✅ SWIFTPAY FINAL PROJECT REPORT

2. Table of Contents

1. Introduction

2. Project Objectives

3. Problem Statement

4. Proposed Solution

5. Functional Requirements

6. Non-Functional Requirements

7. System Architecture

8. UI Prototype (Figma Link)

9. UML Diagrams

10. Version Control & Collaboration

11. Risk Management

12. Monitoring and Progress Tracking

13. Conclusion

14. References

3. Introduction

SwiftPay is a secure, user-friendly FinTech mobile application aimed at simplifying digital transactions for users in Sierra Leone and beyond. The app allows users to send, receive, and manage funds, offering features such as financial education, notifications, multi-currency support, and more.

4. Project Objectives

• To create a reliable FinTech solution for mobile transactions

• To educate users on smart financial decisions

• To offer services such as balance checks, fund transfers, and financial tips

• To ensure system security and user privacy

5. Problem Statement

Many users face delays, security issues, and lack of financial awareness when using traditional mobile money platforms. SwiftPay aims to solve these problems by offering a seamless and educational financial tool.

6. Proposed Solution

SwiftPay will be a mobile-based platform that ensures:

• Fast and secure transactions

• Clean user interface

• Notifications and tips

• User authentication

• Multi-currency support

7. Functional Requirements

• User registration and login

• Money transfer

• Fund withdrawal and deposit

• Transaction history

• Push notifications

• Financial education resources

• Multi-currency support

8. Non-Functional Requirements

• Secure authentication

• High availability and performance

• Responsive mobile design

• Scalability to support growing users

• Offline mode for balance check

9. System Architecture

The system is divided into:

• Frontend (Mobile App - Figma prototype)

• Backend (API and transaction processing)

• Database (User data, transaction records)

• Notification Module (Push updates and messages)

Refer to the architecture diagram for component interaction.

10. UI Prototype

The Figma prototype includes:

• Login Screen

• Registration Screen

• Dashboard

• Send Money

• Withdraw

• Add Funds

• Transaction History

• Notifications

• Financial Tips

Link to Figma Design: [https://www.figma.com/design/D7XU9E2VsMhGOzeUVeqh3I/Untitled?node-id=0-1&t=EyTmM9qg4n1mmEsj-1]

11. UML Diagrams

• Use Case Diagram (with actors: User, Admin)

• Class Diagram

• Activity Diagram

• Sequence Diagram

• All diagrams are saved in the UML\_Diagrams folder of the GitHub repo.

12. Version Control & Collaboration

• Project repository hosted on GitHub: [https://github.com/Gershomisaackingsambo/swiftpay-mobileApp-prototype.git

]

• Branches created: feature/login-authentication, feature/money-transfer, etc.

• Issues and pull requests handled for each feature

• Team collaboration on various aspect of this project

13. Risk Management

Risk Impact Mitigation Strategy

Data loss Medium Regular GitHub backups

Feature overload Medium Focused on core features

Security flaws High Included basic auth & security best practices

14. Monitoring & Progress Tracking

• A Gantt chart was created and updated regularly (see “Project management” folder)

• Milestones were achieved from June 1st – June 11th

• Google Form was created and shared to gather user feedback

• All feedback was considered during prototyping

15. Conclusion

SwiftPay was designed and implemented with real-world usability in mind. The app provides convenience, education, and security in mobile financial transactions. This project reflects strong planning, design thinking, and practical execution in a collaborative environment.

16. References

• Google Forms Feedback

• Figma Prototypes

• GitHub Issues and Pull Requests

• UML Tools (Draw.io)

• Financial APIs (Referenced in architecture)