Contents

List of Figures

1. Introduction

1.1 Problem Definition

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 Document Conventions

The format of this document is referred from the standard IEEE guidelines:

- Font face: Times New Roman
- Font size:
    - Heading: 18
    - Sub-heading: 14
    - Description: 12
- Bold face and indentation is used on general topics and or specific points of interest including the heading and sub-heading.

1.3 Intended Audience and Reading Suggestions

This document is mainly intended for project guides. The sequence for reading the document begins with the overview sections and proceeding through the sections that are most pertaining to each reader type.

1.4 Purpose

The purpose of TradeConnect is to develop a software system that addresses the challenges faced by a retail business in managing its inventory, sales, and connections with other retailers. The software solution aims to streamline operations, improve inventory management, facilitate sales transactions, and enable efficient collaboration with other retailers. By implementing this system, the retail business aims to enhance efficiency, reduce costs, increase customer satisfaction, and drive growth in the competitive retail industry.

1.5 Scope

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

2. Overall Description

2.1 Product Perspective

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.

2.2 Product 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.

2.3 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.

2.4 Assumptions 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.

3. Specific Requirements

3.1 Interface Requirements

The user needs to download the app. Then he/she needs to register to the system by providing a valid phone number, otherwise he/she won't be able to use the app. Then, to benefit from the app the user need to start utilizing its functionalities.

3.2 Functional Requirements

This section outlines the use cases for each user registered to the app (larger system).

3.2.1 Use Case

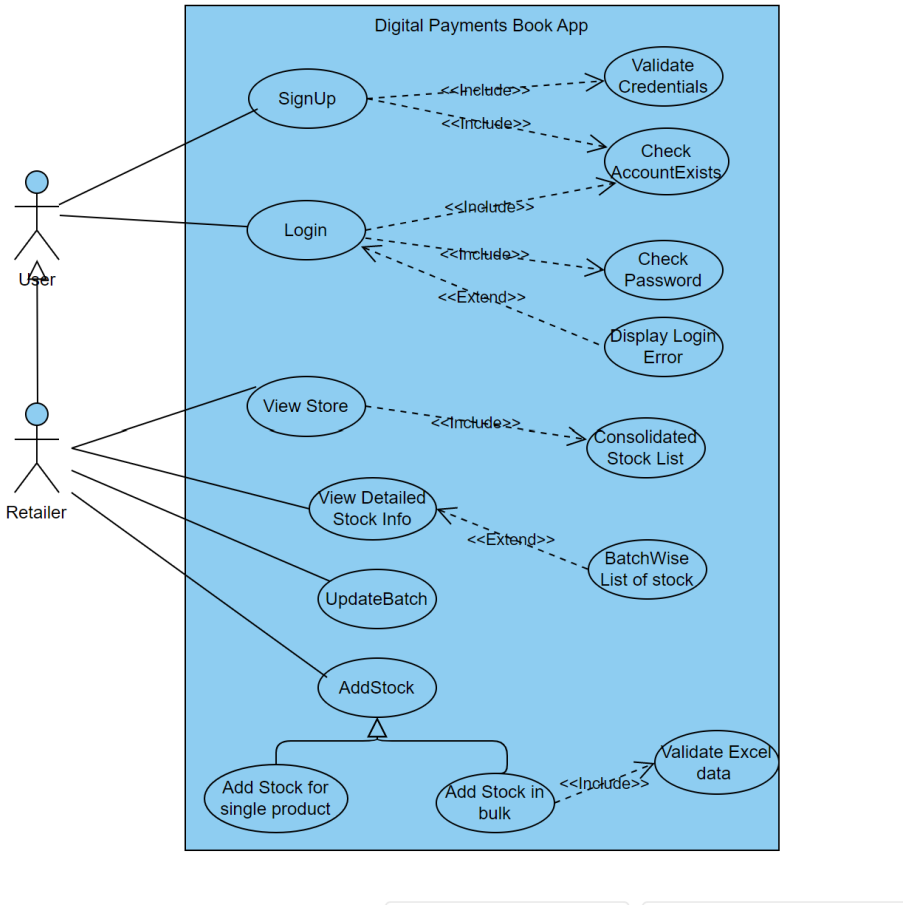

Figure 1: Usecase Diagram(1)

User can create their account on the app by verifying their phone number and filling details that are mandatory to their selected role in the system. If user has previously an account on the app then he has to just verify his phone number by OTP verification.

Users can view the consolidated product list that are currently available in his Inventory. Further he can view batchwise detail of a particular product by clicking on product rows in the listing. Here he can update the batch details individually. Coming to product listing he has two options if he wants to add product stock in his inventory, either he can add batch by batch or he can upload a excel sheet in a provided format filled with batch details of the stock.

Figure 2: Usecase Diagram(2)

Later on, in the Main Menu options retailer can view his orders, here orders can be order created by the user and orders created for user. These two types are separated in two listings for better user experience.

User can create an order from the provided “create order” option, here he can select one of his connections as order receiver and choose batch which are open to sale by order receiver.

User can also create connections with other known retailers using my connections tab. While creating a connection he can simply search by retailer name, his business name or his email address, upon matching with any one of these three fields user can choose the recipient. He can accept or reject the incoming requests. He can also view receivables from his connections, if it is negative he has due on his connection and positive value means connection has to pay the Use

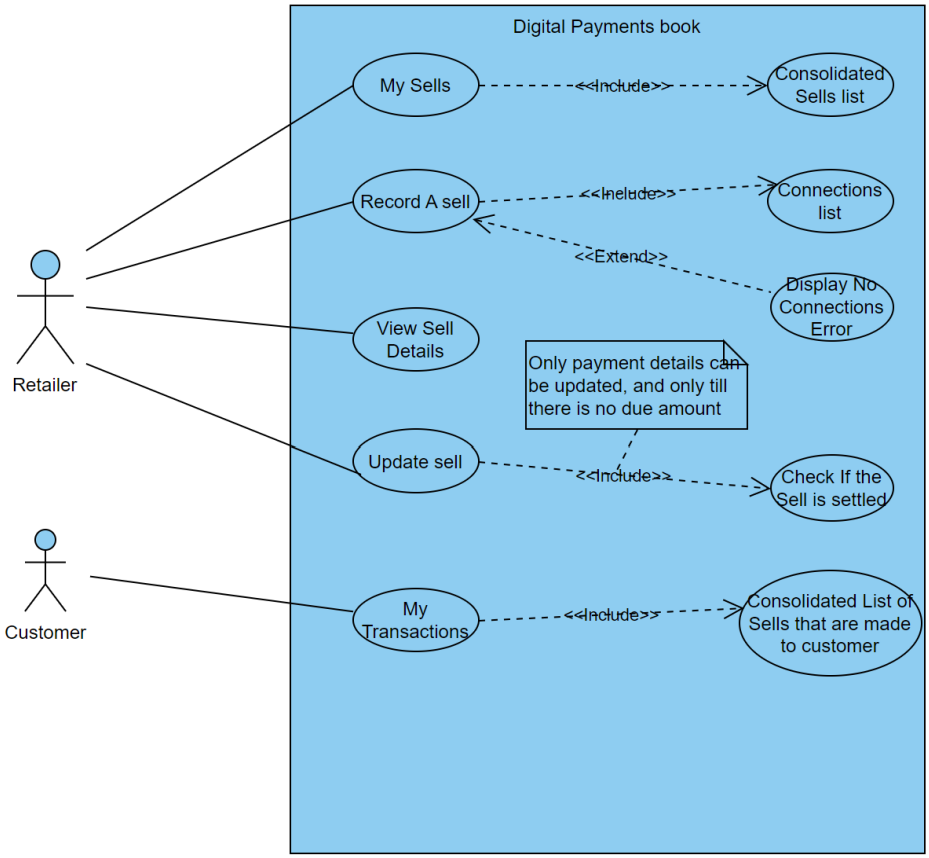

Figure 3: Usecase Diagram(3)

On clicking My Sells button in the menu options user can view sales records that he has made over the time, he can view the due amount that is pending from the other person. User will be provided with two listings here Sell that has due left and Sell that has been settled. User can also record a sell by choosing create sell, he can choose if he wants to create a sell for his customer or to his connection.

