Reduce the Amount of Push Notifications Require for E- Commerce Apps

1.Ganugapenta Muralidhar Reddy, 2.Thota Karthik, 3.Cherukuru Pavan, 4. V Gowtham Reddy, 5. Kalla Nitish

Ms Smitha SP, Asst.Professor, Presidency University, Bengaluru - 566064

Abstract:
Effective communication is essential in the electronic commerce sector for achieving customer satisfaction and overall operational success. Multiple push notifications often bombard the users and may lead to app fatigue and subsequent uninstalls. Difficulties in managing sellers' inventory and admins having to resort to various communication channels challenge these incumbents. This project, therefore, aims to build an advanced notification system that incorporates user preferences and relevance. This proposed solution will establish a single platform for notifications for better organization and delivery timing for maximum engagement. This will aid the user in receiving crucial notifications, the sellers will work more efficiently, while the admins regain their control. This increases user retention, satisfaction, and platform success.

1. Introduction

With the rapid development of e-commerce, push notifications have become one of the important means of communication between the platform and the users. The amount of notifications is increasing rapidly but poses many difficulties and often results in user fatigue and abandonment of applications.

The project implements a smart notification management system that creates a balance between two things-the engagement and frequency of notifications. As a result, with the help of very effective filtering algorithms, users are provided the most appropriate and timely updates about their orders, wishlist items, and important promotions. The targeted approach considerably reduces the fatigue of notifications and establishes effective communication between the platform, users, sellers, and administrators.

The system architecture basically divides into three modules: User, Seller, and Admin. The User Module focuses on providing personalized notifications for order updates and stock alerts, while the Seller Module facilitates inventory management and order processing communications. The Admin Module helps find a holistic view to oversee and control the notification structure of the platform.

The project will be implemented using the latest Android development technologies, including Android Studio, Kotlin, MySQL with PHP-to-MySQL. The system is well-equipped with a hearty testing framework that checks for reliability and performance under various user situations. We finally provide a solution that specifies a few steps for ameliorating the e-commerce experience with the utmost functionality.

Through reduction of redundant notifications along with the intelligent delivery of notifications that addresses one of the serious pain points for the e-commerce platforms today, this project keeps user retention high while allowing businesses to maintain meaningful connections with customers without overwhelming notifications.

2. LITERATURE REVIEW

- **Current Research & Methods**

A wide range of research studies detail the current, accepted norms and theories regarding notification management in the e-commerce sector. For instance, P. Yogananth et al. (2017) investigated the baseline architecture of e-commerce platforms in India and noted how traditional notification systems relied primarily on basic push applications. The study showed that while e-commerce platforms boasted brisk growth, notification systems seemed pathetically rudimentary.

Rahman et al. (2021) introduced further advanced techniques with their study on client-vendor platforms. They presented new approaches for constructing modular notification systems customizable for different organizational needs, whether governmental or commercial. These works set the stage for modern notification management in institutional settings.

The current scene harbors many different methods for notification management, each with its own unique features. The standard push notification system, employed on e-commerce platforms, automatically disseminates alerts in real time for all activities without filtering. This method offers ease of tracking and has been crippled by creating notification fatigue and leading users to disengagement.

A paper authored by Jemin Lee et al. in 2019 has introduced PASS: a system that mediates notifications between interconnected devices. The study demonstrates a sophisticated technique for deferring and optimizing notifications while leveraging data from over 15,000 notifications to create an intelligent delivery system. The positive outcomes of this research show promise in potential energy savings while upholding user engagement.

Recent research has focused on integrated notification systems combining various communication channels. The currently existing methods keep notifications in separate channels based on update types: example, order confirmation, shipping updates, promotional offers, inventory alerts, etc. This segregation provides organization but creates redundancy and thus overwhelms users.

The research has been able to demonstrate that although current notification systems have improved over the years, they still have yet to provide any satisfactory means of balancing the focus on either communication needs or user experience. Current systems are yet functional and, at best, simple; they fail to enthuse users enough to prevent notification fatigue. Understanding the limitations of these current notification systems has led to the development of more advanced systems capable of utilizing user behavioral analytics and predictive algorithms to improve notification delivery.

Various limitations restricting the efficiency of the methods are pointed out by each researched article: lack of personalized notifications; disregarding user preference; lack of coordination among various notification types, which have led to the use of more intelligent systems aided by user behavior analysis and predictive algorithms for optimizing notifications.

3. METHODOLOGY

3.1 Current State System Analysis

The present e-commerce notification ecosystem works according to a simple push-based architecture. Its sophistication in the delivery of messages and user preferences is rather low. In fact, there are very few platforms that operate with a generic notification system sending common alerts for all activities irrespective of relevance or time. Consequently, therefore, the information is sent to users in an excessive volume where many may not have any relevance for their immediate knowledge and interest. This is putting immense pressure on the system to differentiate between critical data and general information while treating any given notification with the same gravity, so that it is not able to prioritize them accordingly.

3.2 System Limitations

There are well-identified, critical weaknesses inherent in the current notification system that undermine user experience and overall platform efficiency.

