UCS2313 – Object Oriented Programming Lab

MINI PROJECT

Truck Reservation System

Madhumitha E 3122 22 5001 068

Micheal Berdinanth M 3122 22 5001 071

Nandhalal S 3122 22 5001 075



Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam – 603110

**Contents:**

1. Problem Statement

2. Motivation

3. Scope and Limitations

4. Design of the Solution (Class Diagram)

5. Design Alternatives

6. Identification of process / modules / algorithms

7. Object oriented features used for the application

8. Integration of modules

9. Various Test Cases and Output screenshots

10. Inference and Future extension

11. Learning Outcomes

12. References

**1. Problem Statement**

Develop a Truck Reservation System that automates the logistics and transportation industry. The system should allow users, including logistics providers, truck owners, and agencies, to efficiently book and manage truck services, ensuring seamless cargo transportation while optimizing truck utilization and maintaining real-time status updates.

**2. Motivation**

The motivation behind the Truck Reservation System project lies in addressing the challenges faced by the logistics and transportation industry. By automating and streamlining the process of reserving trucks, the system aims to enhance efficiency, reduce manual efforts, and provide a centralized platform for users.

In comparison to traditional reservation systems for buses, trains, and airplanes, the Truck Reservation System brings a unique focus on the logistics and freight industry. While passenger reservation systems cater to individuals, the Truck Reservation System targets businesses and entities involved in cargo transportation. The motivation lies in addressing the specific needs of freight logistics, optimizing cargo movement, and streamlining truck reservations to facilitate a more efficient and tailored solution for goods transportation.

**3. Scope and Limitations**

The system streamlines the truck booking process, allowing users to easily reserve trucks based on their specific requirements. User wallets facilitate smooth financial transactions, deducting booking amounts and providing refunds in case of cancellations. Users receive real-time updates on the status of their bookings, ensuring transparency and enhancing the overall user experience. The system accommodates different user types, including logistics users and agency users, providing tailored features for each.

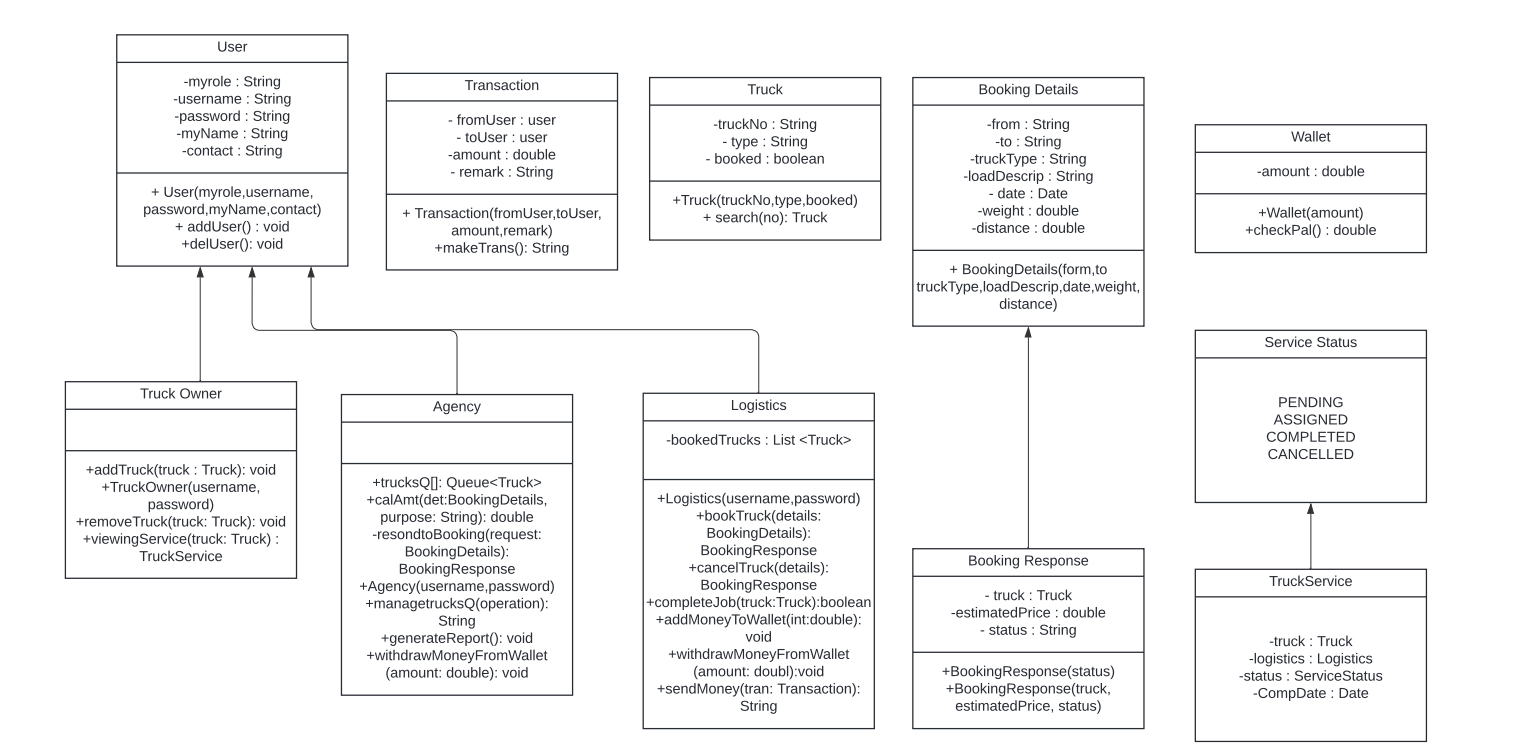
Users have the flexibility to cancel bookings, receiving partial refunds while also ensuring a fair financial distribution between users and the agency. The system accurately reflects the earnings of the agency, providing financial insights and promoting effective fund management.

