PROJECT REPORT

On

Distributed Shopping
SYSTEM

ABSTRACT

The group server and middle server shopping system desktop application is an advanced platform designed to enhance the online shopping experience through a robust, scalable architecture. This system leverages the power of both group servers and middle servers to ensure optimal performance, security, and user satisfaction.

The middle server plays a critical role in processing transactions and managing communications between the user interface and the backend systems. It ensures a seamless checkout process by supporting multiple payment methods, including credit cards, debit cards, and digital wallets, all processed through a secure payment gateway. Real-time order tracking and notifications keep users informed about the status of their purchases from confirmation to delivery.

The application also integrates with inventory management systems through the middle server, providing real-time stock updates to prevent overselling and ensuring that users have accurate information about product availability. The middle server facilitates efficient load balancing and distribution of requests, enhancing the application's performance and

scalability, even during high traffic volumes.

Customer support is enhanced with robust features, including chatbots and live customer service representatives, accessible through the group server. This ensures that users can quickly and effectively resolve any issues or inquiries they may have.

Enhanced security measures, such as encryption and secure data storage, protect user information at all times. Regular updates and maintenance through the group and middle servers ensure that the application remains secure, reliable, and up-to-date.

In summary, the group server and middle server shopping system desktop application offers a secure, efficient, and user-friendly platform for an optimal online shopping experience.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipkart or ebay. Since the application is available in the Smartphone it is easily accessible and always available.

LIST OF CONTENTS

4. STUDY OF THE SYSTEM

2. SYSTEM ANALYSIS

1. TCP Iterative Client:
2. TCP Concurrent Group Serve
3. Dedicated Servers (Silver, Gold, Platinum):
4. UDP Iterative Client-Server Interaction:
5. Security and Data Management
6. Performance and Scalability
7. User Experience
8. EXISTING SYSTEM
9. PROPOSED SYSTEM
10. SYSTEM REQUIREMENT SPECIFICATION
11. NON FUNCTIONAL REQUIREMENTS
12. Description of feature

2. SYSTEM DESIGN

1.Core Components

1. INTRODUCTION

The group server and middle server shopping system desktop application is designed to optimize the online shopping experience through secure and efficient inter-process optimize the online shopping experience through secure and efficient inter-process communication. Utilizing TCP and UDP protocols, the system ensures robust data transmission and user authentication. The TCP-Group server manages login verification and credit point updates, while specialized servers handle the shopping process. This distributed architecture guarantees seamless user interaction and high performance, making it an ideal platform for sophisticated online shopping needs.

1.1 PROJECT OBJECTIVE:

The objective of this project is to learn inter-process communication using TCP and UDP protocols through the development of a distributed group-based shopping system in Java. This involves implementing a TCP iterative client, a TCP concurrent middle-server, and a UDP iterative client-server interaction. By completing this project, you will gain a foundational understanding of the steps required to develop a distributed application, focusing on secure data transmission, user authentication, and seamless communication between different servers to manage and execute shopping transactions efficiently.

1.2 PROJECT OVERVIEW:

The objective of this project is to master inter-process communication via TCP and UDP protocols by developing a distributed group-based shopping system in Java. You will implement a TCP iterative client, a TCP concurrent middle-server, and a UDP iterative client-server interaction. The project will enhance your understanding of secure data transmission, user authentication, and efficient server communication. Additionally, you will

learn to design and manage a scalable system capable of handling multiple transactions simultaneously, ensuring a smooth and reliable shopping experience for users. This foundation will prepare you for more advanced distributed application development.

1.3 PROJECT SCOPE:

The project scope involves creating a distributed group-based shopping system using TCP and UDP protocols in Java. This includes implementing a TCP iterative client, a TCP concurrent middle-server, and a UDP iterative client-server interaction. The system will handle user authentication, credit point management, and communication with dedicated servers (Silver, Gold, Platinum) for the shopping process. It will support secure data transmission and real-time updates, ensuring a seamless and reliable user experience. The project will also include designing a robust architecture capable of managing high traffic and multiple simultaneous transactions efficiently.

1.4 STUDY OF THE SYSTEM

The distributed group-based shopping system is designed to showcase the implementation of inter-process communication using TCP and UDP protocols in Java. This system involves multiple components working together to provide a seamless and efficient online shopping experience, emphasizing secure data transmission, user authentication, and efficient server communication.

