

**Table of Contents**

[1 Introduction 1](#_Toc151373794)

[1.1 Objective 1](#_Toc151373795)

[1.2 Scope 1](#_Toc151373796)

[2 Problem Overview 2](#_Toc151373797)

[3 Requirements 2](#_Toc151373798)

[3.1 Functional Requirements 2](#_Toc151373799)

[3.2 Non-Functional Requirements 3](#_Toc151373800)

[4 Research and design 3](#_Toc151373801)

[4.1 Use-case diagram 4](#_Toc151373802)

[4.2 Database Diagram 5](#_Toc151373803)

[4.3 ERD (Entity Relationship Diagram) 5](#_Toc151373804)

[5 Object-Oriented Programming (OOP) 6](#_Toc151373805)

[5.1 Abstraction 6](#_Toc151373806)

[5.2 Encapsulation 7](#_Toc151373807)

[5.3 Inheritance 8](#_Toc151373808)

[5.4 Polymorphism 8](#_Toc151373809)

[6 Implementation 9](#_Toc151373810)

[6.1 System Snippets 9](#_Toc151373811)

[6.1.1 Login Interface 9](#_Toc151373812)

[6.1.2 Signup interface 10](#_Toc151373813)

[6.1.3 Forget Password Interface 11](#_Toc151373814)

[6.1.4 User Dashboard 12](#_Toc151373815)

[6.1.5 Purchase Item Interface 12](#_Toc151373816)

[6.1.6 Purchase Items with Cart Table 13](#_Toc151373817)

[6.1.7 Receipt or invoice print 14](#_Toc151373818)

[6.1.8 Receipt Generation 15](#_Toc151373819)

[6.1.9 Purchase Details interface 16](#_Toc151373820)

[6.1.10 User Profile Management interface 17](#_Toc151373821)

[6.1.11 Admin Dashboard 18](#_Toc151373822)

[6.1.12 Manage Category Interface 19](#_Toc151373823)

[6.1.13 Product Management Interface 20](#_Toc151373824)

[6.1.14 User Management Interface 21](#_Toc151373825)

[6.1.15 Supplier Management Interface 22](#_Toc151373826)

[6.1.16 Supplier Selection Page 23](#_Toc151373827)

[6.1.17 Transaction View Page 24](#_Toc151373828)

[6.1.18 Supplier Dashboard 25](#_Toc151373829)

[6.1.19 Delivery List and confirmation Interface 26](#_Toc151373830)

[6.1.20 Delivered Items Details Interface 27](#_Toc151373831)

[6.1.21 Supplier Account Management 28](#_Toc151373832)

[7 Techniques used in the development of RMI application 28](#_Toc151373833)

[7.1 Code Architecture 29](#_Toc151373834)

[7.2 RMI interface implementations 30](#_Toc151373835)

[7.3 Implementation of remote method 30](#_Toc151373836)

[7.4 Implementation of RMI methods in Client Program or interface 31](#_Toc151373837)

[7.5 RMI Server Setup 33](#_Toc151373838)

[7.6 RMI Client Setup 34](#_Toc151373839)

[7.7 Server Initialization before running Application 36](#_Toc151373840)

[7.8 Error if the Remote Server is not running 36](#_Toc151373841)

[8 Testing 36](#_Toc151373842)

[9 Relation: Distributed system and Blockchain 37](#_Toc151373843)

[10 Conclusion 39](#_Toc151373844)

[11 Future Enhancement 40](#_Toc151373845)

[12 References 41](#_Toc151373846)

[13 Appendices 42](#_Toc151373847)

[13.1 Gantt Chart 42](#_Toc151373848)

[13.2 Workload Matrices 42](#_Toc151373849)

**Table of Figures**

[Figure 1: Functional Requirements 3](#_Toc151373850)

[Figure 2: Non-Functional Requirements 3](#_Toc151373851)

[Figure 3: Use-Case Diagram 4](#_Toc151373852)

[Figure 4: Schema Diagram 5](#_Toc151373853)

[Figure 5: ERD 6](#_Toc151373854)

[Figure 6: Abstraction Example 7](#_Toc151373855)

[Figure 7: Encapsulation Example 7](#_Toc151373856)

[Figure 8: Inheritance Example 8](#_Toc151373857)

[Figure 9: Polymorphism Example 8](#_Toc151373858)

[Figure 10: Login Interface 9](#_Toc151373859)

[Figure 11: Signup Interface 10](#_Toc151373860)