Limitations of the Truck Reservation System: The email notification feature relies on external email services, and if there are issues with the email configuration or services, it might impact the communication aspect of the system. The financial model is simplified, and for a more sophisticated system, additional features such as detailed financial reports and analytics could be considered.

While the system ensures basic security measures, it might lack advanced security features required for handling sensitive information in a production environment. The system assumes the existence of a single agency user ("Mmb") for simplicity. In a real-world scenario, a more comprehensive user management system with multiple agencies would be needed.

The reliability of email delivery is subject to the configuration of the email server. Issues with email servers might lead to delays or failures in sending notifications. The current system has a basic command-line interface. For a more user-friendly experience, a graphical user interface (GUI) could be implemented in future versions.

**4. Design of the Solution (Class Diagram)**

****

**5. Design Alternatives**

User Interface: Developing a web-based user interface accessible through browsers ensures broad compatibility. Creating a mobile app for truck reservation facilitates on-the-go access for users.

OAuth Integration: Instead of a simple username/password authentication, the system could integrate OAuth for more secure and standardized user authentication.

Role-Based Access Control (RBAC): Implementing RBAC could provide more granular control over user permissions.

Real-Time Updates: Implementing push notifications can provide real-time updates to users. Employing a polling mechanism to periodically check for updates is an alternative approach.

User Authentication: Implementing robust user authentication mechanisms, possibly with multi-factor authentication, enhances security.   
   
Collection Framework: We can use LinkedList, HashMap, TreeMap instead of ArrayList.

**6. Identification of process / modules / algorithms**

User Authentication and Authorization

The system ensures secure user access through a robust authentication process. The User class, along with subclasses Logistics, TruckOwner, and Agency, implements a secure login system. Role-based authorization is managed through user types, allowing specific functionalities for each category.

Truck Booking Process

The performBooking method processes user requests, matching them with available trucks based on parameters like type, load, and distance. Users are promptly notified of successful bookings through the System.out.println statements, providing immediate feedback.

Truck and service management is handled through methods like performAddTruck and performRemoveTruck. These methods ensure accurate and real-time updates to the list of trucks. The FileHandler class manages data persistence.

Payment Processing

The payment processing module is encapsulated within the performBooking method. The system checks the user's wallet balance, deducts the booking amount, and ensures financial transactions are accurate and secure. Exception handling is implemented for insufficient funds.