The system architecture consists of a TCP iterative client, a TCP concurrent middle-server, and a UDP iterative client-server interaction. The TCP iterative client is responsible for collecting user input, including the "TCP-Group server" IP and port, login credentials, and current member credit points. This client then sends the data to the TCP-Group server for

processing.

The TCP-Group server, a concurrent server, handles multiple client requests simultaneously. It displays the received messages, verifies the login credentials, and updates the member's credit points if the credentials are correct. Upon successful login, the TCP-Group server communicates with dedicated Silver, Gold, or Platinum servers to manage the shopping process. Each of these servers is responsible for handling specific types of shopping transactions, ensuring that the system can efficiently process a high volume of transactions.

The UDP iterative client-server interaction is used for message transmission between the different components of the system. This protocol is chosen for its efficiency in handling real-time data transmission, ensuring that users receive timely updates about their transactions.

The system design emphasizes secure data transmission and storage. The use of TCP for critical operations, such as user authentication and credit point updates, ensures reliable and ordered delivery of messages. Meanwhile, UDP is used for less critical but time-sensitive operations, providing a balance between reliability and performance.

This study of the system provides valuable insights into the development of distributed applications, highlighting the importance of choosing appropriate communication protocols, managing server concurrency, and ensuring data security. By understanding these principles, developers can create scalable and efficient distributed systems capable of

handling complex tasks and high traffic volumes. This foundational knowledge is crucial for advanced distributed system development and real-world application scenarios.

Fig 1.1: server module

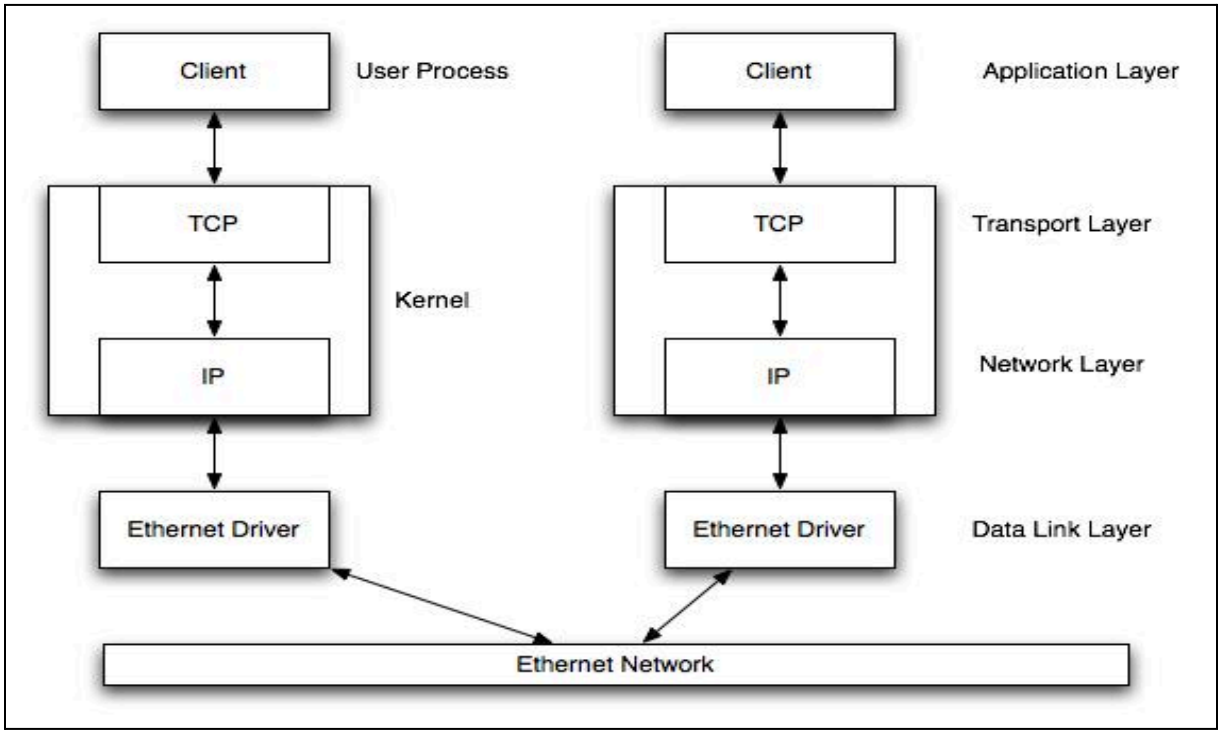


Fig 1.3: Manage Products

2. SYSTEM ANALYSIS

The distributed group-based shopping system is an intricate application designed to facilitate inter-process communication using TCP and UDP protocols in Java. This system's architecture and functionality are critically analyzed to understand its components,

interactions, and overall performance.