[Figure 12: Forgot Password Interface 11](#_Toc151373861)

[Figure 13: User Dashboard 12](#_Toc151373862)

[Figure 14: Purchase Item Interface 12](#_Toc151373863)

[Figure 15: Purchase Items with cart Table Interface 13](#_Toc151373864)

[Figure 16: Receipt or invoice print Interface 14](#_Toc151373865)

[Figure 17: Receipt Generation Interface 15](#_Toc151373866)

[Figure 18: Purchase Details Interface 16](#_Toc151373867)

[Figure 19: User Profile Management Interface 17](#_Toc151373868)

[Figure 20: Admin Dashboard 18](#_Toc151373869)

[Figure 21: Manage Category Interface 19](#_Toc151373870)

[Figure 22: Product Management Interface 20](#_Toc151373871)

[Figure 23: User Management Interface 21](#_Toc151373872)

[Figure 24: Add Supplier Interface 22](#_Toc151373873)

[Figure 25: Manage Supplier Interface 22](#_Toc151373874)

[Figure 26: Supplier Selection Page 23](#_Toc151373875)

[Figure 27: Transaction View Page 24](#_Toc151373876)

[Figure 28: Supplier Dashboard 25](#_Toc151373877)

[Figure 29: Delivery List and Confirmation Interface 26](#_Toc151373878)

[Figure 30: Delivered Items Details Interface 27](#_Toc151373879)

[Figure 31: Supplier Account Management Interface 28](#_Toc151373880)

[Figure 32: Code Architecture 29](#_Toc151373881)

[Figure 33: RMI interface implementations 30](#_Toc151373882)

[Figure 34: Implementation of Remote Method 31](#_Toc151373883)

[Figure 35: Implementation of RMI methods in Client Program or interface Example 32](#_Toc151373884)

[Figure 36: RMI Server Setup code snippet 33](#_Toc151373885)

[Figure 37: RMI client setup code snippets 35](#_Toc151373886)

[Figure 38: Server Initialization 36](#_Toc151373887)

[Figure 39: Error Message if server not running 36](#_Toc151373888)

[Figure 40: Gantt Chart 42](#_Toc151373889)

[Figure 41: Workload Matrix 42](#_Toc151373890)

# Introduction

CKF Group Malaysia, which is also referred as CKF is a global e-commerce enterprise that focuses on providing a wide range of famous products in many categories. These categories include of best-selling goods like cell phones, home appliances, accessories, and baby supplies. CKF has become a household name in the area by attracting loyal consumers and growing at an exponential rate. But the rapid expansion of CKF has also brought up some difficulties which is mainly related to product delivery and order fulfillment. The product delivery system (PDS) which CKF is currently using has not been able to support the company's growth trajectory. There are lot of difficulties that CKF is facing from which lack of user-friendliness, scalability constraints, and security difficulties are the common difficulties. These flaws make it more difficult for CKF to satisfy client demands and effectively handle the rising demand for its goods.

The task for our team is to use Remote Method Invocation (RMI) technology to develop a new Product Delivery System (PDS) for CKF. RMI will simplify distributed system development by enabling communication between client and server applications. Java programming will be used to build the system and will guarantee the username verification and secure communication during the registration process. The client application will provide options for both a graphical user interface and a command line which will provide flexibility. SSL or TLS technology will be used to protect communication between the client and server components in order to increase security.

## Objective

* To develop PDS for CFK making sure it has increased efficiency and provide seamless communication between client and server applications.
* To improve delivery processes, shorten delivery times and enter new markets to increase revenue.
* To develop user-friendly and reliable PDS for customer satisfaction.
* To develop more secure PDS for customers.

## Scope

The scope of this project is to develop and implement advanced Product Delivery System (PDS) by developing a user-friendly solution, utilizing technology to improve logistics as well as by conducting phased testing of project.

# Problem Overview

In the constantly evolving world of e-commerce, CKF Group Malaysia's Product Delivery System (PDS) is facing drawback which is forcing in compromising in its efficiency. The major issue has developed in its usability due to complex user interfaces in the system which is also impacting negatively in its internal operating efficiency and customer experience. Not only in usability, the PDS has also encounter difficulties with scalability which has made difficult to efficiently adjust to the growing market demand for CKF's varied product categories. Delivery delays, issues with inventory control, and an overall incapacity to keep up with the market's explosive expansion are some examples of the difficulties. Additionally, inefficiency in delivery processes has cause order fulfillment to be delayed and error-prone.