Agency Operations

The performAgencyOperations method allows agencies to view booking requests, respond to them, and manage trucks. Specific algorithms within this method cater to agency functionalities.

Wallet Management

Wallet management is facilitated through the Wallet class, where users can view balances, add funds (addMoneyToWallet), withdraw money (withdrawMoneyFromWallet), and send money to other users.

Database Interaction

Data interactions with trucks, services, and bookings are managed through the FileHandler class. This class reads and writes data to text files, providing a simple form of database interaction.

Security Measures

Security measures include basic user authentication and authorization, ensuring that only authorized users can access specific functionalities. Further enhancements can be made by implementing more secure authentication mechanisms.

Error Handling and Exception Handling

Error handling is evident in the code, particularly in scenarios such as insufficient funds (InsufficientFundsException). Exception handling is implemented to detect and gracefully handle errors, enhancing the system's reliability.

Mail

Once the booking process is done, we will send an email to the user regarding booking confirmation with truck details.

**7. Object oriented features used for the application**

Data Abstraction

Data abstraction is achieved through the creation of classes such as Truck, User, and its subclasses (Logistics, TruckOwner, Agency). These classes encapsulate the essential attributes and behaviors, allowing the system to represent real-world entities in a simplified manner.

Inheritance

Inheritance is prominently utilized to establish a hierarchy among user types. The User class serves as the base class, while Logistics, TruckOwner, and Agency classes inherit common attributes and methods. This promotes code reuse and ensures a consistent structure.

Polymorphism

Polymorphism is demonstrated through method overloading and overriding. For instance, the calculatePrice method in the Main class is overloaded to handle different truck types. Additionally, the toString method in classes like BookingDetails is overridden to provide customized representations.

Data Encapsulation

Data encapsulation is achieved by defining private attributes in classes and providing public methods (getters and setters) for controlled access to these attributes. This ensures that the internal details of a class are hidden, promoting information hiding.

Method Overloading and Overriding

Method overloading is evident in the calculatePrice method, which is overloaded based on the truck type. Method overriding is used in the toString method of the BookingDetails class, providing a specific implementation for object representation.

Exception Handling

Exception handling is implemented to enhance the robustness of the system. For example, the InsufficientFundsException is thrown and caught in the performBooking method, ensuring proper handling of insufficient funds scenarios.

Collection Framework

The Collection Framework is employed for managing collections of objects. Lists, such as trucks, services, and bookings, are used to store and manipulate instances of trucks, services, and booking details. This facilitates efficient data storage and retrieval.

Multi-Threading

Multi-threading has been employed to enhance the user experience during the booking process. A separate thread is initiated to simulate a typewriter effect when displaying messages like "Please wait for agency confirmation" and "Booking in progress." This approach ensures a visually engaging and responsive user interface.

Java Activation Framework (JAF)

The Java Activation Framework (JAF) is used to handle data types in a modular and extensible manner. In or project, JAF is utilized for managing various data types related to truck reservations.

Java Mail API

Java Mail API is used for sending and receiving emails in Java applications. In the Truck Reservation System,we used Java Mail API to notify users about their bookings and confirmations. This enhances communication and keeps users informed.

**8. Integration of modules**

The Truck Reservation System is designed with a modular architecture, where distinct modules collaborate seamlessly to provide a comprehensive and efficient booking experience. User Interaction Module: The user interacts with the system through a well-defined interface, entering booking details, specifying cargo requirements, and confirming transactions.  
  
Booking Management Module: Manages the booking lifecycle, from user request to agency confirmation. Utilizes helper methods to find available trucks, calculate pricing, and update booking status.  
  
Truck and Truck Owner Management Module: Handles the addition and removal of trucks by truck owners. Maintains truck availability status and assigns trucks to booking requests.

Agency Operations Module: Enables agency users to view booking requests, respond to them, manage truck owners' fleets, and handle wallet operations.

Multi-Threading Module: Enhances user feedback during time-consuming processes by implementing a multi-threaded approach. For example, a separate thread is initiated to simulate a typewriter effect when displaying progress messages.

