

SwiftPay:UPI

Project submitted to the
SRM University – AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering
School of Engineering and Sciences

Submitted by

SHAIK MOHAMMAD AAKHIL	AP23110011427
MANNALA KARTHIK	AP23110011441
POLAMADA ARUN SAI KUMAR	AP23110011458
DONGALA BALAIAH	AP23110011409



Under the Guidance of
Kavitha Rani Karnena

SRM University–AP
Neerukonda, Mangalagiri, Guntur
Andhra Pradesh – 522 240
[November, 2024]

Certificate:

Date: 20-Nov-24

This is to certify that the work present in this Project entitled “**SwiftPay**” has been carried out by **Shaik Mohammad Aakhil, Dongala Balaiah, Polamada Arun Sai Kumar and, Mannala Karthik** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Co-supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Table of Contents:

Acknowledgments

Abstract

Introduction

Methodology

Key Features

Conclusion

Acknowledgements

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide, Faculty Name, Department of Computer Science & Engineering, SRM University, Andhra Pradesh, for her kind help and for giving me the necessary guidance and valuable suggestions in completing this project work.

Abstract:

SwitPay is a comprehensive UPI (Unified Payments Interface) application designed to facilitate seamless and secure digital financial transactions. This project leverages Object-Oriented Programming (OOP) principles in C++ to build a robust, user-friendly payment system that meets the contemporary demands for efficient cashless transactions.

Objective: The primary goal of SwitPay is to create an application that allows users to send and receive money, make bill payments, and manage transaction history with enhanced security and ease. The project aims to integrate key OOP concepts such as Classes and Objects, Inheritance, Polymorphism, and Exception Handling to ensure modularity, code reusability, and reliability.

Introduction:

The SwiftPayApp project is designed to simulate a user-friendly and efficient UPI (Unified Payments Interface) application using C++. The main goal of the project is to provide users with the ability to conduct secure digital transactions, check balances, and manage account details seamlessly. The development of SwiftPayApp utilizes core object-oriented programming principles such as classes, inheritance, and polymorphism to create a modular and maintainable code structure.

The UPI system has become a vital part of the financial ecosystem, allowing users to perform transactions directly from their bank accounts without needing to input card details. This project aims to reflect real-world payment applications by incorporating key features such as user authentication, premium services, and robust exception handling. Through the SwiftPayApp project, we not only demonstrate essential programming concepts but also simulate a practical, real-world application that emphasizes user experience and security.

By leveraging inheritance and polymorphism, we create a scalable framework that allows for feature extensions such as adding different types of user accounts or new transaction functionalities. The project serves as an educational tool, merging theoretical learning with practical implementation to deepen understanding and showcase the power of C++ in building complex applications.

Methodology:

Here are three key methodologies for the SwiftPayApp project:

- 1. Object-Oriented Design and Development:**

The project leverages core principles of object-oriented programming, including encapsulation, inheritance, and polymorphism. This ensures modular code with reusable components such as User, RegularUser, and PremiumUser classes. This structure promotes scalability and maintainability, making it easier to extend features in the future.

- 2. Data Management and Security:**

User data and transaction histories are managed using file I/O operations, ensuring persistence across sessions. Security measures include password protection and UPI PIN verification for all critical transactions, emphasizing user data privacy and secure financial operations.

3. User-Centric Interface and Features:

The app is designed with user convenience in mind, offering functionalities such as bank account linking, UPI PIN setup, balance checking, money transfers, and cashback rewards. Frequent recipients and transaction history features enhance user experience, simplifying repeated tasks and fostering user loyalty.

Key Features:

Here are the key features of the SwiftPayApp project:

1. User Account Management:

Supports account creation for both regular and premium users.

Enables user authentication through password protection.

Allows users to link and manage multiple bank accounts.

2. Secure Transactions:

Facilitates secure money transfers using UPI with mandatory PIN verification.

Ensures all transactions are logged and can be reviewed through a transaction history feature.

3. Balance Management:

Provides an option for users to check their current bank balance.

Includes mechanisms for updating balances post-transactions with real-time feedback.

4. Transaction History and Frequent Recipients:

Keeps a record of past transactions that users can view for tracking spending and managing finances.

Supports a feature to save frequent recipients for quicker and more efficient future transfers.

5. Error Handling and User Feedback:

Includes comprehensive exception handling to manage issues such as insufficient balance or invalid input.

Provides clear and user-friendly feedback for errors, improving the overall user experience.

These features collectively make SwiftPayApp a robust, user-centric, and secure UPI-based payment solution.

SwiftPay

Bank account
No. *****8910

Select your bank

Search for a bank

SBI Bank

Canara Bank

Next

SwiftPay - Create account

Your name

Email address

Phone number

Password

By continuing, you agree to the Terms of Use. Read our Privacy Policy.

