"TRADECONNECT"

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of of Information Technology

Project Mentor:
Ms. Dolly Mittal
Associate Professor, IT

Submitted By :
Raghav Sharma (19ESKIT073)
Mukul Jangid (19ESKIT058)

Department of of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2022-2023

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that **Mr Mukul Jangid**, a student of B.Tech(Information Technology) VIII semester has submitted his Project Report entitled **"TradeConnect"** under my guidance.

Mentor	**Coordinator**
Ms. Dolly Mittal	Mrs. Sanju Choudhary
Associate Professor, IT	Associate Professor, IT
Signature	Signature

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that **Mr Raghav Sharma**, a student of B.Tech(Information Technology) VIII semester has submitted his Project Report entitled **"TradeConnect"** under my guidance.

Mentor	**Coordinator**
Ms. Dolly Mittal	Mrs. Sanju Choudhary
Associate Professor, IT	Associate Professor, IT
Signature	Signature

DECLARATION

We hereby declare that the report of the project entitled "**TradeConnect**" is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of **Ms. Dolly Mittal** (Dept. of Information Technology) and coordination of **Mrs. Sanju Chaudhary** (Dept. of Information Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Information Technology.It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members **Signature**

Raghav Sharma (19ESKIT073)

Mukul Jangid (19ESKIT058)

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Ms. Dolly Mittal. She has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Mrs. Sanju Choudhary for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Prof. Dr. Anil Chaudhary, HOD, Department of Information Technology, for facilitating, motivating and supporting us during each phase of development of the project. Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:
Raghav Sharma (19ESKIT073)
Mukul Jangid (19ESKIT058)

Abstract

General Description

The retail management system developed in this project aims to revolutionize the way retailers operate and interact within a networked environment. It provides a platform for retailers to connect, collaborate, and streamline their business processes. The system offers a range of features, including batch-level inventory management, order creation and management, in-app payments, and advanced analytics.

The system includes automated reminders for batch expiry dates, ensuring timely actions and minimizing product wastage.

Retailers can create and manage orders within their network of connections, facilitating seamless transactions and efficient supply chain management. The system provides order analytics and insights, enabling retailers to make data-driven decisions and improve their inventory planning and forecasting.

Overall, this retail management system empowers retailers to streamline their operations, collaborate with other retailers, optimize inventory management, facilitate seamless transactions, and gain valuable insights. It aims to enhance the retail experience for both retailers and customers, fostering a connected and efficient retail ecosystem.

Uniqueness

The developed retail management system offers unique features and capabilities that set it apart from traditional systems. It promotes collaboration and communication among multiple retailers, creating a unified retail ecosystem. Retailers can connect, share information, and facilitate seamless transactions. The system includes batch-level inventory management, allowing detailed tracking and management of individual product batches. This level of granularity enables efficient inventory control, including monitoring batch expiry dates and optimizing stock levels. Valuable order analytics and insights empower data-driven decision-making for enhanced inventory management and overall business performance. In-app payment functionality streamlines transactions, eliminating the need for external payment gateways.

The system integrates various communication channels to facilitate seamless and efficient communication between retailers and customers. User experience is prioritized, offering an intuitive interface, easy navigation, and responsive features for an enjoyable and user-friendly experience. In summary, this project provides a comprehensive and innovative retail management solution by fostering collaboration, customization options, batch-level inventory management, insightful analytics, automated reminders, integrated communication channels, and a focus on user experience.

Social Impact

Starting a business not only requires money but innovative ideas and adaptability to changes in the market, which will help the business flourish. Several small business owners start a business and run it for a few days. But when it comes to managing finances, they find it difficult, as all are not good at accounting.

Especially in small businesses, people tend to buy things and pay later. Such debit-credit transactions involve more complications for the business owner, who must keep track of the pending payments and collect them appropriately. This is additional to the regular accounting like inventory calculations, stock supply, infrastructure rent, etc. Our solution provides relief to business owners by doing all accounting and managing end-to-end cash transactions. There is not much technical knowledge required to use this app, and just understanding how to use it will help every individual enjoy all the features of it.

Technology Stack

The application would be divided into three parts: frontend, backend, and database management. The frontend would be comprised of an Android app built using the Kotlin programming language and Android Studio. The frontend would have a user-friendly UI (implemented through XML), which would help the customer and the retailer easily access the functionality. The backend would be comprised of Node.js, which provides a runtime environment, Express as a backend framework, and JWT Token for authentication purposes. The database used is MongoDB.

Table of Content

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Problem Statement and Objective

Local shops have existed for a long time in India. The Local Retailers depend on multiple other wholesaler retailers to buy goods. They are not so big firms that can afford a computer at their desk, due to these limitations they are not able to optimize their business. Also, they have to maintain contact with other salesmen for goods supply. They are not able to maintain their stock properly, they have to keep track of each product manually.

The objective of this project is to develop a comprehensive software system that addresses the challenges faced by a retail business in managing its inventory, sales, and connections with other retailers. The current manual processes and fragmented systems used by the business are inefficient, error-prone, and time-consuming. Therefore, there is a need for a robust and integrated software solution that can streamline the operations, improve inventory management, facilitate sales transactions, and enable efficient collaboration with other retailers.

Retailers also sell goods to other retailers or customers on Credit. But in order to maintain the records of lending they had to maintain a handwritten notebook, in which they would add the details of each retailer and customer and the items that have been lent to them. Initially this was good, but soon the data began to enlarge in size. Retailers had to maintain a separate notebook for each month or a year and had to separate sections for each customer. Also customers didn't have a record, so there was always a fear of forging. The maintenance of the data in notebooks was very tiresome for retailers.

Furthermore, the business lacks a systematic way to establish and manage connections with other retailers in the supply chain. The current process of sending and receiving connection requests is manual, leading to delays and miscommunication.

There is a need for a standardized and automated system that allows retailers to send connection requests, track their status, and facilitate collaboration.
To address these challenges, the proposed software system will provide features such as inventory management, sales management, connection management, and reporting/analytics. The system will integrate various modules, including batch management, order processing, payment integration, and communication channels, to ensure a seamless and efficient workflow for the retail business. The system will also prioritize data security, user authentication, and access control to protect sensitive business information.

1.2 Literature Survey /Market Survey/Investigation and Analysis

Introduction

The retail management system developed in this project aims to revolutionize the retail industry by providing a comprehensive platform for retailers and customers to connect, transact, and manage their operations efficiently. This market analysis assesses the viability and growth prospects of the project.

Market overview

The retail industry is a vast and rapidly growing sector, with significant opportunities for innovative solutions. The retail market consists of various segments, including brick-and-mortar stores, e-commerce platforms, and multi-channel retailers. TradeConnect targets retailers and customers looking for streamlined operations, improved connectivity, and enhanced customer experiences.

Target market

The primary target market for TradeConnect includes small to medium-sized retailers across various sectors such as fashion, electronics, home decor, and more. These retailers can benefit from the platform's inventory management, order processing, and sales tracking features. The secondary target market comprises customers who frequent retail stores and seek a seamless shopping experience, personalized recom-

mendations, and access to transaction history

Competitive Analysis

- Key Competitors

  The market is competitive, with existing solutions catering to different aspects of retail management. Competitors may include established point-of-sale (POS) systems, e-commerce platforms, and inventory management software providers. However, TradeConnect differentiates itself by offering a comprehensive solution that integrates inventory management, order processing, and customer engagement.

- Competitive Advantage

  TradeConnect stands out through its user-friendly interface, real-time data synchronization, robust analytics capabilities, and connection-based features that facilitate collaboration between retailers and customers.

Conclusion

The market analysis indicates a favorable market environment for TradeConnect. The project addresses the evolving needs of retailers and customers in a digitally–driven retail landscape. With its competitive advantages and market trends supporting its growth, TradeConnect is positioned to capture a significant share of the retail market and become a leading solution in the industry.

1.3 Introduction to Project

The retail management system developed in this project aims to revolutionize the way retailers operate and interact within a networked environment. It provides a platform for retailers to connect, collaborate, and streamline their business processes. The system offers a range of features, including batch-level inventory management, order creation and management, in-app payments, and advanced analytics. With the batch-level inventory management functionality, retailers can effectively track and manage their stock at the individual batch level. This allows for precise control over

inventory levels, reducing waste and optimizing stock management. The system also includes automated reminders for batch expiry dates, ensuring timely actions and minimizing product wastage. Retailers can create and manage orders within their network of connections, facilitating seamless transactions and efficient supply chain management. The system provides order analytics and insights, enabling retailers to make data-driven decisions and improve their inventory planning and forecasting. The system also offers communication channels to foster effective communication between retailers and customers. Retailers can send notifications, updates, and promotional messages to their customers, enhancing customer engagement and loyalty. Advanced analytics provide retailers with valuable insights into their sales, inventory, and customer behavior. These insights enable retailers to identify trends, make informed business decisions, and enhance their overall operational efficiency. Overall, this retail management system empowers retailers to streamline their operations, collaborate with other retailers, optimize inventory management, facilitate seamless transactions, and gain valuable insights. It aims to enhance the retail experience for both retailers and customers, fostering a connected and efficient retail ecosystem.

1.4 Proposed Logic / Algorithm / Business Plan / Solution / Device

Proposed Business Logic for TradeConnect: The proposed business logic of TradeConnect revolves around creating a comprehensive platform that connects retailers and customers, streamlines retail operations, and enhances the overall shopping experience. The key components of the business logic include:

1. Retailer Management:
    - Retailer Registration: Retailers can register on the platform by providing their business details, such as business name, type, and tax ID.
    - Inventory Management: Retailers can add batches of stocks to their account and manage inventory details, including batch numbers, quantities, and prices.

- Order Processing: Retailers can create orders to fulfill their goods requirements, specifying the batch details, quantities, and prices.
- Sales Record: Retailers can record sales made to their connections, tracking the batches sold, quantities, and prices.

2. Customer Engagement:

   - Connection Establishment: Customers can view and connect with retailers, enabling them to receive personalized offers, recommendations, and updates.
   - Transaction History: Customers can view their transaction history, including purchases made from different retailers, providing a consolidated view of their shopping activities.
   - Retailer Queries: Customers can directly communicate with retailers, seeking information, clarifications, or making specific requests.

3. Automated Reminders and Notifications:

   - Batch Expiry Reminders: Retailers receive automated reminders about batch expiries, helping them manage stock freshness and minimize wastage.
   - Payment Reminders: Retailers receive due payment reminders for sales made to their connections, ensuring timely and accurate payment settlements.

4. Analytics and Insights:

   - Order and Sales Analytics: Retailers can access analytics and reports to gain insights into their orders, sales performance, popular products, and customer preferences. This information can drive informed business decisions and strategies.

The proposed business logic aims to create a seamless and efficient retail ecosystem where retailers can manage their inventory, process orders, engage with customers, and analyze sales data. Customers, on the other hand, can connect with retailers, view transaction history, and enjoy personalized shopping experiences. The

automation of reminders and notifications enhances operational efficiency, while analytics provide valuable insights for strategic decision-making.

1.5 Scope of the Project

The system will provide features to track and manage inventory, including batch management, stock levels, expiration dates, and movement of products across different retailers. It will enable real-time visibility, accurate stock tracking, and efficient inventory management.

The system will automate the sales process, including order processing, payment tracking, invoicing, and sales analytics. It will streamline sales transactions, improve billing accuracy, and provide insights for informed business decisions.

The system will provide comprehensive reporting and analytics capabilities, allowing users to generate various reports and gain insights into inventory performance, sales trends, and overall business metrics.

Deliverables: The project deliverables include the UI of the application, business logic of the app, database schema, backend business logic and REST APIs construction.

Timelines: The project starts with the building of a database schema and UI design, followed by the business logic of the backend, including the REST APIs. Then the business logic of the Android app would be built, which consumes the REST APIs to interact with the backend. This will be followed by testing of end-to-end functionalities and bug fixation, if any.

Milestones:

1. Completion of database schema
2. Completion of the UI
3. Completion of business logic on the backend
4. Completion of REST APIs
5. Completion of the business logic of the Android App
6. Completion of Testing of end-to-end functionalities and bug fixation

Chapter 2
Software Requirement Specification

2.1 Overall Description

The software system is designed to be a standalone solution that caters to the specific needs of a retail business. It will serve as a central platform for managing various aspects of the business, including inventory, sales, and retailer connections. The system will be developed using modern technologies and will be scalable to accommodate future growth and enhancements.
From a technical perspective, the system will utilize a combination of android-based technologies and a database management system to store and retrieve data. It will have a user-friendly interface that allows retailers to easily navigate and perform their tasks. It will provide retailers with a robust and efficient tool to manage their business operations effectively.
The software system will be designed with a modular architecture, allowing for easy customization and integration of additional features or modules as required by the business. It will adhere to industry best practices and standards to ensure data integrity, security, and reliable performance.

2.1.1 Product Perspective

2.1.1.1 System Interfaces

The TradeConnect app will have interfaces with external systems, such as Firebase and external servers. These interfaces will enable the app to fetch order, sells, connection, store and transaction data, authenticate user, and facilitate secure network calls.

2.1.1.2 User Interfaces

1. SignUp Screen

- This screen contains a phone number input field, in which user enters a phone number and receives an OTP for verification. After successful verification user is prompted to choose the role (Retailer, Cutomer).

2. SignIn Screen

   - This screen contains a phone number input field, in which user enters a phone number and receives an OTP for verification and a Google SignIn button.
   - User can sign in using any of the above two methods. Once successfully signed in user is taken directly to MainScreen.

3. Main Screen

   - This screen contains Retailer's shopname as a heading. And various other button to navigate the app.
   - The buttons include, MyProfile, ViewStore, MyOrders, MySells, MyConnections.
   - In case of a customer SignIn, the user will see the list of all the retailers to whom some transactions have been made.

4. ViewStore Screen

   - This screen shows the entire detail of the retailer's inventory. When the user clicks any product, its batch details are shown to the user.
   - Retailer has the option to add product with single batch or bulk upload through the excel sheet.

5. MyOrder Screen

   - This screen shows all the orders which are in the inactive state.
   - The retailer has the option to create order requests to its connections.

6. Create Order Screen

- This screen shows the existing connections of the retailer. Based on the connection selection by the retailer, the product list is updated.
- Retailer can choose the product and its batch, and create requeust for some quantity of that batch. Multiple products can be added.

7. PendingOrders Screen

   - This screen shows two tabs namely Sent and Received.
   - The sent tab contains all the orders that have been sent by the current user to other retailers, and are still not accepted or rejected by the other.
   - The received tab contains all the orders that have been received by the current user from other retailers, and are still not accepted or rejected by the user.

8. MySells Screen

   - This screen shows the entire list of all the sells that have been created by the user. Each sell item show the due amount if there's any, else show the status of Paid.
   - The user has the option to update the sell. The paid amount can be updated any number of times.
   - The retailer has the option of creating sells for either some other retailer or for some customer entity.

9. MyInvitations Screen

   - This screen shows two tabs namely Sent and Received.
   - The sent tab contains all the connection requests that have been sent by the current user to other retailers, and are still not accepted or rejected by the other.
   - The received tab contains all the connection requests that have been received by the current user from other retailers and are still not accepted or rejected by the user.

10. MyConnections Screen

    - This screen shows all the connections of the current user. User can from here go to the MyInvitations screen to view all connection requests or Add new connection.

11. AddConnection Screen

    - This screen has an input field where retailer can search for other retailers to connect with based on their Name, Phone number or Shop Address.
    - The list is updated in realtime, the user can select any retailer and send the connection request to the user.

12. MyInvitations Screen

    - This screen shows two tabs namely Sent and Received.
    - The sent tab contains all the connection requests that have been sent by the current user to other retailers, and are still not accepted or rejected by the other.
    - The received tab contains all the connection requests that have been received by the current user from other retailers and are still not accepted or rejected by the user.

13. MyProfile Screen

    - This screen shows all the personal details of the user including the no. of Connections, Orders and Total sales.
    - From here Retailer can go to the Analytics screen, to get some insights.

14. Analytics Screen

    - This screen shows data insights in graphical form.
    - Bar and Pie charts are generated for the Orders and Sales data.
    - Graph data can be aggregated based on the last 7 days, last 6 months and last 3 years.

2.1.1.3 Hardware Interfaces

Minimum Requirements

Client Side			
	Processor	RAM	Disk Space
Android Device	Android Version 5.0(Lollipop) and above	500 MB	250 MB
Server Side			
	Processor	RAM	Disk Space
NPM v6	Intel Pentium III or AMD -800 MHz	1 GB	3.5 GB
MongoDB- v4+	Intel Pentium III or AMD -800 MHz	256 MB	500 MB (Excluding Data Size)

Table 2.1: Minimum Hardware Requirements

Recommended Requirements

Client Side			
	Processor	RAM	Disk Space
Android Device	Android Version 5.0(Lollipop) and above	2 GB	1 GB
Server Side			
	Processor	RAM	Disk Space
NPM v7	All Intel or AMD -2 GHz	2 GB	3.5 GB
MongoDB- v4.4	Intel or AMD -2 GHz	512 MB	500 MB (Excluding Data Size)

Table 2.2: Recommended Hardware Requirements

2.1.1.4 Software Interfaces

- **Client**

  Any Android Mobile Device

- **Operating System**

  Android (5.0 and above)

- **Data Base Server**

  MongoDB atlas, supported on any OS

- **Development End**
  Android Studio, JDK 17

2.1.1.5 Communications Interfaces

Our system is a mobile-based application and hence it requires only a basic smartphone. This system supports all Android ($\geq$ 5.0) enabled devices.
Communication standards and Network server communications protocols used are: HTTP, HTTPS

2.1.1.6 Memory Constraints

Memory constraints in the TradeConnect app can refer to the limitations or considerations related to the app's memory usage.

1. Device RAM: The TradeConnect app should be designed to operate efficiently within the available random access memory (RAM) of the user's device. The app should optimize its memory usage to prevent excessive RAM consumption, which can lead to performance issues and potentially cause the app to crash or slow down.

2. Storage Space: The app should be mindful of the user's device storage space. Saving excessive data, cache files, or temporary files can consume significant storage space. The app should implement appropriate storage management techniques to prevent storage constraints and ensure smooth functioning.

3. Garbage Collection: The TradeConnect app should manage memory through proper garbage collection techniques. Unused objects and resources should be promptly released to free up memory and prevent memory leaks.

4. Compatibility with Low-memory Devices: The app should be designed to work smoothly on devices with lower memory capabilities. Optimizing resource usage and implementing memory-efficient algorithms can ensure the app's compatibility with a wider range of devices.

5. Background Processes: The app should be mindful of background processes and services that consume memory resources. It should prioritize essential processes and minimize unnecessary background operations to conserve memory.

6. Server Specifications: The hardware specifications of the backend server also affect memory constraints. Servers with limited RAM capacity may struggle to handle memory-intensive applications or high traffic loads.

7. Concurrent Requests: The backend application receives a high volume of concurrent requests, each request typically requires memory to handle. If the server does not have enough memory to handle the concurrent load, it may lead to performance issues or even crashes.

2.1.1.7 Operations

This TradeConnect enables various operations to facilitate the smooth functioning of the system. Some of the key operations include:

1. User Management: The project allows for the creation, registration, and management of users, such as retailers and customers. It involves functionalities like user authentication, user profile management, and access control.

2. Inventory Management: Retailers can add, update, and track their inventory of products. They can manage stock levels, track product availability, and view detailed information about each product, including batch details, quantities, and prices.

3. Order Management: Retailers can create and manage orders for their products. They can create new orders, view existing orders, track order status, and update order details as needed. Customers can view their order history and track the status of their orders.

4. Sales Management: The project provides functionality to record and manage sales transactions. Retailers can record sales made to customers, including the products sold, quantities, prices, and payment details. They can generate sales reports and track sales performance.

5. Connection Management: Retailers can establish connections with other retailers or customers. They can send connection requests, accept or decline requests, and manage their connections. This facilitates collaboration, communication, and business relationships within the retail ecosystem.

6. Reporting and Analytics: The project includes reporting and analytics features to provide insights into various aspects of the retail business. Retailers can generate reports on sales, inventory, and financial performance. They can analyze data to identify trends, make informed business decisions, and optimize their operations.

7. Notifications and Reminders: The system sends automated notifications and reminders to retailers and customers. This includes batch expiry reminders, payment due reminders, order updates, and other relevant notifications to keep stakeholders informed and updated.

These operations collectively contribute to streamlining the retail management process, improving efficiency, and enhancing the overall customer experience.

2.1.1.8 Project Functions

Retailer

1. Add Batch Wise Stocks: Retailers can add details of new batches of products to their inventory, including batch number, quantity, price, and other relevant information.

2. Manage Inventory: Retailers can view and manage their inventory, including updating stock quantities, modifying batch details, and tracking batch expiry dates.

3. Create Orders: Retailers can create orders for new goods or products based on their requirements. They can specify the desired quantity and preferred batch details for each item.

4. Approve or Decline Orders: Retailers have the ability to review and approve or decline orders received from their connections. They can consider factors such as stock availability, pricing, and customer demand before making a decision.

5. Create Sales Records: Retailers can record sales made to their connections. They can specify the sold batch details, quantity, price, and other relevant information. This helps in maintaining a record of all transactions.

6. Receive Expiry Reminders: Retailers will receive automated reminders for batch expiry dates. This helps them stay updated and take necessary actions such as removing expired batches from their inventory or offering discounts to clear them before expiry.