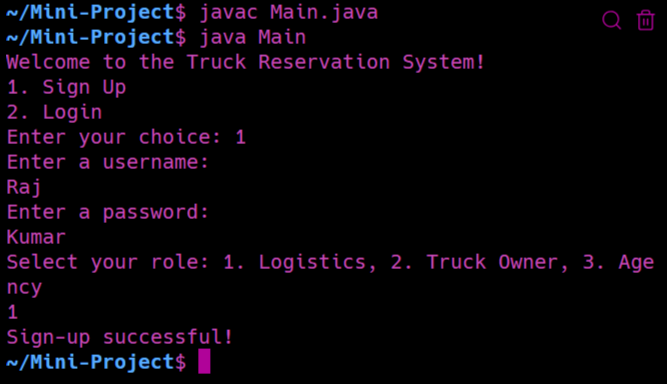
File Handling Module: Manages the persistence of crucial data such as truck details, services, and booking history. Ensures data integrity and allows the system to recover from interruptions.

Exception Handling Module: Implements robust exception handling mechanisms to gracefully manage unexpected scenarios, ensuring the system's stability.

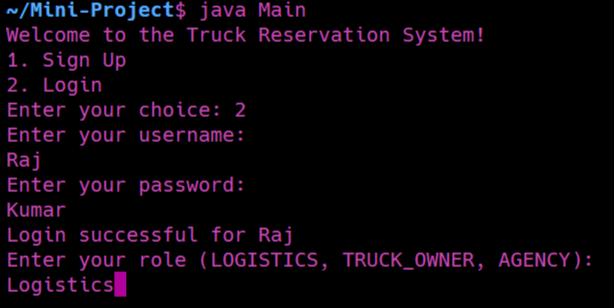
The seamless collaboration of these modules results in a cohesive Truck Reservation System that delivers a user-friendly and reliable experience.

**9. Various Test Cases and Output screenshots**

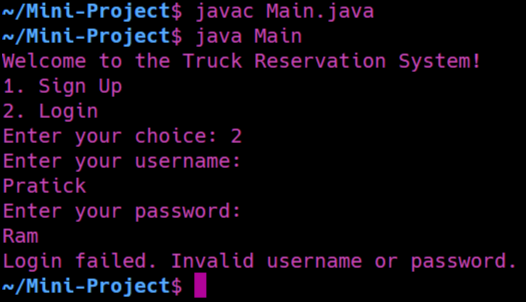
1. Signing up as a new user.



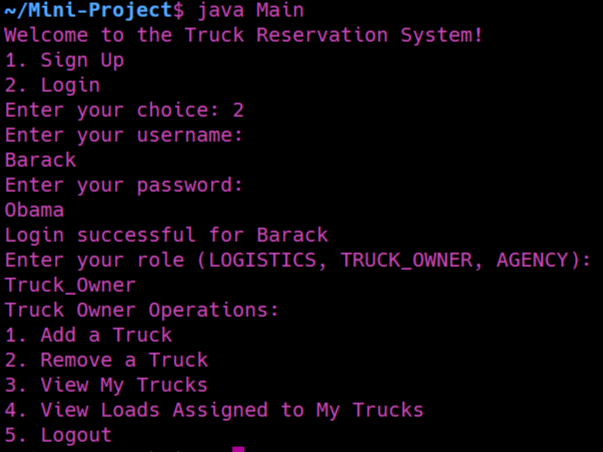
1. Logging in as existing user.



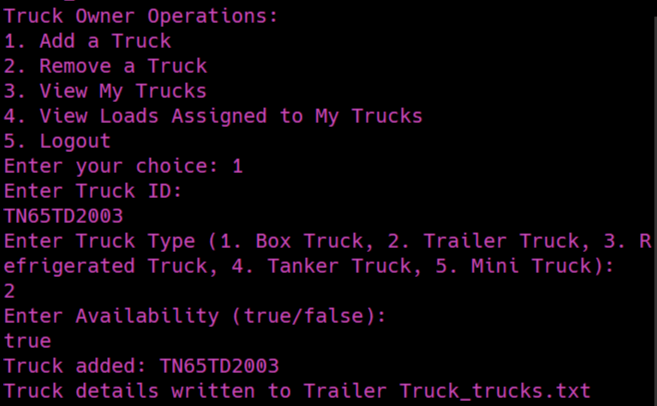
1. Logging in without creating an account (without signing in).