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

Performance of making the backend servers is very important as there would be only a single instance of the server. The app is distributed to all the users individually, so the app performance is not a huge matter of concern and it depends on the specification of the user's device. Besides, our web service should handle multiple users at the same time.

3.3.2 Design Constraints

In the implementation process of this system, Kotlin Programming Language will be the main development language for the frontend and Javascript/Node would be main development language on the backend side. Since Kotlin is selected to be the main development language, Kotlin Programming Language Code Convention published by Jetbrains is chosen as a standard for the development process of the system. In the process of the documentation of the system, IEEE standards will be used and UML standard will be used while designing the diagrams.

Since this system will be a part of much larger system, it must be portable to this larger system. That's why portability is one of the most important attributes of this system. Since the larger system is a website that has the potential of increasing its number of users, user traffic and number of products, this system needs to be scale up with the website in the correct order.

Therefore, scalability must be the number one attribute that system will have.

3.3.3 Security

Database has to be reached securely and its data should not be broken. It also should not change except inter-agent updates. Moreover, since our dataset contain some personal information of user

such as user phone, email, address, orders and connections details, security design is important in the web service.

3.3.4 Usability

The scope of the product is widespread. The only requirement is downloading mobile application and using the app. Besides, people from every age and shall easily use the system.

4. External Interface Requirement

4.1 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)

- 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)

4.2 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

4.3 Communication Interfaces

Our system is a mobile-based application and hence it requires only a basic smartphone. This system supports all Android (>5.0) enabled devices.

5. Data Model and Description

5.1 Data Description

5.1.1 Data Objects

This subsection of the document explains system's classes and their relations with each other.