Sign Up



Transaction history

Sent ₹500 to Arun

Mar 9, 2028



Sent ₹200 to Balaiah

Feb 14, 2028



Received ₹100 from Karthik

Jan 15, 2028



Sent ₹50 to Arun

Dec 31, 2027



Received ₹70 from Karthik

Nov 27, 2027



Enter UPI PIN

Enter your 6 digit UPI PIN to view your balance

0

0

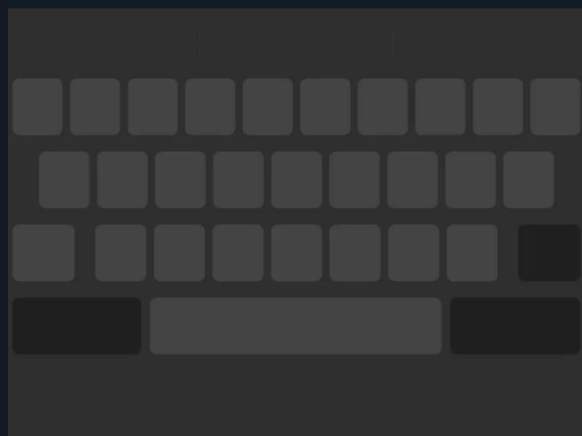
0

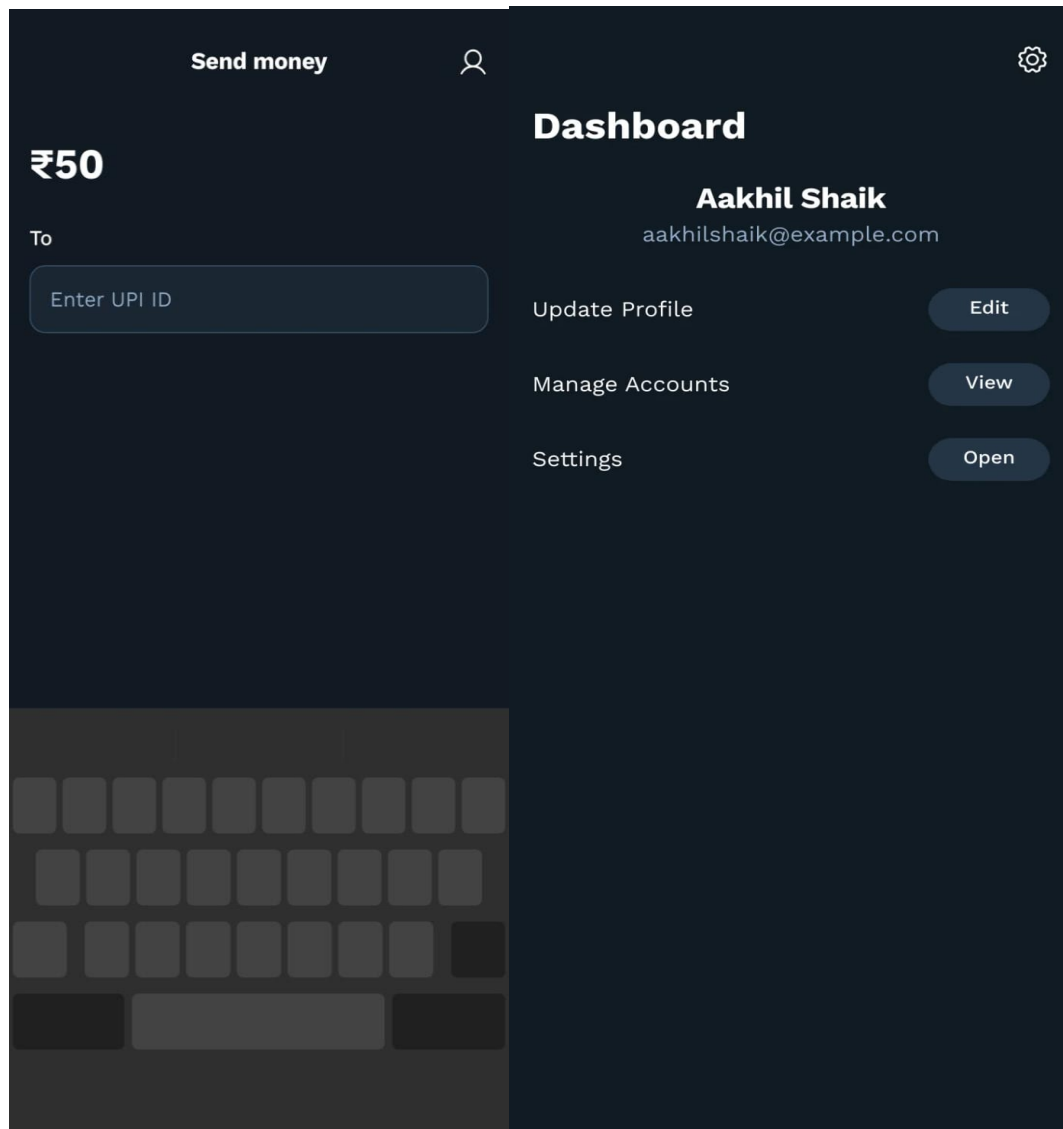
0

0

1

[Forgot your PIN?](#)





Conclusion:

The SwiftPayApp project serves as a comprehensive demonstration of key software engineering principles, including inheritance, polymorphism, and exception handling. By developing this UPI-based payment system, we successfully integrated various C++ concepts such as object-oriented programming, secure transaction management, and user authentication. The implementation of inheritance and polymorphism not only allowed for cleaner code but also promoted reusability and scalability, especially with the inclusion of premium user features.

This project highlights the importance of building user-friendly and secure financial applications. Through robust error handling and clear user feedback, we ensured a seamless and reliable experience. Additionally, the architecture is designed for potential future enhancements, enabling easy integration of new features and functionalities.

Overall, the SwiftPayApp project not only meets its objectives of facilitating secure transactions and efficient balance management but also lays the groundwork for future developments and scalability, making it a practical and effective solution in the digital payment landscape.