Firstly, the system sends too many notifications without any filtering mechanisms, resulting in user fatigue and disengagement. Secondly, sellers face serious challenges aimed at timely communicating information regarding inventory to the end-users, which often leads to delays or duplicate notifications. Most importantly, the admin side, being critical in overlooking various notification dispatching processes throughout the platform, lacks snug tools to control and manage these processes. Finally, the fact that real human preference disclosure is inhibited means that users receive standardized push notifications rather than focused and focused alerts conditioned by their preferences and actions.

3.3 Proposed System Details

To counter these challenges, the proposed solution aims at introducing a competent notification management system, anchored on three prime modules:

The User module provides filtering intelligence as it pushes notifications according to the user's individual preferences and behavior patterns. Real-time updates of orders and stock-out alerts should be given higher priority, ensuring that critical information can reach users on time. The Seller module serves as the communication effective in the dispatching of inventory management information as it allows for inventory updates and order processing. The Admin module displays an overview of all operations, empowering platform administrators with all the appropriate knowledge and power to keep their notifications rolling and at optimized times amongst their categories.

3.4 System Advantages

Consequently, the proposed system presents substantial improvements over existing solutions:
Personal service notifications significantly enhance user engagement and satisfaction. Consolidated updates avoid redundancy and minimize notification fatigue. Real-time inventory management improves operational efficiency for sellers. Broad controls for admins enhance platform management and optimization. The system's modular architecture offers easy scale and enhancement in the future.

3.5 Technical Implementation

This approach makes use of modern technological framework to achieve adequate performance.
The Android Studio was used as the primary development environment with Kotlin programming for enhanced programming support. The backend runs MySQL with PHP integration for efficient data management and processing. The system uses complex

algorithms for notification filtering and delivering optimized messages. Regular test cases are employed to check for system reliability and performance for all identified use cases.

It provides a comprehensive methodology to tackle the aforementioned challenges in e-commerce notification management with a solid foundation for future enhancement and scalability.

4. OBJECTIVES

4.1 Main Objective

The primary aim of this project is to deliver an e-commerce notification delivery system that dramatically reduces notification fatigue while establishing substantial communication between the user, seller, and administrator. This system aims to improve user engagement through smart notification management without interfering with essential updates and platform functionality.

4.2 User Experience Improvement

The project builds a personalized notification framework that respects the user preferences and behavior. The system will therefore filter notifications so the user will receive relevant and timely updates on his/her order, items in their wishlist, and important promotional messages. This aims to further improve user satisfaction and platform retention.

4.3 Operational Efficiency

This project intends to create a means of operational communication for sellers and admins through smart notification management so as to ease communications; for instance, timely updates on inventory and order processing and platform management notices. It aims to enhance business efficiency and lessen unnecessary communication overhead.

4.4 Technical Implementation

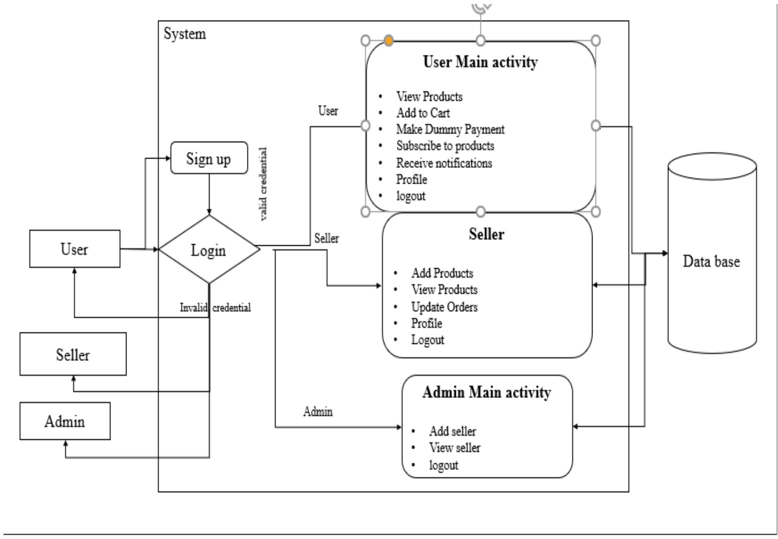
This project aims for a solid technical basis using modern Android development tools and technologies. These include designing an architecture scalable so that complex notification filtering and delivery mechanisms can be tackled, all while ensuring system performance and reliability. The aim is to implement a maintainable and extendable framework that shall accommodate the platform's requirements.

4.5 Performance Metrics

The system would aim at achieving measurable improvements in key performance indicators: reduced number of notifications, increased user engagement, and improved operational efficiency. These goals will be supported by thorough testing protocols so that the system's reliability and effectiveness for various application scenarios are eminent.

In this regard, by way of specific objectives, the project addresses major issues in e-commerce notification management and thus establishes a platform for further platform enhancements for greater user experience.

5. ARCHITECTURE



(fig:1)

5.1 User Mode (Buyer Mode):

- **Purpose**: For customers who browse and purchase products.
- **Role**: Consumers of the platform's products.

5.2 Seller Mode:

- **Purpose**: For merchants who list and manage their products.
- **Role**: Providers of products for sale on the platform.

5.3 Admin Mode:

- **Purpose**: For platform administrators who manage the overall system.
- **Role**: Oversees users, sellers, and platform operations.

6. SYSTEM DESIGN

6.1 Introduction of Input design

Input Design refers to the process of defining how data will be entered into the system, ensuring it is captured in a way that is both efficient and secure. This phase connects the user with the system and focuses on simplifying data entry while preventing errors. The goal is to make data collection as straightforward and error-free as possible while maintaining privacy and security.