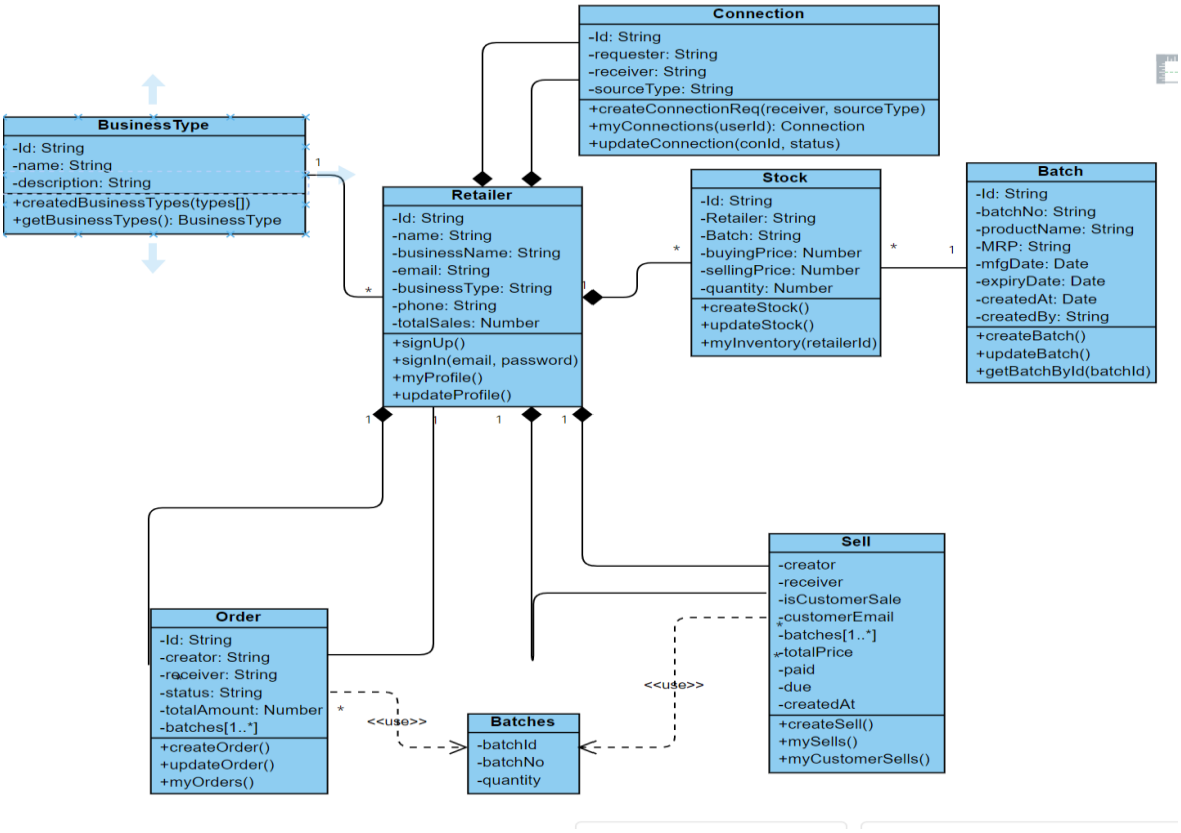

Figure 4: Class Diagram

5.1.2 Entity Relationship

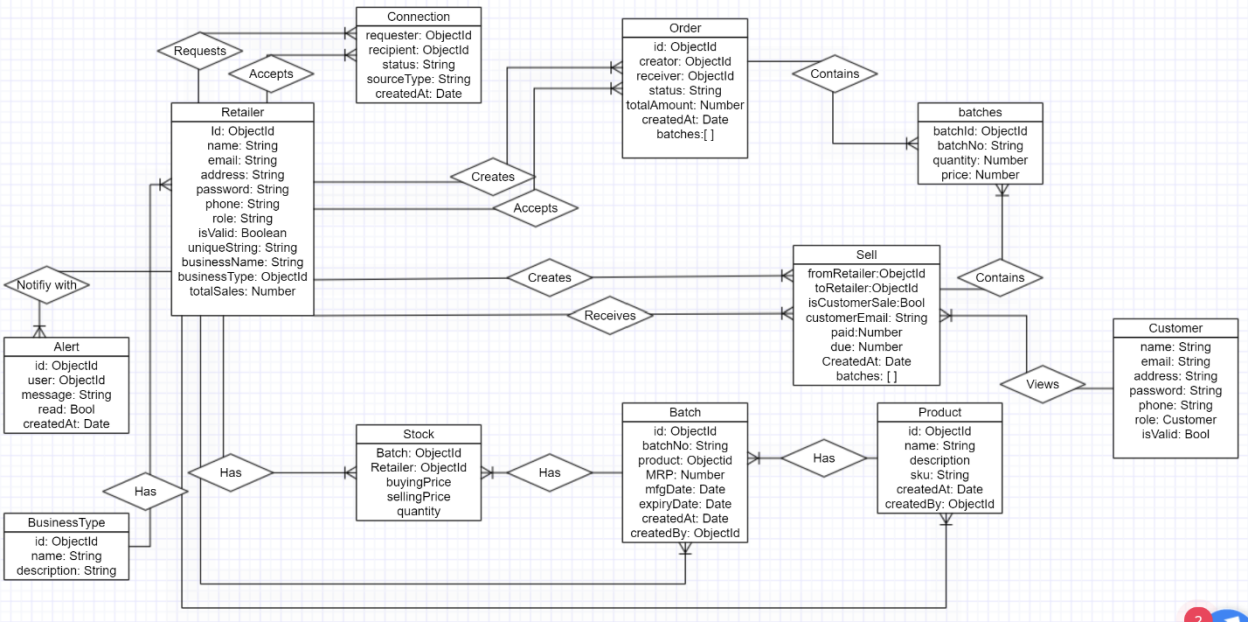

Figure 5: Entity Relationship Diagram

6. Behavioral Model and Description

6.1 Description for Software Behavior

This subsection describes the major events and states of our software. This app is a retail management system that facilitates the interaction between customers and retailers. It allows customers to view their transaction history and details for the purchases they have made from different retailers. Customers can access the app, log in to their accounts, and navigate to the "Transactions" section to review their past transactions. The app provides a user-friendly interface for filtering, sorting, and viewing transaction records. Customers can easily track their purchases, review specific transaction details, and download or print transaction records as needed. This feature enables customers to stay informed about their shopping history and make informed decisions based on their past experiences.

6.2 Sequence Diagrams

This is the UML sequence diagram of Digital Payments Book which shows the interaction between the objects of Users (Retailer, Customer) and Server/Databases.