7. Receive Payment Reminders: Retailers will receive automated reminders for due payments from their customers. This ensures timely payment collection and helps in managing cash flow.

8. Order and sales analytics: Retailers can view real time orders and sales analytics in the form of charts and graphs. This will help them gain insights of their business about number of orders or sales, received / created for any timeline they want.

Customer

1. View all associated retailers/shops: Customers can access a list of all the retailers or shops they are associated with. This allows them to quickly find and connect with their preferred retailers for their purchasing needs.

2. View transactions: Customers can view their transaction history, including details of their purchases from different retailers. They can review the batch details, quantities purchased, prices, and payment information for each transaction

2.1.1.9 User Characteristics

As a user of the TradeConnect app, you are likely to possess certain characteristics that influence your interaction and experience with the application. These user char-

acteristics can vary based on your retail knowledge, habits, preferences and role. Here is a detailed explanation of user characteristics relevant to the TradeConnect app:

1. Retail & Store Management Literacy: Your level of retail management system knowledge and literacy plays a significant role in how you utilize the TradeConnect app. Whether you are a novice or an experienced retailer, the app should cater to your understanding and provide features and functionalities that align with your goals and needs.

2. Technology Proficiency: Your familiarity and proficiency with technology impact how easily you can navigate and utilize the TradeConnect app. If you are comfortable with smartphones, applications, and digital tools, you may quickly adapt to using the app's interface and features. However, if you have limited experience with technology, the app should offer a user-friendly and intuitive interface to accommodate your needs.

3. Ownership of Retail Business: This determines whether you would be using the TradeConnect app as a customer or a retailer. The app provides different screens and functionalities for each user.

4. Data Accuracy and Consistency: As a user, you are responsible for providing accurate and consistent data to the app. This includes inputting correct product details, order and sell details. The app relies on your cooperation to ensure the accuracy and consistency of data across the whole system .

5. Privacy and Security Awareness: Your awareness of privacy and security concerns is crucial while using the TradeConnect app. You should be cautious about sharing personal financial information and follow best practices for data security, such as setting OTPs, enabling device security features, and using secure networks. The app should prioritize data encryption, secure authentication, and robust privacy measures to instil confidence in its users.

6. Multilingual and Localization Preferences: Depending on your language preferences and geographic location, the TradeConnect app may need to support

multiple languages or localization options. Providing localized currency formats, language options, and regional financial regulations can enhance your user experience.

2.1.1.10 Constraints

- The Internet connection is a constraint for the application. Since the application fetches data from the server over the Internet, it is crucial that there is an Internet connection for the application to function.
- The backend will be constrained by the capacity of the database. Since the database is shared with the larger system, it may be forced to queue incoming requests and as a result, increase the time it takes to fetch data.
- The mobile must be equipped with the TradeConnect App.
- Execution time for the algorithm should take no longer than one second.
- All Kotlin code shall conform to the Kotlin Code Convention standards.
- Users shall be required to log in to use the app.

2.1.1.11 Assumption and Dependencies

Every system requires some certain parameters to work, to work as per the requirement, our system also requires some parameters, and we assume them as fulfilled before using this system, which is as:

- It is assumed that the system will be accessed by authorized users only, and appropriate authentication and authorization mechanisms will be implemented.
- It is assumed that the users of the system have the required knowledge and training to effectively use the system's features and functionalities.
- Software is dependent on access of Internet, as it is a remote application, it is necessary to have internet access.
- Assume that all the information entered by the user will be correct. If any wrong information is found, then the system will notify an alert.

- The system is required to save the generated reports

Chapter 3
System Design Specification

3.1 System Architecture

The system architecture of the TradeConnect App:

1. Presentation Layer:
    - The Presentation Layer is responsible for handling the user interface and user interactions.
    - It includes activities, fragments, and views that correspond to each screen in the app.
    - The MyProfile, Analytics, SignUp, SignIn, MyOrders, MyConnections, ViewStore, MySells, CreateOrder, CreateSell, AddStock and MyInvitations are implemented as separate UI components within this layer.
    - Each screen interacts with the business logic layer to retrieve and update data.
2. Business Logic Layer:
    - The Business Logic Layer contains the core functionality and business rules of the TradeConnect app.
    - It handles the processing and manipulation of Inventory data of a user based on their interactions.
    - Each screen in the presentation layer interacts with the business logic layer to perform operations specific to that screen.
    - For example, the ViewStore screen shows the Inventory details of a user, whereas the MyInvitations screen shows all the connection requests for that user.

3. Data Access Layer:

    - The Data Access Layer is responsible for accessing and managing the app's data.
    - It interacts with the database or data storage mechanism, through network REST APIs to perform data operations.
    - Each screen in the business logic layer interacts with the data access layer to retrieve, store, and update the app data.
    - The data access layer ensures data integrity and provides efficient methods for data retrieval and storage.

4. External Services/Interfaces:

    - The TradeConnect app may integrate with external services or interfaces to enhance its functionality.
    - The authentication system is handled by Firebase, which is responsible for handling phone and mail authentication methods.
    - The Push Notification services are also handled by Firebase Cloud Messaging.
    - Integration with external services is facilitated through appropriate APIs and protocols.

5. Network Layer:

    - The network layer makes network calls to the server through REST APIs and performs data communication.
    - It is implemented by a well know network library called Retrofit which uses OkHTTP.
    - This layer include components and functionalities such as making HTTP requests, handling authentication and authorization, processing responses from the API, error handling, and any necessary data transformation or serialization/deserialization tasks.

3.2 Module Decomposition Description

The module decomposition for TradeConnect involves breaking down the system into distinct modules or components to facilitate better organization, maintainability, and modularity. The key modules identified for the project are as follows:

1. User Management Module:
    - Responsible for user registration, authentication, and authorization.
    - Handles user profile management, including personal information, contact details, and preferences.
2. Retailer Management Module:
    - Handles retailer-specific functionalities, such as retailer registration, business information management, and inventory management.
    - Manages batch-wise stocks, including details like batch numbers, quantities, prices, and expiration dates.
    - Facilitates order processing, sales record creation.
3. Customer Management Module:
    - Manages customer-related functionalities, including customer registration, profile management, and connection establishment with retailers.
    - Provides features for customers to view transaction history.
4. Order Management Module:
    - Handles the creation, processing, and tracking of orders placed by retailers.
    - Provides features for customers to view transaction history.
    - Tracks batch details, quantities, prices, and payment information associated with each order.
5. Notification Module:

- Responsible for sending automated reminders and notifications to retailers and customers.
- Sends batch expiry reminders to retailers to manage stock freshness and minimize wastage.
- Sends order, connection and sells 's requests and update notifications.

6. Analytics Module:

- Collects and processes data related to orders, sales.
- Generates reports and analytics to provide retailers with valuable insights into their business performance.

These modules work together to create a comprehensive retail management system that connects retailers and customers, streamlines inventory and order management, and provides valuable analytics for informed decision-making. Each module has specific responsibilities and interfaces with other modules to ensure seamless functionality and a smooth flow of data and operations throughout the system.

3.3 High Level Design Diagrams

3.3.1 Use Case Diagram

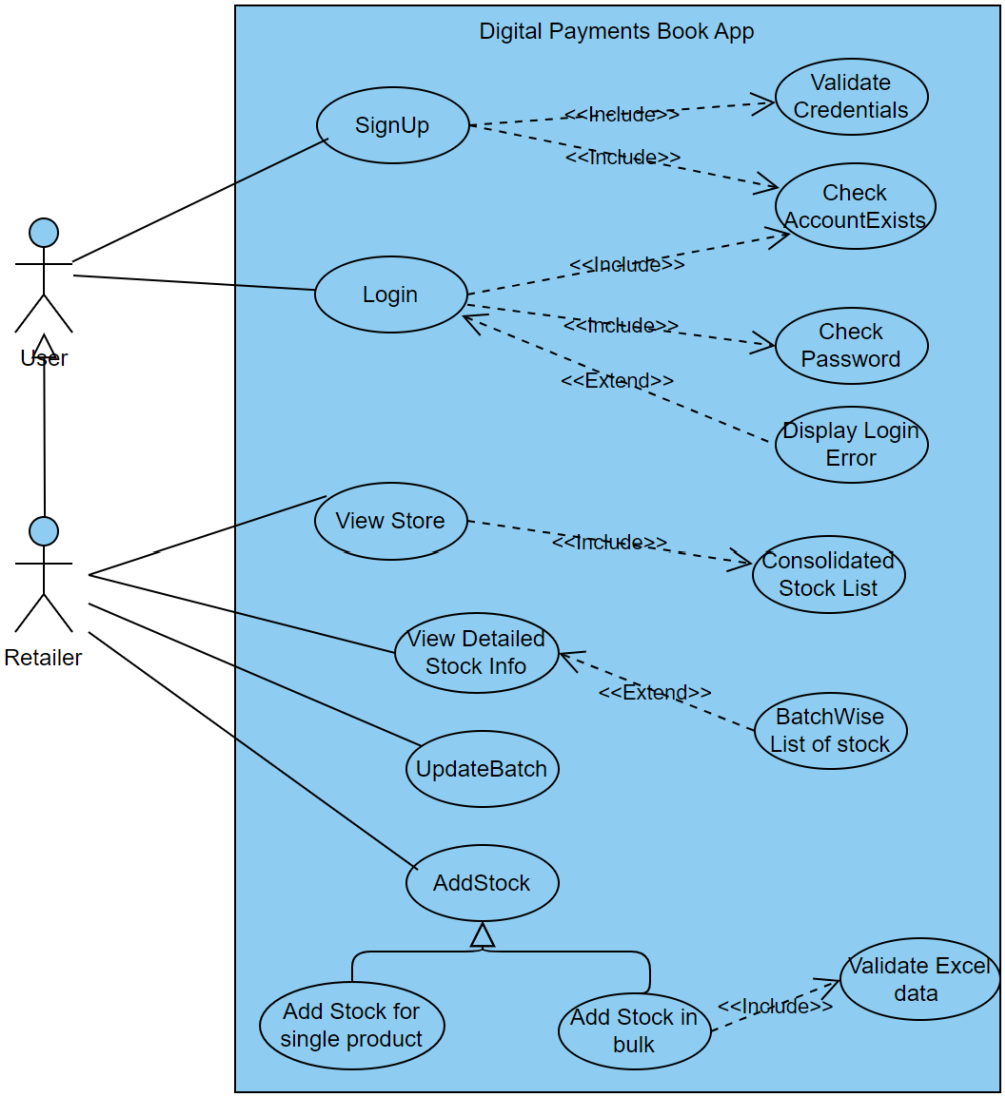

Figure 3.1: Use Case (1)

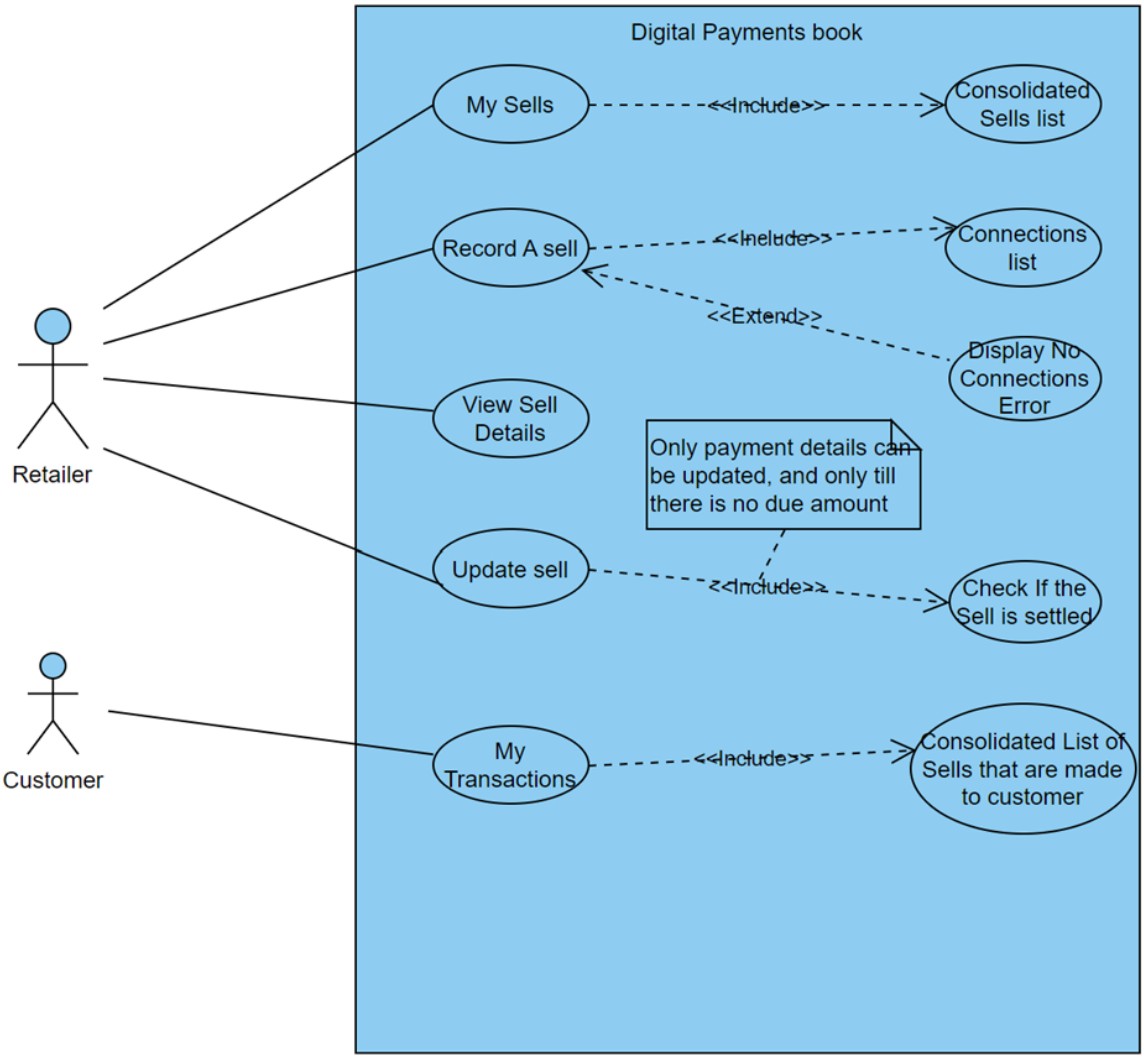

Figure 3.2: Use Case (2)

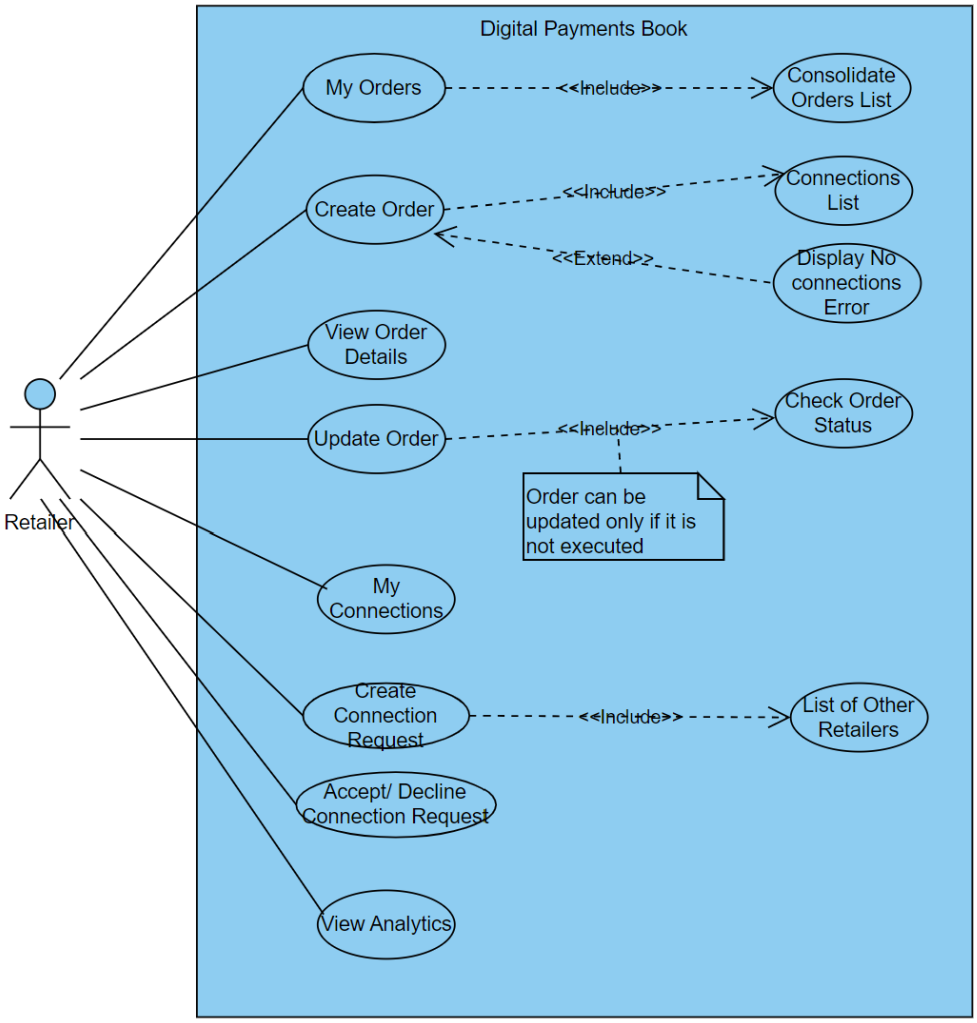

Figure 3.3: Use Case (3)