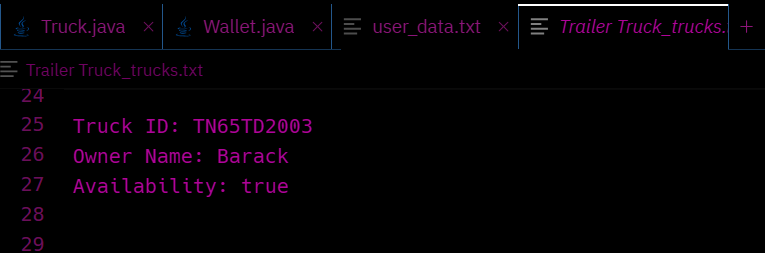
1. Truck Owner’s Operations



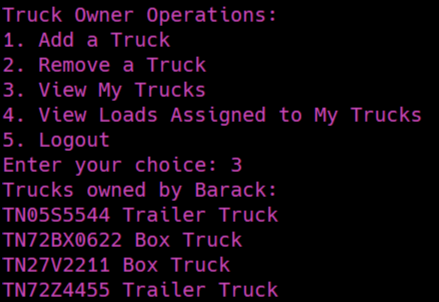
1. Adding a new truck



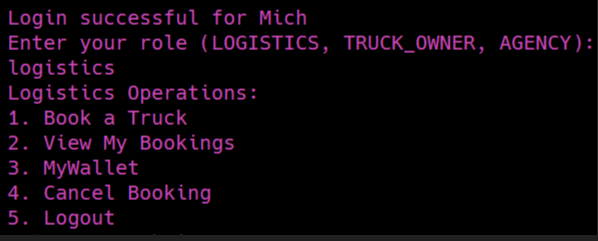
These truck details are written in the text file.



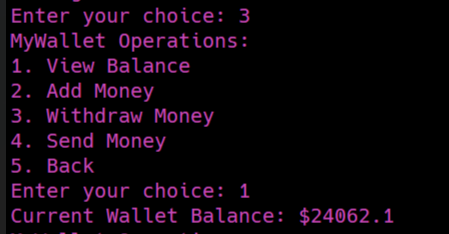
1. Viewing trucks and its details by the truck owner.



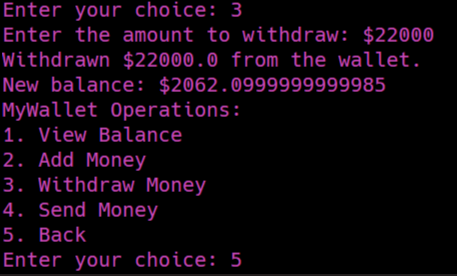
1. Logging in as logistics and logistic operations



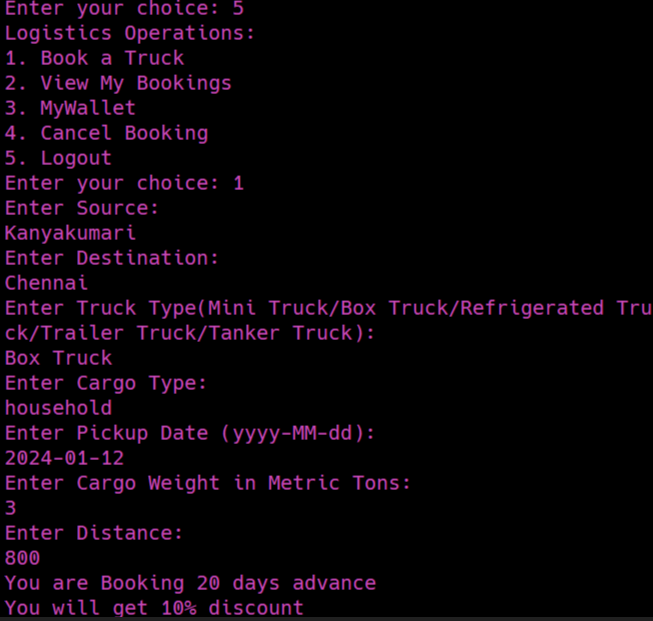
1. Viewing MyWallet, seeing current balance