As a result, these difficulties has made customers dissatisfied which damages trust and can cause revenue loss for CKF. In order to overcome these difficulties, CKF has understood the necessity of developing a new PDS with a user-friendly interface, scalability, effective control over delivery operations, strong client account management, and extensive reporting abilities.

In order to resolve these present issues, CKF has plan to build up new PDS system with the help or by using RMI and Java programming language for long-term success in the competitive

e-commerce industry by gaining consumer happiness and streamlining internal operations.

# Requirements

To address the drawbacks of the current PDS system and in order to meet the needs of CKF Group Malaysia, our team has developed the following project specifications:

## Functional Requirements

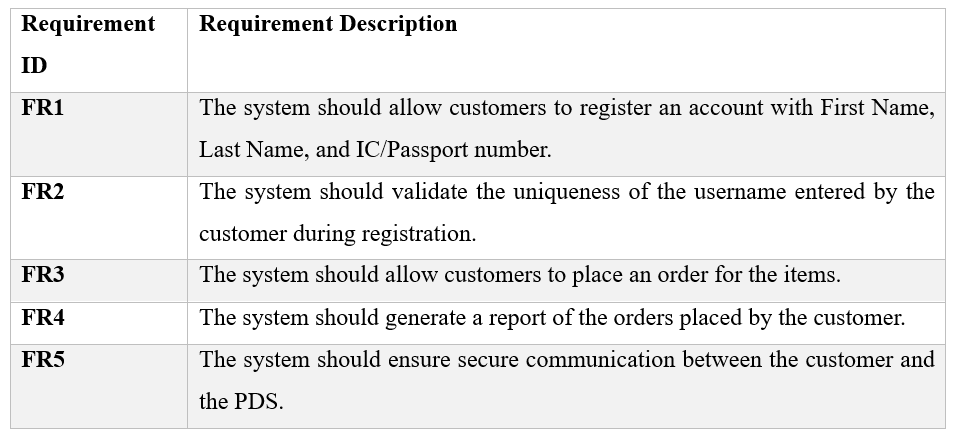
****

Figure 1: Functional Requirements

## Non-Functional Requirements

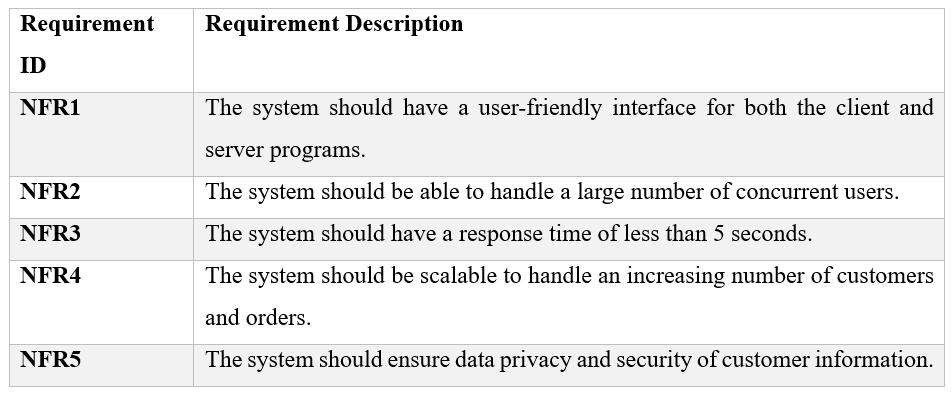
****

Figure 2: Non-Functional Requirements

# Research and design

Research is required to create a new Product Delivery System (PDS) for CKF's e-commerce platform. It comprises investigating various tools and technologies that might be used to create a trustworthy and secure system that meets the demands of the firm and its clients. The team started by figuring out what the new Product Delivery System needed using things like use-case, ER, and database diagrams. Our team did significant research to find out what features and stuff or requirement the system had to have. We also looked into different tools and technologies and ended up picking Java and RMI for the project (McCombes, 2023).

## Use-case diagram

This diagram displays that the features used are available to both customers and administrators. Customers can sign up, make orders, check their order status, and create reports right away. Administrators can regulate products, process or regulate orders, manage user accounts, and produce reports. The graphical diagram illustrates how users can interact with the system, highlighting essential functions and their connections (Paradigm, 2020).