6.2 UML DIAGRAM

UML (Unified Modeling Language) is a standardized visual language used in object-oriented software design. It helps describe, visualize, and document the structure and behavior of a system. UML is widely used for designing and modeling complex software systems.

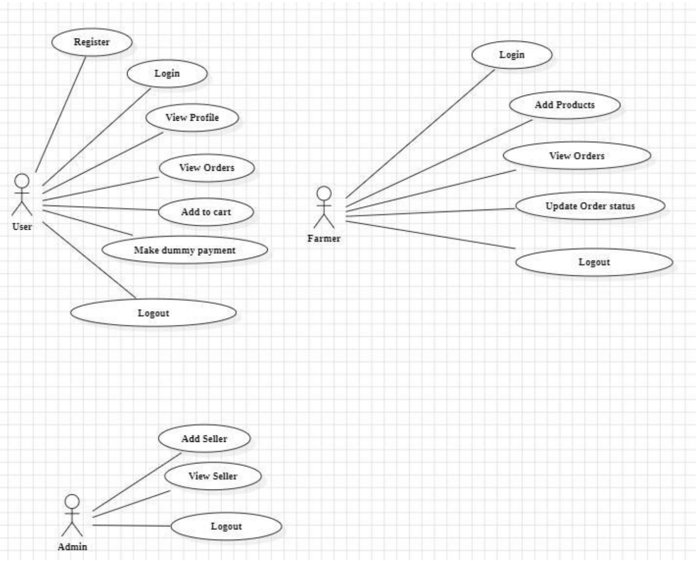
Goals of UML Design:

- Provide a clear and expressive visual representation of software designs.
- Enable the extension and specialization of models as needed.
- Remain independent of specific programming languages or processes.

- Facilitate important concepts like collaboration, frameworks, and components in the software development process.

6.3 USE CASE DIAGRAM

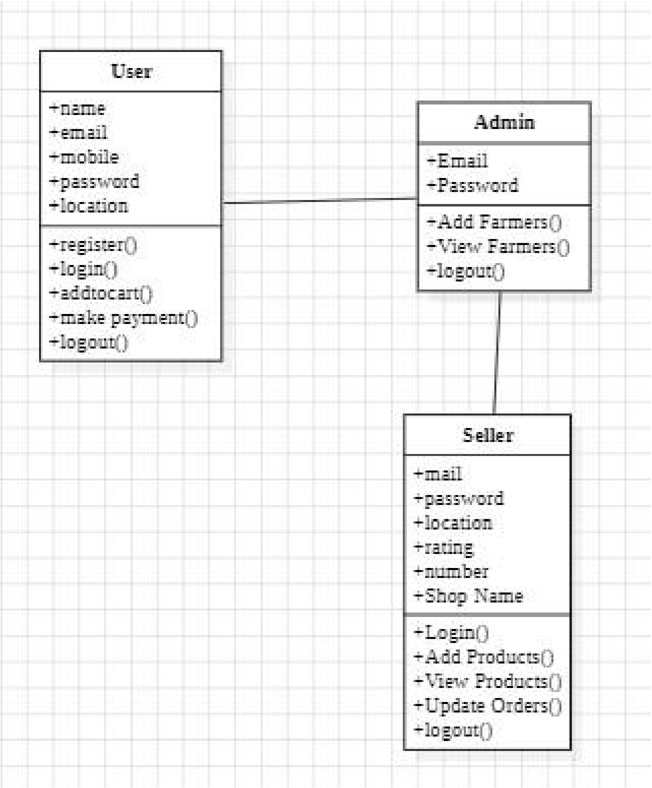
It highlights the system's actors (users or other systems) and the interactions they have with the system's functionalities (use cases). The diagram's primary purpose is to clarify which tasks the system performs and who interacts with the system.



(fig:2)

6.3.1 CLASS DIAGRAM

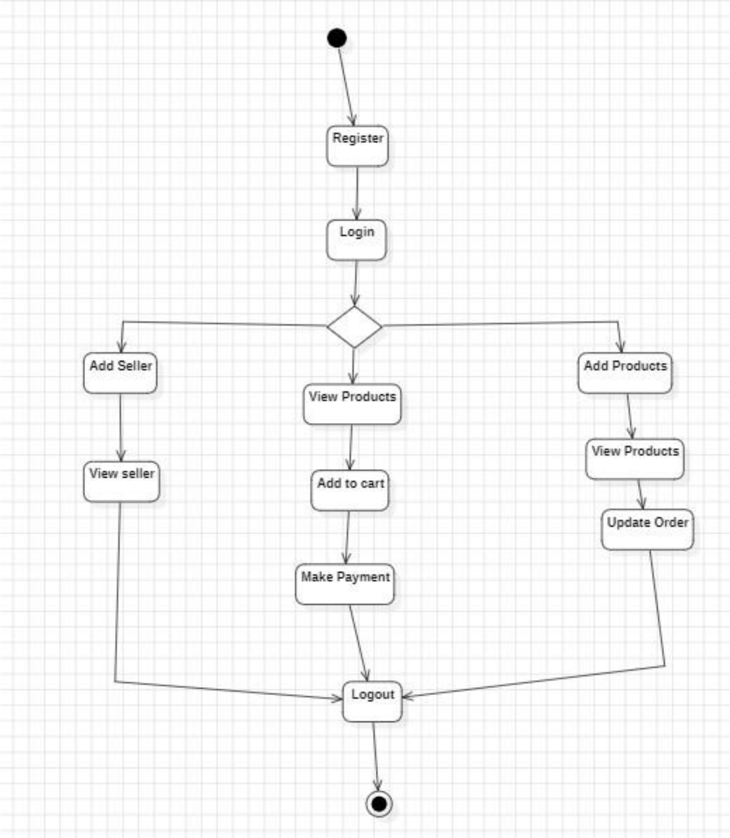
The **Class Diagram** illustrates the static architecture of a system, depicting the system's classes, their properties, functions, and the relationships among them.. It is essential for illustrating how data and behaviors are structured in the system.



