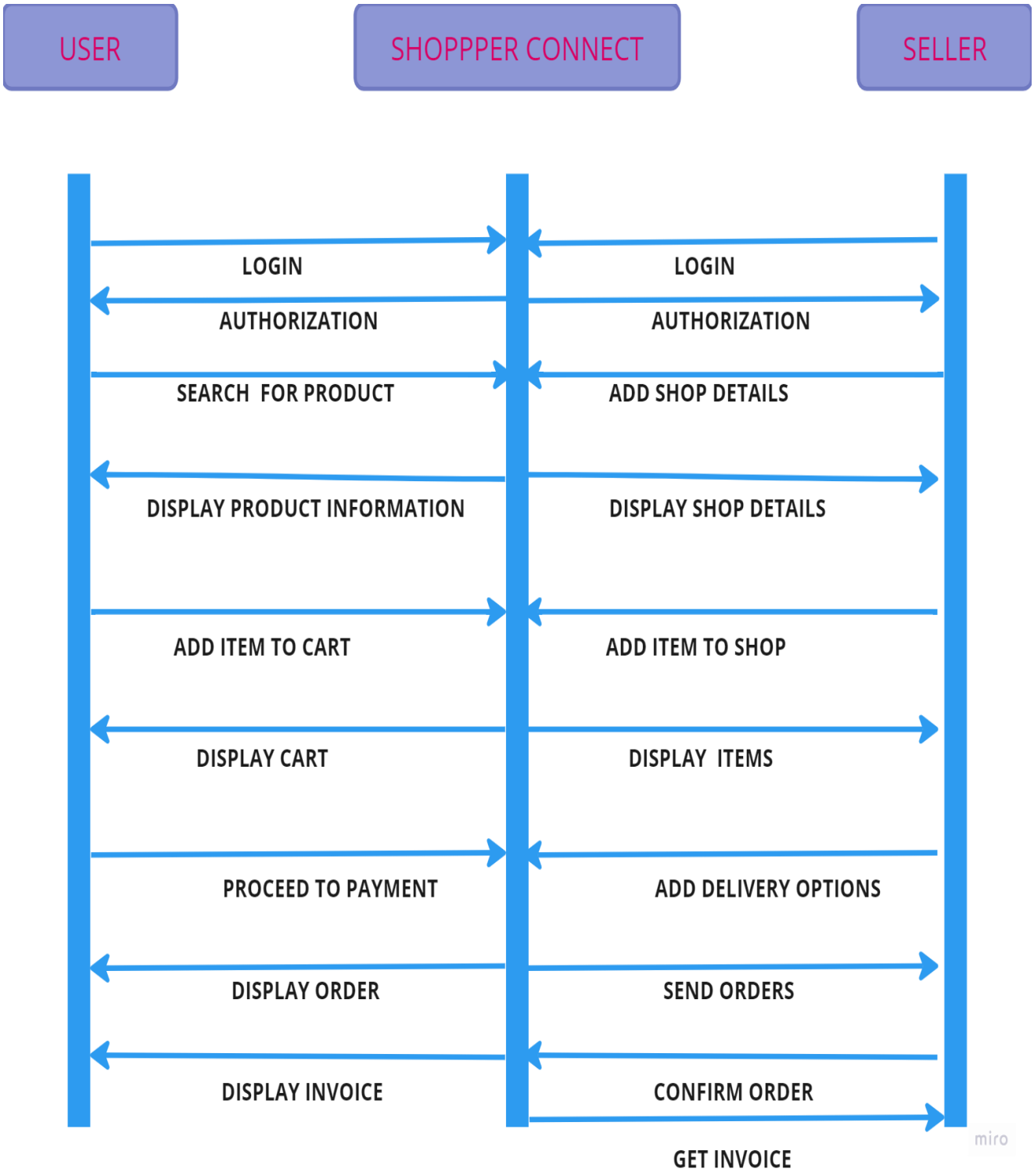

Figure 12 Sequence Diagram

3.3 UI &UX Development

- Finalizing Display Size
- Overall Screen width 1440px
- Working area screen width
- Can be referred to as the minimum overall screen width 1140 px
- Major work area :<1140 px (generalized)
- 1024px in most precise cases

Based on the above specification, the current version of the platform will be suitable for:

- Computer
- Laptop
- Another big screen

3.3.1 Finalizing Display Size

3.3.2 Finalization of Color palette for Platform

F ***Figure 14 Colour Palette*** *ı*

Reasons for choosing this palette:

- Color psychology:
- Blue being a color of shorter wavelength is considered cool and comforting. Symbolizes factors like calm, money, luck, motivation, growth etc.
- Being connected to nature, the color also gives a feeling of refreshens.
- B Rather than using proper blue more comforting shades are used to reduce eye strain and maximize screen time.
- The color palette is compatible with dark mode for future scope.