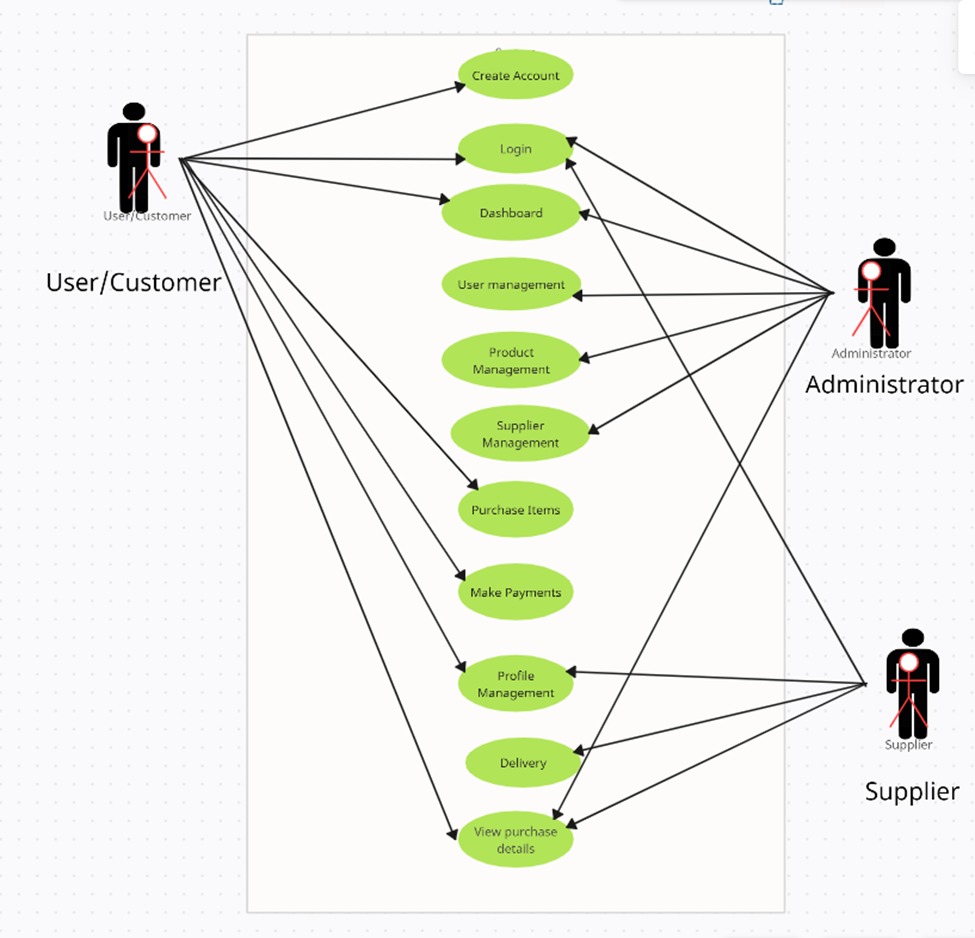


Figure 3: Use-Case Diagram

## Database Diagram

The project's database diagram presents an organized data architecture which is designed in such a way that the architecture will facilitate the effective administration of CKF Group Malaysia's product delivery system. The architecture comprises of interconnected tables which are responsible for storing and structuring data associated with deliveries, customers, orders, items, and different product categories (Mortier, 2022).

A diagram of a computer

Description automatically generated

Figure 4: Schema Diagram

## ERD (Entity Relationship Diagram)

Our product delivery system which has the Entity-Relationship Diagram (ERD) illustrates crucial relationships among entities such as customers, orders, items, and details. These entities are interlinked which offering a transparent representation of the data organization within the system. The ERD plays a vital role in guiding the design and development of the system by presenting entities and their interconnections between them (Ravikiran A S, 2023).