(fig:3)

6.3.2 SEQUENCE DIAGRAM

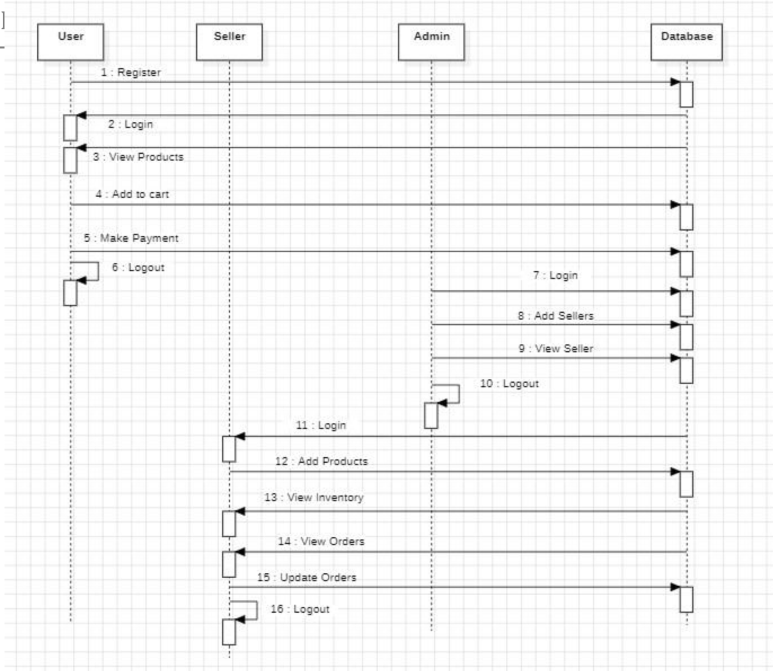
A **Sequence Diagram** is a type of interaction diagram that illustrates how objects interact with each other over time. It shows the order in which events occur, helping developers understand the flow of messages between components.



(Fig: 4)

6.3.3 COLLABORATION DIAGRAM

The **Collaboration Diagram** is similar to a sequence diagram but emphasizes the organization of objects within the system. It indicates the sequence of method calls among objects and their relationships.



6.3.4 ACTIVITY DIAGRAM

(fig: 5)

An **Activity Diagram** visualizes workflows and step-by-step activities within a system. It is used to describe both business processes and operational workflows, highlighting decisions, iterations, and concurrency.

6.3.5 COMPONENT DIAGRAM

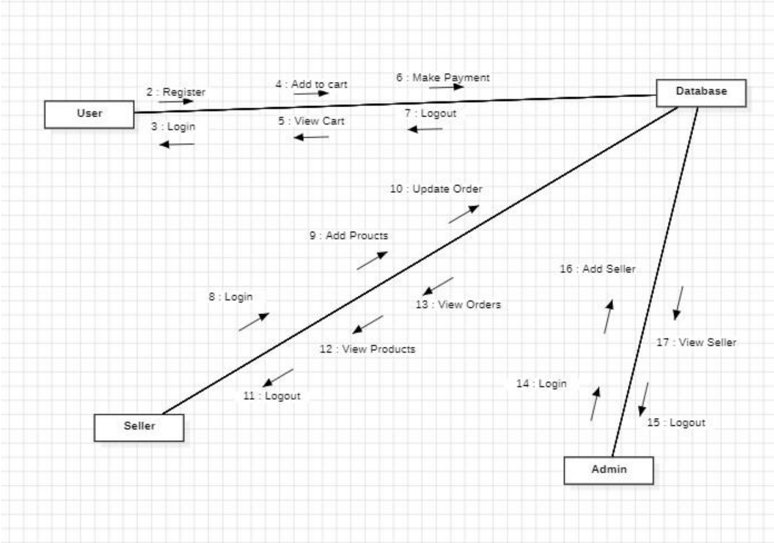
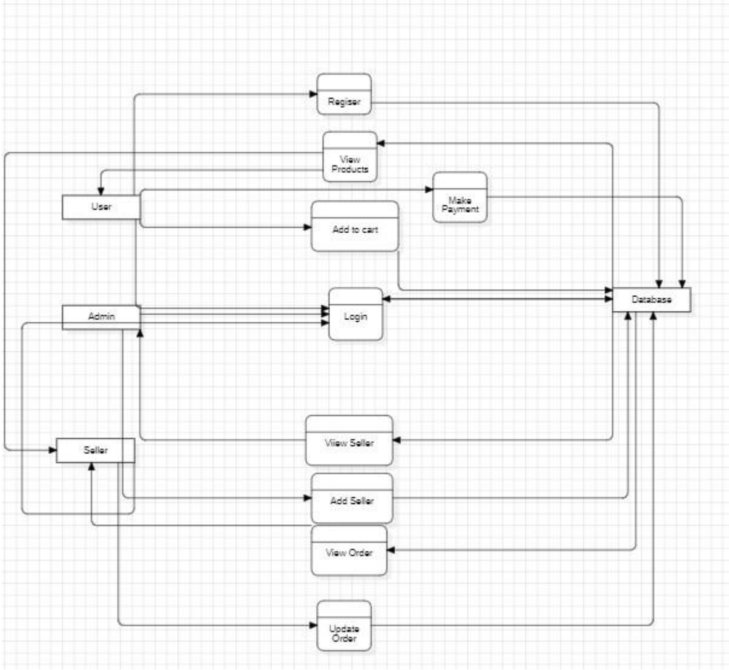