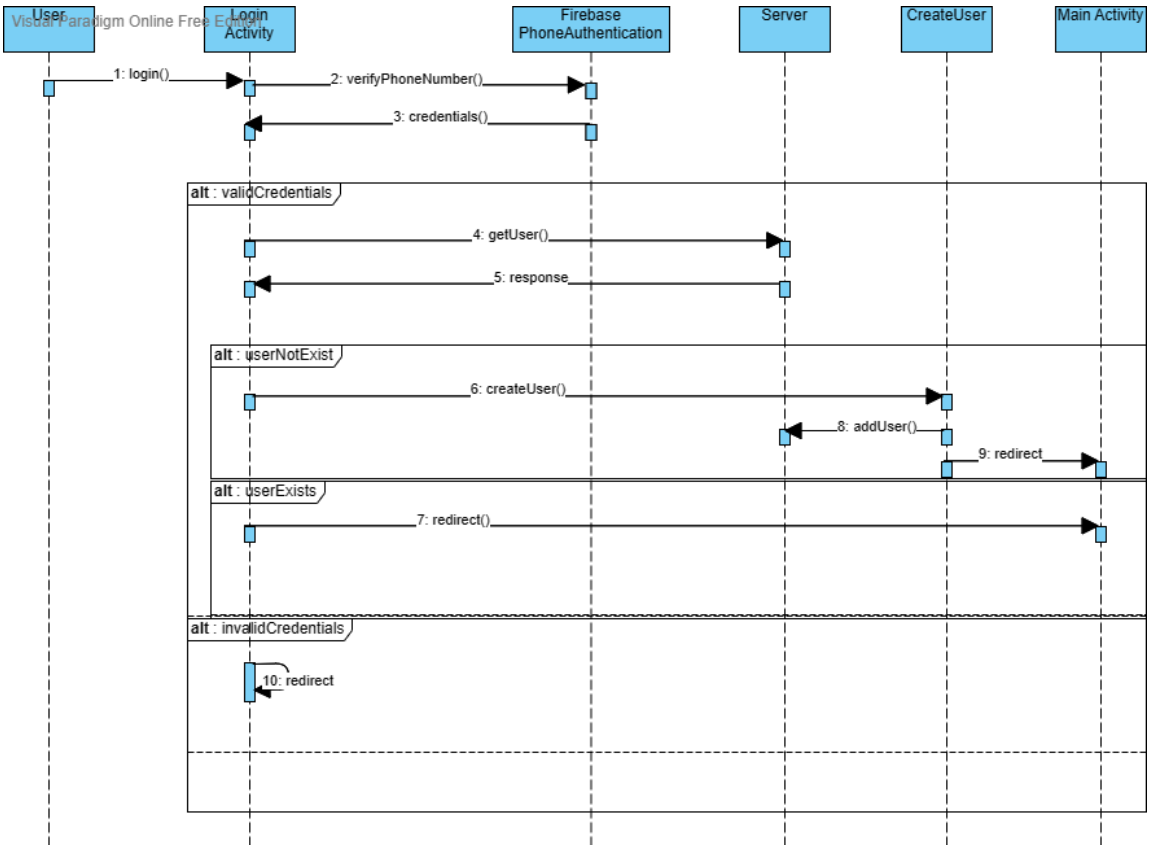

Figure 6: Sequence Diagram for Login

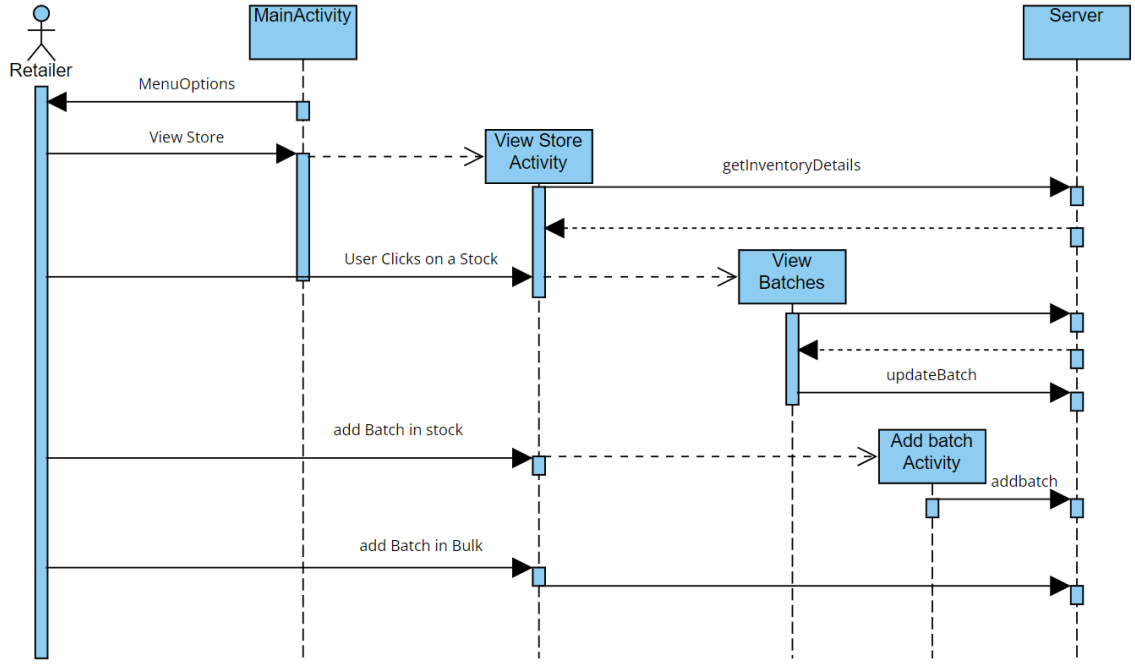

Figure 7: : Sequence Diagram for View Store

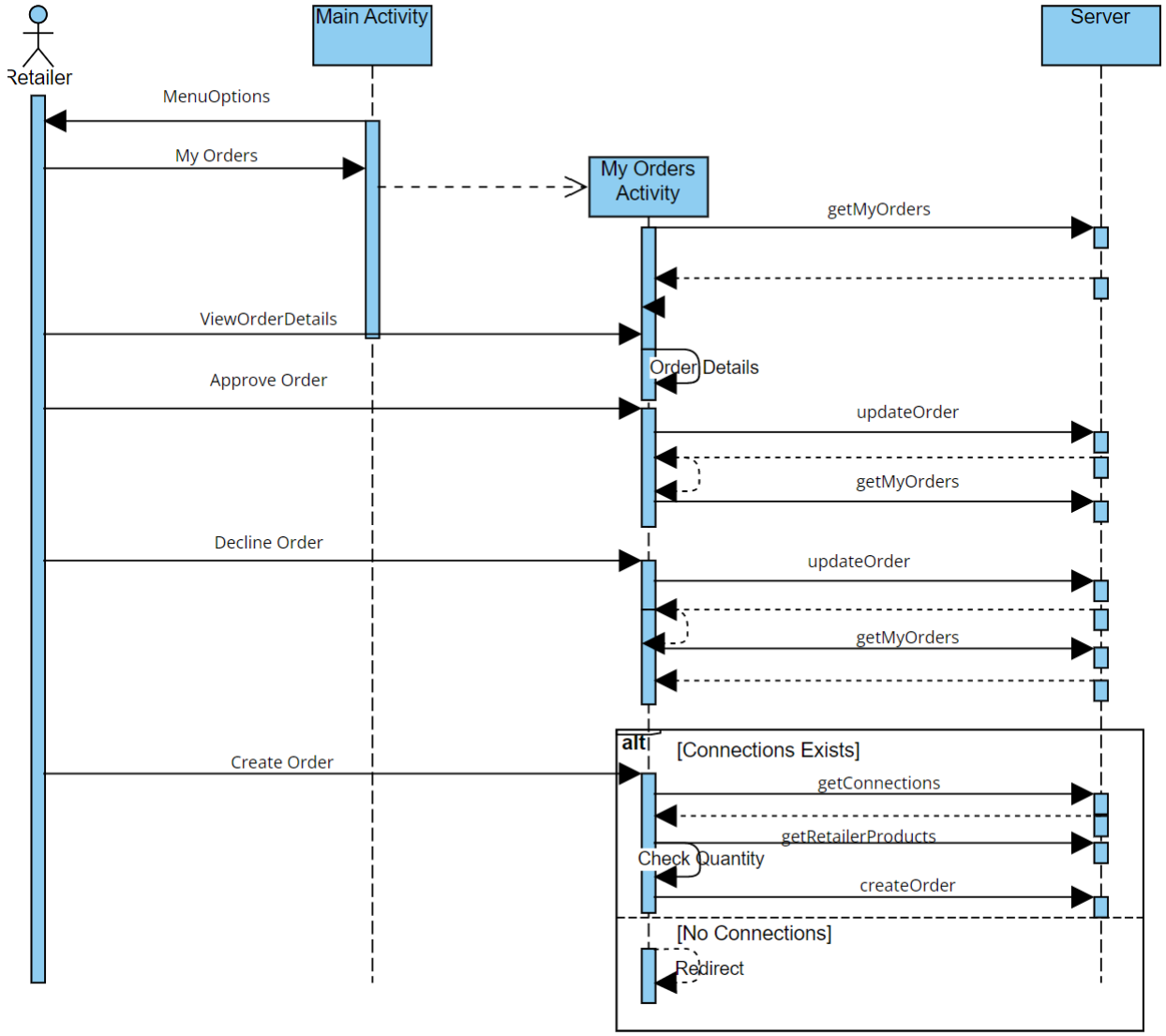

Figure 8: Sequence Diagram for My Orders

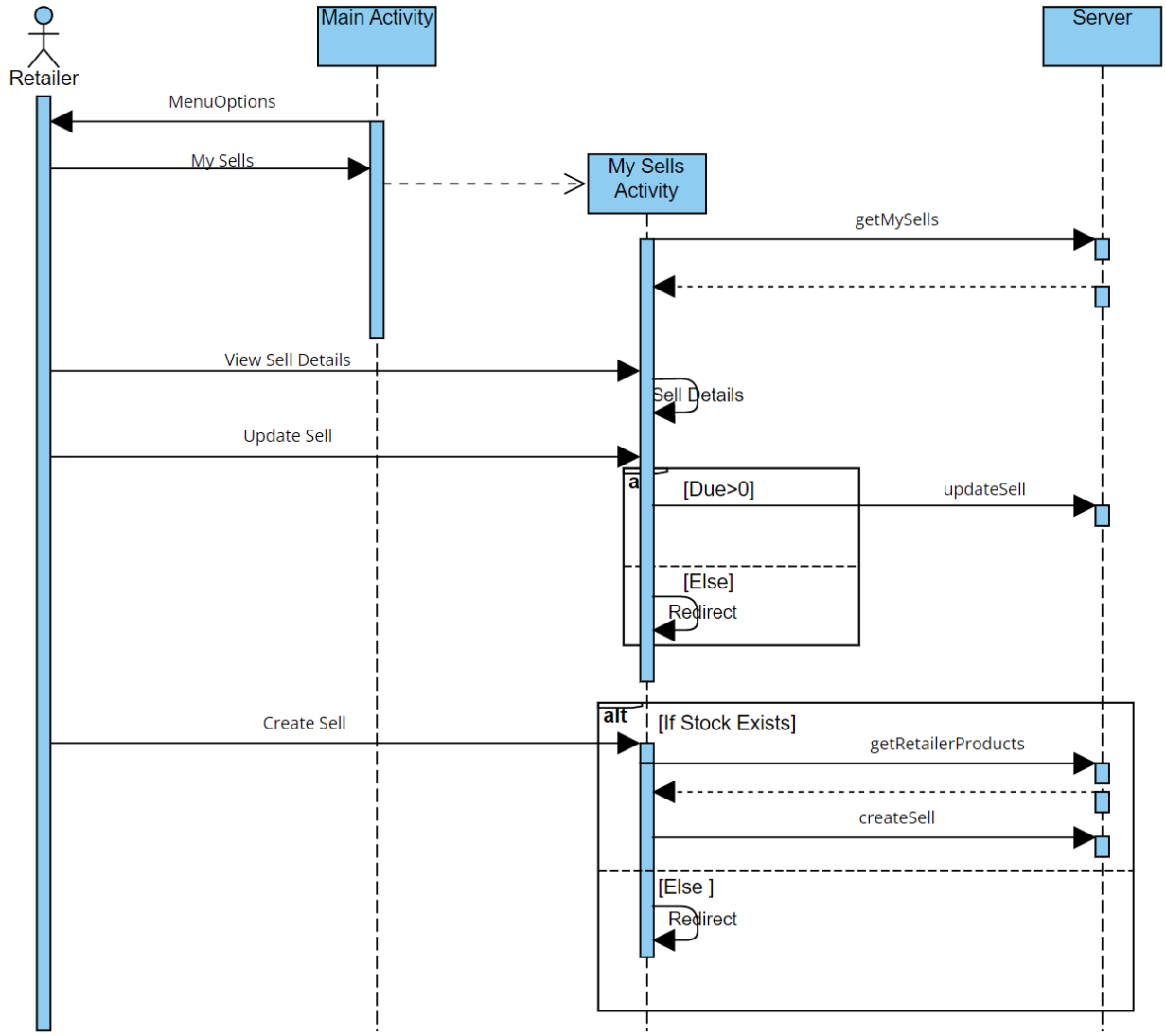

Figure 9: Sequence Diagram for Transactions for Customer and Retailer

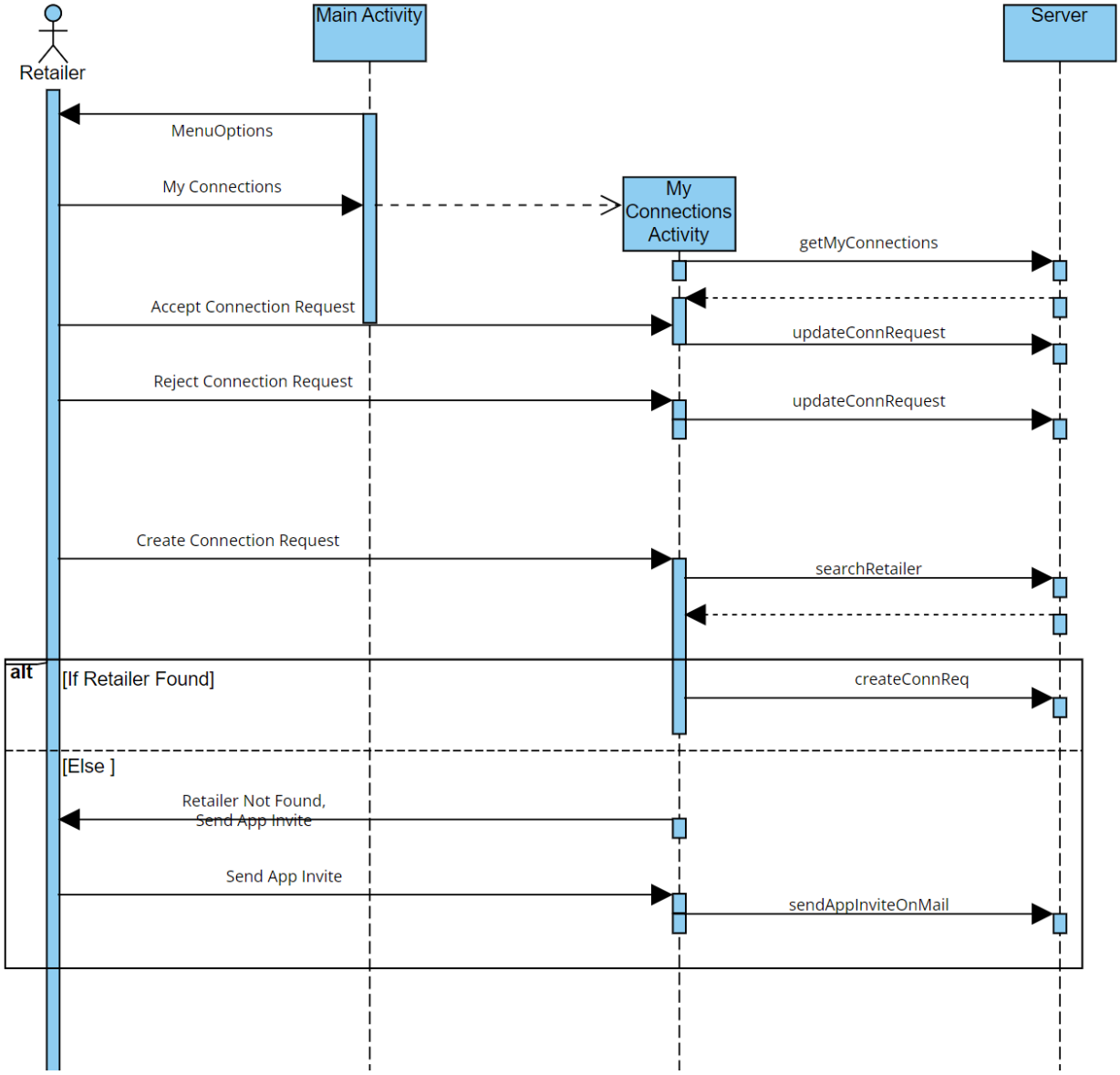

Figure 10: Sequence Diagram for My connections

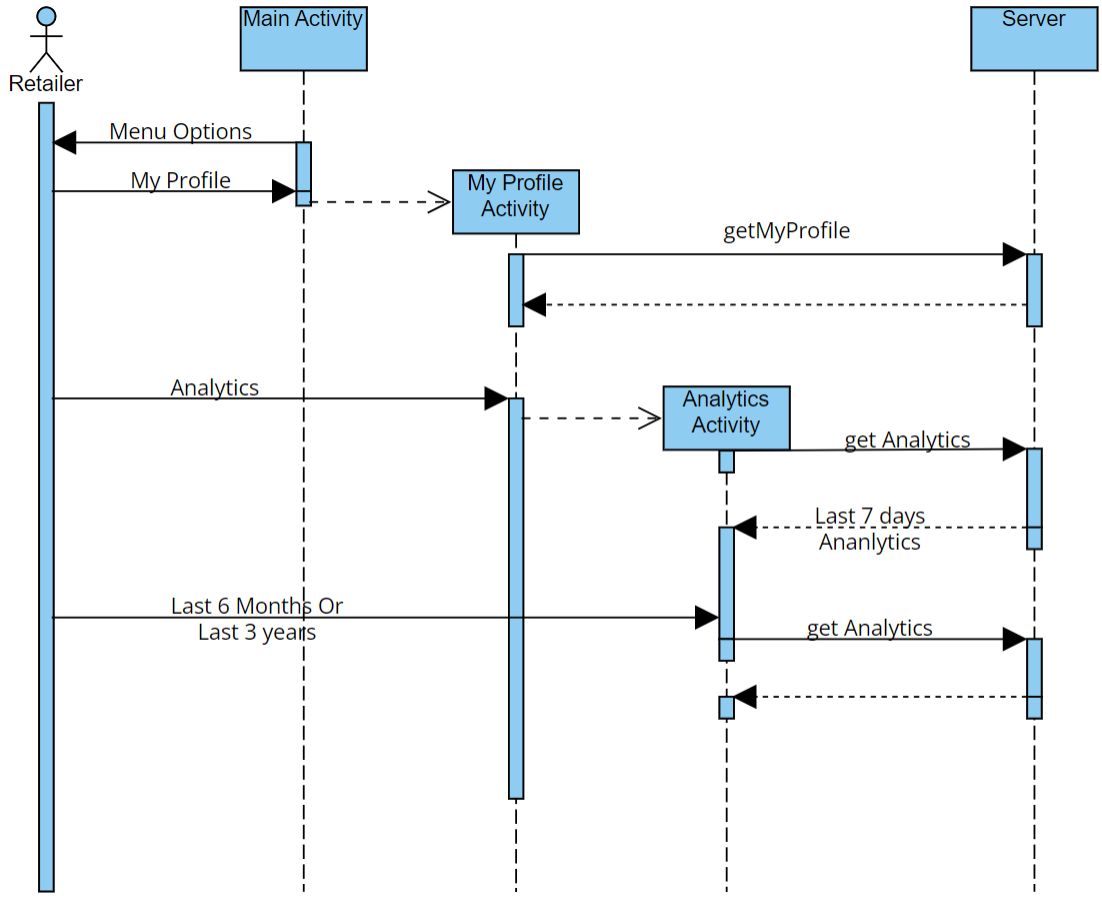