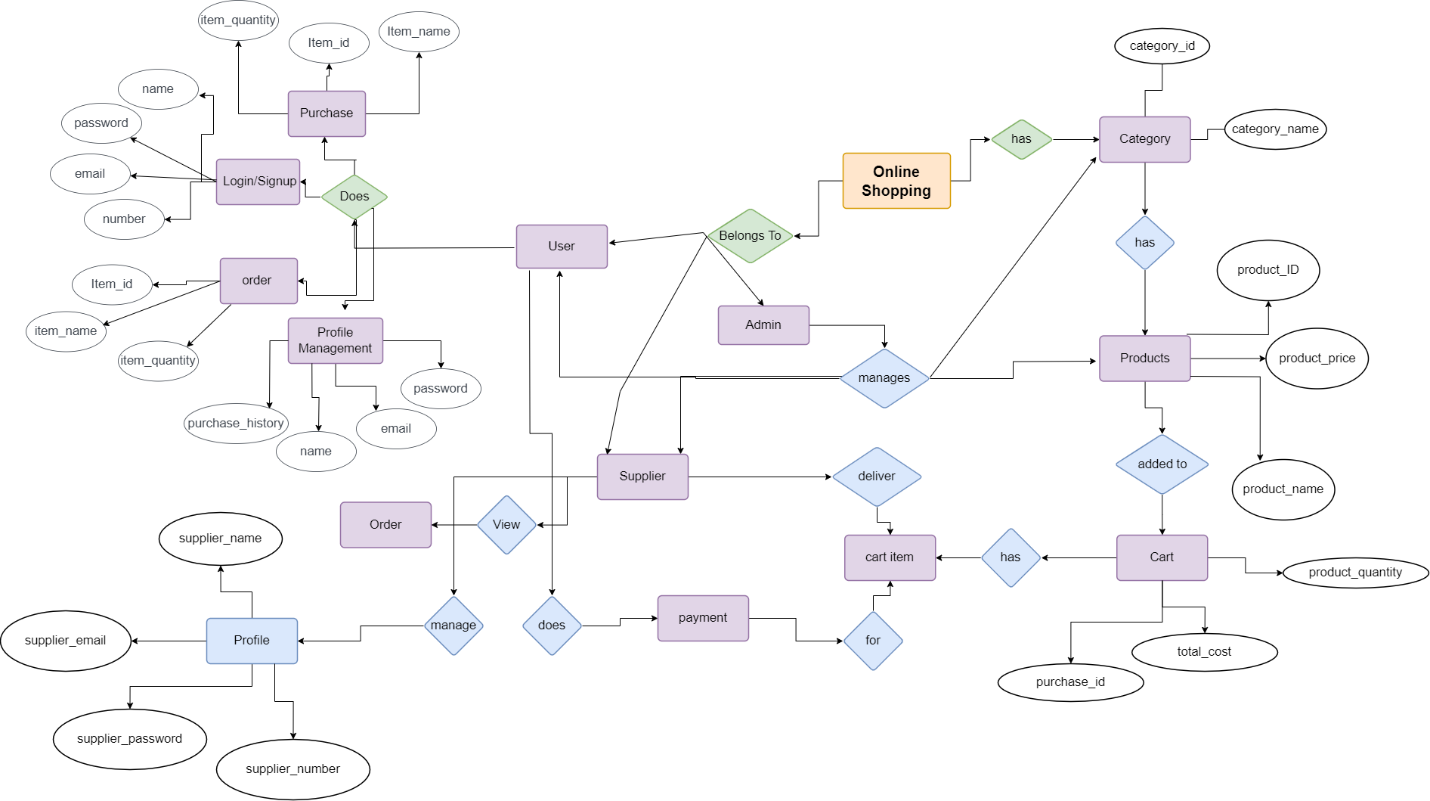


Figure 5: ERD

# Object-Oriented Programming (OOP)

Object-oriented programming (OOP) is a programming style that prioritizes data or objects above functions and logic. By linking data and functions, it aims to apply real-world concepts in programming, such as inheritance, encapsulation, and polymorphism. Encapsulation, inheritance, polymorphism, and abstraction are four fundamental ideas of OOP—provide framework for easily extensible as well as easy-to-maintain software design (BasuMallick, 2022).

## Abstraction

Abstraction is the process of concealing an application's internal workings from the outer world. An object has specific methods and properties. To hide them from users, we can utilize access modifiers. Only the essential features and functions can be provided to the other programs. This is how most OOPS abstractions are implemented (Costello, 2018).

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Figure 6: Abstraction Example

## Encapsulation

Encapsulation is one of the key concepts of object-oriented programming, or OOP. The practice of combining data and methods that manipulate it into a single entity, like a Java class is known as encapsulation. This concept is widely used to hide an object's internal representation or state from the outside world. This is referred to as information hiding. Software system flexibility, maintenance, and security are all enhanced by encapsulation (Janseen, 2022).

A computer screen shot of text

Description automatically generated

Figure 7: Encapsulation Example

## Inheritance

In object-oriented programming, inheritance refers to concept of one class inheriting attributes as well as methods of another. It's a programming approach that allows you to reuse code by using an object's data and actions. To put it another way class that inherits from another class shares every property and method of referred class (Hartman, 2023).