3.3.3 Platform Logo Selection

Figure 15 Logo of Website

3.3.4 Design prototype

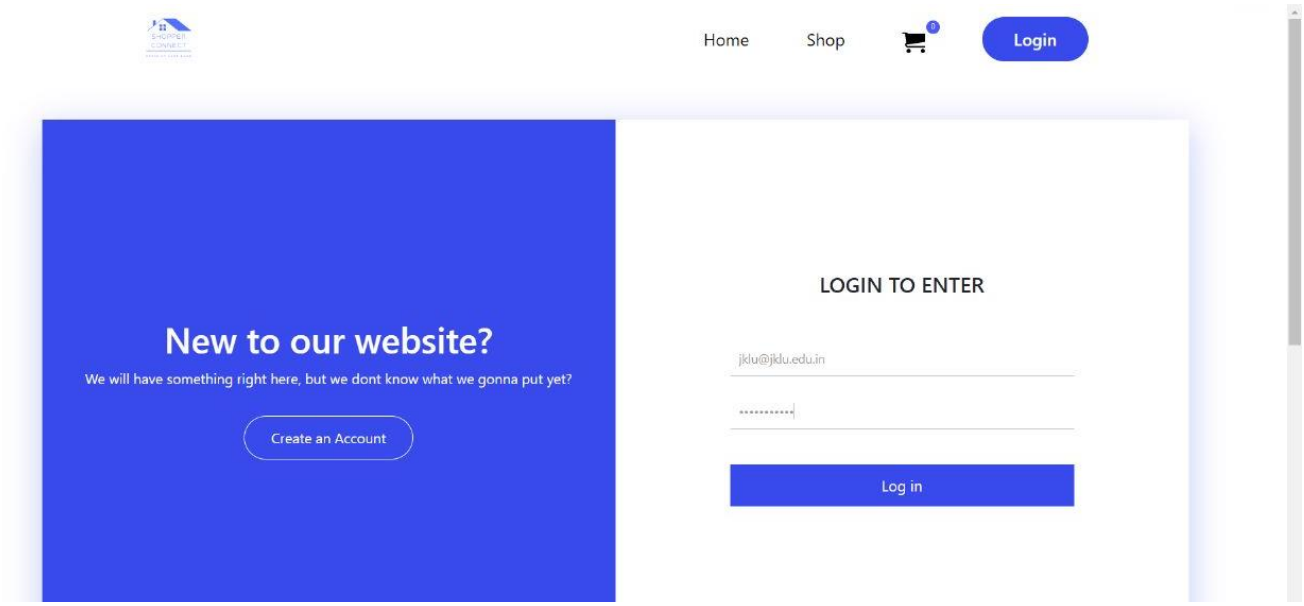

Figure 16 Website Login

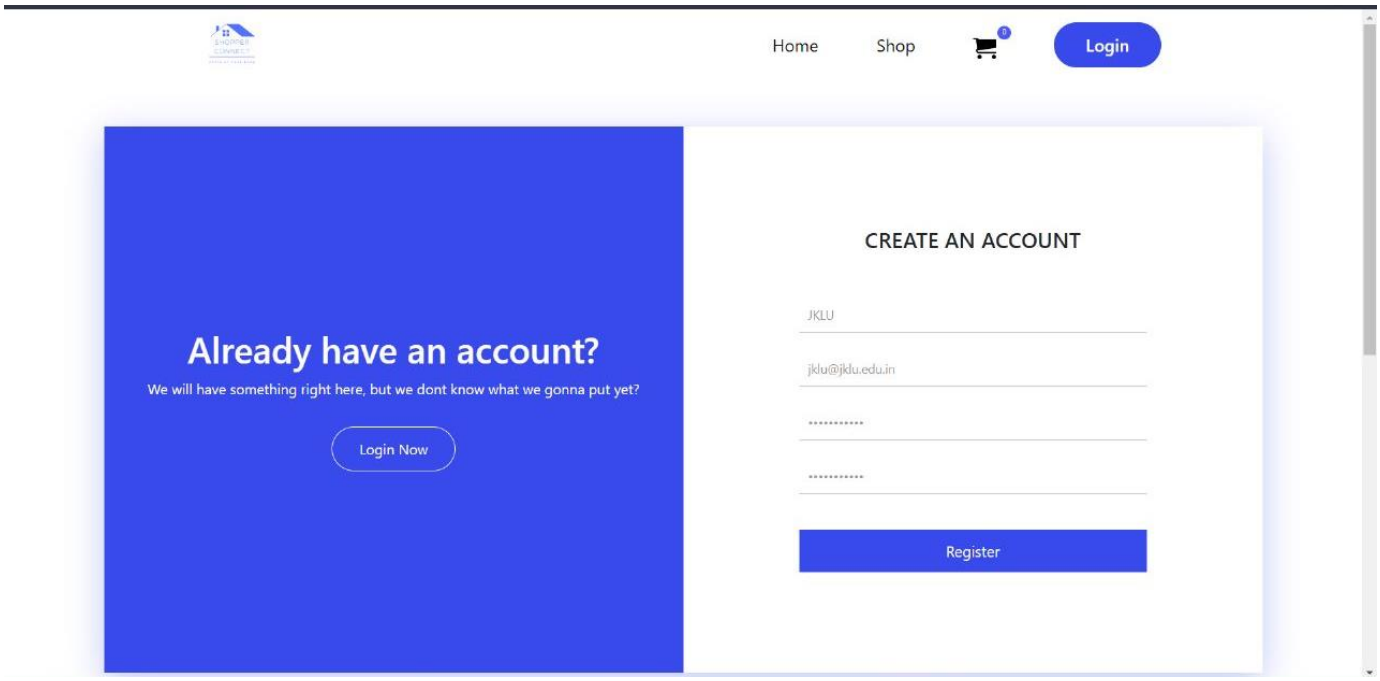

Figure 17 Register Website

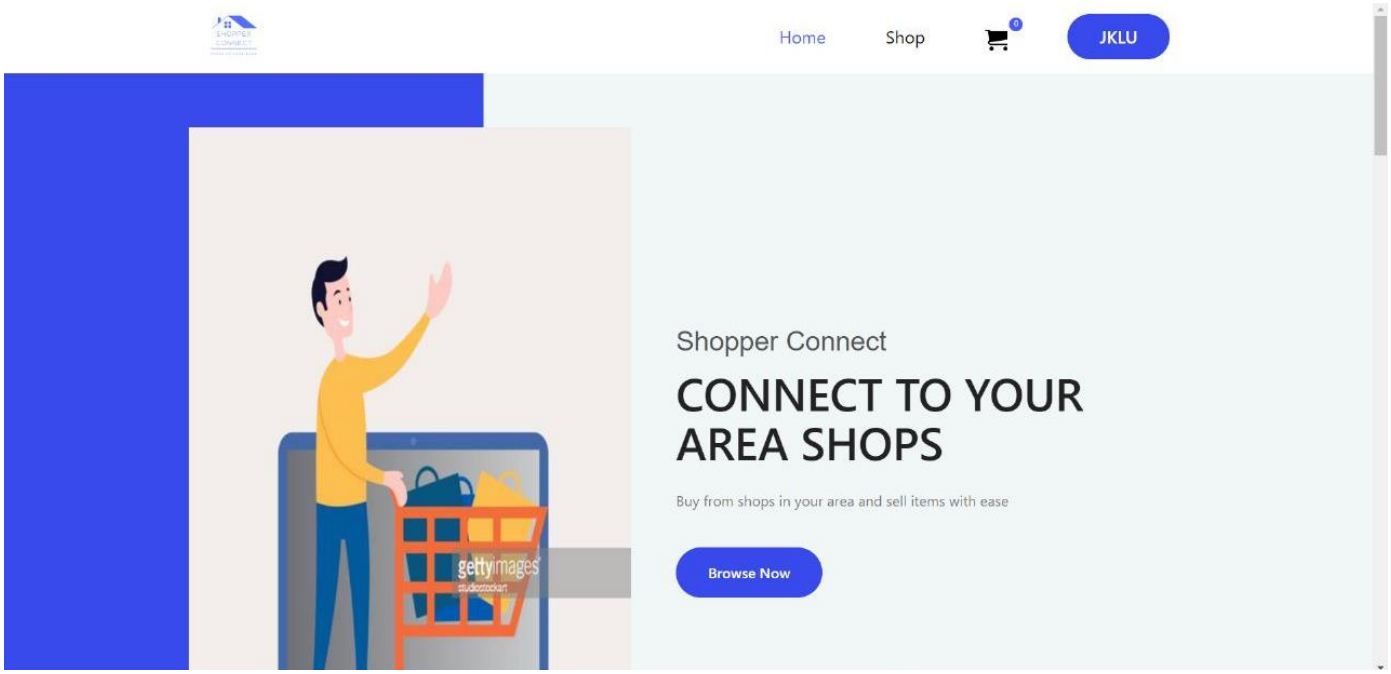

Figure 18 Front User page

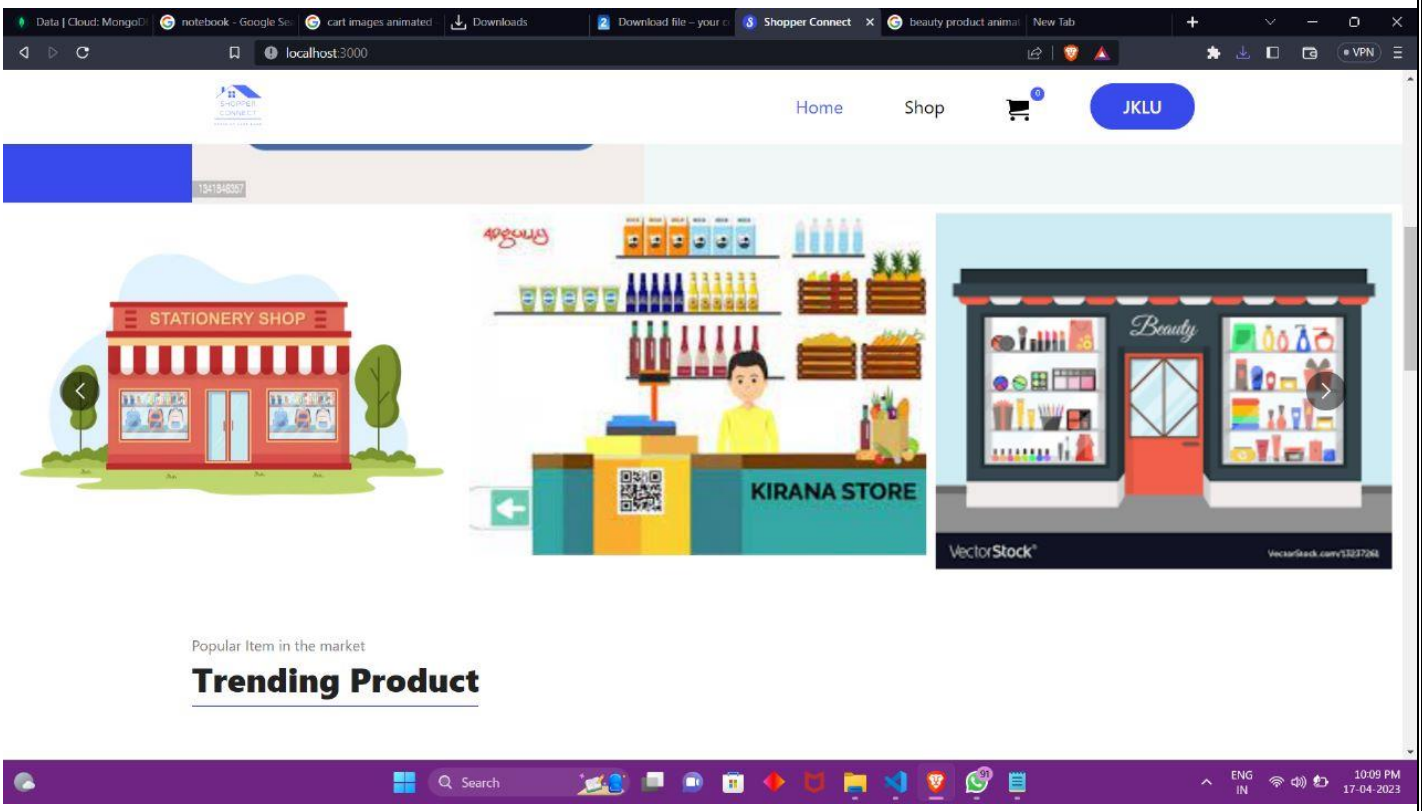

Figure 19 Seller Front page

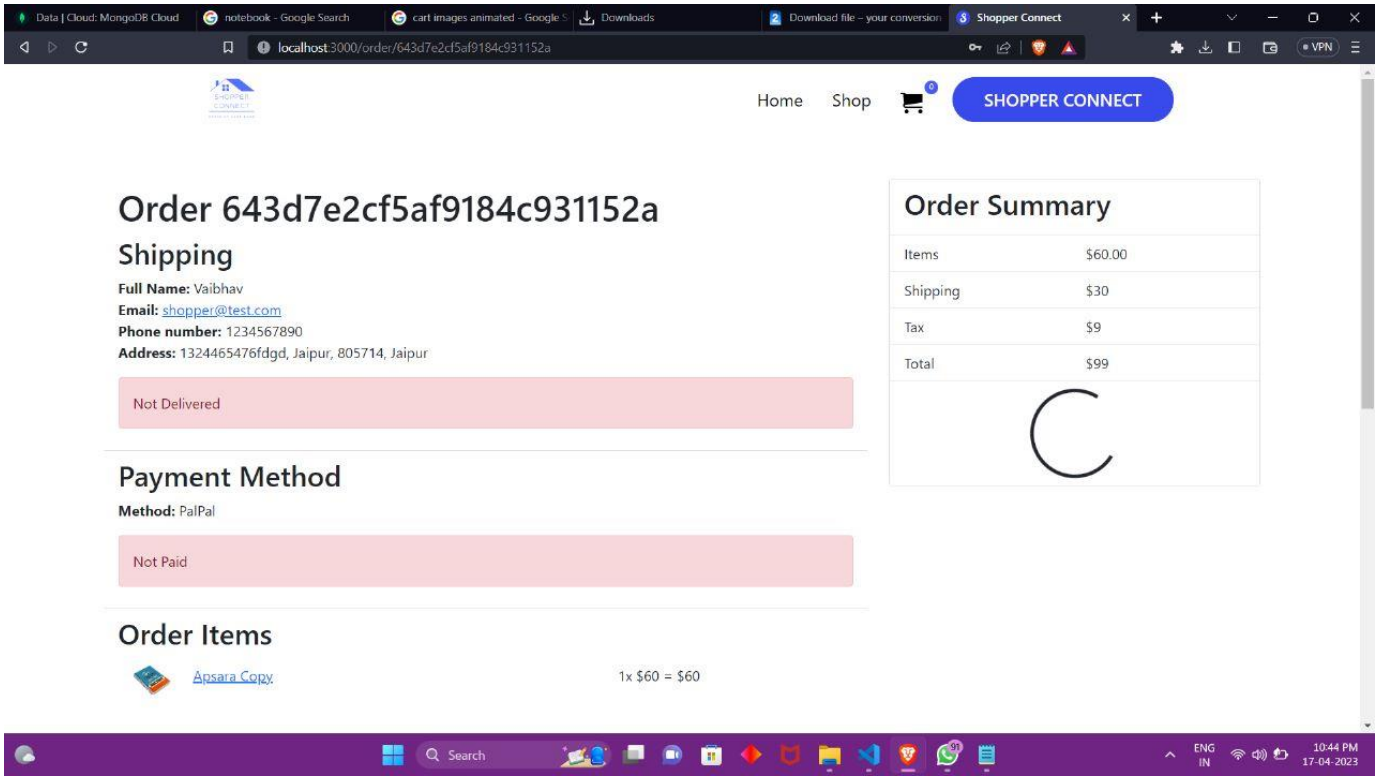

Figure 20 Payment and order details

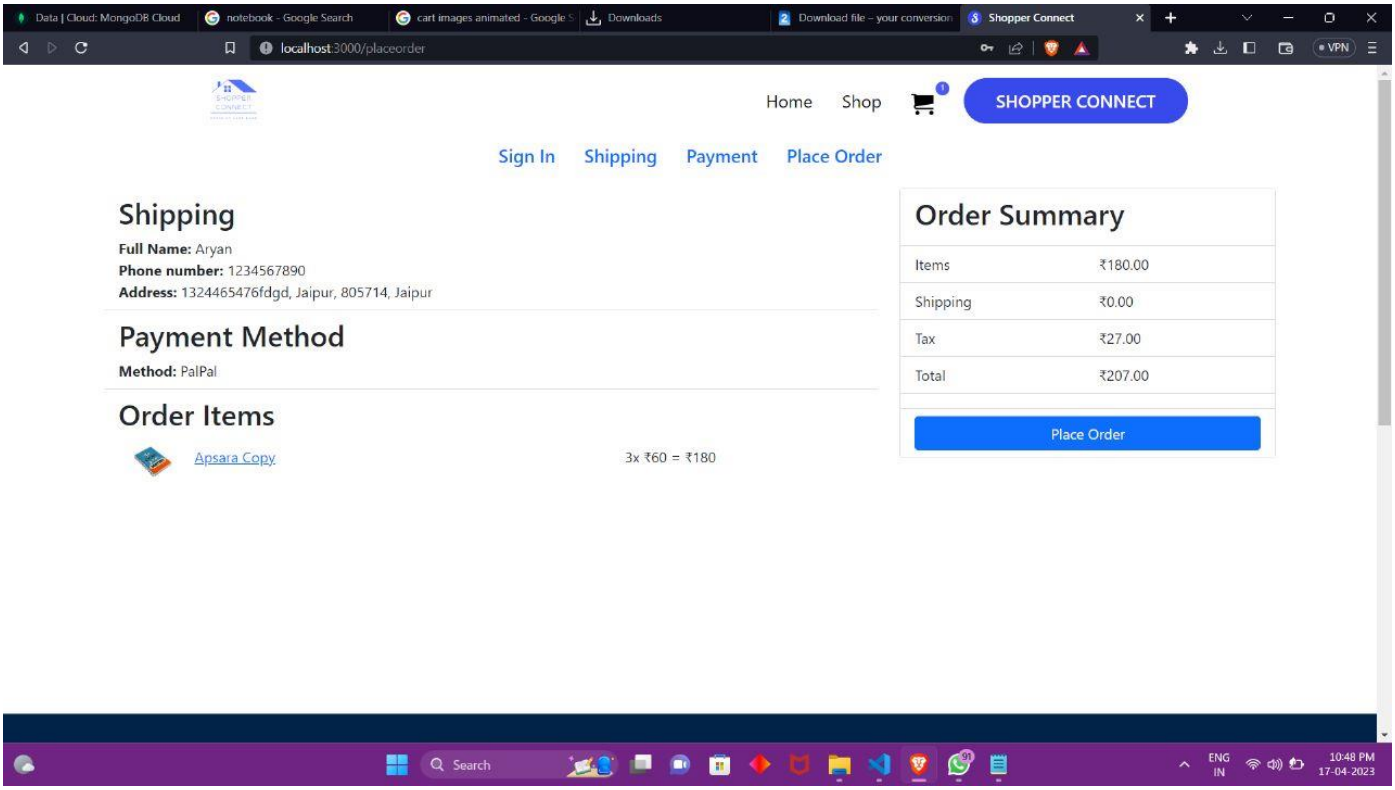

Figure 21 Payment Gateway

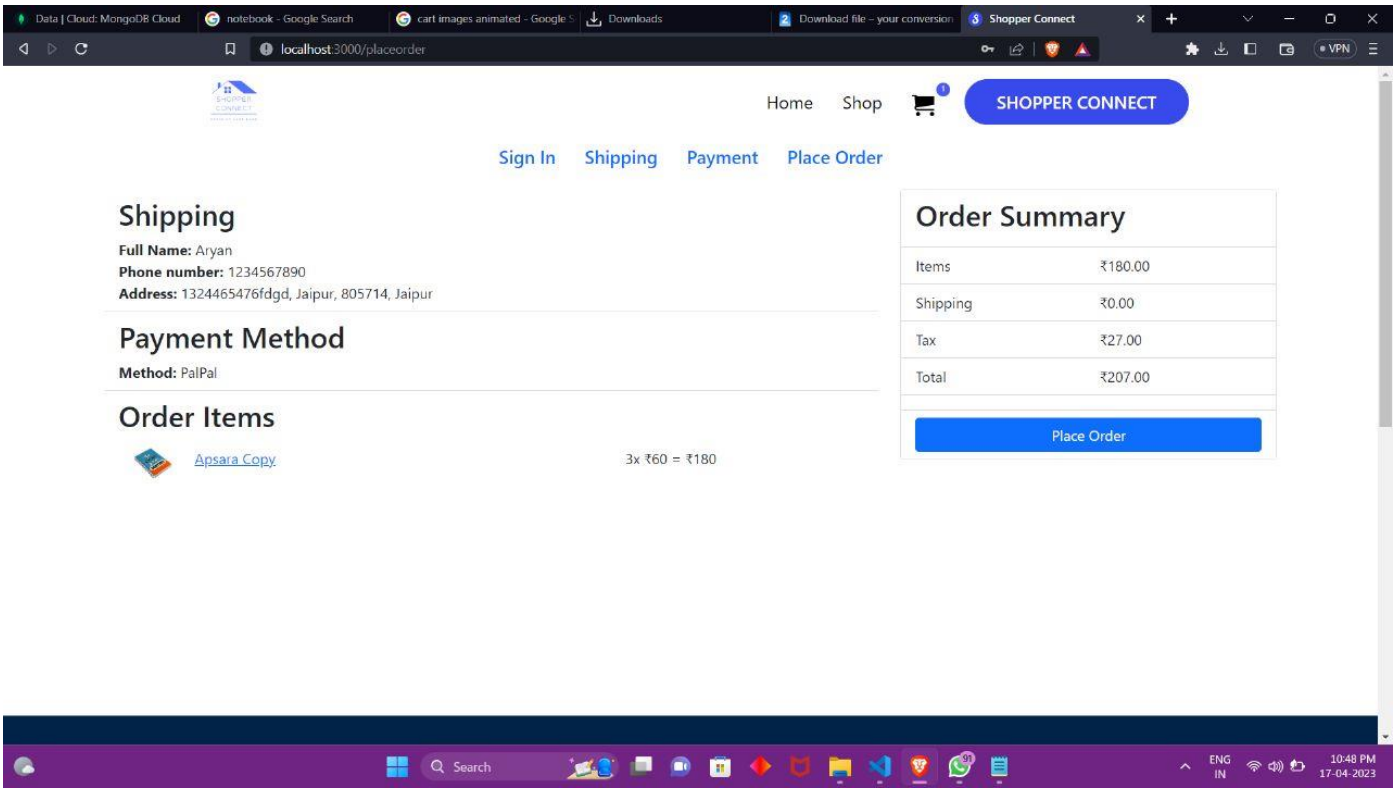

Figure 22 Seller To manage Order

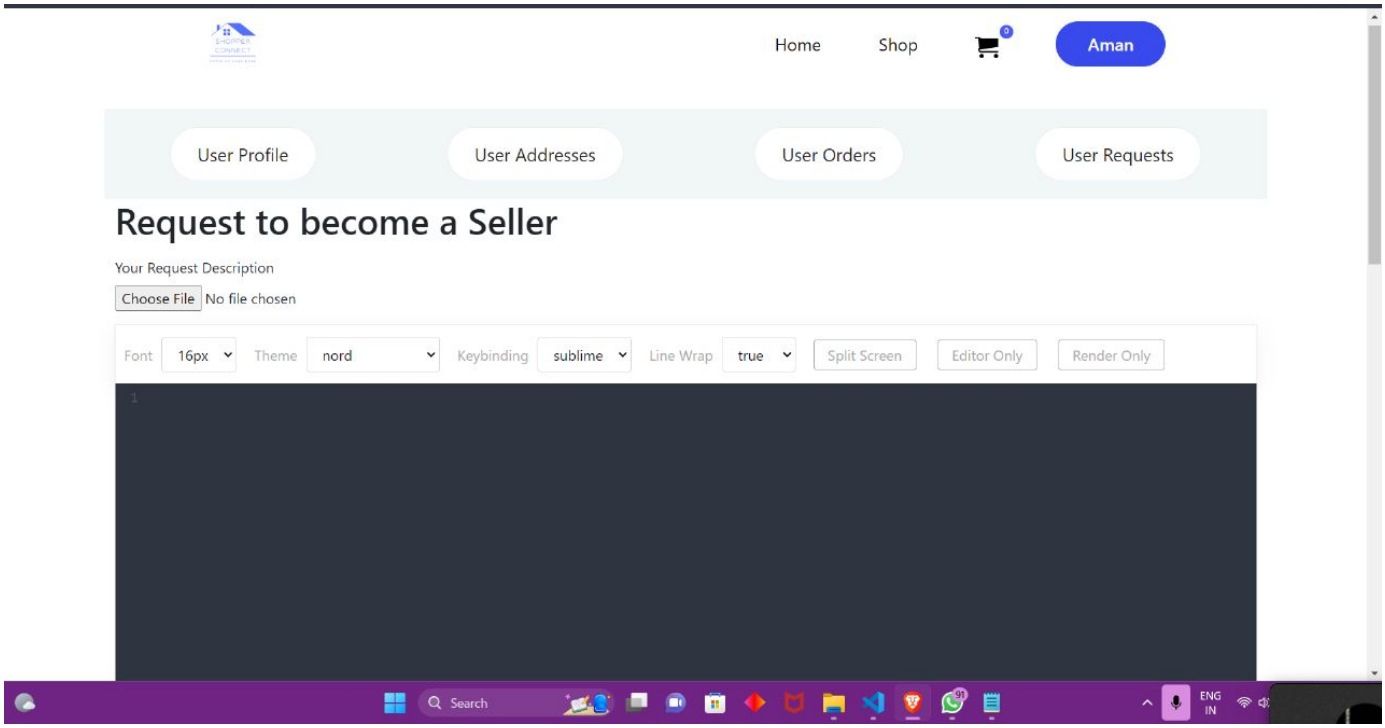

Figure 23 To become a Seller

Software Development

4.1 Implementation Strategy

1. **Selection of appropriate technology stack:** The first step in implementing the solution for the above problem statement is to select an appropriate technology stack. The MERN stack, which includes MongoDB, Express.js, React, and Node.js, can be used to develop a scalable and robust solution.
2. **Building a RESTful API:** The solution should be built using a RESTful API architecture, which allows for easy communication between the client and server. The API should follow industry-standard conventions for endpoints and data formats, such as JSON.
3. **Creating a database schema:** A database schema should be created to store the data related to shops and their details. MongoDB can be used as a NoSQL database for this purpose, as it provides flexibility and scalability for storing unstructured data.