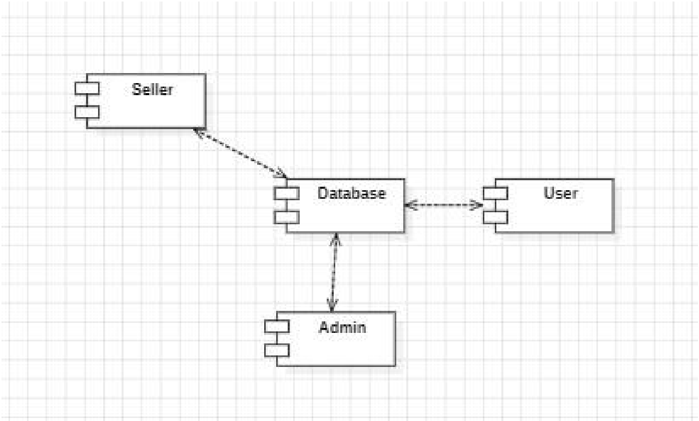
fig:6)



(fig:7)

6.3.6 DEPLOYMENT DIAGRAM

6.3.7 ER DIAGRAM



(fig:8)

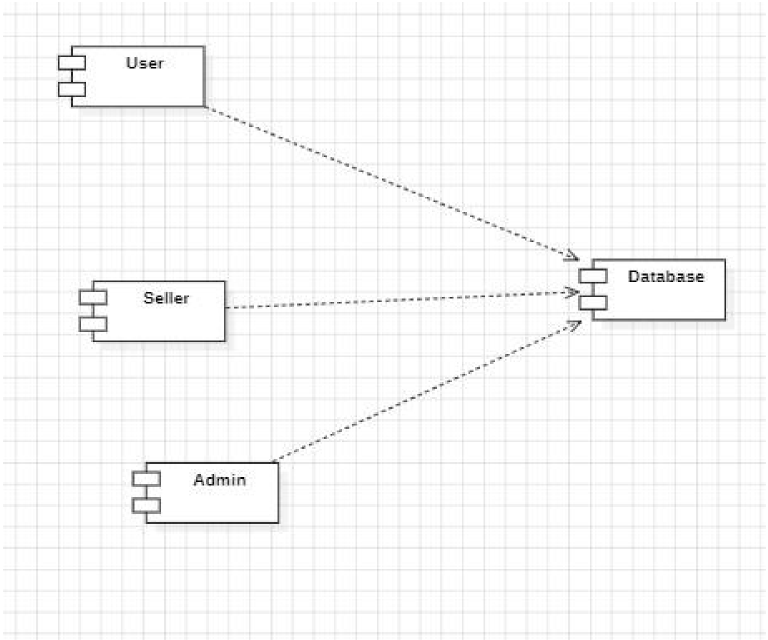
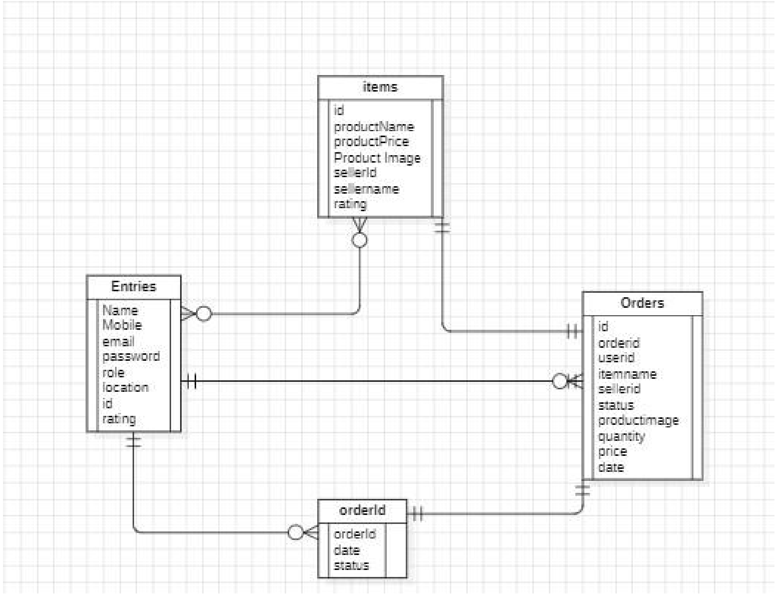


fig:9)

6.4 DATA FLOW DIAGRAM



(fig:10)

7. SYSTEM STUDY AND TESTING

7.1 Feasibility study

A feasibility study is conducted to analyze whether the proposed system can be implemented effectively and economically. The goal is to evaluate if the system is financially viable and technically feasible, and if it aligns with the company's objectives. During this phase, a detailed business proposal is outlined, including project costs and a general plan.

