Social Swap | Your Crypto Companion

# 24-FYP-106



**Session 2020-2024** **BACHELOR OF SCIENCE** **IN** **COMPUTER SCIENCE**

# SUBMITTED BY

Muhammad Sajid Bashir 20-NTU-CS-1168

Hammad Ahmad 20-NTU-CS-1057

# SUPERVISED BY

Dr. Muhammad Asif Habib

Ms. Atiab Ishtiaq

**Department of Computer Science**

**National Textile University, Faisalabad.**

**DEDICATION**

We dedicate the Social Swap project to all individuals who embrace the power of social crypto trading, and who value truth, trust, and integrity in the digital realm. Our dedication extends to the researchers, developers, and professionals who have tirelessly contributed to the field of social crypto trading, as well as the users who rely on this technology to navigate the world of digital assets. May our efforts in the Social Swap project actively contribute to the ongoing battle against misinformation and uphold the principles of transparency, accuracy, and accountability in the realm of cryptocurrency trading.

# CERTIFICATE

**NATIONAL TEXTILE UNIVERSITY, FAISALABAD**

This synopsis written by **Muhammad Sajid Bashir (20-NTU-CS-1168), Muhammad Hammad (20-NTU-CS-1057)** under the direction of their supervisors and approved by all the members of the synopsis committee, has been presented to and accepted by the Director, Graduate Studies, and Research, in the partial fulfillment of the requirement of the degree of Bachelor of Science in Computer Science.

Dr. Muhammad Asif Habib Ms. Atiab Ishtiaq

Supervisor Co-Supervisor

Internal Examiner External Examiner

Coordinator/ HoD Dean/Director,

Graduate Studies & Research

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

# ACKNOWLEDGMENTS

First, In the Name of Allah, the Most Beneficent, the Most Merciful. All the praises and thanks be to Allah. A lot of love for our beloved Holy Prophet MUHAMMAD (S.A.W), his guidance always helps us to get the right path. After that, we also proudly declare that the success parameters of this project are the endless prayers and support of our parents. We extend our sincere gratitude to our project supervisor, **Dr. Muhammad Asif Habib**, and co-supervisor, **Ms. Atiab Ishtiaq**, for their invaluable guidance and support throughout our project. We would also like to acknowledge the faculty members of the Department of Computer Science for their feedback and encouragement. Without the collective effort of all mentioned, this project would not have been possible.

**Muhammad Sajid Bashir**

**20-NTU-CS-1168**

**Hammad Ahmad**

**20-NTU-CS-1057**

# TABLE OF CONTENTS

DEDICATION .................................................................................................................. ii

CERTIFICATE ................................................................................................................. iii ACKNOWLEDGMENTS ............................................................................................... iv TABLE OF CONTENTS ................................................................................................. v LIST OF ABBREVIATIONS .......................................................................................... viii LIST OF TABLES ........................................................................................................... ix

LIST OF FIGURES .......................................................................................................... x ABSTRACT ..................................................................................................................... xi Chapter 1 .......................................................................................................................... 1

Introduction ...................................................................................................................... 1

1.1 Overview …............................................................................................................ 1

1.2 Problem Statement ................................................................................................. 1

1.3 Purpose ................................................................................................................... 1

1.4 Project Goals .......................................................................................................... 2

1.5 Project Objective .................................................................................................... 2

1.6 Project Scope .......................................................................................................... 2

1.7 Project History ........................................................................................................ 3

1.8 Project Significance................................................................................................. 3

1.9 Proposed Solution ................................................................................................... 3

1.10 Project Scheduling ................................................................................................ 4

1.11 Risks & Risk Mitigation ....................................................................................... 5

*1.11.1* *Technology Risks* ........................................................................................ 5

*1.11.2* *Financial Risks* ........................................................................................... 5

Chapter 2 ........................................................................................................................... 6

Theoretical background ..................................................................................................... 6

2.1 Literature review ..................................................................................................... 6

*2.1.1* *Overview of Literature review* .......................................................................... 6

2.3 Systematic literature review (SLR) ......................................................................... 7

2.3 Critical Analysis ...................................................................................................... 9

Chapter 3 ........................................................................................................................... 10

System Requirements......................................................................................................... 10

3.1 Functional Requirements ........................................................................................ 10

*3.1.1* *User Management*............................................................................................. 10

*3.1.2* *Wallet Managemet*............................................................................................. 10

*3.1.3* *Chat Functionality*............................................................................................ 10

*3.1.4* *Crypto Transactions*.......................................................................................... 10

*3.1.5* *Coin Graphs*...................................................................................................... 10

*3.1.6* *Trading Bot*....................................................................................................... 10

*3.1.7 News Updates*................................................................................................... 11

3.2 Non-Functional Requirements ............................................................................... 11

*3.2.1* *Secure Communication* ...................................................................................11

*3.2.2* *Cross Platform Compatibility* ......................................................................... 11

*3.2.3* *Responsive Design* .......................................................................................... 11

*3.2.4* *User Interface* .................................................................................................. 11

3.3 Use Case Diagram .................................................................................................. 12

3.4 Use Case Description ............................................................................................. 12

*3.4.1* *Use Case of User* ............................................................................................. 13

*3.4.2* *Use Case of Wallet* ........................................................................................... 14

*3.4.3* *Use Case of Crypto Trading Model* ................................................................. 15

*3.4.4* *Use Case of Suggestion bot* ............................................................................. 15

*3.4.5* *Use Case of Chat Functionality* ...................................................................... 16

*3.4.6* *Use Case of Latest News Updates* ................................................................... 17