System Components

2.1 TCP Iterative Client:

- **Function:** This component collects user input, including the IP and port of the TCP-Group server, login credentials, and current member credit points.
- **Interaction:** It establishes a connection with the TCP-Group server and sends the collected data for processing.
- **Design Considerations:** Ensures that user inputs are correctly formatted and securely transmitted.

2.2 TCP Concurrent Group Server:

- **Function:** Acts as the central processing unit, handling multiple client requests concurrently.
- **Interaction:** It receives login credentials and credit points from the client, verifies the credentials, updates the credit points, and sends a response back to the client.
- **Design Considerations:** Utilizes multi-threading to manage concurrent client connections efficiently, ensuring that each client's request is processed independently.

2.3 Dedicated Servers (Silver, Gold, Platinum):

- **Function:** Each server handles specific shopping processes based on the user's membership level.
- **Interaction:** Communicates with the TCP-Group server to receive and process

shopping requests, ensuring that the correct server is engaged based on user credentials.

- **Design Considerations:** Ensures scalability and specialized processing, providing tailored services for different membership levels.

2.4 UDP Iterative Client-Server Interaction:

- **Function:** Facilitates message transmission between the various components.
- **Interaction:** Used for real-time updates and non-critical operations that require fast data transmission.
- **Design Considerations:** Balances between reliability and speed, ensuring that real-time information is delivered promptly.

2.5 Security and Data Management

- **User Authentication:** The system ensures secure user authentication by verifying login credentials through the TCP-Group server. This process includes encrypted transmission of sensitive data to prevent unauthorized access.
- **Data Integrity:** Credit points and transaction details are securely managed and updated to maintain data integrity. The use of TCP ensures reliable and ordered delivery of critical information.
- **Secure Communication:** Both TCP and UDP protocols are implemented with encryption to protect data during transmission, ensuring that user information remains confidential.

2.6Performance and Scalability

- **Concurrency Management:** The TCP-Group server's ability to handle multiple client requests concurrently is crucial for performance. Multi-threading ensures that each client is serviced independently, reducing wait times and increasing throughput.
- **Load Balancing:** The distribution of tasks among the Silver, Gold, and Platinum servers ensures that no single server becomes a bottleneck. This load balancing is critical for maintaining high performance during peak usage.
- **Real-Time Updates:** The use of UDP for real-time updates ensures that users receive timely notifications about their transactions, enhancing the overall user experience.

User Experience

- **Intuitive Interface:** The system is designed to provide a user-friendly interface, making it easy for users to navigate, input data, and complete transactions.
- **Real-Time Feedback:** Users receive real-time feedback on their actions, such as login success, credit point updates, and transaction status, ensuring transparency and engagement.
- **Error Handling:** Robust error handling mechanisms are in place to guide users in case of invalid inputs or system errors, enhancing reliability and user trust.

The system analysis of the distributed group-based shopping system reveals a well-structured application that efficiently utilizes TCP and UDP protocols for secure,

reliable, and real-time communication. The design considerations focus on scalability, performance, and user experience, ensuring that the system can handle high traffic volumes and complex transactions effectively. This analysis provides a comprehensive understanding of the system's architecture, security measures, and performance optimizations, serving as a foundation for further

2.6 EXISTING SYSTEM

The current system for shopping is to visit the shop manually and from the available product choose the item the customer wants and buy the item by payment of the price of the item .

1. It is less user-friendly.
2. Users must go to shop and select products.
3. It is difficult to identify the required product.
4. Description of the product limited.
5. It is a time consuming process

2.6 PROPOSED SYSTEM

In the proposed system customers need not go to the shop to buy the products. He can order the product he wishes to buy through the application in his Smartphone. The shop owner will be the admin of the system. Shop owners can appoint moderators who will help the owner in managing the customers and

product orders. The system also recommends a home delivery system for the purchased products.

2.7 SYSTEM REQUIREMENT SPECIFICATION

2.3.1 GENERAL DESCRIPTION

Product Description:

The system consists of two parts .A web application which can provide the online shopping service and an desktop application for the customer to

access the web service from his Smartphone. a desktop application should be able to help the customer for selecting his item and to help the owner in managing the orders from the customers.