****

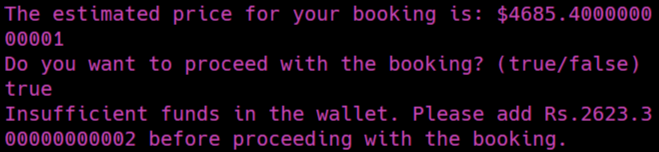
1. Withdrawing money



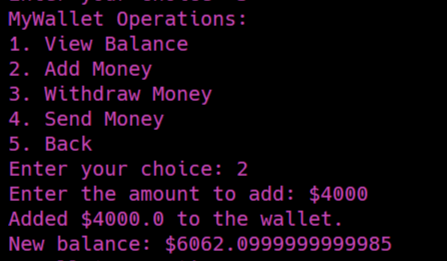
1. Booking process (You get 10 percent discount if you book in advance)



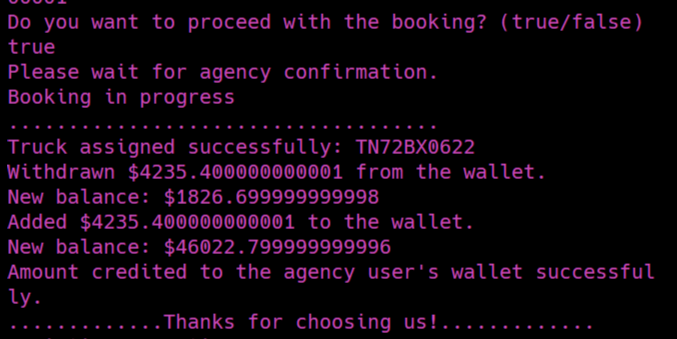
1. Insufficient money in the wallet.