4. **Developing a user interface:** A user-friendly interface should be designed to allow users to search for shops based on their location and category. React can be used to develop a responsive and interactive user interface that works seamlessly across different devices.
5. **Implementing authentication and authorization:** To ensure the security of the platform, authentication and authorization should be implemented. The platform can use OAuth 2.0 or JSON Web Tokens (JWT) to provide secure access to the platform for both users and shop owners.
6. **Testing and quality assurance:** The platform should be tested thoroughly to ensure that it is free from bugs and works as intended. Automated testing tools such as Jest and Enzyme can be used to test the front-end, while tools like Mocha and Chai can be used to test the back end.
7. **Continuous integration and deployment:** A continuous integration and deployment (CI/CD) pipeline should be set up to automate the testing, build, and deployment of the platform. Tools like Jenkins and GitLab can be used for this purpose.
8. **Monitoring and analytics:** The platform should be monitored continuously to ensure that it is performing optimally. Tools like Kibana and Grafana can be used to collect and analyze metrics related to the platform's performance, usage, and errors.

4.2 Code Walkthrough

A code walkthrough is a type of software review that involves a group of developers or team members examining a section of code in detail. The purpose of a code walkthrough is to identify any errors, issues, or potential improvements in the code. For our Project the review is done by fellow mate Saurabh. According to him the code is well documented.

4.3 Code Review

Code review is a semi- formal manner was done to improve our code and to further discuss methodologies used. It was not only about the execution but also how the code looks. it is important to note that no code is truly perfect and there is always room for improvement. Therefore, it is recommended to conduct regular code reviews even if the initial code appears to be error-free. This can help to identify potential areas of improvement, ensure adherence to coding standards and best practices, and enhance overall code quality.

4.4 Code Inspection

The code reviewer begins by reviewing the code for the front-end web application built using ReactJS. They check if all the components are rendered correctly and if there are any syntax errors in the code. They also check if the CSS styling is consistent across all the pages. Next, the reviewer moves on to the backend code