Problem Statement:

As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have an effortless shopping experience and save time through shopping online. To compete with those online brands , If shops are providing an online portal where their customers can shop through the internet and get the products at their doors, it will increase the number of customers.

2.3.2 SYSTEM OBJECTIVES

- To provide an desktop application application for online shopping of products in an existing shop.
- To provide a online shopping web site for the same shop. 2.3.3

SYSTEM REQUIREMENTS

2.3.3.1 NON FUNCTIONAL REQUIREMENTS

i. EFFICIENCY REQUIREMENT

When an online shopping cart desktop application is implemented, customers can purchase products in an efficient manner.

ii. RELIABILITY REQUIREMENT

The system should provide a reliable environment to both customers and owners. All orders should be reached at the admin without any errors.

3. SYSTEM DESIGN

The distributed group-based shopping system is designed with a robust architecture that leverages both TCP and UDP protocols to ensure secure, efficient, and real-time communication.

Core Components

1. **TCP Iterative Client:**
    - **Purpose:** Collects user input (IP, port, login credentials, and credit points).
    - **Function:** Establishes a TCP connection to send data to the TCP-Group server.
2. **TCP Concurrent Group Server:**
    - **Purpose:** Acts as the central hub, managing multiple client requests.
    - **Function:** Utilizes multi-threading to handle concurrent connections, verifies login credentials, updates credit points, and communicates with dedicated servers (Silver, Gold, Platinum).
3. **Dedicated Servers (Silver, Gold, Platinum):**
    - **Purpose:** Manage shopping processes based on user membership levels.
    - **Function:** Perform specialized tasks and transactions, providing tailored services.
4. **UDP Iterative Client-Server Interaction:**
    - **Purpose:** Handles real-time updates and non-critical data transmission.
    - **Function:** Ensures quick, efficient message delivery between system components.

Design Considerations

- **Security:** Data encryption during TCP and UDP transmissions to protect user information.
- **Concurrency:** Multi-threading in the TCP-Group server to manage simultaneous client connections effectively.
- **Scalability:** Distribution of tasks among dedicated servers to prevent bottlenecks and ensure high performance.

- **User Experience:** Intuitive interface with real-time feedback and robust error handling for seamless user interaction.

5. Summary

This system design ensures a scalable, secure, and user-friendly platform, capable of handling high volumes of transactions with real-time efficiency. It provides a foundation for advanced distributed applications through its efficient use of communication protocols and server management.

6 INPUT AND OUTPUT DESIGN

3.1.1 INPUT DESIGN:

Input design is the link that ties the information system into the world of its

users. The input design involves determining the inputs, validating the data, minimizing the data entry and providing a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user originated inputs are converted to a computer based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing.

3.1.2 OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient

manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

3.2 FILE SYSTEM

FILE DESIGN:

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary Key - the field that is unique for all the record occurrences.
- Foreign Key -the field used to set relation between tables. Normalization is a technique to avoid redundancy in the tables.

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

Java Script

JS is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side

scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

Java Script is used to create pop up windows displaying different alerts in the system like “User registered successfully”, ”Product added to cart” etc.

3.3.2 BACK END

An application software called Navicat was used to design the tables in in uses java language

3.4 TABLES

 LOGIN

login

Field Name	Field Type	Default	AllowNull	PriKey	Extra
uid	int(11) FIRST		NO	YES	auto_increment
uname	varchar(50) AFTER `uid`		NO	NO	
upass	varchar(50) AFTER `uname`		NO	NO	
utype	varchar(20) AFTER `upass`		NO	NO	
enabled	varchar(10) AFTER `utype`		NO	NO	