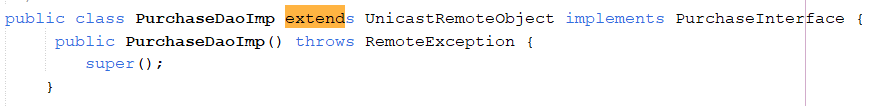


Figure 8: Inheritance Example

## Polymorphism

Polymorphism is a method in an object-oriented programming language that performs different actions based on class of object that calls it. Many class objects get message from polymorphism each object responds appropriately according to class's properties. It utilizes ideas of virtual functions, overriding, and function overloading (Taylor, 2021).

A close-up of a computer screen

Description automatically generated

A close-up of a computer code

Description automatically generated

Figure 9: Polymorphism Example

# Implementation

## System Snippets

### Login Interface

A screenshot of a login page

Description automatically generated

Figure 10: Login Interface

The above snippet shows the login page of PDS system. By providing email and password user, supplier as well as admin can login to the system with their own role and access given to them. Sign up button is also visible in login page where we can sign up if we don’t have account as well as we can do forget password if unable to remember old password.

### Signup interface

A screen shot of a computer screen

Description automatically generated

Figure 11: Signup Interface

The above figure provided is signup page of the system. If we don’t have account, we can signup up by entering the fields seen in above snippet that include user ID, username, Email, password, Phone, security question, answer of security question as well as address line 1 and 2.

### Forget Password Interface

A screenshot of a login form

Description automatically generated

Figure 12: Forgot Password Interface

Above figure shows forget password interface. If we forgot password, we can reset and set new password in shown figure. By providing email, security question set while doing sign up and answer to that question we can set new password. The save and back button can also be seen in this page.

### User Dashboard

A screenshot of a computer

Description automatically generated

Figure 13: User Dashboard

The above provided figure is snippet of user dashboard. In this form all the details can be seen such as; total purchase, total categories, total products as well as all the purchase details.

### Purchase Item Interface

A computer screen with a computer screen and a box with text

Description automatically generated

Figure 14: Purchase Item Interface

The provided snapshot is of purchase item interface. In this form, purchase details can be seen. By searching product in search bar, we can see product details. We can add, purchase items from purchase. In this form add, purchase, print as well as clear button is also can be seen. We can add to purchase list by clicking purchase id, quantity and then by adding. It provides a convenient and user-friendly way for customers to purchase products from the project.

### Purchase Items with Cart Table

A computer screen with a computer screen and a box

Description automatically generated

Figure 15: Purchase Items with cart Table Interface

The provided snippet is of purchase items with cart table. User can see all the details of purchase and total cost with the quantity.

### Receipt or invoice print

A screenshot of a computer

Description automatically generated

Figure 16: Receipt or invoice print Interface

The Receipt or invoice print Form, which includes elements such General where print service, print range as well as copies can be seen which makes it simple for users to print invoices.

### Receipt Generation

A screenshot of a computer

Description automatically generated

Figure 17: Receipt Generation Interface

The Receipt Form, which includes elements such as itemized data and total price, makes it simple for users to process bill payments, calculate, pay, and print invoices.

### Purchase Details interface

A screenshot of a computer

Description automatically generated

Figure 18: Purchase Details Interface

The picture shows an purchase viewing page with payment information as well as comprehensive list of all the things that were ordered, providing comprehensive summary of every purchase details.

### User Profile Management interface

A screenshot of a computer

Description automatically generated

Figure 19: User Profile Management Interface

### Admin Dashboard

A screenshot of a shopping chart

Description automatically generated

Figure 20: Admin Dashboard

The provided picture is of admin dashboard. Admin can view all the details of users, suppliers, as well as the details of all the sales and products that are supplied also can be viewed by admin from admin dashboard.

### Manage Category Interface

A screenshot of a computer

Description automatically generated

Figure 21: Manage Category Interface

Administrators can establish new project product categories on the manage Category Page after successful inclusion, which includes choices for name and description.

### Product Management Interface

A screenshot of a computer

Description automatically generated

Figure 22: Product Management Interface

The image above is a screenshot of the system's "manage product" page, which allows you to update products.

### User Management Interface

A screenshot of a computer

Description automatically generated

Figure 23: User Management Interface

A screenshot of the "User Management" form on the system is shown in the image above. This page lists every user in the system in detail and includes all of their pertinent data.