3.3.2 Entity Relationship Diagram

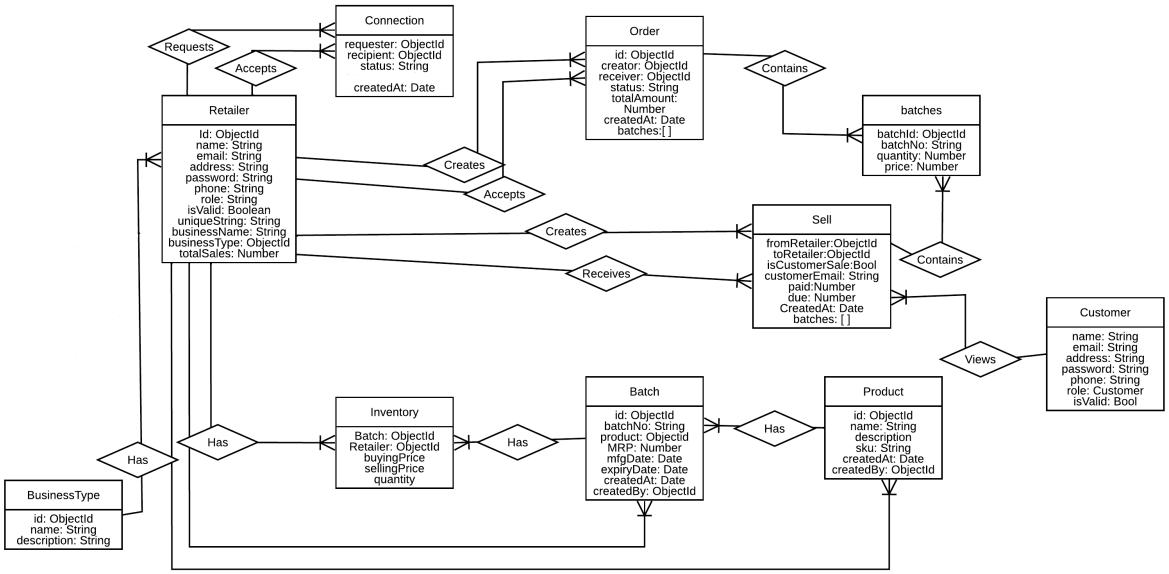

Figure 3.4: Entity Relationship

3.3.3 Activity Diagram

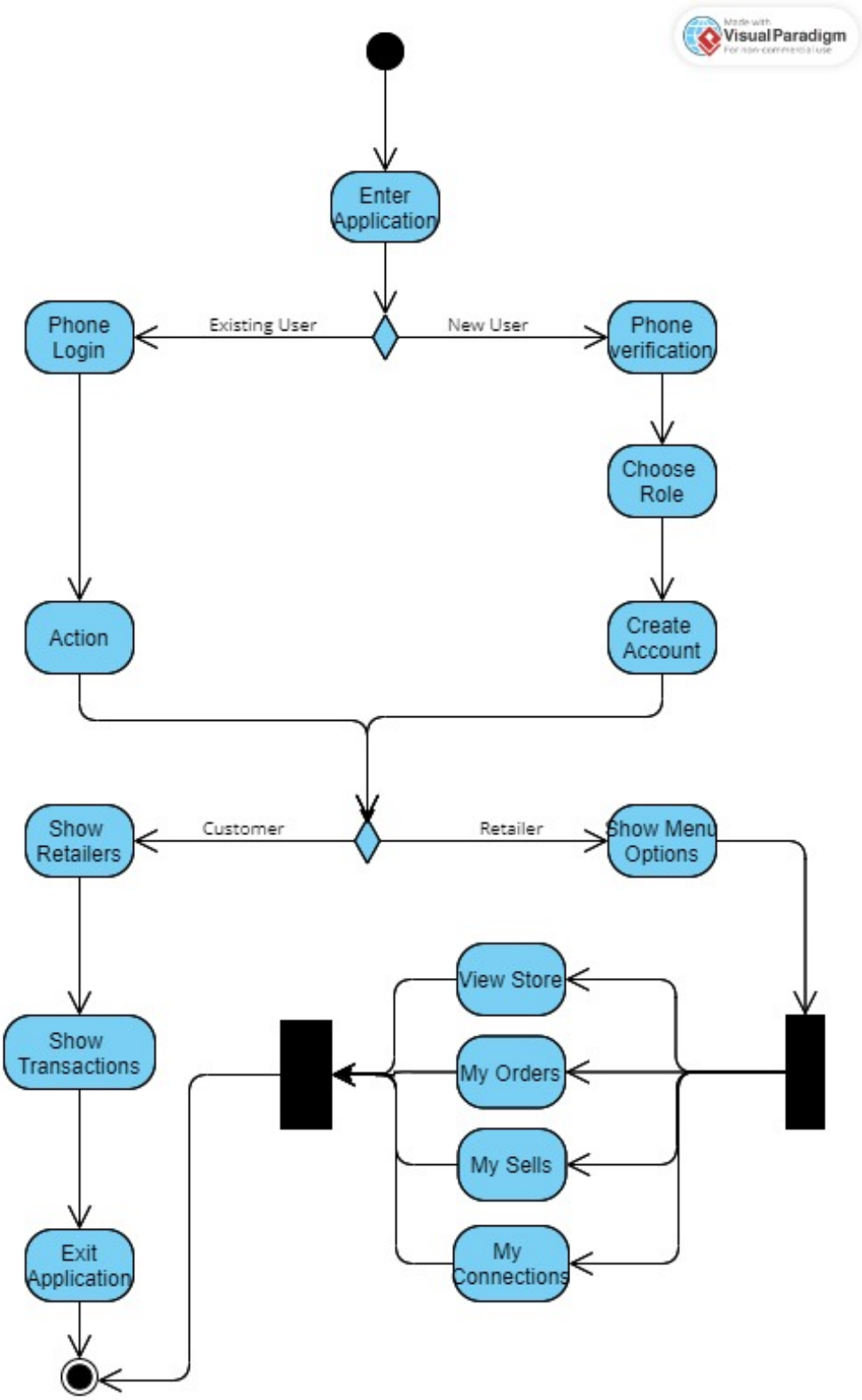

Figure 3.5: Activity Diagram for login and basic overview

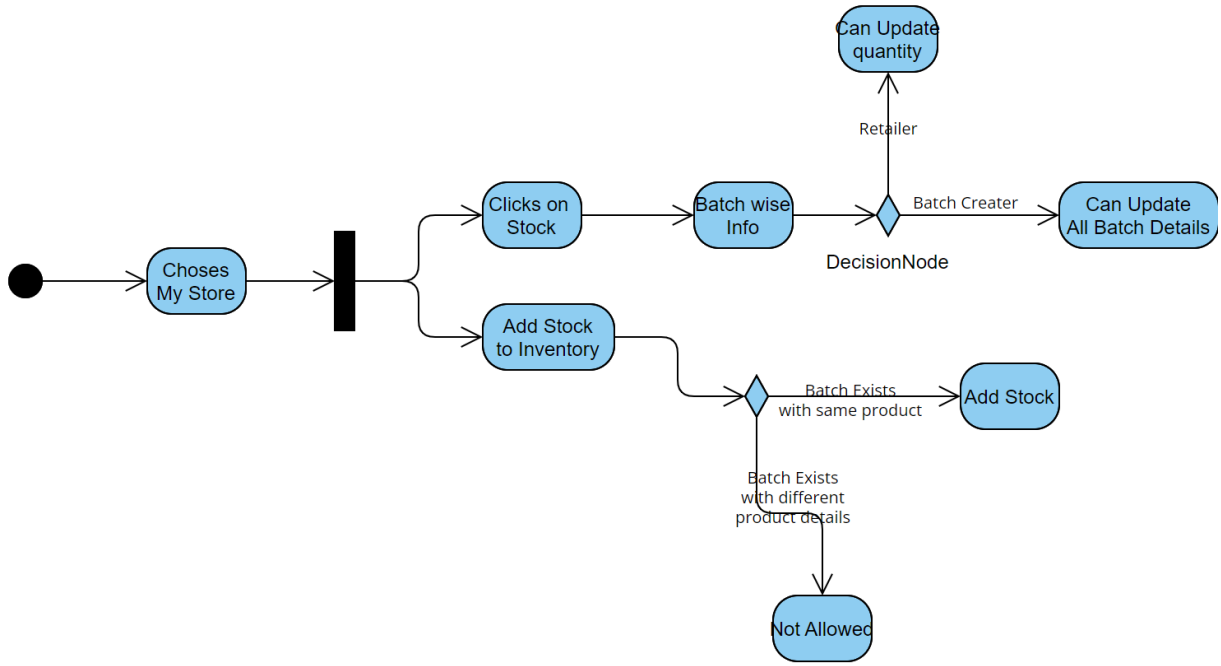

Figure 3.6: Activity Diagram for My store

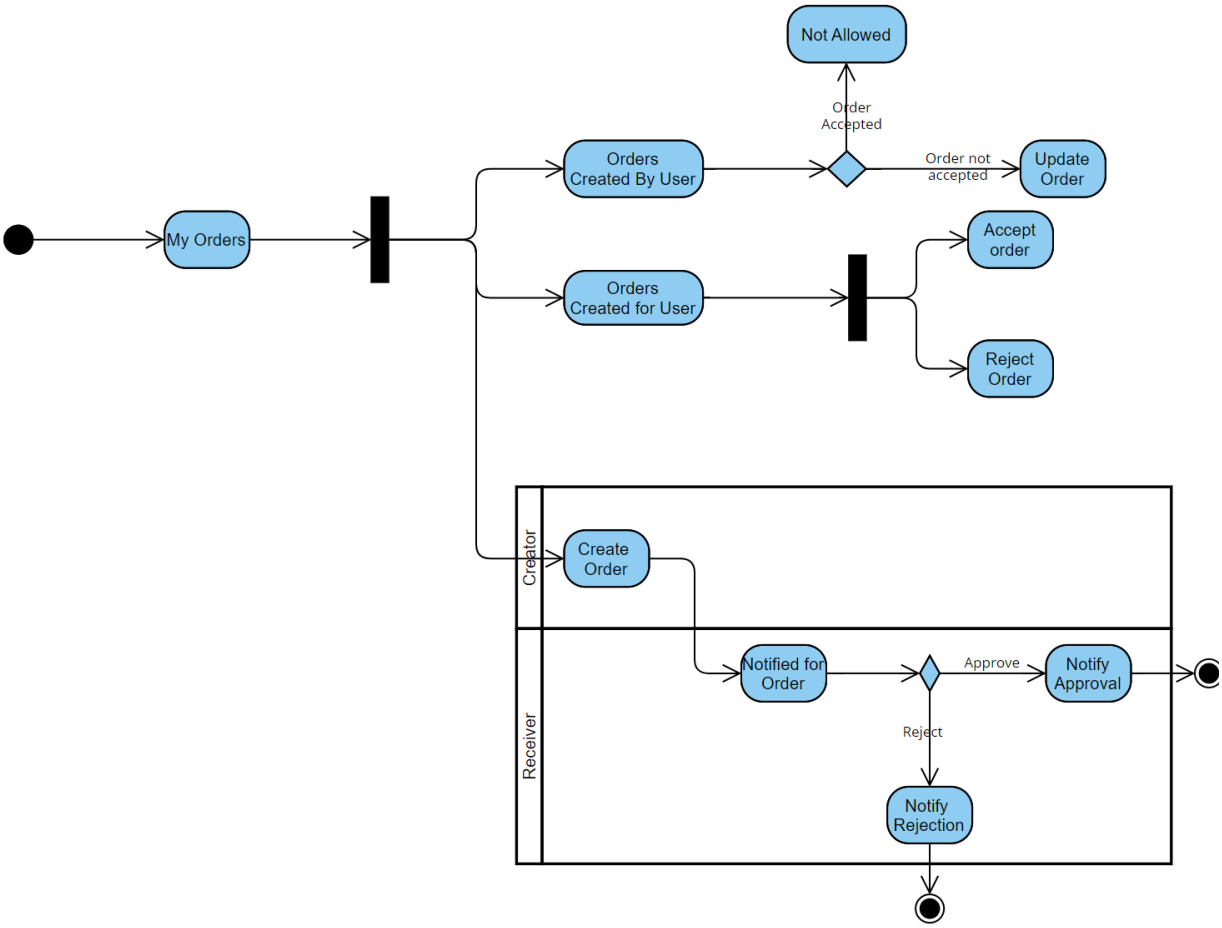

Figure 3.7: Activity Diagram for My Orders

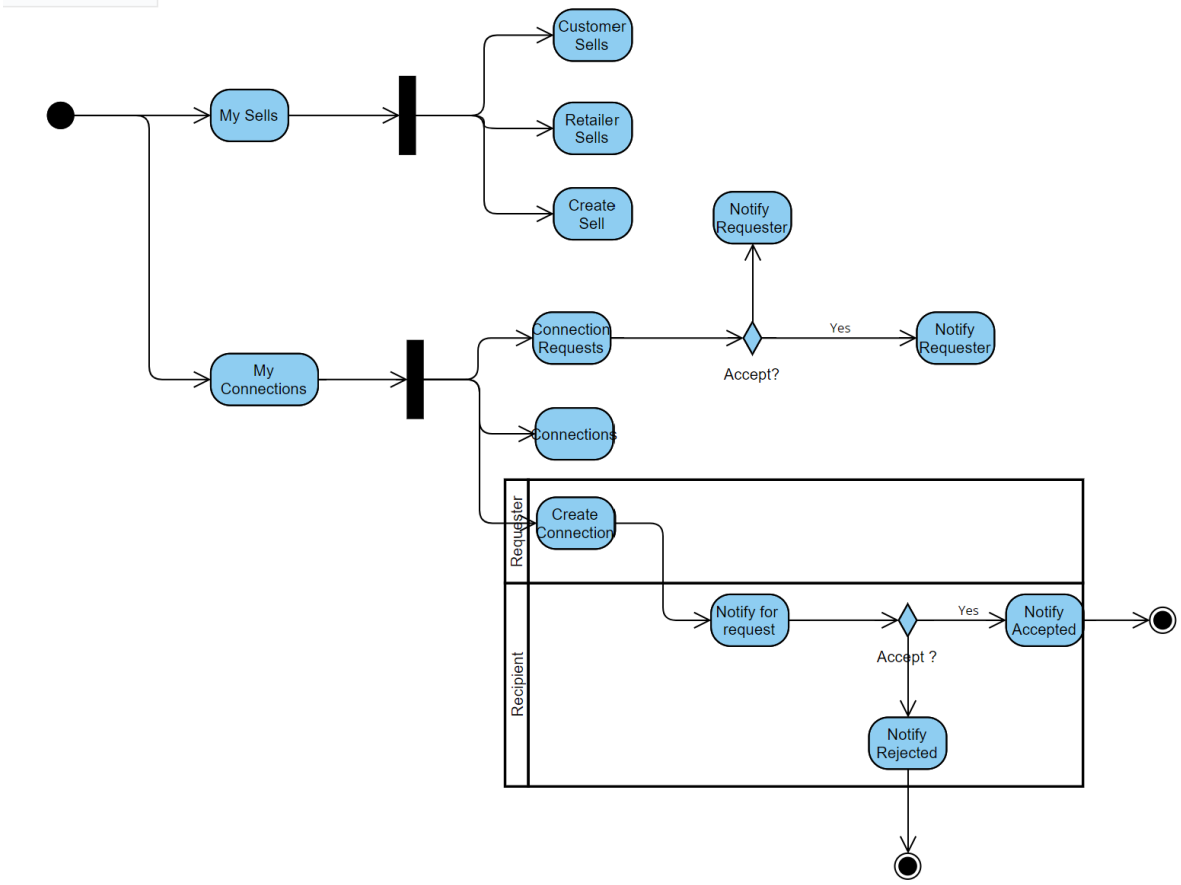

Figure 3.8: Activity Diagram for My Sells and Connections

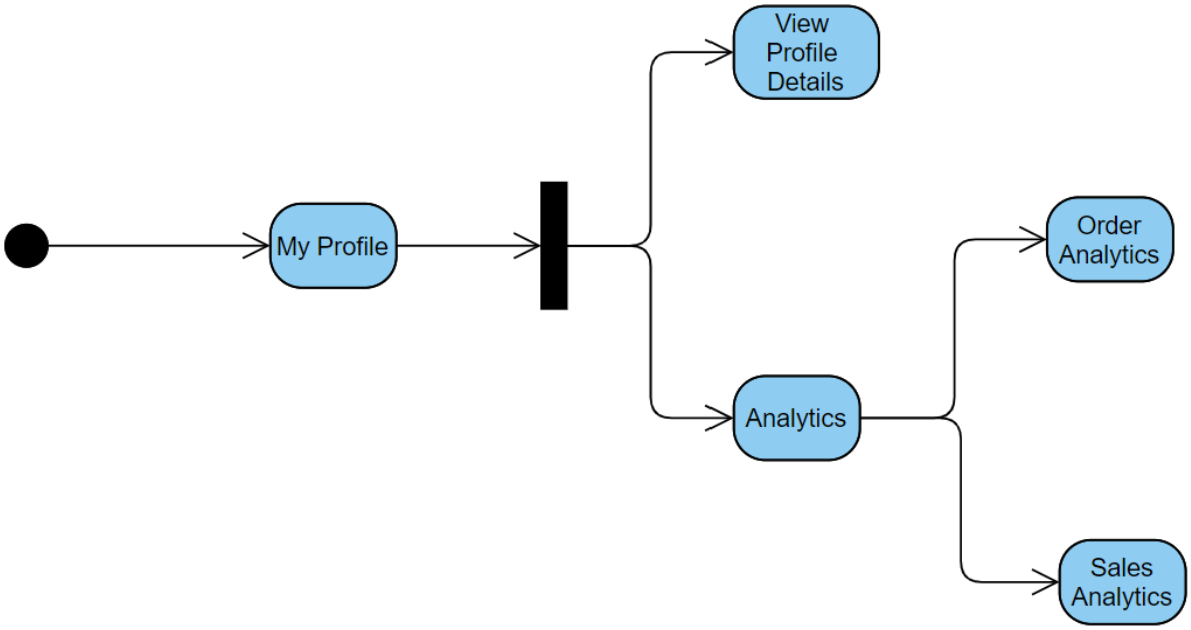

Figure 3.9: Activity Diagram for My Profile

3.3.4 Data-Flow Diagram

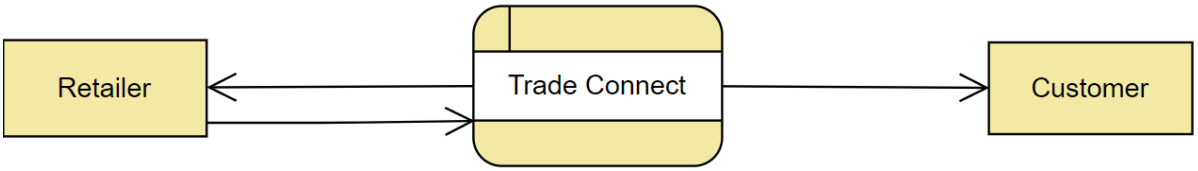

Figure 3.10: Level 0 Block Diagram

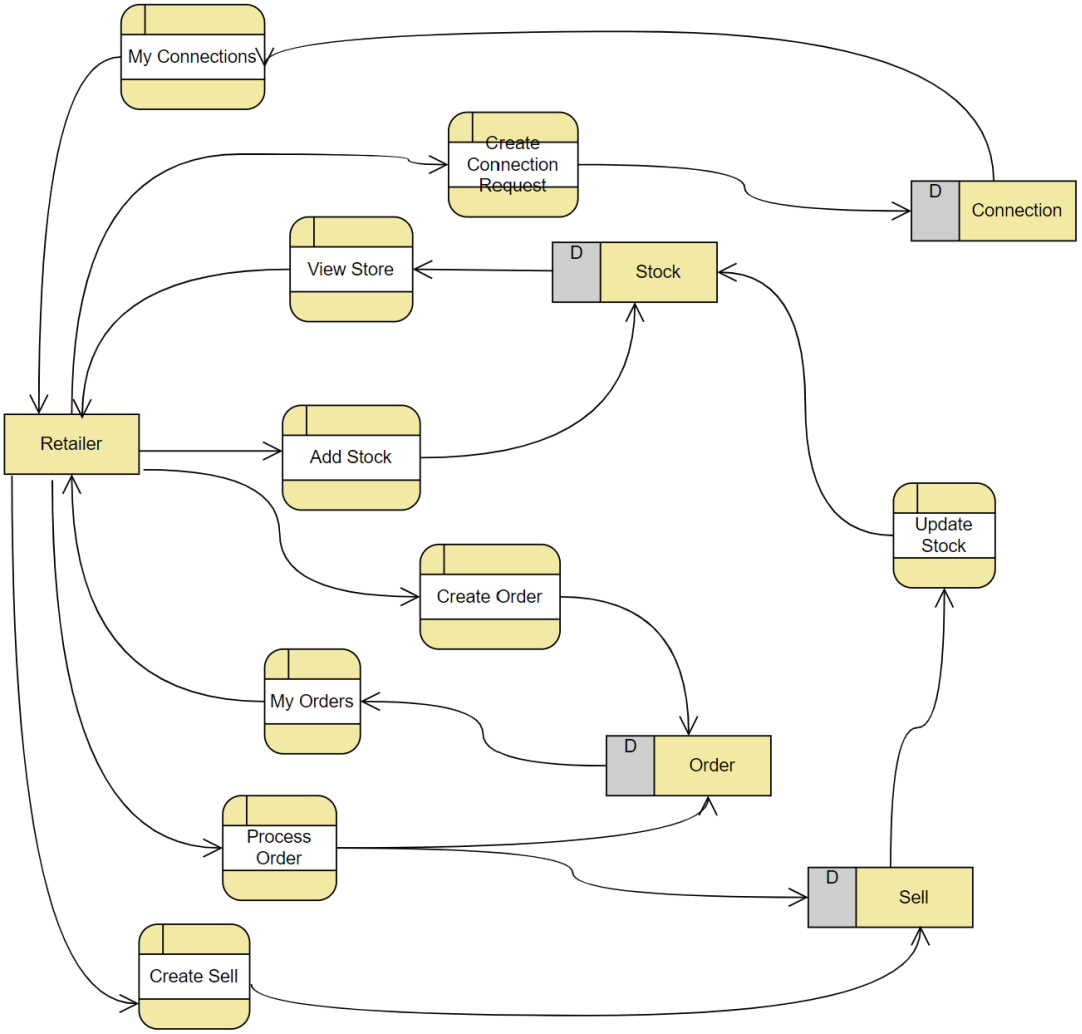

Figure 3.11: Level 1 Block Diagram

3.3.5 Class Diagram

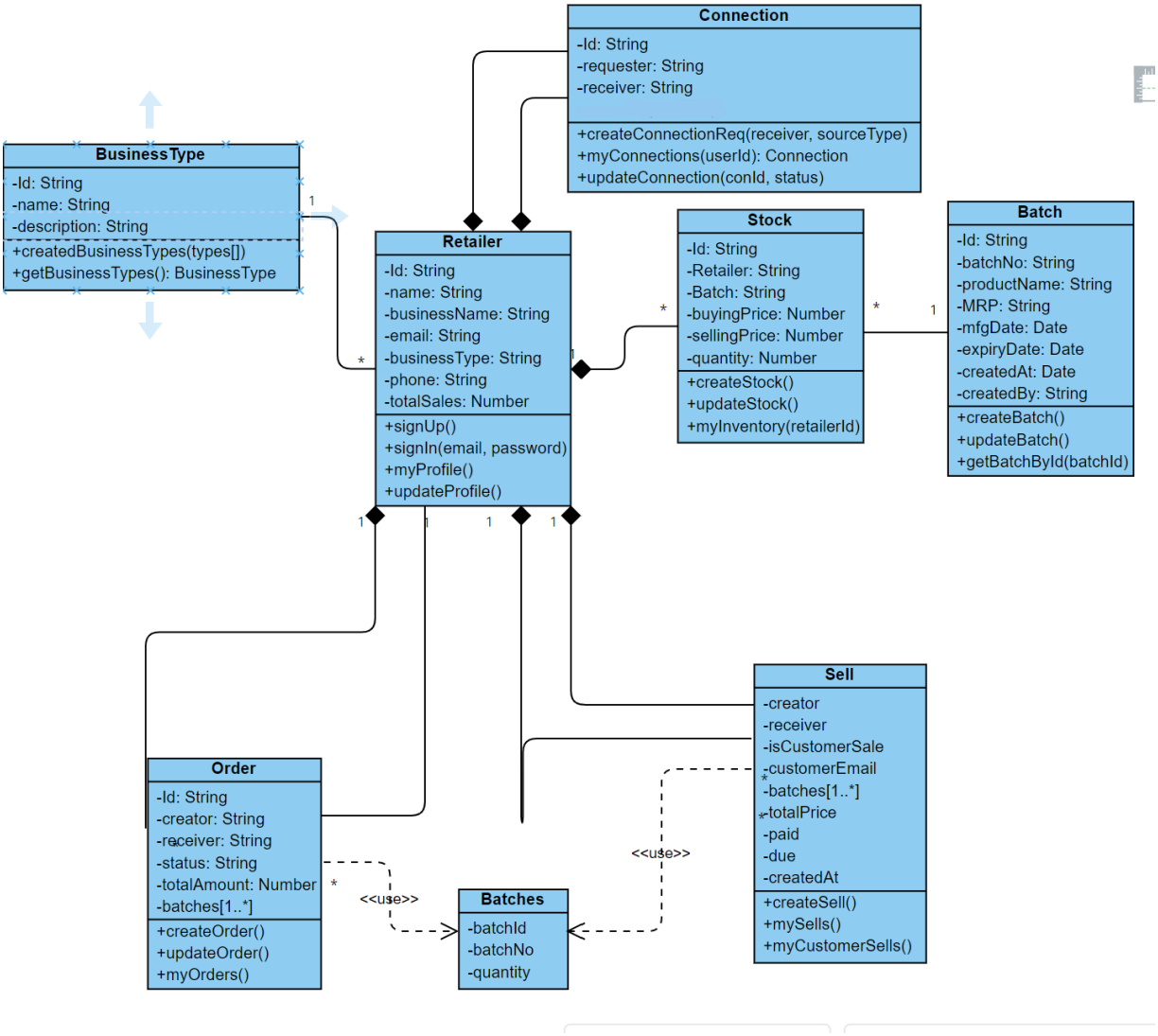

Figure 3.12: Class Diagram