Fig 3.1: Login Table

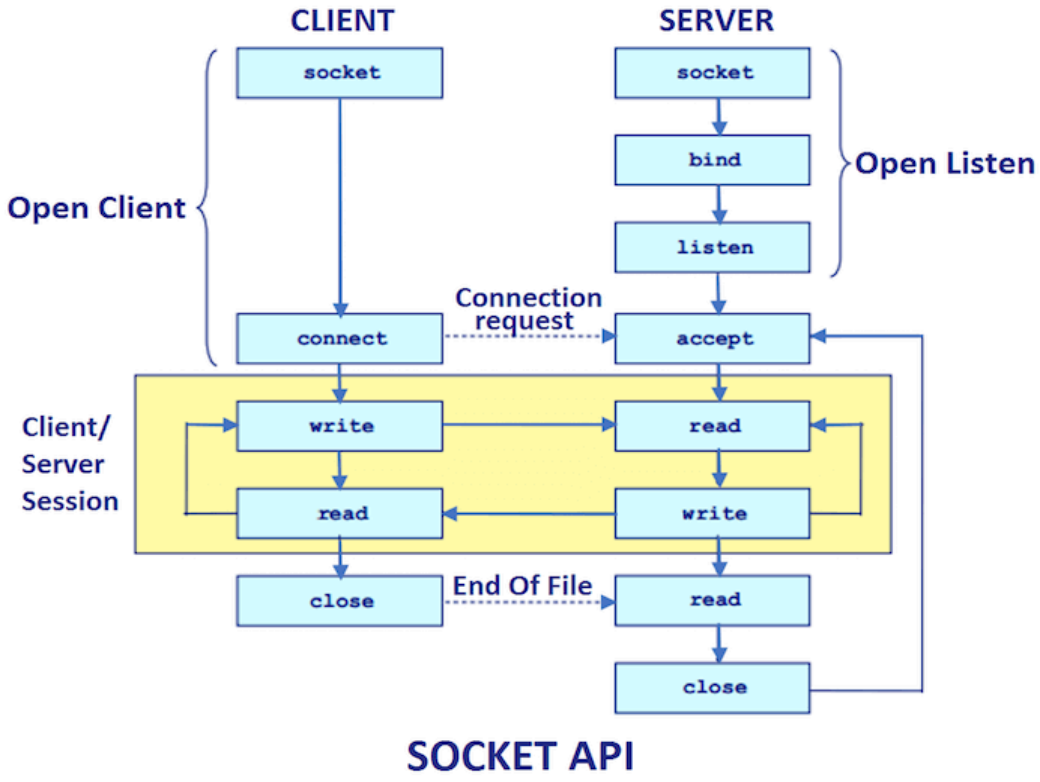
3. E-R DIAGRAMS

1 DATA FLOW DIAGRAM

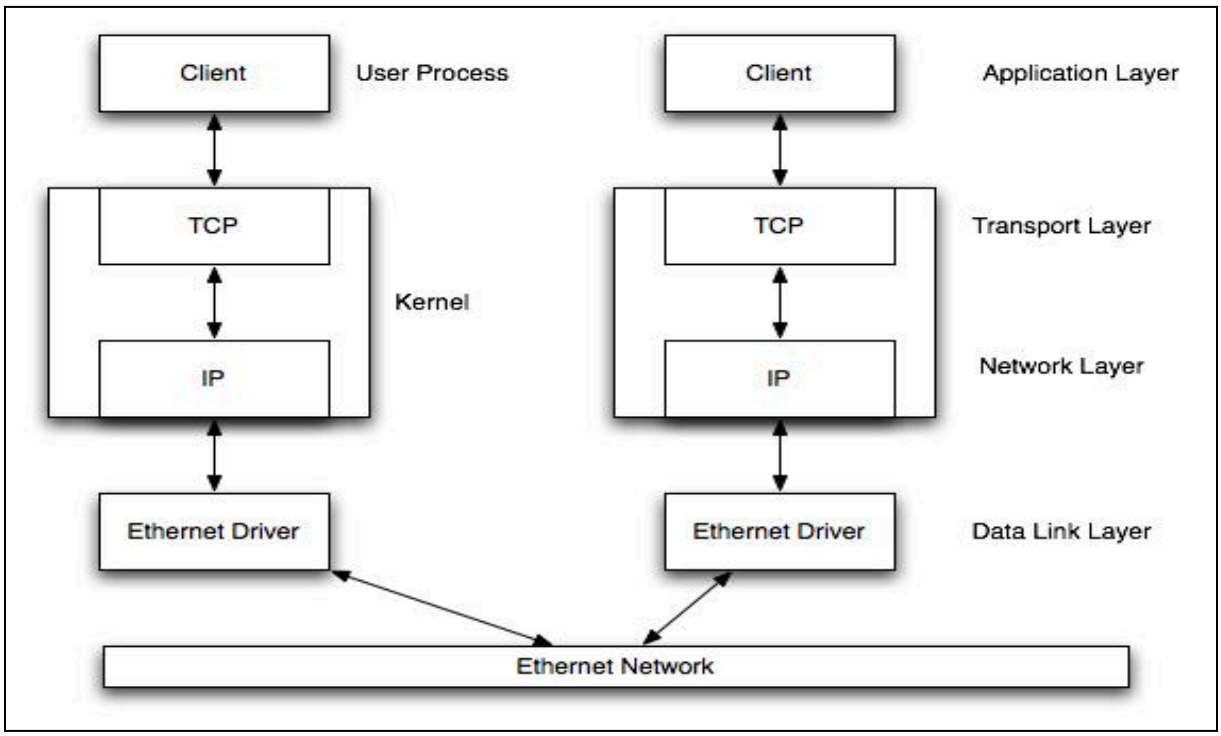
A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical

implementation. It has the purpose of clarifying system

requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to



Admin module



Manage Product

the lowest level of detail. DFD can be considered to be an abstraction of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons DFD's are often referred to as logical data flow diagrams.

EXTERNAL ENTITY

An external entity is a source or destination of a data flow. Only those entities which originate or receive data are represented on a data flow diagram. The symbol used is a rectangular box.

PROCESS

A process shows a transformation or manipulation of data flow within the system. The symbol used is an oval shape.

DATAFLOW

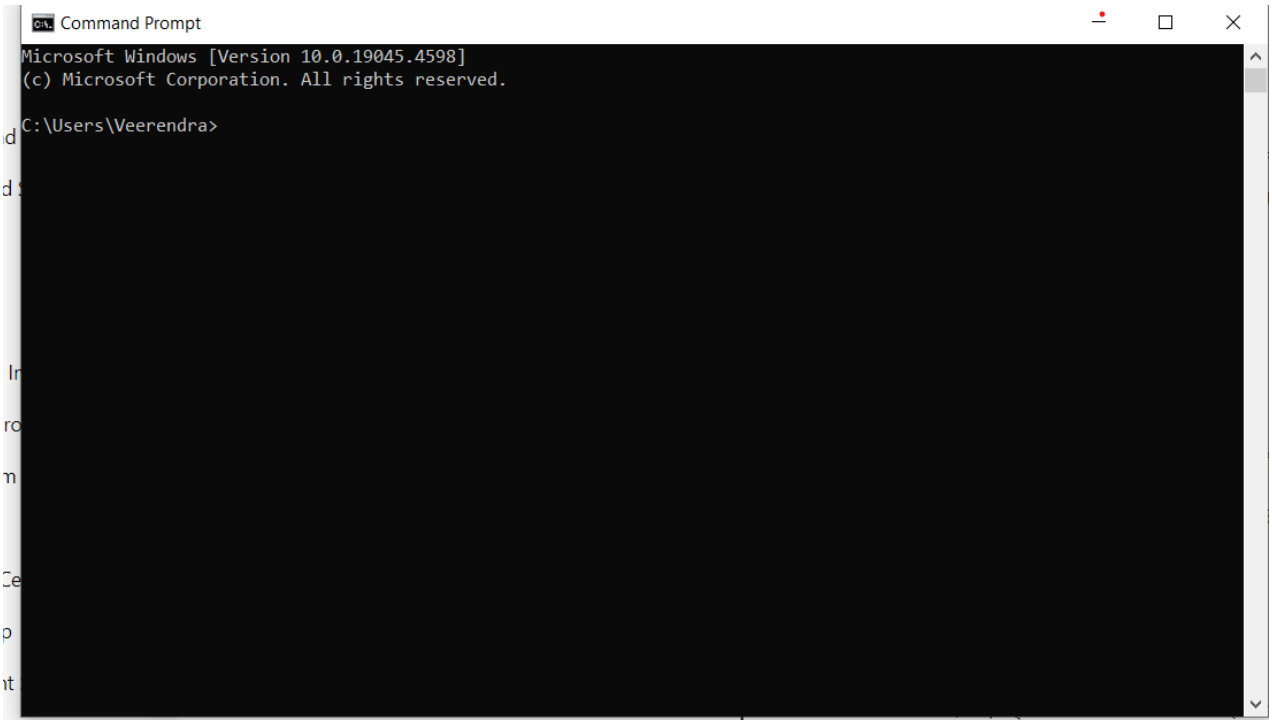
The data flow shows the flow of information from a source to its destination. Data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents.