### Supplier Management Interface

A screenshot of a computer screen

Description automatically generated

Figure 24: Add Supplier Interface

A screenshot of a computer

Description automatically generated

Figure 25: Manage Supplier Interface

The above provide two snippets are of add supplier and manage supplier.

### Supplier Selection Page

A screenshot of a computer

Description automatically generated

Figure 26: Supplier Selection Page

The above snippet is of select supplier. Where user can select supplier of to deliver their order from this page.

### Transaction View Page

A screenshot of a computer

Description automatically generated

Figure 27: Transaction View Page

This page provides a full breakdown of each payment made, giving the user an accurate representation of their transactions.

### Supplier Dashboard

A screenshot of a computer

Description automatically generated

Figure 28: Supplier Dashboard

The provided figure is of supplier dashboard. In this supplier can view his delivery details as well his all his account detail from this page.

### Delivery List and confirmation Interface

A screenshot of a computer

Description automatically generated

Figure 29: Delivery List and Confirmation Interface

The above interface is of delivery list and confirmation interface. In this delivery list can be viewed in this page.

### Delivered Items Details Interface

A screenshot of a computer

Description automatically generated

Figure 30: Delivered Items Details Interface

In this page delivered item list can be viewed.

### Supplier Account Management

A screenshot of a computer screen

Description automatically generated

Figure 31: Supplier Account Management Interface

The above provided figure is of supplier account. From this page supplier detail can be updated as well as his account can also be deleted by admin.

# Techniques used in the development of RMI application

The below screenshots are some examples for RMI application implementation used in our project. We have used same methods and components for RMI interface and implementation, we have provided some snippets for example.

## Code Architecture

A screenshot of a computer

Description automatically generated

Figure 32: Code Architecture

## RMI interface implementations

A text on a white background

Description automatically generated

A screenshot of a computer

Description automatically generated

Figure 33: RMI interface implementations

## Implementation of remote method

A screen shot of a computer

Description automatically generated

Figure 34: Implementation of Remote Method

## Implementation of RMI methods in Client Program or interface

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

Figure 35: Implementation of RMI methods in Client Program or interface Example

## RMI Server Setup

A screenshot of a computer code

Description automatically generated

Figure 36: RMI Server Setup code snippet

## RMI Client Setup

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer code

Description automatically generated

Figure 37: RMI client setup code snippets

## Server Initialization before running Application

A screenshot of a computer

Description automatically generated

Figure 38: Server Initialization

## Error if the Remote Server is not running

A screen shot of a computer

Description automatically generated

Figure : Error Message if server not running

# Testing

Testing represents an essential component of every software development effort which including the PDS system for CKF. The testing process plays a pivotal role in verifying the system's proper functionality which should be aligned with customer requirements with the lack of defects. Presented below are various tests conducted for our project (Kinsbruner, 2023).

|  |  |  |
| --- | --- | --- |
| Genre of testing | Outline | Scenario of our System Test |
| Unit Testing | Individual system components evaluated independently. | Verifying the accuracy of data storage by checking the registration procedure for creating a new consumer account. |
| A close-up of black text  Description automatically generated | Investigating how various system components interact with one another. | Ensuring accurate data transmission by testing the communication between the client program and server program using RMI. |
| A black text on a white background  Description automatically generated | Conducting extensive evaluation of the entire system. | Verifying the proper functionality of the entire system by testing end-to-end processes which includes user registration, product ordering, and report generation. |
| Black text on a white background  Description automatically generated | Analyzing the system without prior knowledge of its internal workings or source code. | Analyzing the user interface of the user program to ensure it is user-friendly and straight forward. |
| A white box testing with black text  Description automatically generated | Reviewing the system while taking into consideration its internal logic or source code. | Reviewing the secure and accurate handling of customer data by testing the logic of the server program. |
| A black text on a white background  Description automatically generated | Verifying that the system meets the needs and requirements of actual users. | Granting a group of user access to the system and collecting feedback on their user experience. |

# Relation: Distributed system and Blockchain

In the process of development of the CKF Product Delivery System (PDS), the relation between a distributed system and blockchain plays a crucial role in order to help to shape the architecture's core principles. To enhance fault tolerance and scalability, Remote Method Invocation (RMI) is utilized for communication between the client and server programs establishes a distributed system, distributing components across nodes. The core idea of blockchain which runs on a decentralized peer-to-peer network is to provide security and transparency which is reflected in this decentralization. Using cryptographic techniques, the distributed system and blockchain both provide secure communication which is very important factor in the PDS since it safeguards data while it is in transit and also guards from illegal access.