3.3.6 Sequence Diagram

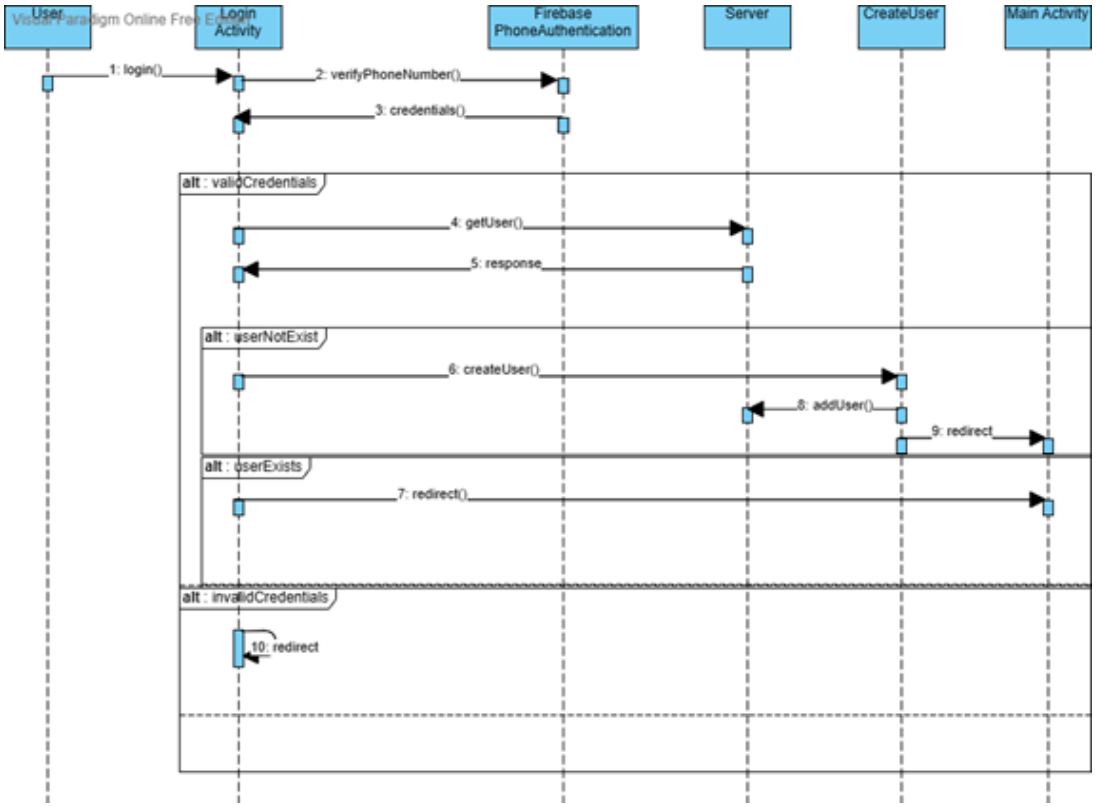

Figure 3.13: Sequence Diagram for Login

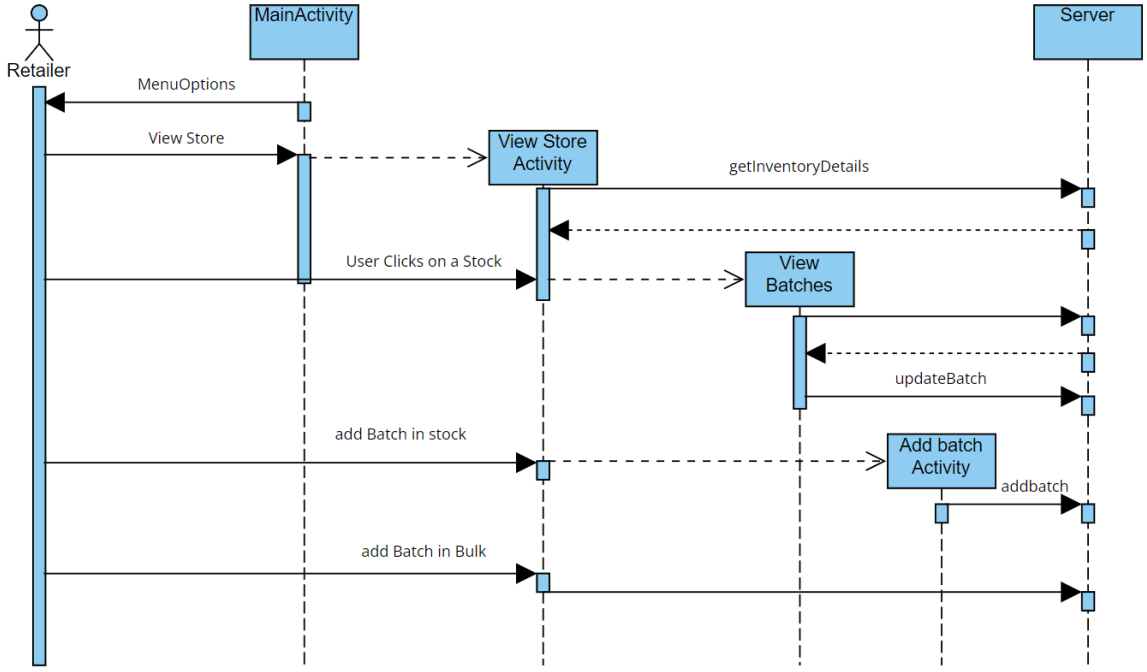

Figure 3.14: Sequence Diagram for My Store option for a Retailer

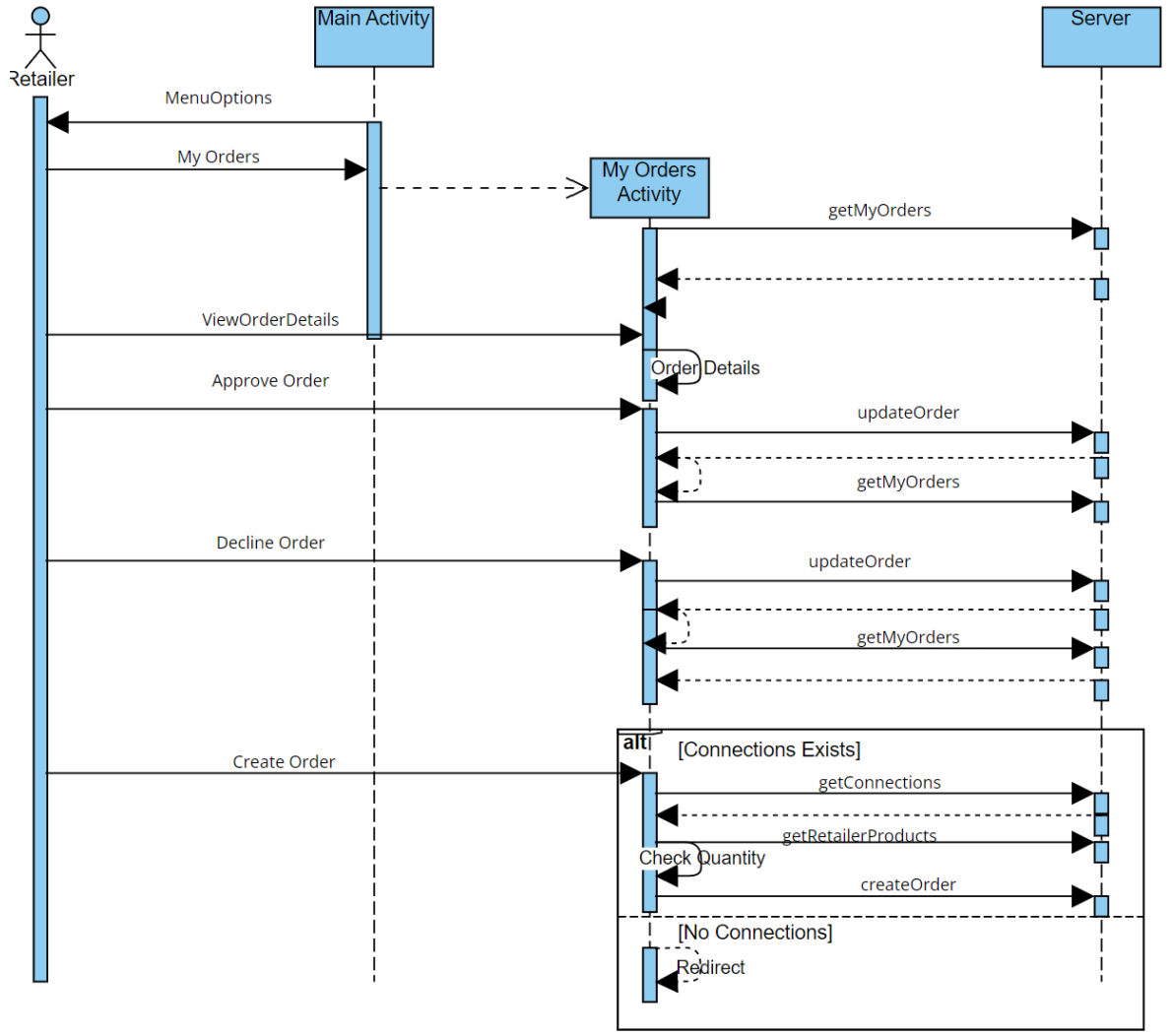

Figure 3.15: Sequence Diagram for My Orders

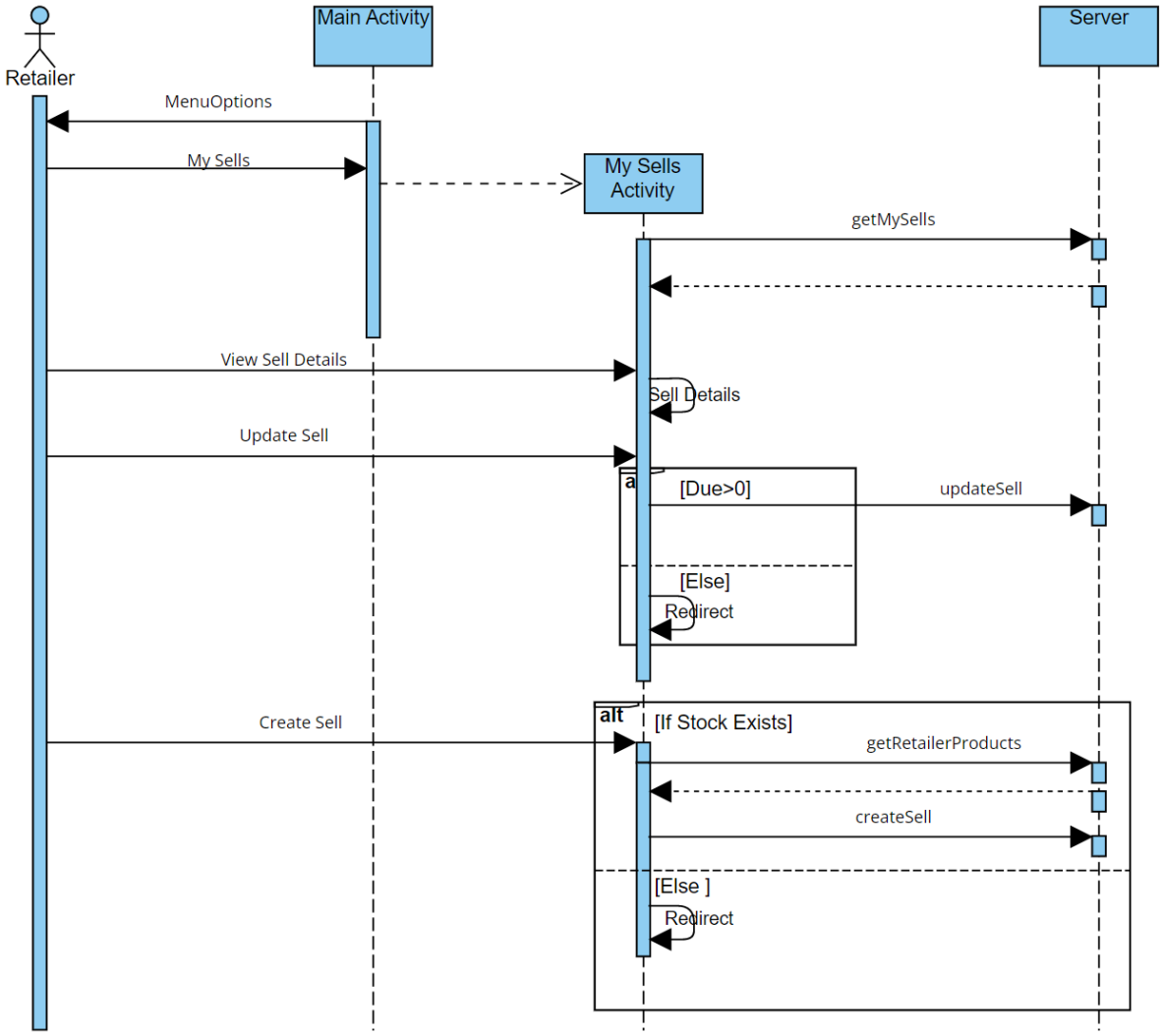

Figure 3.16: Sequence Diagram for Transactions of Customer and Retailer

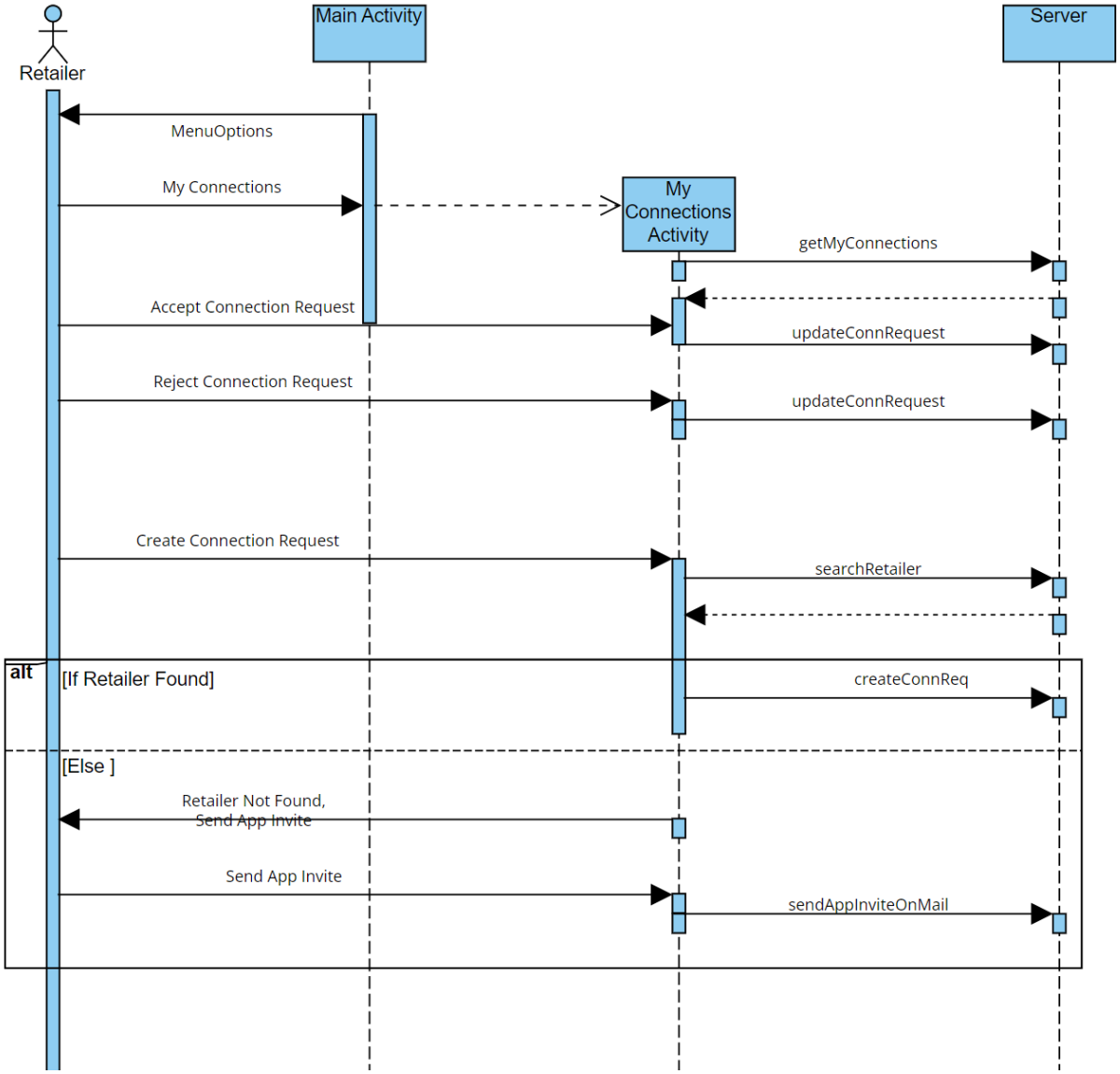

Figure 3.17: Sequence Diagram for Transactions of Customer and Retailer

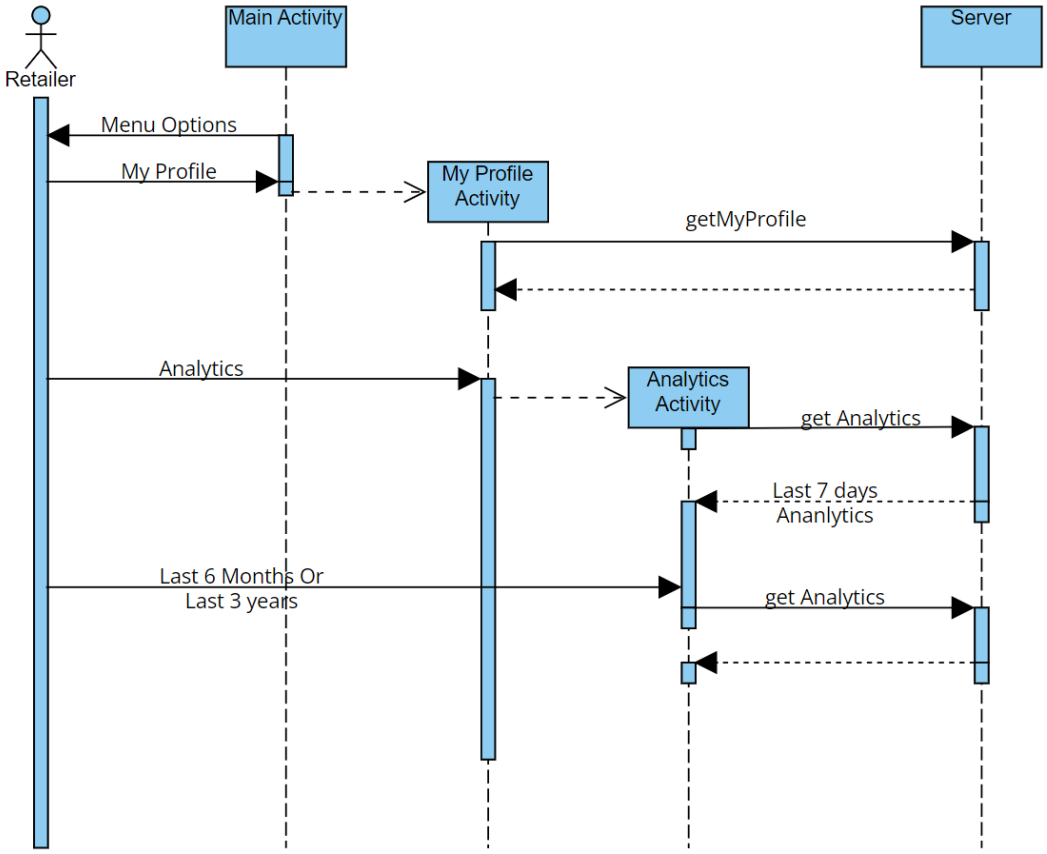

Figure 3.18: Sequence Diagram for Transactions of Customer and Retailer

Chapter 4

Methodology and Team

4.1 Introduction to Waterfall Framework

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as an input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.