DATA STORE

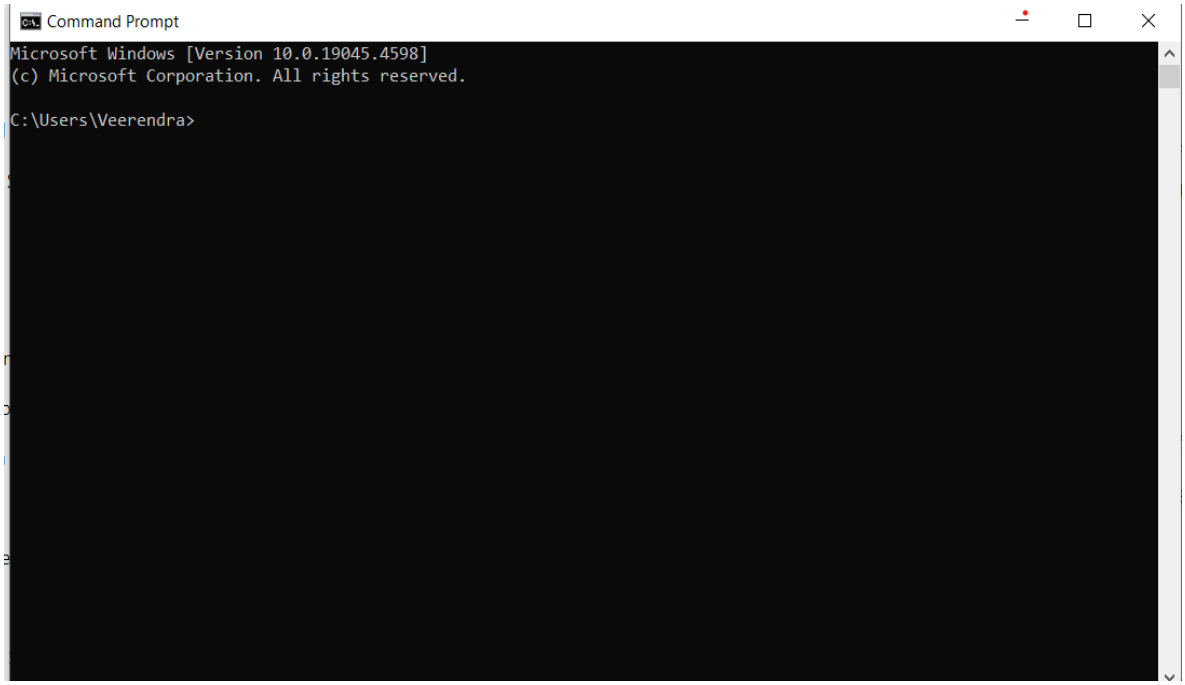
A data store is a holding place for information within the system: It is represented by an open ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an

2 SCREEN SHOTS

 REGISTRATION



➢ Admin



Command Prompt

6. CONCLUSION

The project entitled **Server middel shopping system** was completed successfully.

The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application and an android application for purchasing items from a shop.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using html & css, usage of responsive templates, designing of desktop applications, and management of . The entire system is secured. Also the project helped us understand about the development phases of a project and software development life cycle. We learned how to test different features of a project.

This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications.

There is a scope for further development in our project to a great extend. A number of features can be added to this system in future like providing

7. REFERENCES

1. Java Network Programming by Elliotte Rusty Harold: This book provides a comprehensive introduction to Java networking APIs and covers topics such as TCP/IP protocols, sockets,

and Java's networking classes. It is an essential resource for understanding the fundamentals required for implementing TCP and UDP communication.

2. Distributed Systems: Principles and Paradigms by Andrew S. Tanenbaum and Maarten Van Steen: This book offers in-depth coverage of the principles and design patterns of distributed systems. It includes case studies and practical examples that are relevant to designing a distributed group-based shopping system.
3. Java Concurrency in Practice by Brian Goetz: This book is a detailed guide to writing concurrent applications in Java. It covers best practices for multi-threading and concurrency management, which are crucial for implementing a concurrent TCP-Group server.
4. Computer Networking: A Top-Down Approach by James Kurose and Keith Ross: This textbook provides a detailed look at networking concepts, including TCP/IP and UDP protocols, with practical examples that help in understanding how to implement these protocols in a real-world application.
5. Official Java Documentation (Oracle): The official documentation is an indispensable resource for Java developers. It provides detailed information on the Java Standard Library, including networking classes and APIs used for socket programming.
6. These references collectively provide the foundational knowledge and practical guidance necessary for developing a secure, efficient, and scalable distributed shopping system using