Three primary factors considered in the feasibility analysis include:

Economic Feasibility:
This aspect assesses the financial impact of the proposed system. Given that many technologies used in the project are freely available, the costs are kept minimal, with only a few customized products requiring purchase.

Technical Feasibility:
This phase evaluates the system's technical requirements. The system should not place excessive demands on existing resources. The design ensures that the system's needs are minimal, requiring no significant changes to the current infrastructure.

Social Feasibility:
This examines how well the system will be accepted by users and their ability to effectively interact with it. A critical part of this phase is providing proper user training, ensuring they feel confident and comfortable using the system. The success of the system relies heavily on the user's acceptance and their willingness to adapt to the new technology.

7.2 Types of Tests & Test Cases

1. **Unit Testing**
   Unit testing validates that individual components or functions within the system are working correctly. It involves checking if each program unit operates as expected, based on the input-output specification. These tests are performed after the development of each unit before integration.
2. **Integration Testing**
   Integration testing verifies that various software components function together as a unified system.. The goal is to identify issues that may arise when combining different modules that worked well individually during unit testing.
3. **Functional Testing**
   Functional testing ensures that the system meets the requirements defined by the business and technical specifications. It involves testing the system's functions to confirm they work as expected, including checking the handling of both valid and invalid inputs, as well as expected outputs.
4. **System Testing**
   The goal of testing is to detect any errors and verify that the system operates as intended. It focuses on validating the integrated components and ensuring that the system meets all predefined specifications and works reliably across different configurations.

5. **White Box Testing**

   White box testing is carried out with an understanding of the system's internal design and functionality. It checks the flow of information and logic within the system, ensuring the code executes as expected and that no internal defects are present.

6. **Black Box Testing**

   Black box testing evaluates the system's functionality without any understanding of its internal structure.. It focuses on the functionality of the system by providing inputs and checking the corresponding outputs, verifying if the system meets user requirements.

7. **Acceptance Testing**

   Acceptance testing is the final phase before the system is transferred to users.. It ensures the system meets the defined functional requirements and is ready for deployment. During this phase, end-users are involved in validating that the system fulfills their needs and expectations.

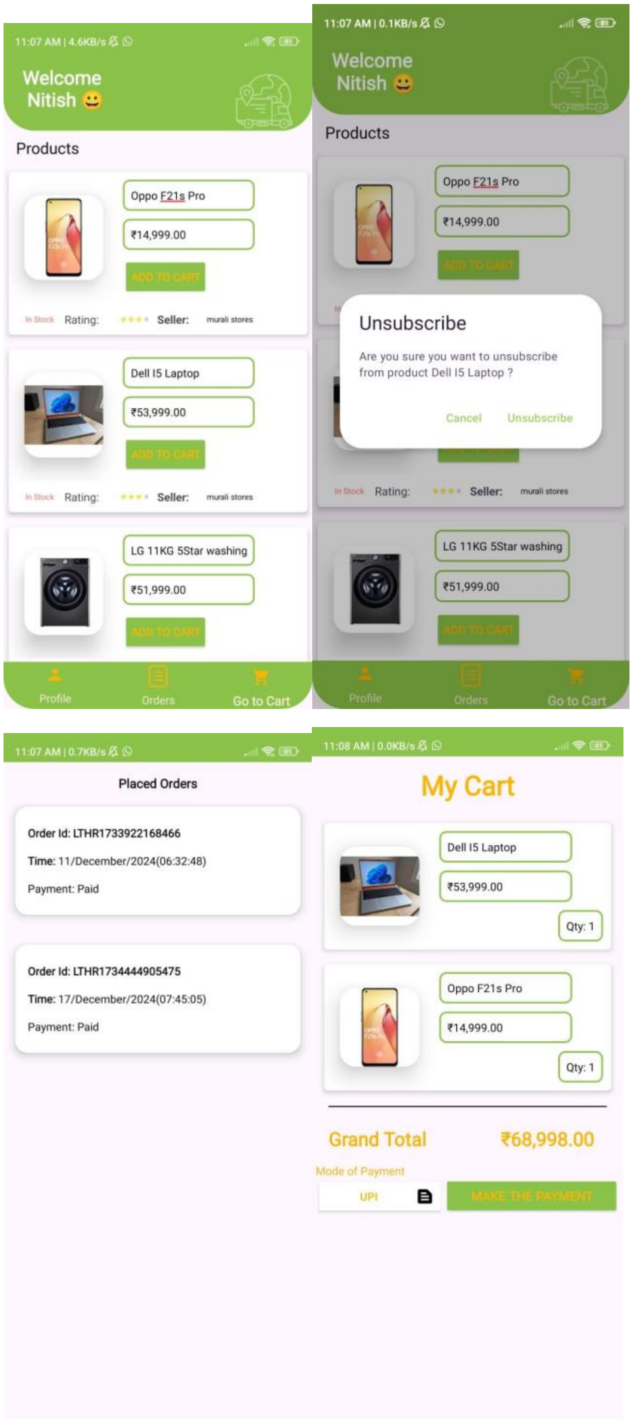
Test Results:

All tests conducted during this phase passed successfully, and no defects were encountered.