Figure 11: Sequence Diagram for My Profile

6.3 Block Diagram

6.3.1 Level 0 Data Flow Diagram

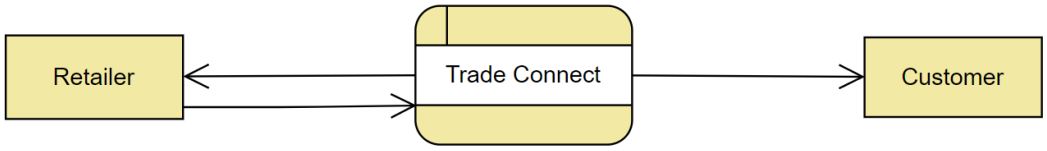

Figure 12: Level 0 Block Diagram

6.3.2 Level 1 Data Flow Diagram

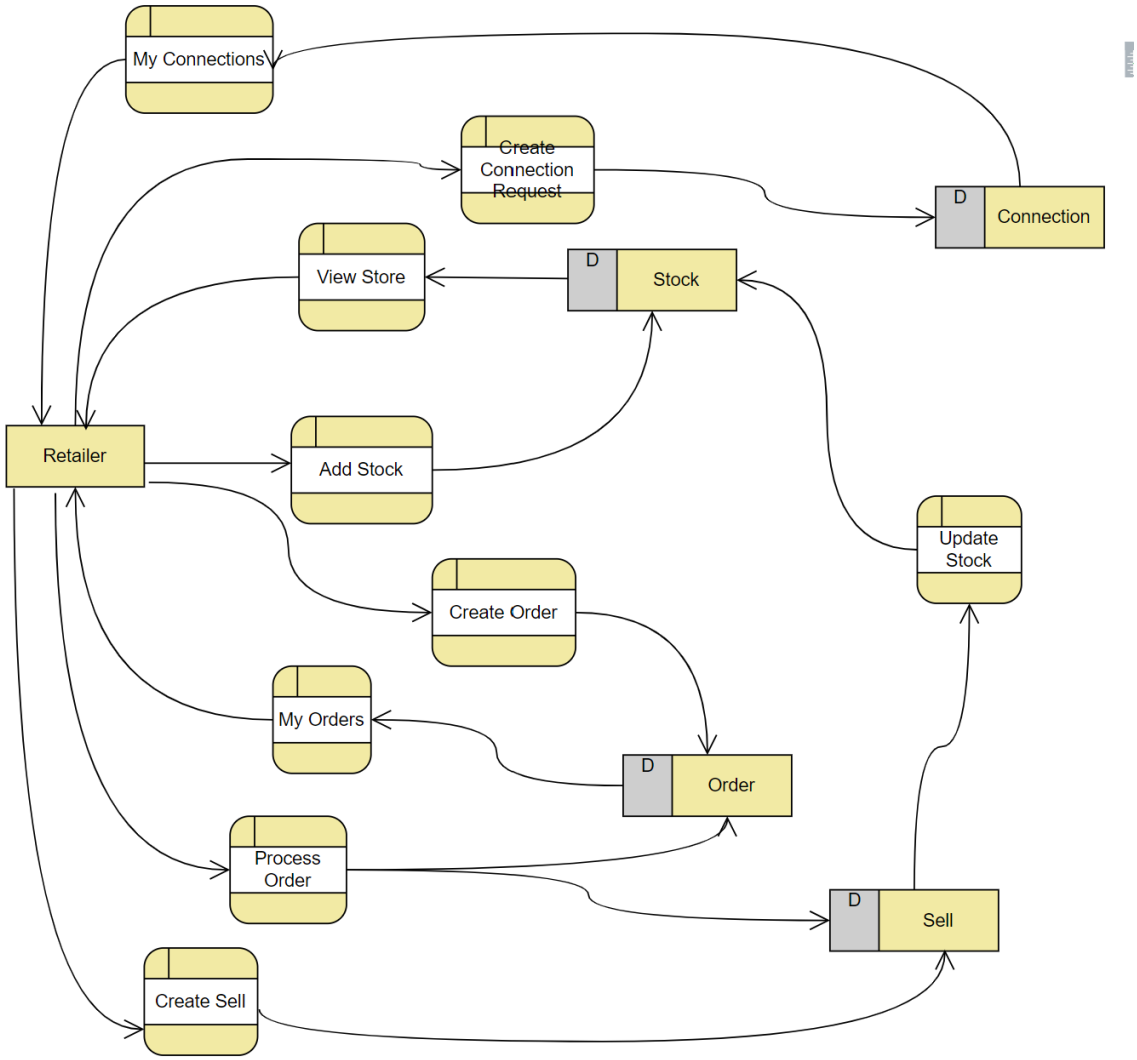

Figure 13: Level 1 Block Diagram

6.4 Activity Diagram

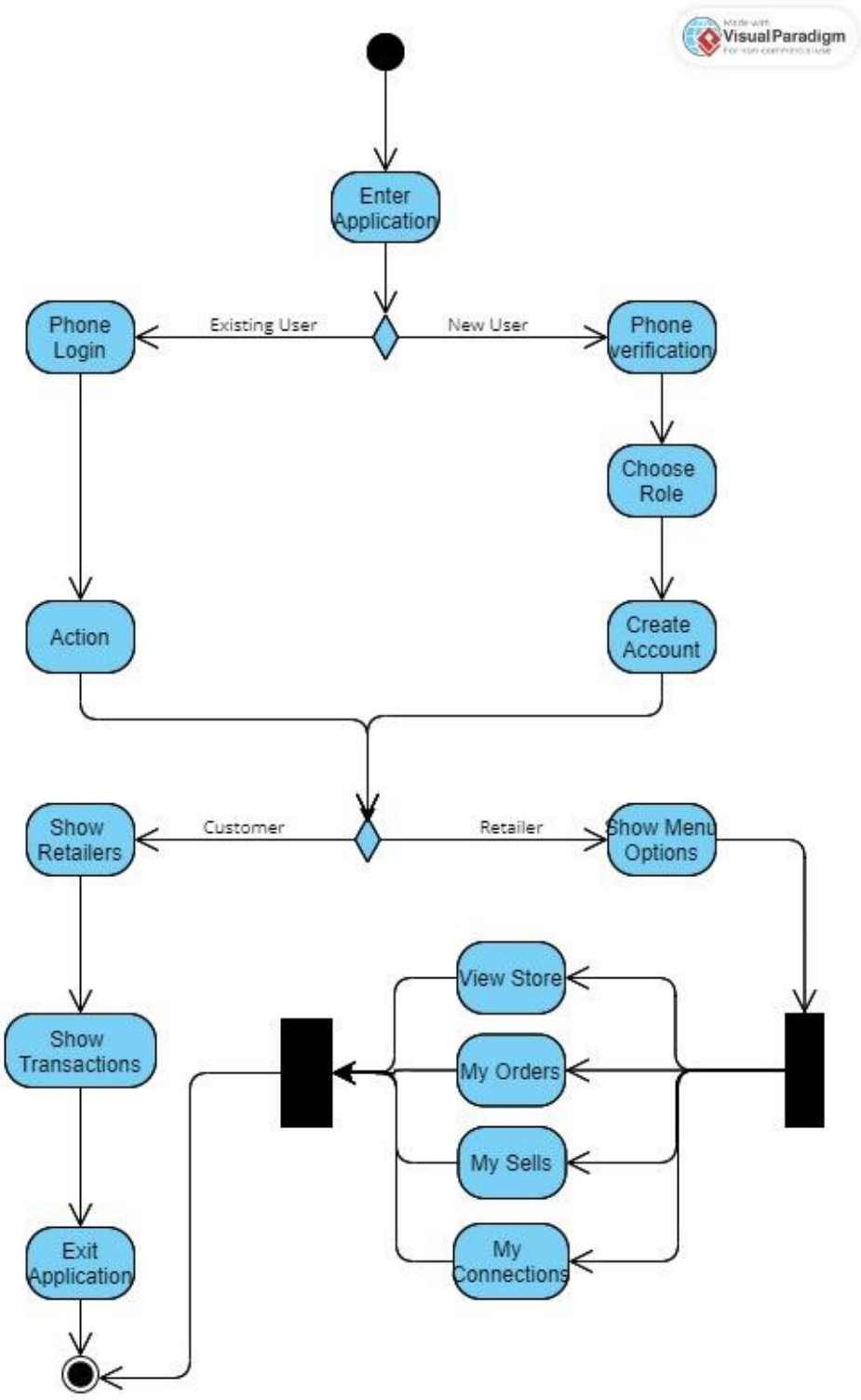

Figure 14: Activity Diagram for Login and basic overview

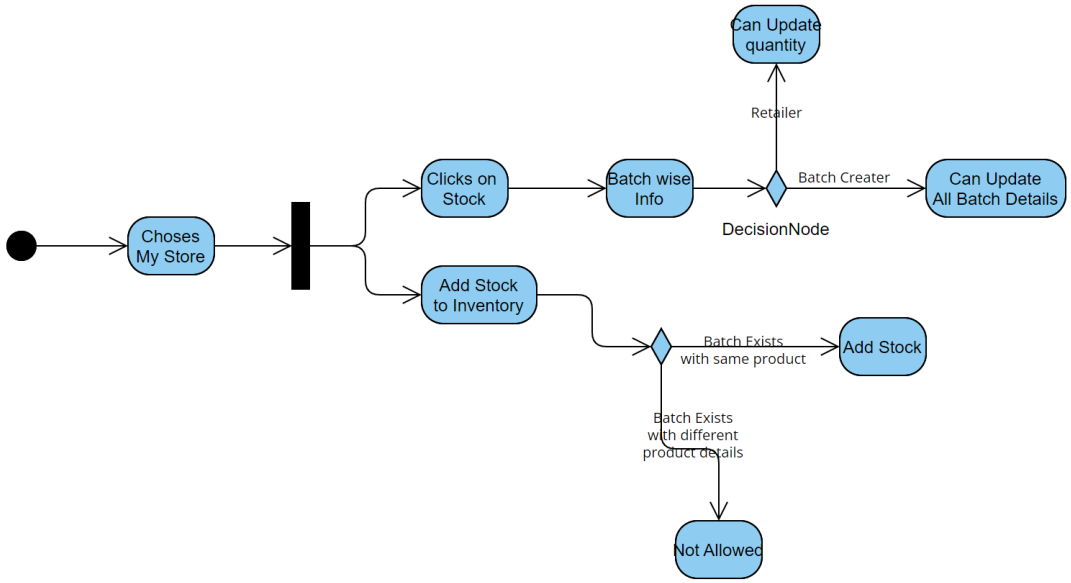

Figure 15: Activity Diagram for My store

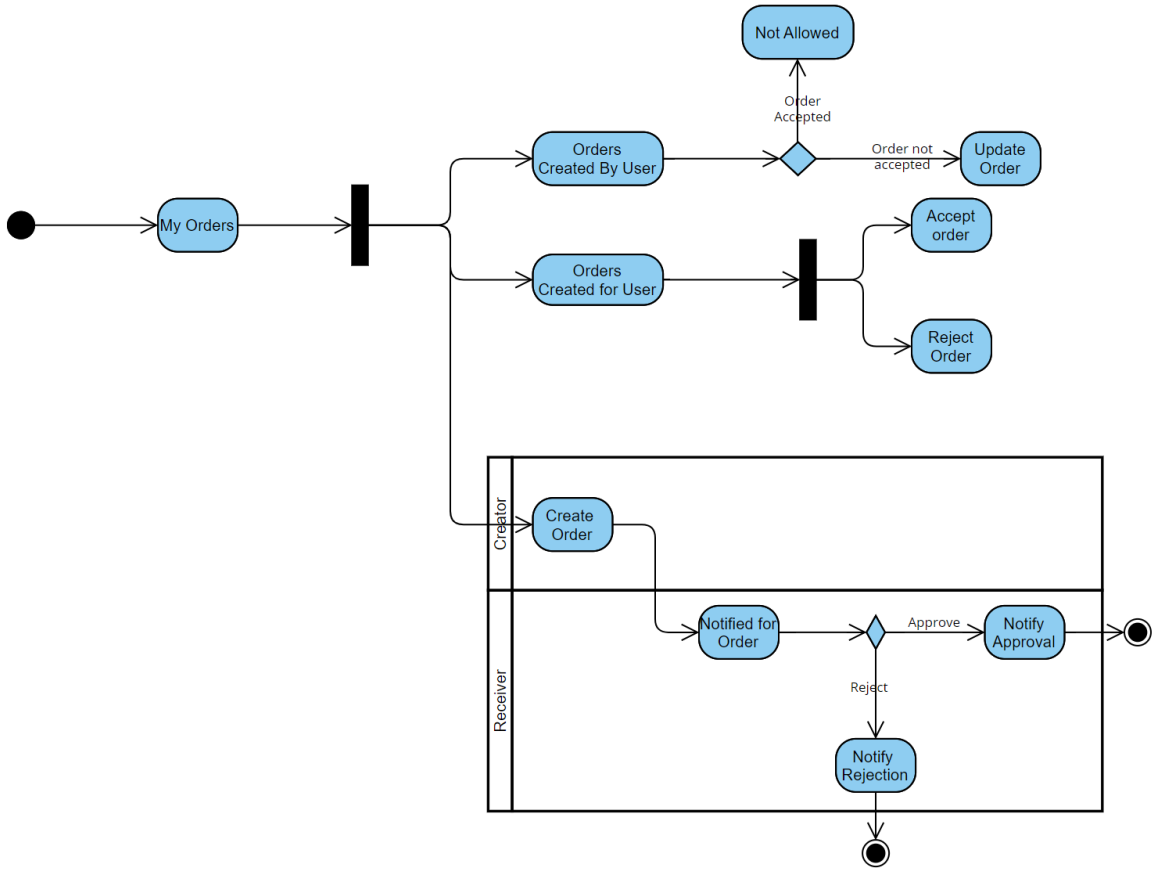