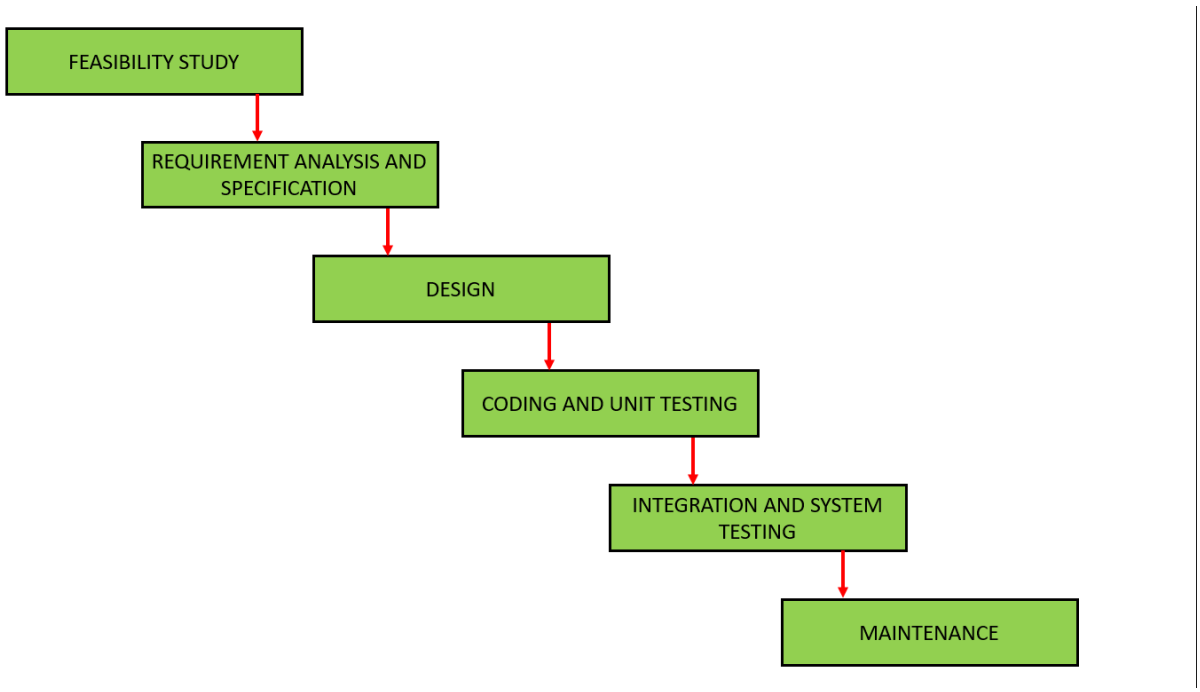

Figure 4.1: WaterFall model

The sequential phases in Waterfall model are-

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

3. **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

4. **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of system:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

6. **Maintenance:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name ”Waterfall Model”. In this model phases do not overlap.

Development Methodology - Waterfall Model for Retailer Management System:

1. Clear Requirements Gathering: The Waterfall Model allows for thorough requirements gathering at the beginning of the project. This is crucial for a retailer management system to understand the specific needs and functionalities required by retailers, such as inventory management, order processing, sales tracking, and customer management.

2. Designing Comprehensive Modules: With the Waterfall Model, the system's architecture and modules can be designed and planned in advance. This ensures that all essential functionalities, such as batch management, sales analytics, connection management, and payment processing, are incorporated into the system design.

3. Sequential Implementation: The Waterfall Model allows for a sequential implementation approach, where each module is developed and tested individually before integrating them into the complete system. This enables focused development and ensures that each module meets the required specifications.

4. Thorough Testing and Quality Assurance: As the Waterfall Model emphasizes completing each phase before moving forward, there is dedicated time for testing and quality assurance activities. This ensures that the retailer management system undergoes comprehensive testing, including unit testing, integration testing, and system testing, to identify and rectify any issues or bugs.

5. Documentation for User Training: The Waterfall Model promotes detailed documentation, which is crucial for a retailer management system. User manuals, guides, and training materials can be developed during the documentation phase, helping retailers understand and effectively utilize the system's features and functionalities.

6. Deployment and Maintenance: The Waterfall Model ensures a systematic approach to deployment, where the system is implemented and made available to retailers. Additionally, the comprehensive documentation produced during the development process aids in system maintenance and future enhancements, making it easier to address any issues or incorporate new features.

7. Time and Cost Management: The Waterfall Model allows for better estimation of project timelines and costs, as each phase is completed before proceeding to the next. This aids in project planning and resource allocation, ensuring that the retailer management system is delivered within the specified time and budget constraints.

8. Scalability and Customization: With a well-defined and planned development approach, the Waterfall Model enables scalability and customization of the retailer management system. As the system requirements are thoroughly understood and designed, it becomes easier to accommodate future growth and tailor the system to meet specific retailer needs.

9. Regulatory Compliance: For a retailer management system, compliance with industry regulations and standards is crucial. The Waterfall Model facilitates incorporating necessary compliance measures, such as data security, privacy regulations, and financial transaction protocols, during the development process.

10. Stakeholder Communication: The Waterfall Model allows for clear communication and collaboration with stakeholders throughout the project. Regular progress updates, milestone achievements, and documentation reviews ensure that all stakeholders, including retailers, can actively participate and provide feedback at appropriate stages.

Overall, the Waterfall Model provides a structured and systematic approach for the development of a retailer management system. It ensures that the system meets the specific requirements of retailers, is thoroughly tested, and can be deployed and maintained effectively.

Waterfall Model Pros & Cons

Advantage

- The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development

and a product can proceed through the development process model phases one by one.

- Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

4.2 Team Members, Roles & Responsibilities

I. Raghav Sharma(19ESKIT073)- Android Developer
Responsibilities

1. User Interface Design: Designing visually appealing and intuitive user interfaces that align with the app's branding and target audience. This includes creating wireframes, mockups, and interactive prototypes to demonstrate the app's layout and navigation flow.

2. User Experience Research: Conducting user research, interviews, and surveys to understand user needs, preferences, and pain points. Using this feedback to inform design decisions and create user-centered experiences.

3. Information Architecture: Organizing and structuring app content in a logical and user-friendly manner. Creating navigation schemes, menus, and information hierarchies to ensure smooth and efficient user interactions.

4. Communicating with backend team: Getting requirements form the backend tema to integrate features in the mobile app, and populating the UI with data fetched through the REST APIs.

5. Continuous Improvement: Staying updated with the latest UI and UX design trends, best practices, and emerging technologies. Continuously refining and

enhancing the UI and UX based on user feedback, analytics, and industry standards.

II. Mukul Jangid - Node.js Developer

Responsibilities

1. Database design: Designing Database schema, relations between the entities in the system. Developing Entity relationship diagrams, writing mongodb scripts for database models.
2. Backend design: Designing the backend system and setting project conventions.Documenting requirements from the frontend team and writing APIs for those requirements.
3. Business Logic development:Developing the business logic according to requirements gathered.Creating reusable controllers and functions.
4. API testing and documentation: Creating documentation for the APIs in the postman tool for frontend team's integration, and testing them on sample data.
5. Continuos improvement: Improving APIs according to any requirement change of the frontend team.

Chapter 5
Centering System Testing

The designed system has been testing through following test parameters.

5.1 Functionality Testing

Functionality testing of the TradeConnect app ensures that all its features and functionalities are working as intended. Here are some key areas to consider when performing functionality testing for the app:

1. Add Products:
   (a) Test the ability to add single batch product or bulk upload data for multiple batches.
   (b) Verify that batch no. search gives the correct data and UI is populated accordingly.
   (c) Test the app's ability to handle various scenarios, such as MFG date greater tah Expiry date, and input of unsupported data.
   (d) Test the bulk upload through excel file. Generating correct request body from the file in predefined format.
2. Analytics:
   (a) Test the app's ability to generate correct graphical data, for both bar and pie charts.
   (b) Test the app's ability to handle different types of graphical data like, weekly, monthly and yearly.
3. Creating Sell and Orders:
   (a) Test the functionality to generate orders and sells for retailers and customers.

(b) Verify that users get notifications on their devices and mails whenever a new order or sell is created for them.

(c) Test the ability to handle various user input edge cases, for example sell or order quantity can't be greater than the batch quantity etc.

4. Connection Requests:

(a) Test the ability for Accepting or Declining connection requests.

(b) Verify that users get notifications on their devices and mails whenever a new connection request is received or accepted.

(c) Test the functionality of searching the user by Name, phone and Shop address and sending connection request.

5. User Interface and Navigation:

(a) Test the app's user interface to ensure it is intuitive, responsive, and visually appealing.

(b) Verify that users can easily navigate through different sections of the app, access key features, and perform actions without confusion.

(c) Test the responsiveness of the app across different devices and screen sizes.

6. Notifications and Alerts:

(a) Test the app's ability to send notifications and alerts to users regarding connection requests, orders and sells.

(b) Verify that notifications are delivered promptly and accurately, helping users stay informed about their financial status.

7. Security and Privacy:

(a) Test the app's security measures, such as encryption of sensitive data and secure authentication processes for example using JWT token for each request that is being made to the server.

(b) Verify that user data is protected from unauthorized access or data breaches.

8. Error Handling and Exception Cases:

   (a) Test the app's behavior in handling invalid inputs, unexpected errors, and exceptional cases.

   (b) Verify that appropriate error messages are displayed, and the app gracefully recovers from errors without crashing or losing data.

5.2 Performance Testing

1. Response Time:

   - The app consistently provides fast response times for key operations such as order, sell creation and populating various lists in the app.
   - Response times meet or exceed the defined performance objectives, ensuring a smooth user experience.

2. Stability:

   - The app exhibits stability and robustness during prolonged testing, sustaining continuous usage without crashes or significant performance issues.
   - Memory leaks, resource leaks, and other stability concerns are minimized, providing a reliable and consistent user experience.

3. Network and Connectivity:

   - The app demonstrates resilience to network disruptions or intermittent connectivity issues, recovering gracefully and maintaining data synchronization.
   - It efficiently handles low bandwidth or high-latency conditions, ensuring optimal performance under different network scenarios.

4. Load Testing:

   - Simulate a high number of concurrent users and transactions to assess the system's performance under heavy load.

- Measure response times and throughput to identify any performance bottlenecks or areas of improvement.
- Verify that the system can handle the expected load without significant degradation in performance.

5. Stress Testing:

   - Push the system beyond its normal operating capacity to determine its stability and resilience.
   - Evaluate the system's behavior under extreme loads, such as a sudden spike in user traffic or a high volume of transactions.
   - Monitor resource utilization, such as CPU and memory usage, to ensure the system can handle the stress without crashes or failures.

6. Database Performance:

   - Evaluate the performance of database operations, such as querying, updating, and inserting records.
   - Measure query execution times and assess the efficiency of database indexes and query optimization.
   - Monitor database resource utilization, such as CPU, memory, and disk I/O, to identify any performance issues.

7. Error Handling and Exception Testing:

   - Simulate various error scenarios, such as invalid requests, network failures, or database errors.
   - Verify that the system gracefully handles errors and exceptions without crashing or compromising data integrity.
   - Evaluate error logging and reporting mechanisms to ensure they provide useful information for debugging and troubleshooting.

7. Performance Monitoring and Analysis:

- Implement performance monitoring tools and collect relevant metrics during testing.
- Analyze performance data to identify performance bottlenecks, resource utilization patterns, and areas for optimization.
- Use performance profiling tools to pinpoint specific areas of code or database queries that may require optimization.

8. Recommendations:

   - Based on the performance testing results, specific recommendations for optimization, scalability improvements, or architectural changes may be provided to further enhance the app's performance.

It's important to note that the actual results of performance testing will depend on various factors such as the app's complexity, underlying infrastructure, network conditions, and the test environment. Conducting thorough performance testing and analyzing the specific metrics and measurements will provide more accurate and actionable results for the app.

5.3 Usability Testing

Usability testing for the TradeConnect app focuses on assessing its user-friendliness, ease of use, and overall user experience. It aims to identify any usability issues or areas for improvement. Here are some key aspects for conducting usability testing:

1. Test Setup:

   - Define the target user group for the app and recruit participants who represent the intended user base.
   - Create realistic test scenarios and tasks that align with the app's primary features and functionalities.

2. User Interface (UI) Evaluation:

- Assess the app's UI design, layout, and visual elements for clarity, consistency, and intuitiveness.
- Evaluate the placement of navigation menus, buttons, and interactive elements to ensure ease of use and discoverability.
- Test the app's responsiveness and adaptability across different devices and screen sizes.

3. Navigation and Flow:
    - Evaluate the ease of navigating through different sections and screens of the app.
    - Test the app's menu structure, search functionality, and back/forward navigation to ensure a seamless user experience.
    - Verify that users can accomplish tasks and access information efficiently without getting lost or confused.
4. Input and Data Entry:
    - Test the ease and accuracy of entering data, such as creating orders and sells.
    - Verify that input fields are labeled clearly, validation messages are displayed appropriately, and error handling is user-friendly.
    - Evaluate the use of input controls, such as date pickers, drop-down lists, for intuitiveness and ease of interaction.
5. Feature Usage and Functionality:
    - Assess how easily users can understand and utilize the app's key features and functionalities.
    - Test the app's ability to perform common tasks, such as adding and updating batches and sells.
    - Verify that users can access and understand the information presented, such as data analytics' graphical data for their orders and sells.

6. User Feedback and Satisfaction:

    - Gather user feedback through surveys, interviews, or feedback forms to understand user satisfaction and identify areas for improvement.
    - Analyze user feedback to identify common pain points, usability issues, or feature requests.
    - Use user feedback as input for iterative design improvements and future enhancements.

Chapter 6
Test Execution Summary

Execution Test Summary Report is an overall view of Testing Process from start to end. Test Plan comes at the starting of project while Test Summary Report comes at the end of the testing process. This report is given to the client for his understanding purpose. The Test Summary Report contents are : Test Case ID generated, Total number of resources consumed, Passed Test Cases, Failed Test Cases, Status of Test Cases, etc.

Test Execution Summary for TradeConnect app:

Test Case ID: TC001
Total Number of Resources Consumed: 10 hours
Passed Test Cases: 25
Failed Test Cases: 3
Status of Test Cases: Completed

Test Case ID: TC002
Total Number of Resources Consumed: 8 hours
Passed Test Cases: 18
Failed Test Cases: 2
Status of Test Cases: Completed

Test Case ID: TC003
Total Number of Resources Consumed: 12 hours
Passed Test Cases: 30
Failed Test Cases: 1
Status of Test Cases: Completed

Test Case ID: TC004
Total Number of Resources Consumed: 6 hours

Passed Test Cases: 15

Failed Test Cases: 0

Status of Test Cases: Completed

Test Case ID: TC005

Total Number of Resources Consumed: 10 hours

Passed Test Cases: 22

Failed Test Cases: 1

Status of Test Cases: Completed

Test Case ID: TC006

Total Number of Resources Consumed: 8 hours

Passed Test Cases: 20

Failed Test Cases: 0

Status of Test Cases: Completed

Test Case ID: TC007

Total Number of Resources Consumed: 6 hours

Passed Test Cases: 12

Failed Test Cases: 3

Status of Test Cases: Completed

Test Case ID: TC008

Total Number of Resources Consumed: 4 hours

Passed Test Cases: 10

Failed Test Cases: 0

Status of Test Cases: Completed

Test Case ID: TC009

Total Number of Resources Consumed: 8 hours

Passed Test Cases: 18

Failed Test Cases: 1

Status of Test Cases: Completed

Test Case ID: TC010

S.No	Test Case Id	Test Case Status	Passed Test Cases	Failed Test Cases	No. of Resources Consumed
1	TC001	Completed	25	3	10h
2	TC002	Completed	18	2	8h
3	TC003	Completed	30	1	12h
4	TC004	Completed	15	0	6h
5	TC005	Completed	22	1	10h
6	TC006	Completed	20	0	8h
7	TC007	Completed	12	3	6h
8	TC008	Completed	10	0	4h
9	TC009	Completed	18	1	8h
10	TC010	Completed	14	0	6h

Table 6.1: Test Execution Summary

Total Number of Resources Consumed: 6 hours

Passed Test Cases: 14

Failed Test Cases: 0

Status of Test Cases: Completed

Overall Test Execution Summary:

Total Number of Resources Consumed: 78 hours

Total Passed Test Cases: 184

Total Failed Test Cases: 11

The test execution for the Money Manager app has been completed, and the summary shows the overall results.