8. IMPLEMENTATION

8.1 Modules:

- **User Module:** The User Module is designed to provide a seamless shopping experience. Users can register and log in securely to browse a variety of products. They can place orders, make payments through a secure payment gateway, and track their current orders. Additionally, users can view their complete order history and manage subscriptions for specific products. Notifications are personalized, ensuring users are alerted when subscribed products are back in stock or their orders are updated. This module focuses on delivering a user-friendly, efficient, and engaging shopping experience.



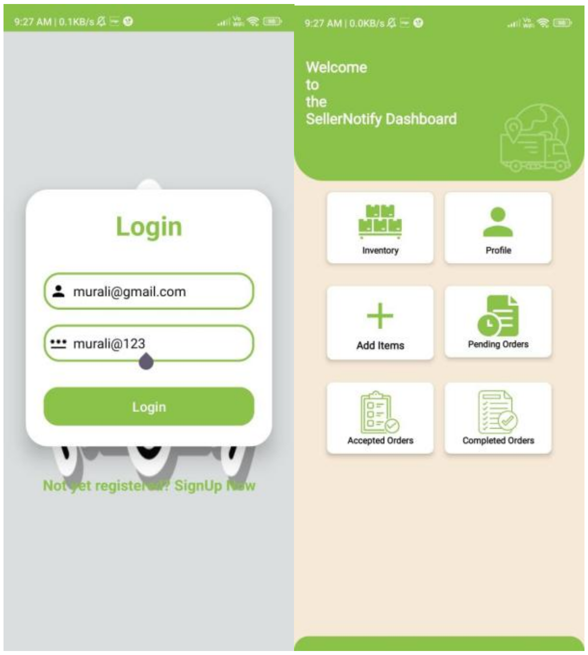
Are you sure you want to unsubscribe from product Dell I5 Laptop ?

Order Id: LTHR1733922168466
Time: 11/December/2024(06:32:48)
Payment: Paid

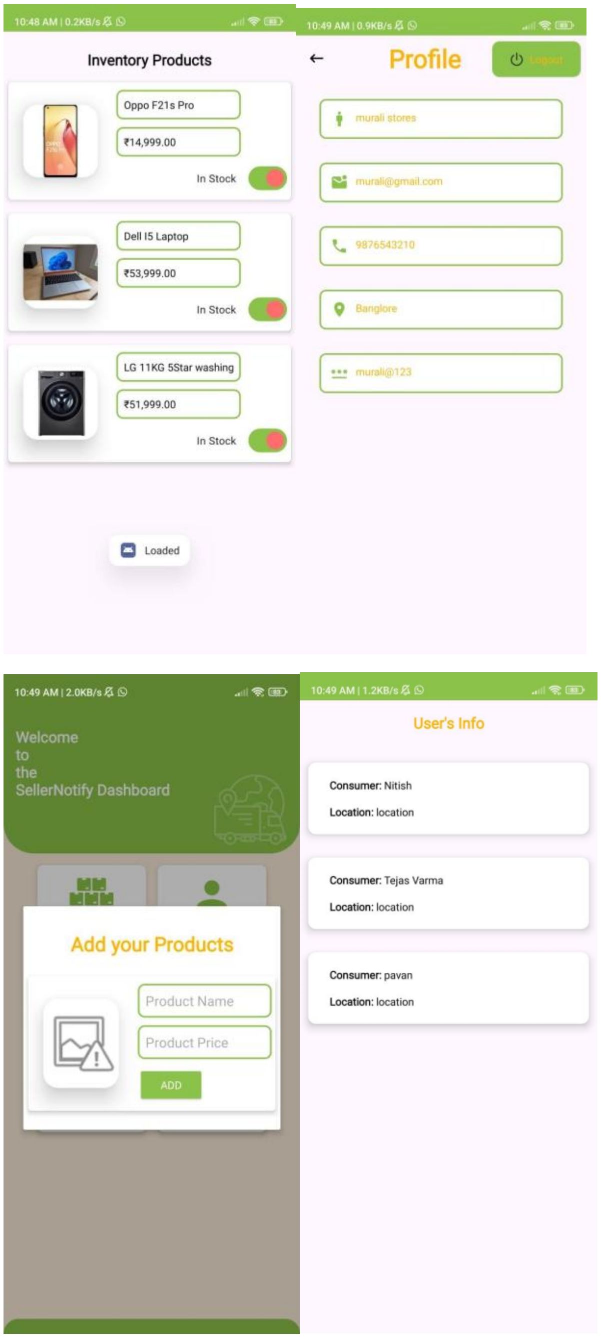
Order Id: LTHR1734444905475
Time: 17/December/2024(07:45:05)
Payment: Paid

- **Seller Module**: The Seller Module empowers sellers to manage their storefront effectively. Sellers can log in to add products, update descriptions, and manage inventory in real-time.