While a distributed system focuses on data consistency and integrity through consensus algorithms and the blockchain takes these principles more further by creating a fixed ledger which is resistant to tampering and helps to provide a transparent history of transactions.

Additionally, the concept of smart contracts which is inherit to blockchain platforms offers a medium for programmable and automated actions which will help to enhance the efficiency and security of certain aspects of the PDS (Thành, 2022).

# Conclusion

The project focused on a development, and implementation of the new Product Delivery System (PDS) in order to mark a significant approach toward overcoming the challenges stated by the previous system. To enhance the scalability, fault tolerance, and user-friendly interfaces of the PSD, Remote Method Invocation (RMI) technology and a distributed system architecture made was utilized. To ensures the protection of sensitive data with the implementation of a robust registration process which adds an additional layer of security was due to the incorporation of secure communication through SSL/TLS technology. The utilization of Java programming language contributed in enhancing the system's flexibility, supporting both command line and graphical user interfaces for the client program. Overall, the project not only resolves the identified shortcomings in the existing PDS but also lays the foundation for future enhancements. All this helped in embracing the principles of decentralization, security, and adaptability in the evolving world of e-commerce.

# Future Enhancement

The CKF e-commerce platform's new Product Delivery System (PDS) may be enhanced with the use of virtualization and cloud computing technologies. This would help with scalability, managing more orders and traffic, and flexible resource management. Additionally, it could strengthen security, disaster recovery, and fault tolerance. Machine learning algorithms have the potential to improve the delivery process by analysing data such as driver availability, traffic patterns, and delivery times. As a consequence, delivery times would be reduced and resources would be used more efficiently, eventually increasing the customer experience. These upgrades could increase the effectiveness and efficiency of CKF's PDS, guaranteeing that it satisfies consumer needs and expectations.

# References

BasuMallick, C. (2022). What Is OOP (Object Oriented Programming)? https://www.spiceworks.com/tech/devops/articles/object-oriented-programming/

Costello, D. (2018). What is abstraction in photography? British Journal of Aesthetics, 58(4), 385–400. https://doi.org/10.1093/aesthj/ayy037

Hartman, J. (2023). Inheritance in Java OOPs: Learn Different Types with Example. Guru99. https://www.guru99.com/java-class-inheritance.html

Janseen, T. (2022). OOP Concept for Beginners: What is Encapsulation. Stackify. https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/

Kinsbruner, E. (2023). 17 Different Types of Testing in Software | Types of Automated Application Testing. https://www.perfecto.io/resources/types-of-testing

McCombes, S. (2023). What Is a Research Design | Types, Guide & Examples. Scribbr, 1–19. https://www.scribbr.com/methodology/research-design/

Mortier, G. (2022). What Is a Database Diagram? | Vertabelo Database Modeler. https://vertabelo.com/blog/what-is-database-diagram/

Paradigm, V. (2020). What is Use Case Diagram? In Visual Paradigm. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/;WWWSESSIONID=612C8B3D05E57E77D4690805B34C35EE.www1

Ravikiran A S. (2023). ER Diagrams in DBMS\_ Entity Relationship Diagram Model \_ Simplilearn. https://www.simplilearn.com/tutorials/sql-tutorial/er-diagram-in-dbms

Taylor, T. (2021). Understanding the role of polymorphism in OOP | TechTarget. https://www.techtarget.com/searchapparchitecture/tip/Understanding-the-role-of-polymorphism-in-OOP

Thành, P. N. (2022). Distributed systems on blockchain - Viblo. https://viblo.asia/p/distributed-systems-on-blockchain-E1XVOXap4Mz

# Appendices

## Gantt Chart

Figure 40: Gantt Chart

## Workload Matrices

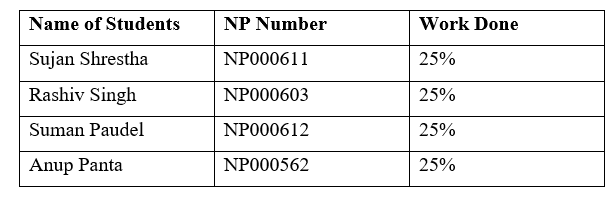


Figure 41: Workload Matrix