Figure 16: Activity Diagram for My Orders

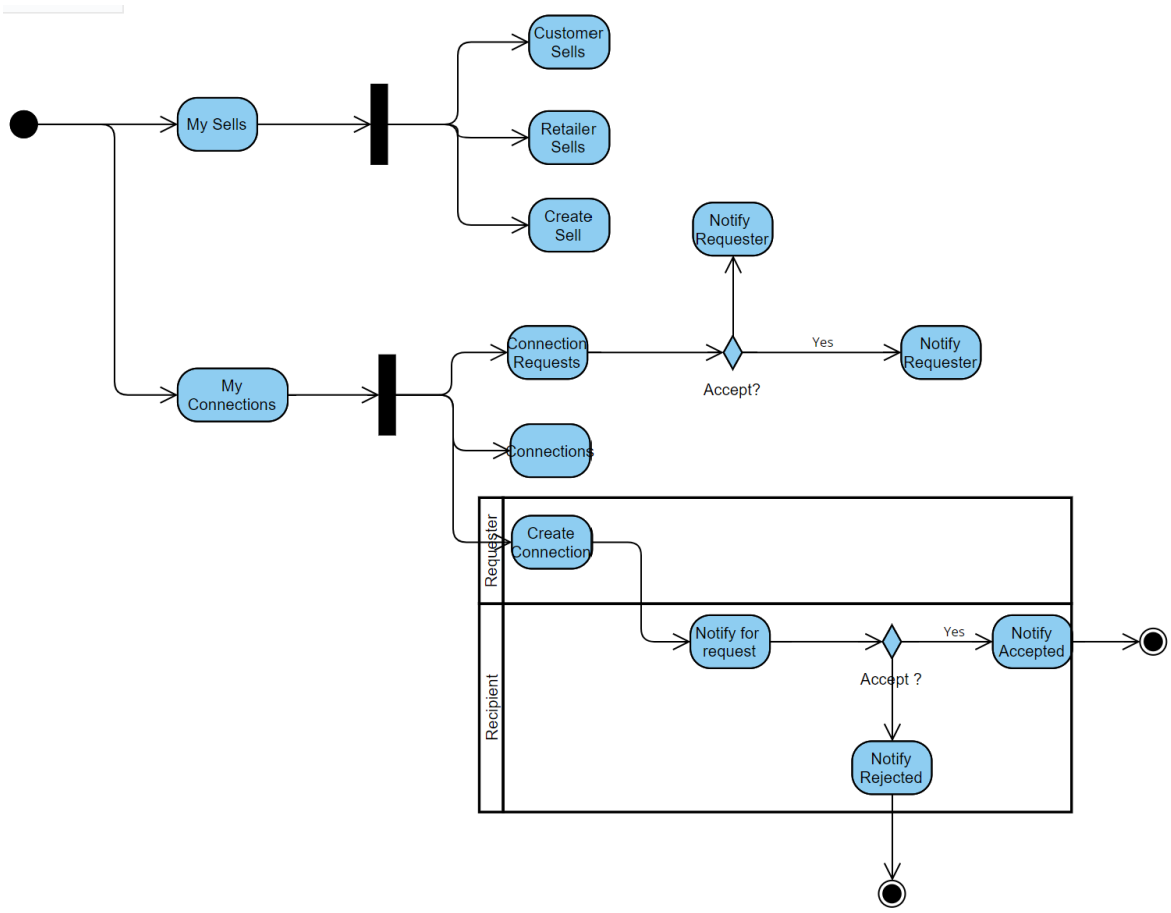

Figure 17: Activity Diagram for My Sells and Connections

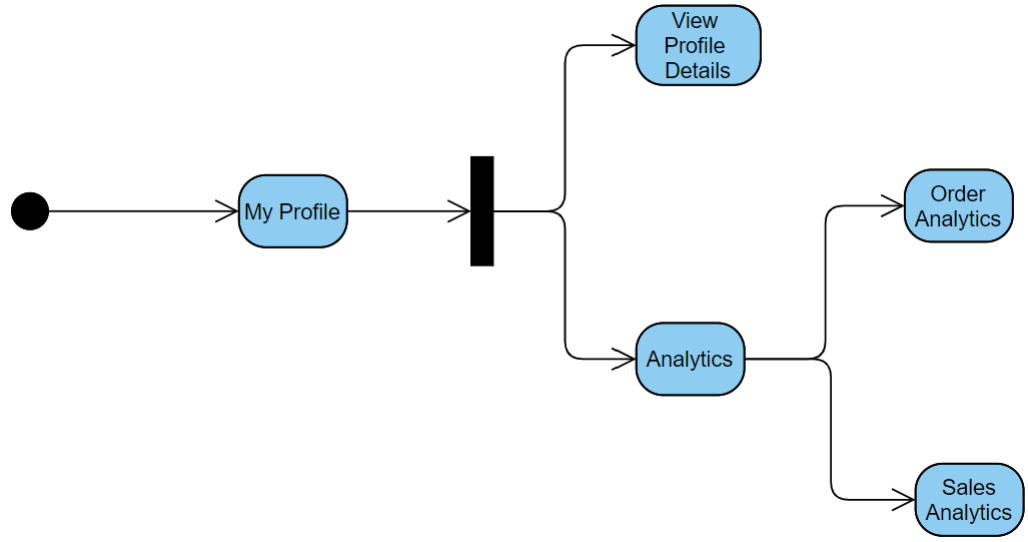

Figure 18: Activity Diagram for My Profile

7. Conclusion

In conclusion, the retail management system developed in this project provides a comprehensive solution for customers and retailers to manage their transactions effectively. The system allows customers to view their transaction history and provides a convenient way to track their purchases from different retailers. Retailers benefit from features such as order management, inventory tracking, and sales record creation. The system streamlines the retail process, enhances transparency, and improves customer-retailer relationships.

8. 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 data-driven 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.

9. References

As Internet is an Ocean of knowledge, we, too, has been helped by the same inter network of system. We've referenced from many a site to get Information/ for Knowledge Gathering to

understand the current scenario of the market, below are the references we have got helped from, and we acknowledge the same:

1. Dr. M. Laxmaiah, A Novel Approach for Digital Online Payment System, January 2019 https://www.researchgate.net/publication/327379378_A_Novel_Approach_for_Digital_Online_Payment_System
2. https://mongoosejs.com/
3. https://www.mongodb.com/home
4. https://www.android.com/intl/en_in/