written in Node.js. They check if the RESTful API endpoints have been implemented correctly and if the code is scalable and maintainable. They also check if the database schema has been designed properly and if there are any queries that can be optimized. After reviewing the code, the reviewer notes down any issues found, such as potential security vulnerabilities or coding standards violations. They also make suggestions for improvements, such as implementing better error handling or refactoring certain parts of the code to make it more modular. Finally, the reviewer communicates their findings to the development team and works with them to address any issues and make the necessary changes to the code.

4.5 Code Inspection Technique

Code inspection is a technique used for reviewing and evaluating software code to identify defects, improve the quality of the code, and ensure that it meets the requirements and specifications. The process involves a group of reviewers analyzing and discussing the code to identify errors, inconsistencies, and potential problems.

There are several techniques used for code inspection, including:

- **Peer review**: In this technique, a developer reviews the code of another developer in the team. The developer looks for errors, coding standards, and the implementation of the business logic. This technique helps to identify errors early in the development process and improves the overall quality of the code.
- **Walkthrough:** In this technique, a group of developers and stakeholders review the code together to identify issues and suggest improvements. The code is presented and discussed in a meeting, and each person provides feedback and suggestions for improvement.
- **Pair programming:** In this technique, two developers work together to write and review the code simultaneously. One developer writes the code while the other reviews it, and they switch roles frequently. This technique helps to identify and fix errors quickly and promotes collaboration and knowledge sharing among the developers.
- **Code review tools**: There are various code review tools available that help to automate the code inspection process. These tools analyze the code for errors, code standards, and best practices. They provide feedback and suggestions for improvement and help to ensure that the code meets the requirements and specifications.

Software Technology

5.1 Testing Strategies

Testing strategies are the approach or plan that a software testing team decides to follow to test a software application. The main purpose of testing strategies is to ensure that software meets the expected requirements, quality standards, and is bug-free before it is released to the end-users. Here are some common testing strategies:

Manual Testing: In this type of testing, testers execute test cases manually, without the help of any automation tool.

Automated Testing: Automated testing is the use of software tools to execute tests, without human intervention.

Black Box Testing: Black box testing is a testing technique in which the tester does not have access to the internal workings of the system being tested. The tester only interacts with the system's inputs and outputs.

White Box Testing: White box testing is a testing technique in which the tester has access to the internal workings of the system being tested. The tester can see how the system processes inputs and generates outputs.

Integration Testing: Integration testing is a type of testing in which individual components of a software application are combined and tested as a group.

Regression Testing: Regression testing is a type of testing that is done to ensure that new changes to a software application do not break existing functionality.

Performance Testing: Performance testing is a type of testing that is done to evaluate the performance of a software application under different workloads.

Security Testing: Security testing is a type of testing that is done to ensure that the software application is secure and protected against unauthorized access and attacks.

User Acceptance Testing: User acceptance testing (UAT) is the final stage of testing before the software application is released to end-users. In this type of testing, the end-users test the software application to ensure that it meets their requirements and is easy to use.