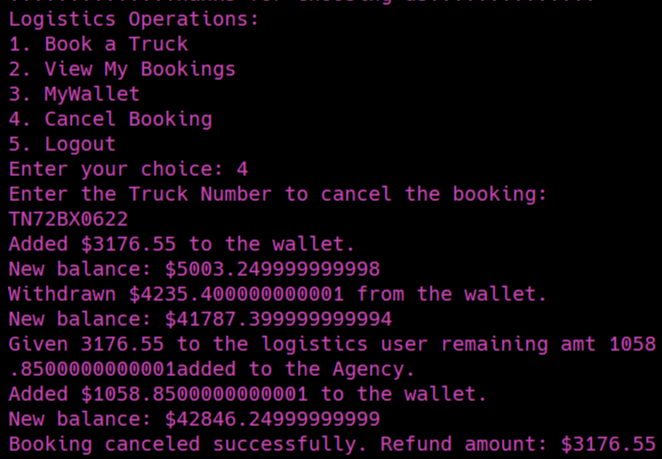
1. Adding money to my wallet



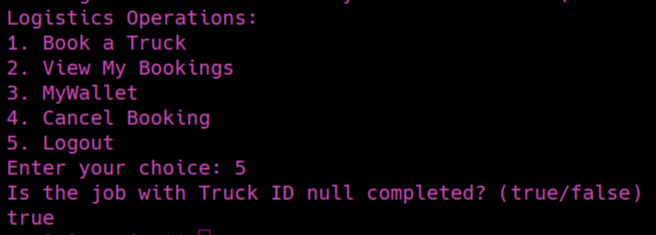
1. Booking process



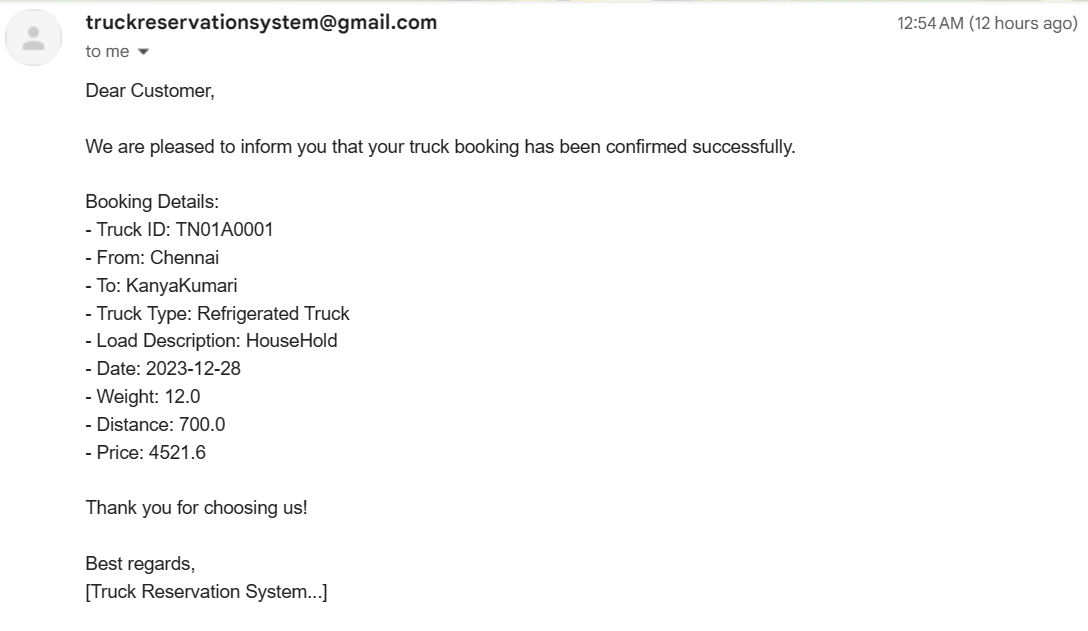
1. Cancel Booking (Money added to the wallet of logistics and agency)



1. Asking whether job competed before quitting



1. Email sent to the customer once he has done the booking



All the details are written in files.

**10. Inference and Future extension**

The development of the Truck Reservation System has proven to be a valuable learning experience. The system effectively meets its objectives, offering a user-friendly platform for logistics and truck owners. Overcoming technical challenges, the project team successfully implemented features such as real-time booking, truck assignment, and wallet management. User feedback played a crucial role in refining the system, ensuring its efficiency and reliability. Throughout the development process, skills in Java programming, object-oriented design, and multithreading were honed, contributing to personal and professional growth. The project management approach demonstrated adaptability, allowing for adjustments to the initial plan.

We plan to extend this project in the future by giving improvements to the user interface for a more intuitive and visually appealing experience. Exploring the possibility of adding advanced booking options, such as recurring bookings, custom cargo handling, or real-time tracking. Investigate integrating the system with external databases or APIs to provide more comprehensive and up-to-date information. We also consider in developing a mobile application to extend the reach of the Truck Reservation System and provide on-the-go access. Also the inclusion of data analytics features to provide insights into booking patterns, popular routes, and other relevant metrics.

We consider making the system adaptable to different languages and regions, enhancing its accessibility on a global scale.

**11. Learning Outcomes**

1. This project helped us strengthen our knowledge in Object Oriented Programming using Java language.
2. This project improved our logical thinking, problem solving and knowledge in algorithmic construct immensely.
3. It helped in effective teamwork.
4. It helped us in designing the idea by doing class diagram which is the basis of our project.
5. Applied and reinforced knowledge of object-oriented design principles such as encapsulation, inheritance, polymorphism, and abstraction to create modular and maintainable code.
6. Developed project management skills by breaking down the system into manageable modules, collaborating with team members, and ensuring seamless integration of components.
7. Demonstrated proficiency in multi-threading concepts to enhance user experience during time-consuming operations, showcasing responsiveness in the application.
8. Implemented robust exception handling mechanisms to identify and address potential issues, contributing to the reliability of the system.
9. Acquired knowledge of user interface design by creating an interactive and user-friendly system for booking trucks.
10. Gained insights into the logistics and truck reservation domain, understanding the complexities and requirements of a real-world application.

**12. References**

[GeeksforGeeks | A computer science portal for geeks](https://www.geeksforgeeks.org/)

[Online Truck Booking India, Transporters, Logistics services (truckguru.co.in)](https://truckguru.co.in/)

[Online Load, Truck Booking, Transporters, Logistics - Trukky](https://www.trukky.com/)

Logic First Tamil - YouTube channel