They are notified of new orders instantly and can accept or reject them based on availability. Sellers also have the capability to update order statuses, keeping users informed. The module ensures that sellers have all the tools needed to monitor stock levels and streamline their operations, reducing delays and enhancing customer satisfaction.



murali@gmail.com
murali@123



Inventory Products

Oppo F21s Pro
₹14,999.00
In Stock

Dell I5 Laptop
₹53,999.00
In Stock

LG 11KG 5Star washing
₹51,999.00
In Stock

Loaded

Profile

murali stores
murali@gmail.com
9876543210
Banglore
murali@123

Welcome to the SellerNotify Dashboard

Add your Products

Product Name
Product Price
ADD

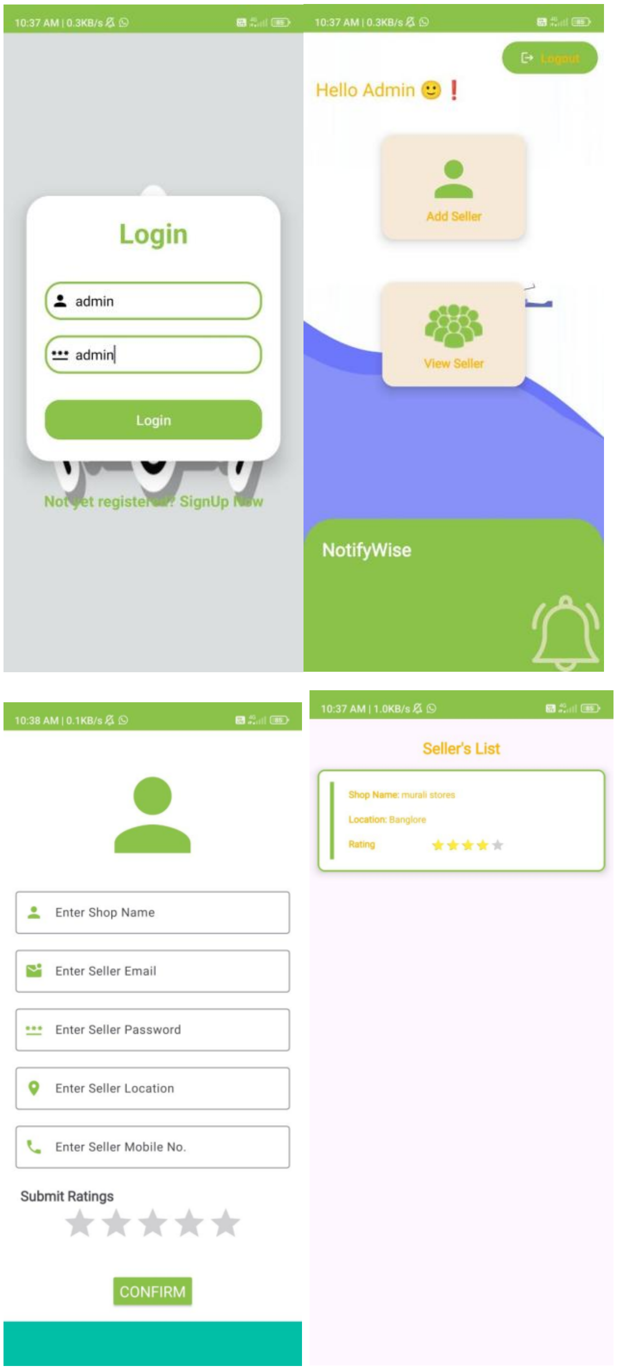
User's Info

Consumer: Nitish
Location: location

Consumer: Tejas Varma
Location: location

Consumer: pavan
Location: location

- **Admin Module:** The Admin Module oversees the platform's overall functionality and integrity. Admins can add new sellers, monitor accounts, and ensure compliance with platform standards. They maintain quality control by reviewing product listings and assigning ratings to sellers based on their performance.



Login
admin
admin
Login
Not yet registered? SignUp Now

Hello Admin 🙂❗
Add Seller
View Seller
NotifyWise

Enter Shop Name
Enter Seller Email
Enter Seller Password
Enter Seller Location
Enter Seller Mobile No.
Submit Ratings
☆☆☆☆☆
CONFIRM

Seller's List

Shop Name: murali stores
Location: Banglore
Rating ★★★★☆

9. CONCLUSION

This project aims to revolutionize e-commerce notifications by introducing a smarter, more user-centric system. By reducing redundant alerts and enhancing communication efficiency, the solution significantly improves user retention and satisfaction while streamlining operations for sellers and admins. This approach ensures sustainable platform growth and user trust.

10. ACKNOWLEDGMENT

The authors would like to acknowledge the support of Presidency University for providing resources and facilitating this research project. We are also grateful to the university librarians, professors, and research assistants for their assistance.

11. REFERENCES

1. P Yogananth; K. Priyadharshini; S. Mahalakshmi; R. Udhayasanthiya; A. Shilpasree; Customer demanding products in online shopping — A novel framework; 16-18 February 2017.

2. Hafizur Rahman; Eshan Barua; Samanta Afrin; Ashikur Rahman; Mohammad Monirujjaman Khan; Bid & Buy: An Effective Online Based Platform for Client and Vendor; 08-10 April 2021.

3. Vincentius Riandaru Prasetyo; Exploring the Cheapest Products Across Three E-Commerce Platforms Using the K-Means Algorithm; 30-31 August 2018.

4. F. Wegscheider || Minimizing unnecessary notification traffic in the IMS presence system || 16-18 January 2006