It is important to have testing strategies in place to ensure that software is functioning properly and meets the requirements of the stakeholders. Testing helps to identify defects and errors in the software, which can be costly if left undetected and cause potential harm to users. Additionally, testing strategies help to improve the overall quality and reliability of the software, as well as increase confidence in the product. Testing strategies may vary depending on the type of software being developed, the development process, and the intended use of the software.

5.2 Test Case

A test case is a specific set of instructions or conditions under which a tester will determine whether an application, software system or one of its features is working as intended. It consists of a set of input values, execution preconditions, expected results, and post-execution state. Test cases are used to ensure that the software or application under test meets the specified requirements, is free from defects and is working correctly. They provide a structured approach for identifying and correcting defects in software development, help to ensure the software meets the user's needs, and can be used to measure the quality of the software.

Test Case 1

Test Case ID	CM_001	**Test Case Description**	Test Login functionality of user and seller defirently		
Created By	Vaibhav	**Reviewed By**	Aryan	**Version**	1.0

QA Tester's Log

Tester's Name	Aryan	**Date Tested**	March 22, 2023	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites:
1	Access to Chrome Browser
2	
3	
4	

S #	Test Data
1	Userid =vaibhavtomar@jklu.edu.in
2	Pass = qwerty
3	
4	

Test Scenario Verify on entering valid userid and password, the person can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the Web App	Site should open	As Expected	Pass
2	Enter Userid & Password	Credential can be entered	As Expected	Pass
3	Choose type	User or Seller should be selected	As Expected	Pass
4	Click Submit	Logged in	As Expected	Pass

Test Case -2

Test Case ID	CM_002	Test Case Description	To select Store		
Created By	Aryan	Reviewed By	vaibhav	Version	2.0

QA Tester's Log

Tester's Name	Vaibhav	Date Tested	March 22, 2023	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Access to the information of store and item
2	
3	
4	

S #	Test Data
1	List of all store and product
2	
3	
4	

Test Conditions

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Select store	Software should succesfully able to input information.	As Expected	Pass
2	Show data of item store	Software should succesfully able to show data to user	As Expected	Pass
3	Should show the price of item and quantity	the quantity of item should be exact that has been entered	As Expected	Pass

Test Case -3

Test Case ID	CM_001	Test Case Description	To delete student information or entry by admin.		
Created By	Vaibhav	Reviewed By	Aryan	Version	1.0

QA Tester's Log

h

Tester's Name	Vaibhav	Date Tested	March 22, 2023	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Access to the information of delivery.
2	
3	
4	

S #	Test Data
1	Entries of all order done by Store.
2	
3	
4	

Test Conditions

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Write the order detail	Software should succesfully show the list of the order and details	As Expected	Pass
2	Show delivery or pickup detaisl	Software should succesfully able to show data	As Expected	Pass
3	Cancel the order	Software should succesfully cancel the order	As Expected	Pass

Summary

6.1 Conclusion

The challenge of connecting nearby shops can be addressed by utilizing technology to create a platform that allows customers to find and connect with local businesses quickly and conveniently. The platform must be designed with an intuitive user interface and provide relevant information about each shop, including the products and services they offer, their location, hours of operation, and contact details. It is essential that the platform is scalable and adaptable to support the needs of businesses of all sizes, from small independent retailers to large multinational corporations. Proper implementation and testing strategies can ensure that the platform is effective, reliable, and meets the needs of both customers and businesses. This can include conducting thorough testing of the platform's features, ensuring its compatibility with different devices and browsers, and performing load testing to ensure it can handle high levels of traffic. By solving the problem of connecting nearby shops, we can help to support local businesses and communities by providing a more convenient and efficient way for customers to find and connect with the shops they need.

6.2 Future Scope

- **Integration with Augmented Reality**: With the integration of augmented reality (AR) technology, users can get a more immersive shopping experience by browsing nearby shops in real-time, virtually exploring the products, and accessing relevant information through their devices.
- **AI-Powered Recommendation System**: By integrating an AI-powered recommendation system, the platform can provide personalized recommendations to users based on their preferences, previous purchases, and search history.
- **Integration with Social Media Platforms**: The integration of social media platforms can help to promote local businesses and provide users with more information about the shops they are interested in.
- **Advanced analytics**: Advanced analytics can be used to gather and analyze data on user behavior, shop performance, and market trends, providing insights that can inform business decisions and drive growth.

References

https://www.javatpoint.com/software-engineering

https://www.mongodb.com/mern-stack#:~:text=MERN%20stands%20for%20MongoDB%2C%20Express,a%20client%2Dside%20JavaScript%20framework

https://www.simplilearn.com/tutorials/mongodb-tutorial/what-is-mern-stack-introduction-and-examples

https://www.udemy.com/course/mern-stack-front-to-back/

https://trello.com/?&aceid=&adposition=&adgroup=97922000102&campaign=9952959484&creative=432962144093&device=c&keyword=softwear%20tools&matchtype=e&network=g&placement=&ds_kids=p53400093765&ds_e=GOOGLE&ds_eid=700000001741695&ds_e1=GOOGLE&gclid=CjwKCAjw3POhBhBQEiwAqTCuBptNz7tDClo_KzuiI0k-Hsmgq2E3woWUL2MymuO5IcNbogsLYgcd1hoCKtkQAvD_BwE&gclsrc=aw.ds