Chapter 7

Project Screen Shots

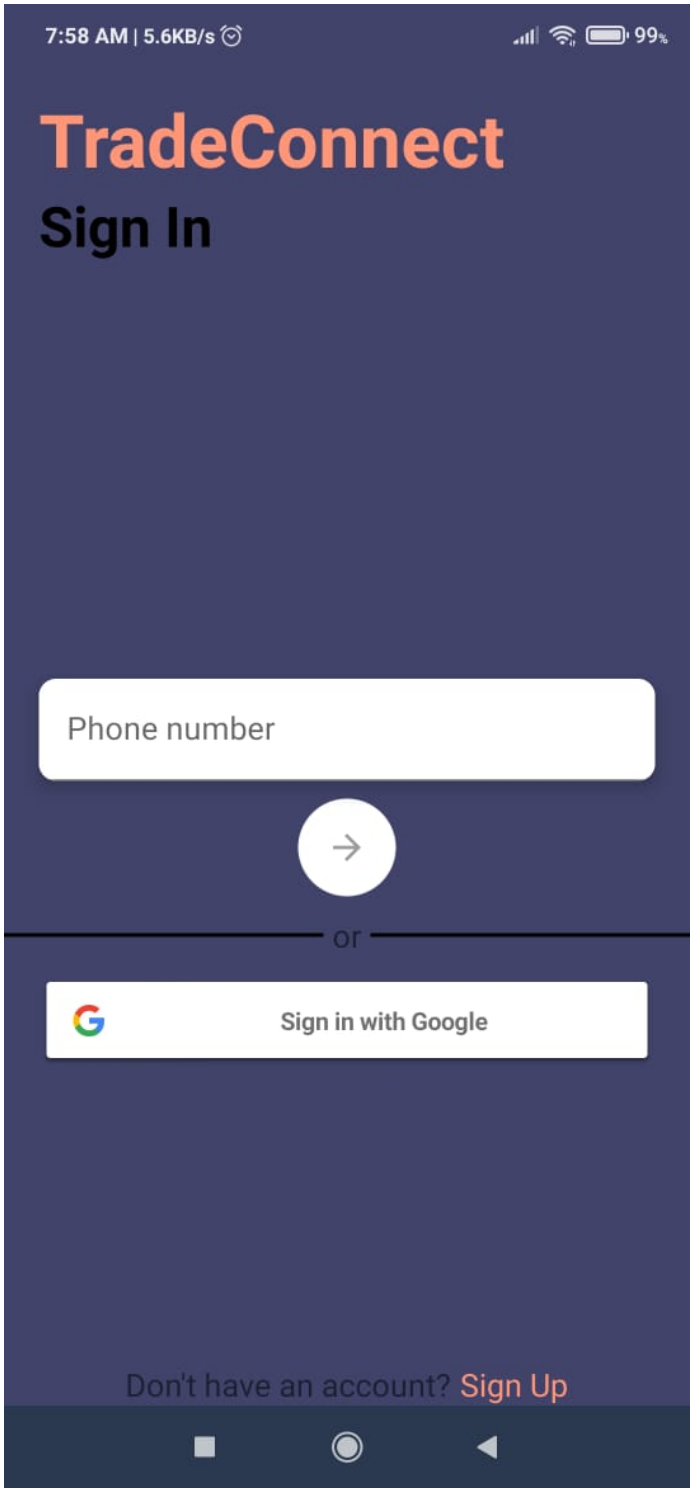

Figure 7.1: SignIn

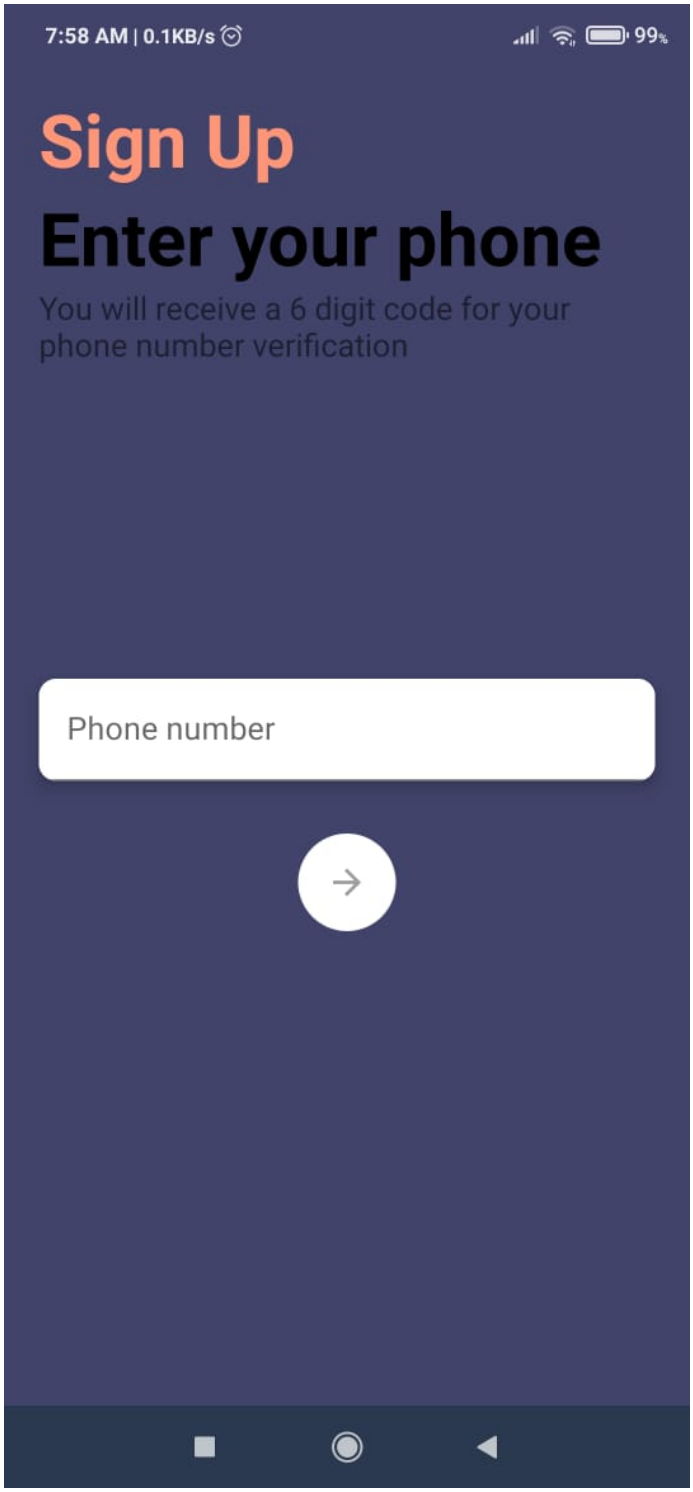

Figure 7.2: SignUp

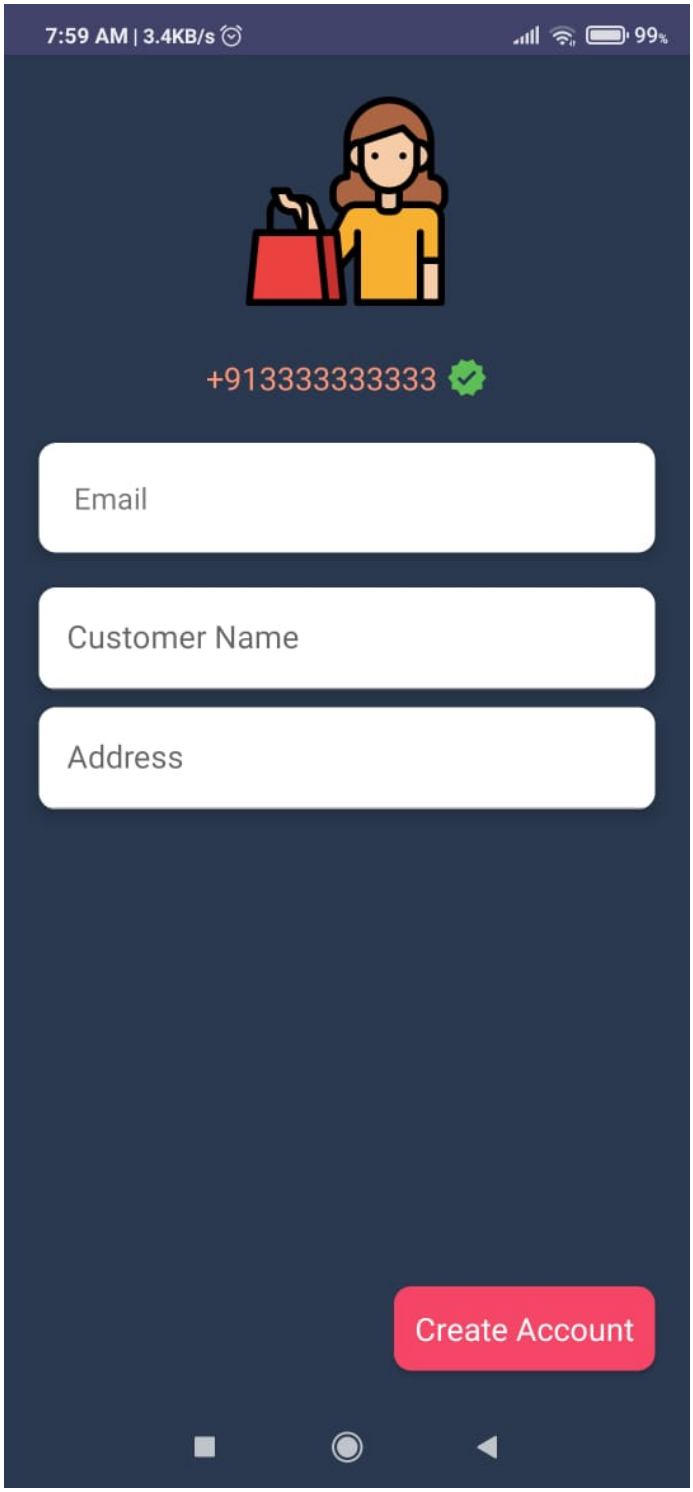

Figure 7.3: Create Customer

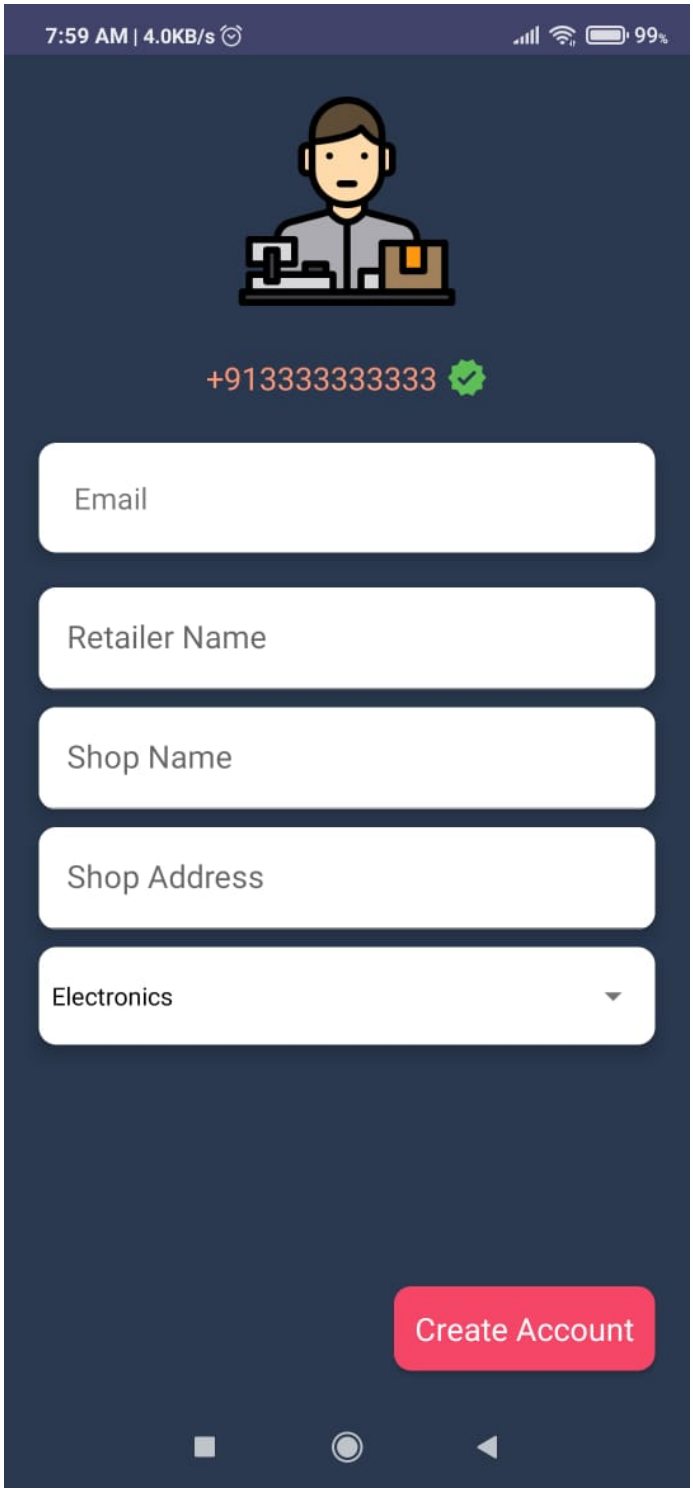

Figure 7.4: Create Retailer

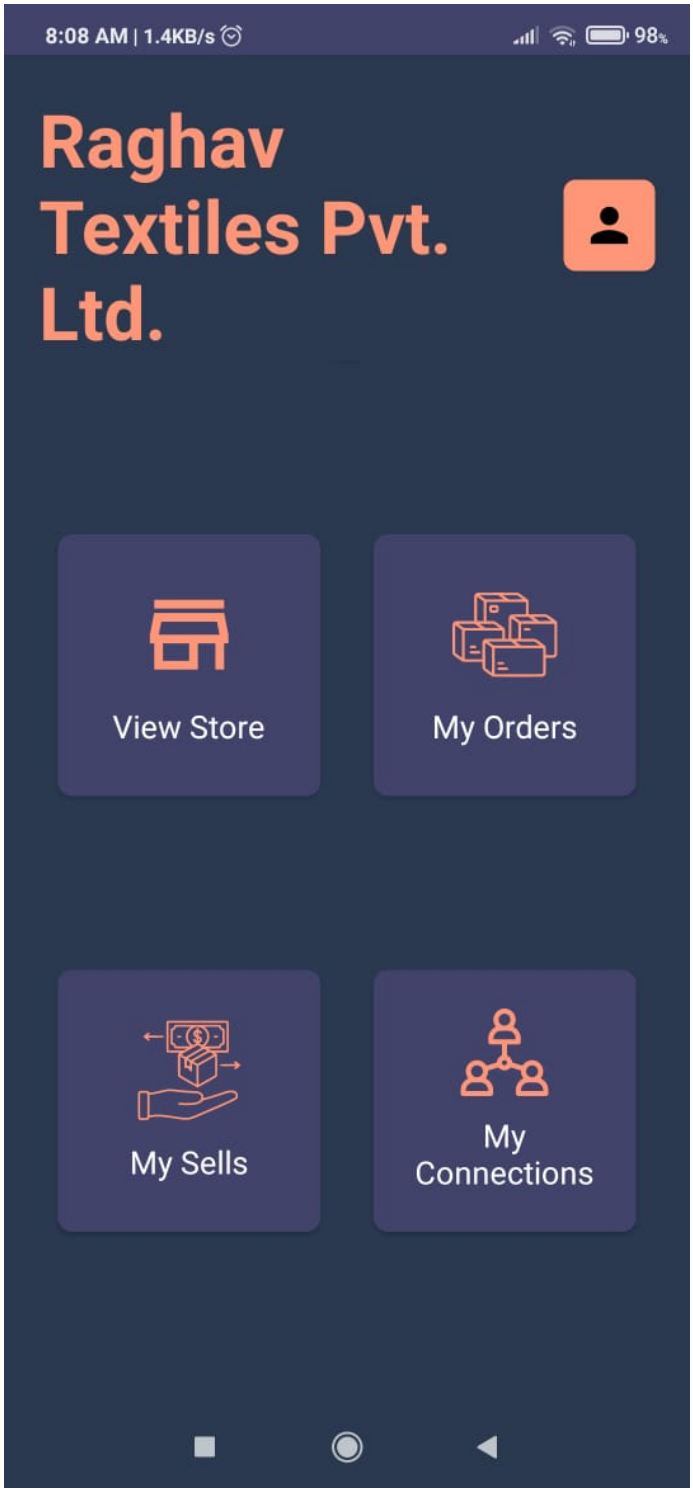

Figure 7.5: Home Screen

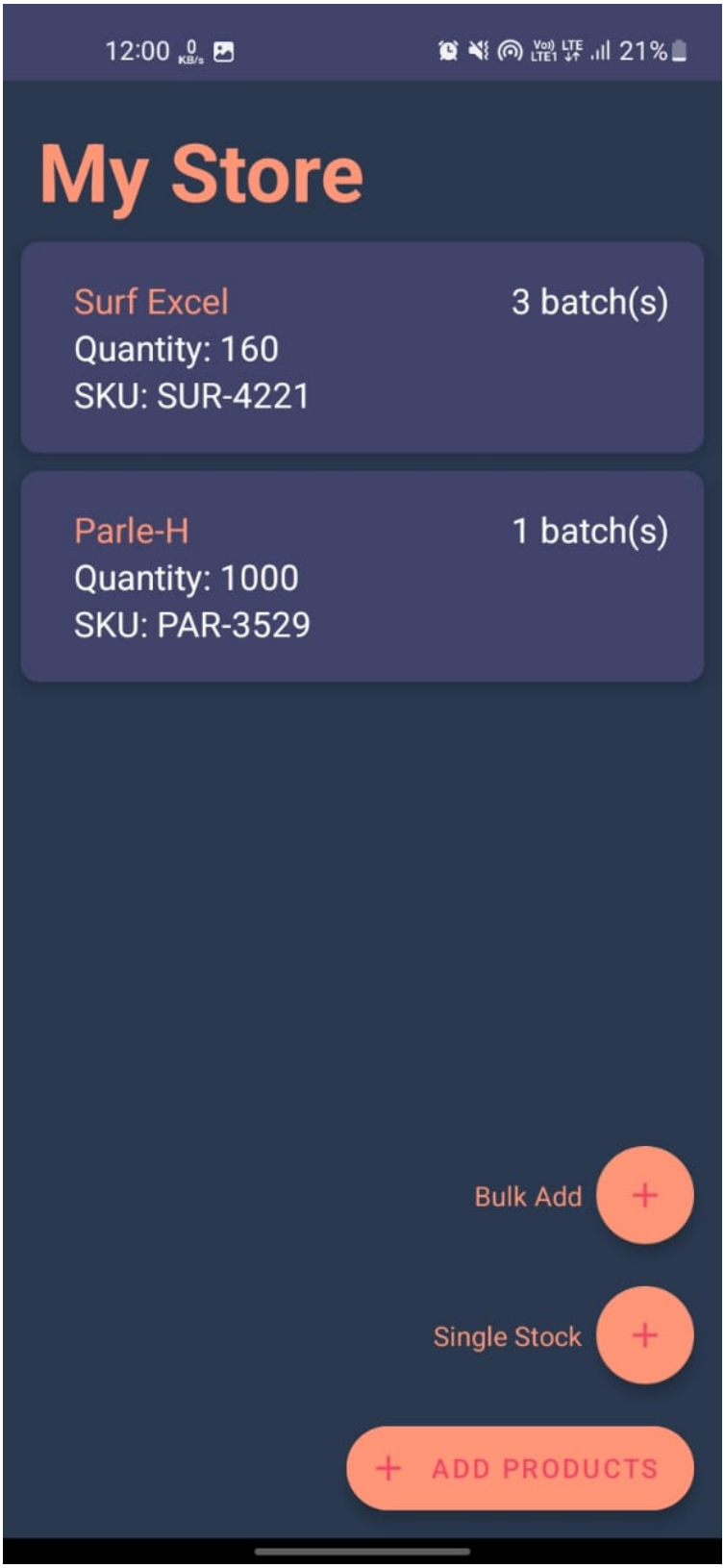

Figure 7.6: MyStore

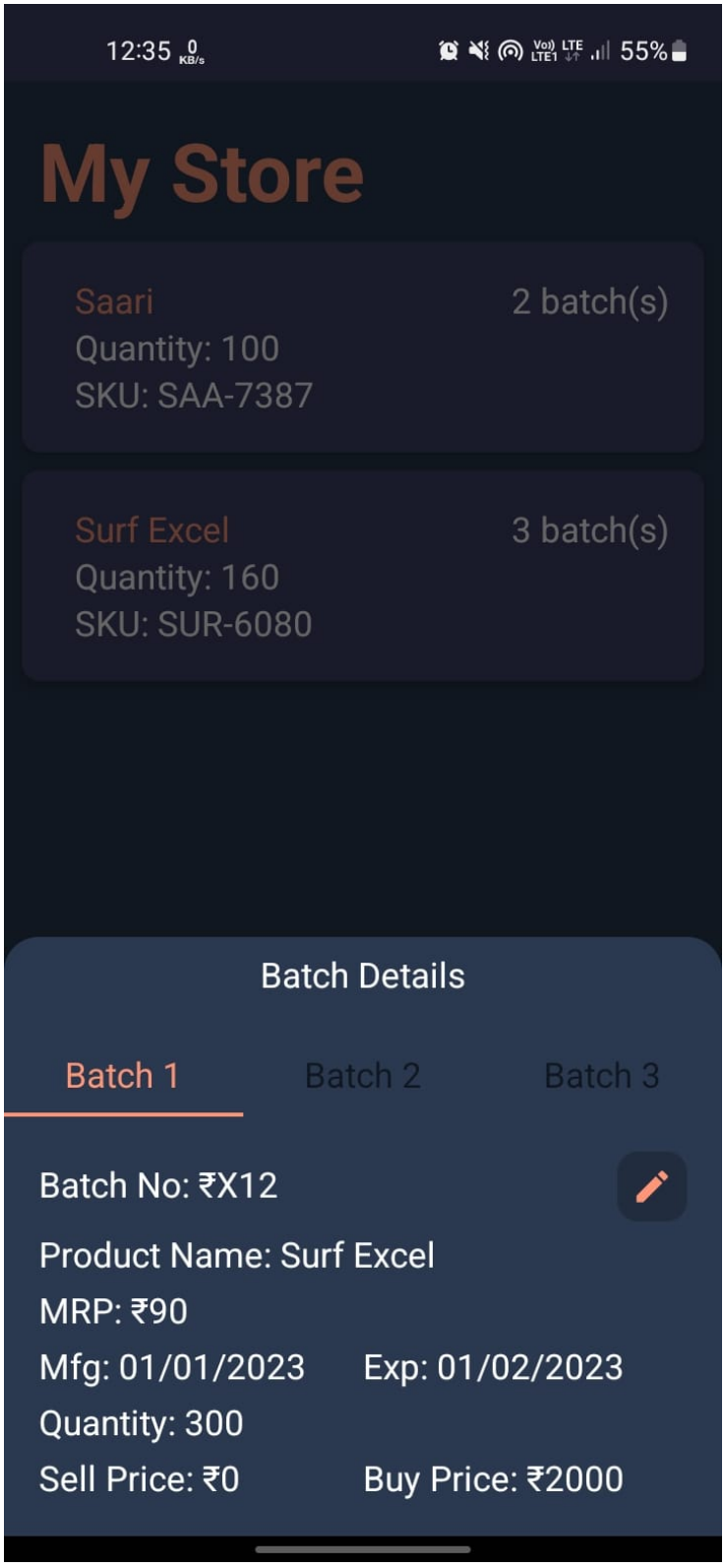

Figure 7.7: My Store Batch Details

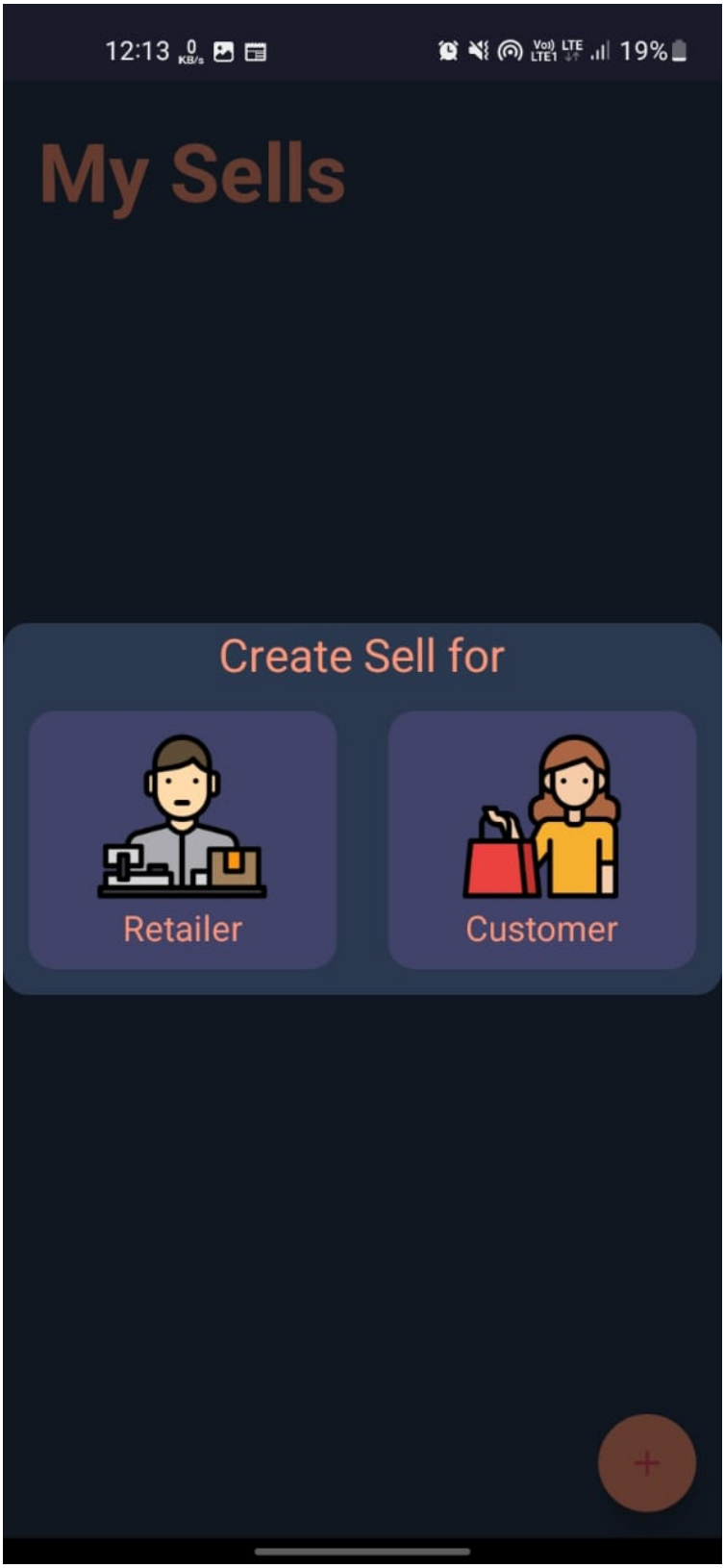

Figure 7.8: My Sell Create Sell

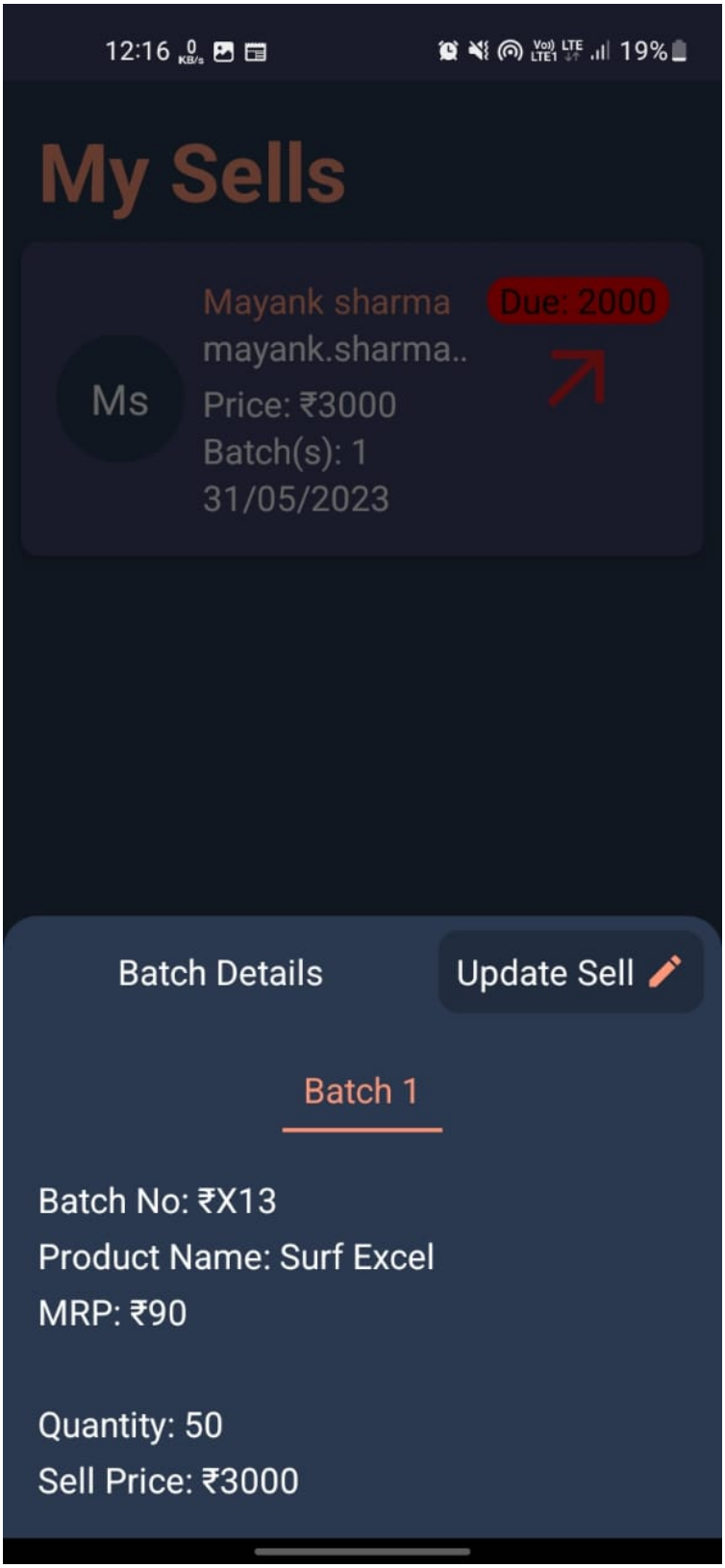

Figure 7.9: My Sell Batch Details

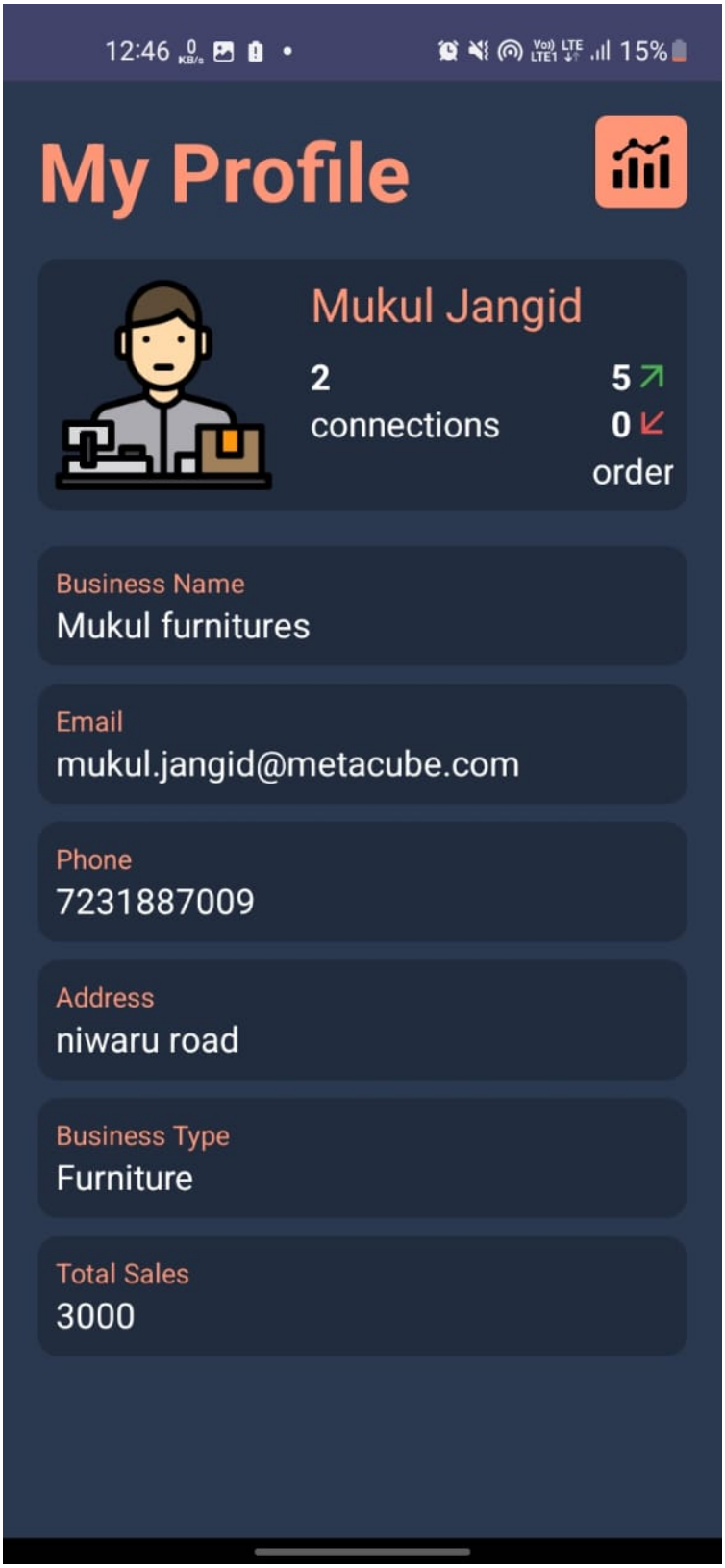

Figure 7.10: MyProfile

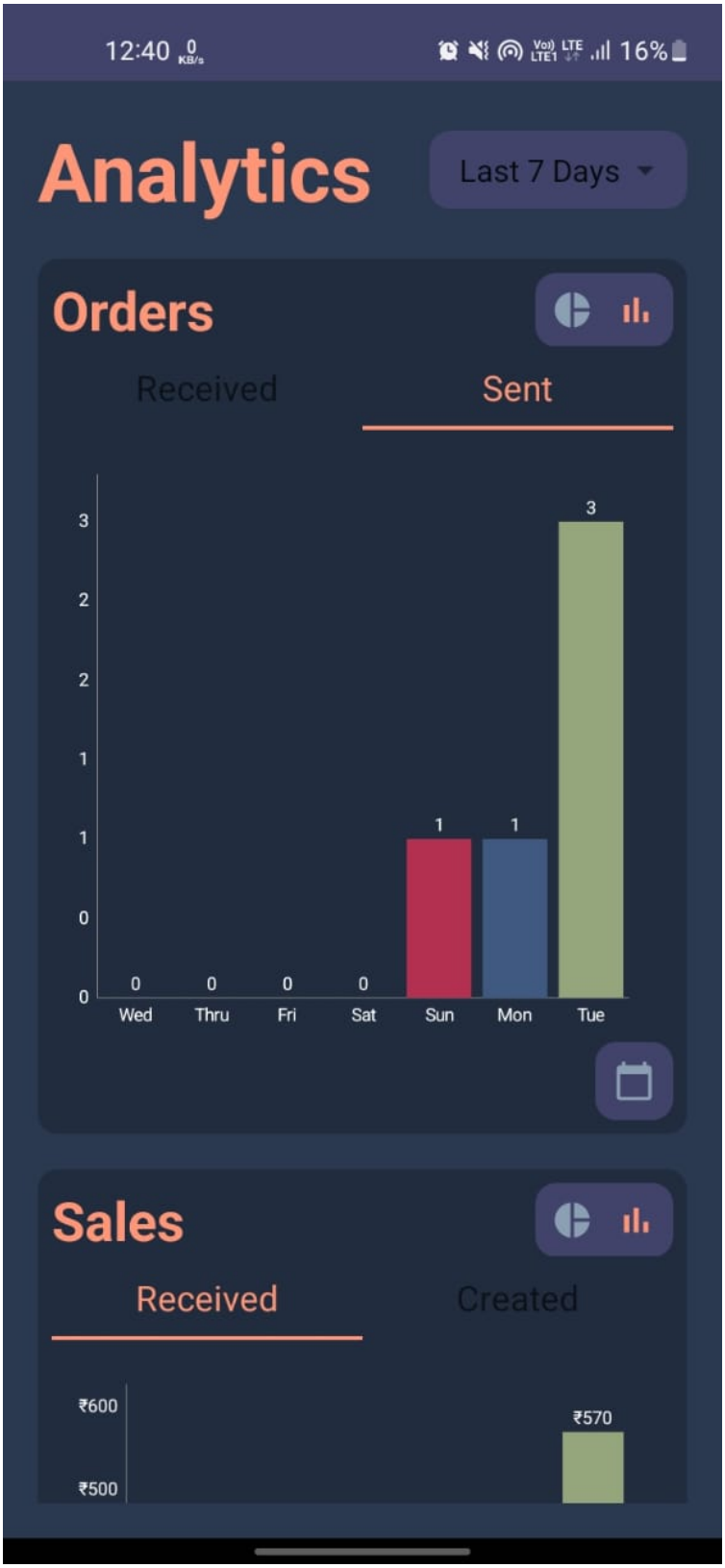

Figure 7.11: Analytics Orders

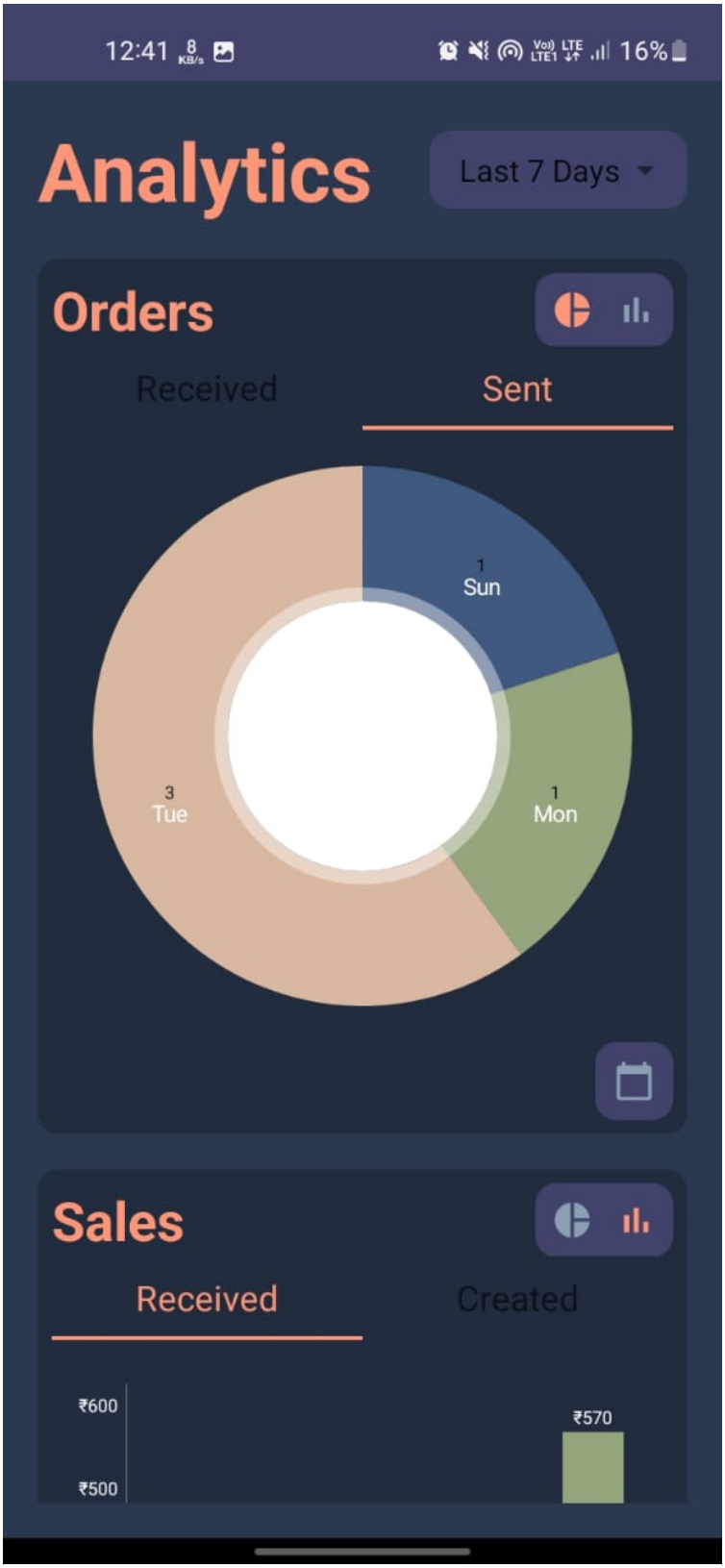

Figure 7.12: Analytics Orders Pie

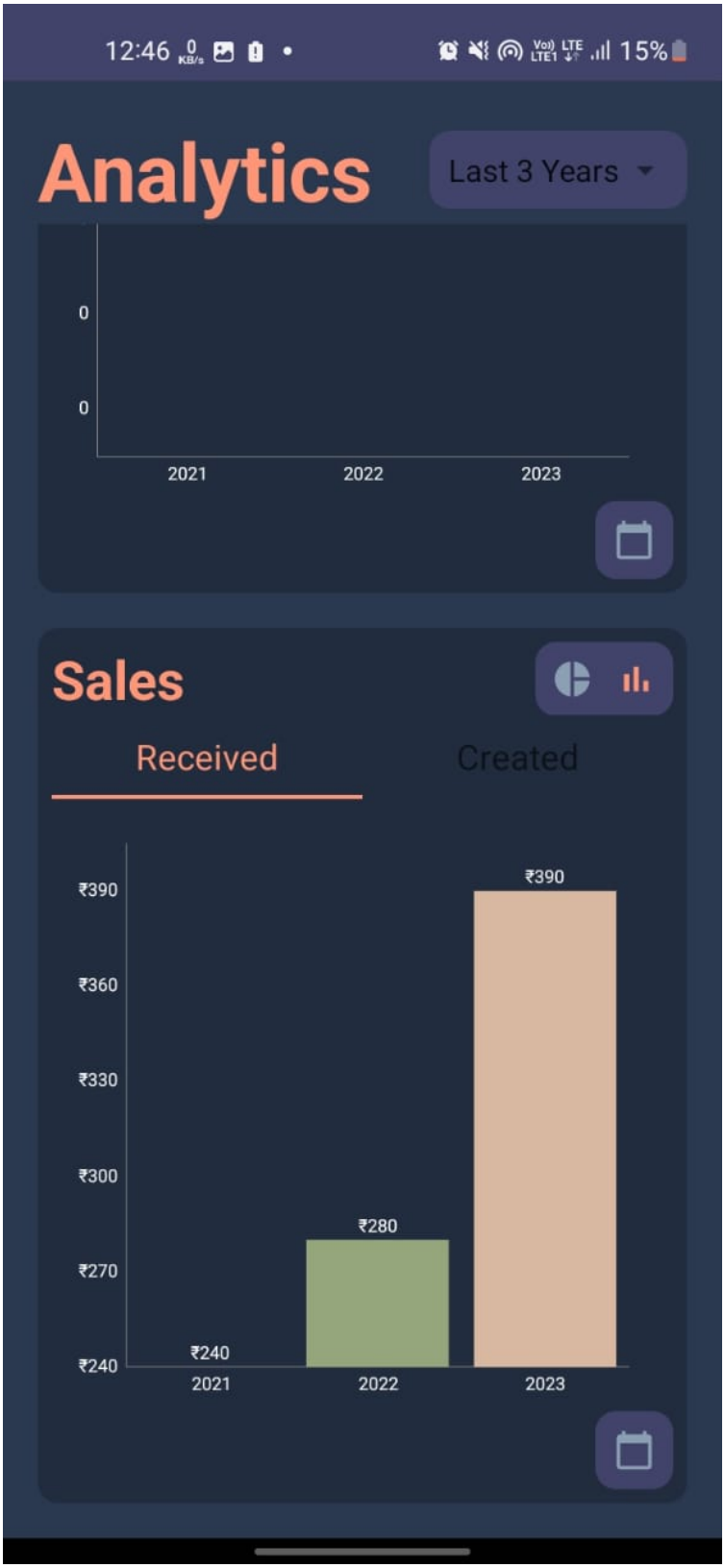

Figure 7.13: Analytics Sales

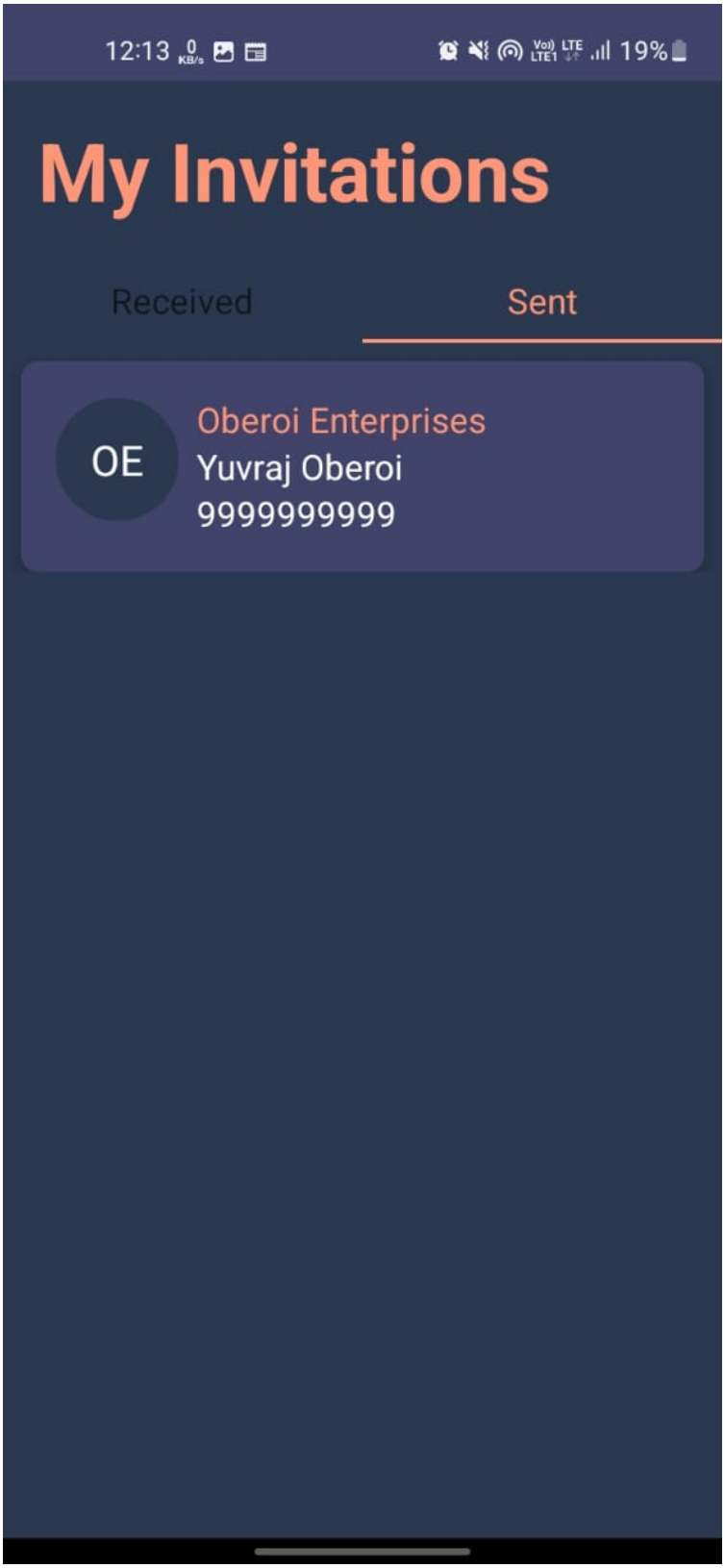

Figure 7.14: My Invitations

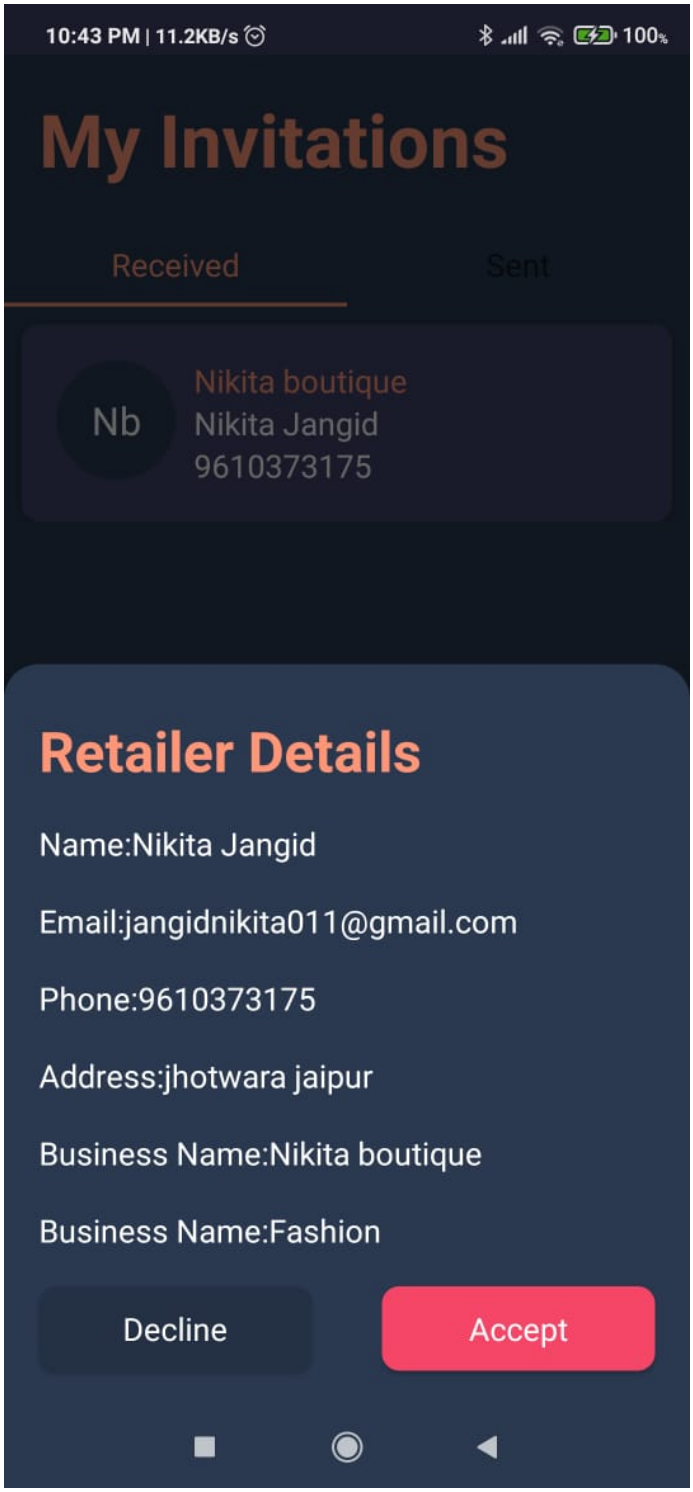

Figure 7.15: My Invitations Accept Request

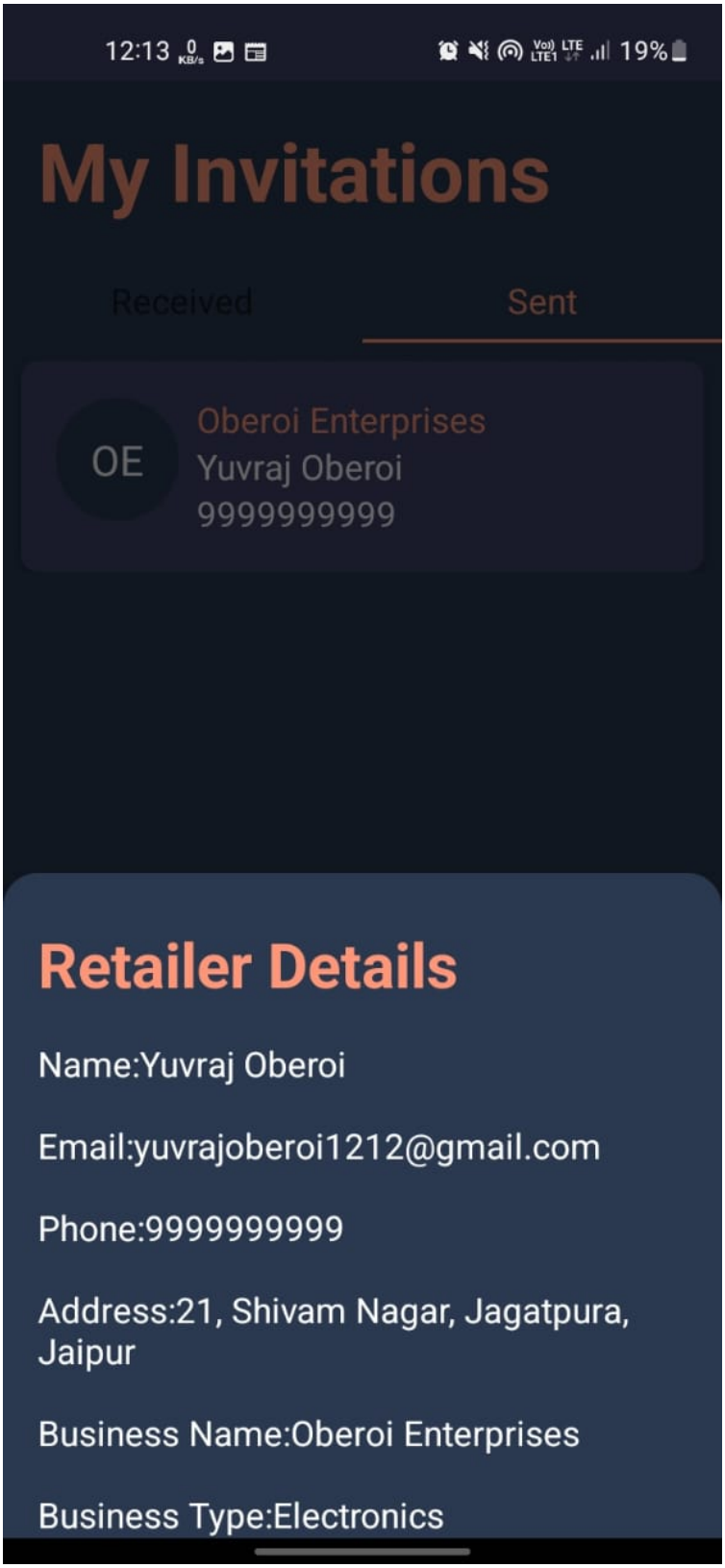

Figure 7.16: My Invitations Retailer Details

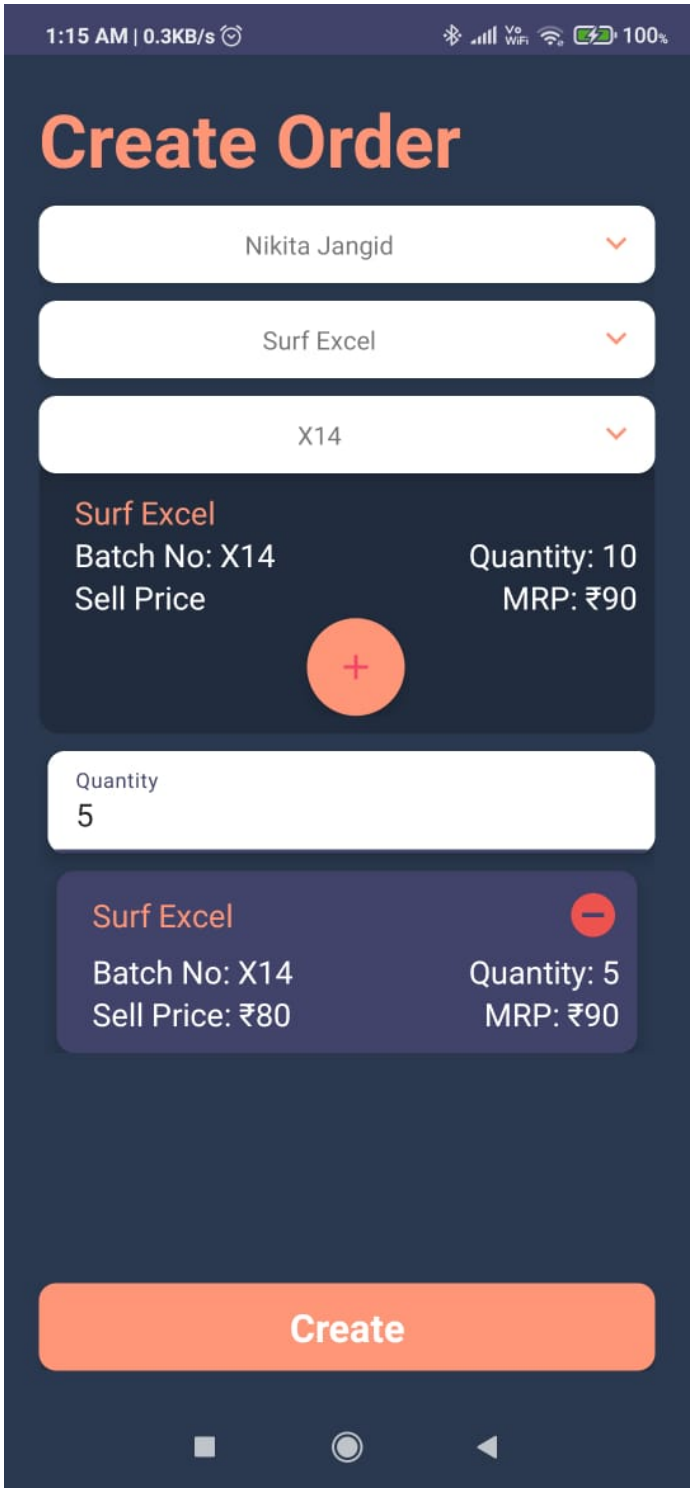

Figure 7.17: Create Order

Figure 7.18: UPending Orders

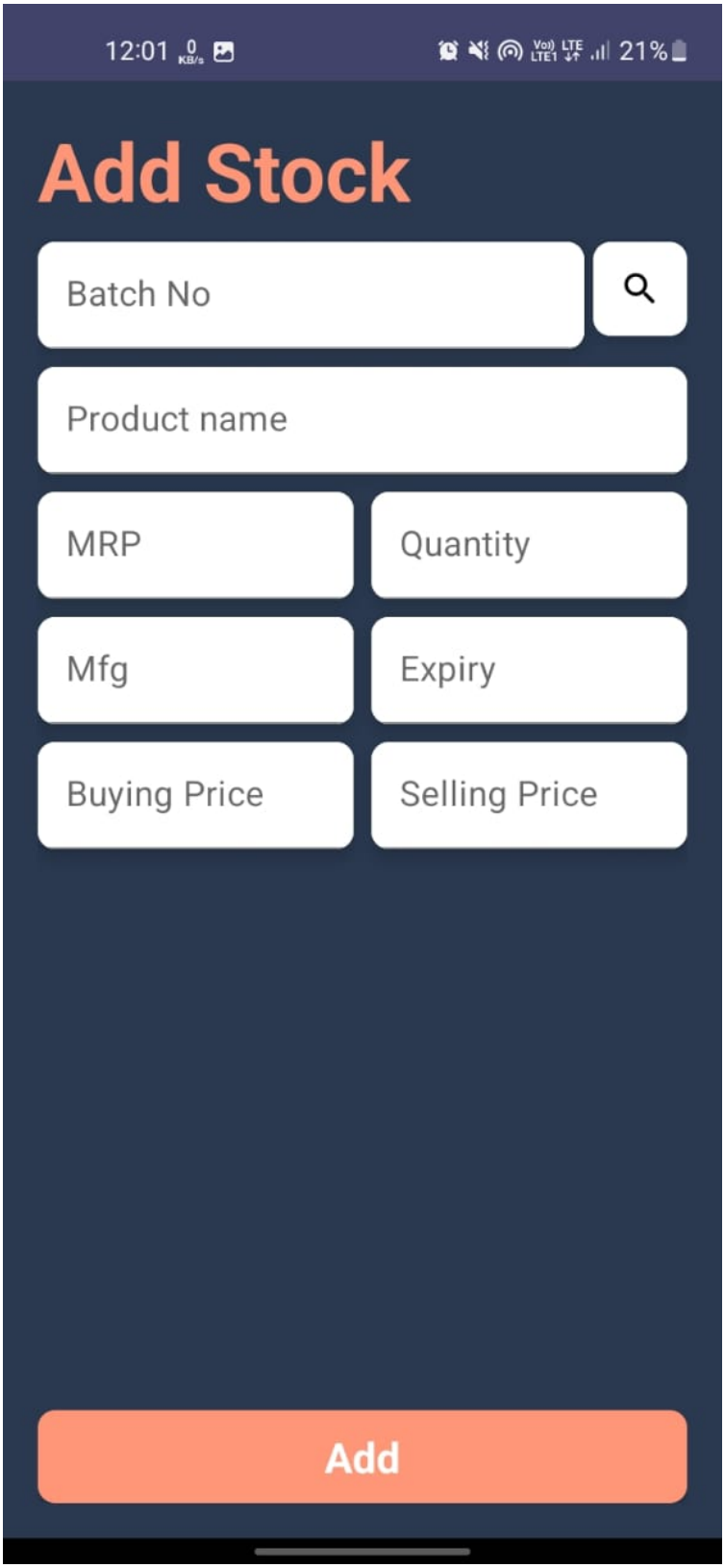

Figure 7.19: AddStock

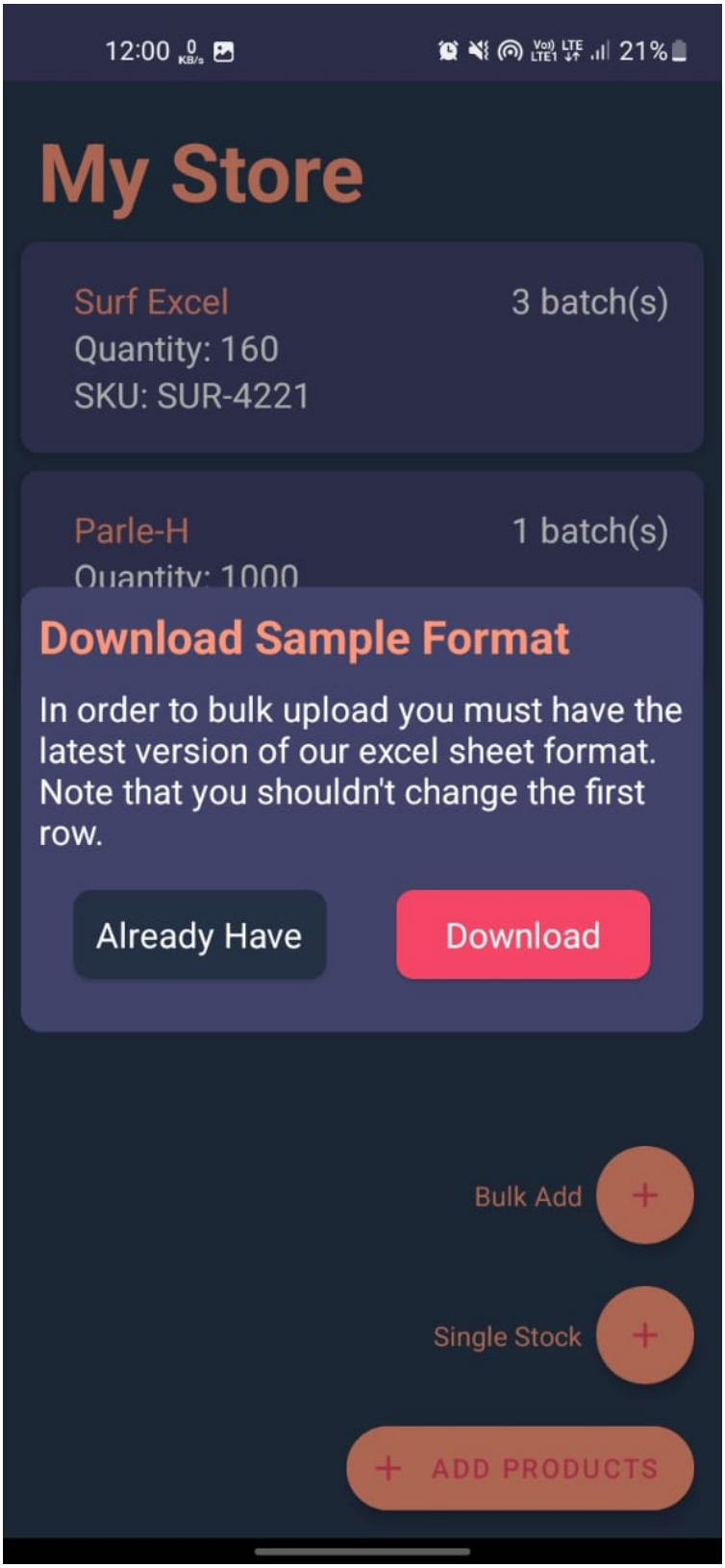

Figure 7.20: Excel Sheet Download

Chapter 8
Project Summary and Conclusions

The TradeConnect is an integrated system designed to streamline and automate various aspects of the retail business. It provides a platform for retailers to manage their inventory, track sales, connect with customers and other retailers, and facilitate transactions. The project includes a mobile app for customers to view and interact with retailers, as well as a backend system to handle data storage, processing, and business logic.

Throughout the project, we have developed and implemented various features and functionalities, including batch-wise stock management, order creation and management, sales recording, connection requests, and customer-retailer interactions. The system allows retailers to efficiently manage their inventory, fulfill customer orders, track sales performance, and establish connections with other retailers for collaboration and growth.

8.1 Conclusion

In conclusion, TradeConnect provides an end-to-end solution for retailers to streamline their operations, enhance customer experiences, and drive business growth. The project has successfully achieved its objectives by delivering a comprehensive set of features that cater to the needs of retailers and customers.

The system enables retailers to efficiently manage their inventory, create and manage orders, record sales transactions, and establish connections with other retailers. It empowers customers to easily discover and interact with retailers, view transaction histories, and communicate with their preferred retailers.

Throughout the project, we have prioritized reliability, security, and performance, ensuring that the system operates smoothly even under high loads and stringent security requirements. The modular and scalable architecture allows for future

enhancements and integration with other systems.

This project not only simplifies retail operations but also fosters collaboration and growth within the retail community. It brings efficiency, convenience, and improved customer experiences to the retail industry, contributing to its overall development and success.

Chapter 9
Future Scope

1. Integration with payment gateways: Implementing secure and seamless integration with popular payment gateways will enable customers to make online payments directly through the system, enhancing convenience and efficiency.

2. Advanced analytics and reporting: Enhancing the system with advanced analytics capabilities will provide retailers with valuable insights into customer behaviour, sales trends, inventory management, and more. Detailed reports and visualizations can assist retailers in making datadriven decisions and optimizing their operations.

3. Loyalty program integration: Incorporating a loyalty program into the system can incentivize customer loyalty and repeat purchases. Retailers can offer rewards, discounts, or exclusive offers to customers based on their transaction history.

4. Expansion to multiple retail sectors: The system can be expanded to cater to various retail sectors such as fashion, electronics, groceries, etc., accommodating specific industry requirements and providing tailored features.

References

[1] *Dr. M. Laxmaiah, A Novel Approach for Digital Online Payment System, January 2019* URL: https://www.researchgate.net/publication/327379378_A_Novel_Approach_for_Digital_Online_Payment_System

[2] *https://developer.android.com/*

[3] *https://firebase.google.com/docs*

[4] *Head First Android Development*

[5] *Android App Development for Dummies*

[6] *https://mongoosejs.com/s*

[7] *https://www.mongodb.com/home*

[8] *https://www.android.com/intl/en¡n/*

[9] *https://www.geeksforgeeks.org/*

[10] *Internet (Google, Bing, etc) & AI language models (ChatGPT, Bard, etc)*