*3*.4.7 *Use Case of Profile Update* ............................................................................. 18

Chapter 4 .......................................................................................................................... 20

Methodology .................................................................................................................... 20

4.1 Project Planning ..................................................................................................... 20

4.2 Methodology for Software Development .............................................................. 20

*4.2.1 Existing Methodologies* .................................................................................. 21

*4.2.2* *Selected Methodology* ..................................................................................... 21

4.3 Reason for Selected Methodology ......................................................................... 22

*4.3.1 Quick Response to Change* .............................................................................. 23

*4.3.2 Regular Demonstration* ................................................................................... 23

*4.3.3 Agile as Mindset* .............................................................................................. 23

*4.3.4 Creating Tangible Values* ................................................................................ 23

Chapter 5 ........................................................................................................................... 24

System Architecture .......................................................................................................... 24

5.1 Architecture ............................................................................................................. 24

*5.1.1 Microservices* .................................................................................................... 24

*5.1.2 Microservices in Action* ..................................................................................... 24

5.2 Activity Diagram ..................................................................................................... 26

5.3 UML Class Diagram ................................................................................................ 27

5.4 Sequence Diagram ................................................................................................... 27

Chapter 6 ............................................................................................................................ 28

System Implementation ..................................................................................................... 28

6.1 System Tools and Techniques ................................................................................. 28

*6.1.1 System Tools* ...................................................................................................... 28

*6.1.2 System Technologies* .......................................................................................... 29

6.2 Class Diagram ......................................................................................................... 31

6.3 Deployment Diagram .............................................................................................. 32

Chapter 7 ............................................................................................................................ 33

System Testing ................................................................................................................... 33

7.1 System Testing ........................................................................................................ 33

*7.1.1 Black Box Testing* ............................................................................................... 33

7.2 Test Cases ................................................................................................................ 34

*7.2.1 User Registration Case* ...................................................................................... 34

*7.2.2 User Login Test Case* ......................................................................................... 34

*7.2.3 AI/ML Prediction Test Case* ............................................................................... 35

*7.2.4 Chat Functionality Test Case* ............................................................................. 36

*7*.2.5 *Profile Update Test Case* ...................................................................................... 36

*7*.2.6 *Latest News Update Test Case* ............................................................................. 37

*7*.2.7 *Wallet Test Case* ................................................................................................... 38

Chapter 8 ............................................................................................................................ 39

Application Prototype ........................................................................................................ 39

8.1 Login In ................................................................................................................... 40

8.2 Sign Up .................................................................................................................... 41

8.3 Portfolio ................................................................................................................... 42

8.4 News Feed ............................................................................................................... 43

8.5 User Profile ............................................................................................................. 44

8.6 Create Wallet ........................................................................................................... 45

8.7 Chats ........................................................................................................................ 46

8.8 Market ..................................................................................................................... 47

8.9 Perdictions................................................................................................................ 48

Conclusion ......................................................................................................................... 49

References ......................................................................................................................... 50

# LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| API | = | Application Programming Interface |
| AWS | = | Amazon Web Services |
| CLI | = | Command Line Interface |
| CPU | = | Central Processing Unit |
| CRUD | = | Create, Read, Update, Delete |
| CSS | = | Cascading Style Sheets |
| CV | = | Computer Vision |
| EC2 | = | Elastic Compute Cloud |
| FYP | = | Final Year Project |
| GANs | = | Generative Adversarial Networks |
| GPU | = | Graphics Processing Unit |
| GUI | = | Graphical User Interface |
| HTML | = | Hypertext Markup Language |
| HTTPS | = | Hypertext Transfer Protocol Secure |
| HP | = | Hewlett-Packard |
| IDE | = | Integrated Development Environment |
| OS | = | Operating System |
| PPO | = | Preferred provider organization |
| RAM | = | Random Access Memory |
| SLR | = | Systematic Literature Review |
| SSD | = | Solid State Drive |
| SQL | = | Structured Query Language |
| SWAP | = | Swap Space |
| UI | = | User Interface |
| URL | = | Uniform Resource Locator |
| UML | = | Unified Modeling Language |
| UX | = | User Experience |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF TABLES** |  |
| Table 2.2 | Systematic literature review (SLR) | 7 |
| Table 3.1 | Description of User actor’s Register use case | 13 |
| Table 3.2 | Description of User Wallet Management use case | 14 |
| Table 3.3 | Description of User Crypto Transactions use case | 15 |
| Table 3.4 | Description of User Suggestion bot use case | 15 |
| Table 3.5 | Description of User Chat Functionality use case | 16 |
| Table 3.6 | Description of User Latest News Updates use case | 17 |
| Table 3.7 | Description of User Profile Update use case | 18 |
| Table 7.1 | Test Case for User Registration | 34 |
| Table 7.2 | Test Case for User Login | 34 |
| Table 7.3 | Test Case for AI/ML Model prediction | 35 |
| Table 7.4 | Test Case for Chat Functionality | 36 |
| Table 7.5 | Test Case for Profile Update | 36 |
| Table 7.6 | Test Case for Latest News Update | 37 |
| Table 7.7 | Test Case for Wallet | 38 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| Figure 1.1 | Project Scheduling | 4 |
| Figure 3.1 | Use Case Diagram of User actor | 12 |
| Figure 4.1 | Agile Methodology | 21 |
| Figure 5.1 | Architecture Diagram of Social Swap | 25 |
| Figure 5.2 | Activity Diagram of Social Swap | 26 |
| Figure 5.3 | Sequence Diagram | 27 |
| Figure 6.1 | UML Class Diagram | 31 |
| Figure 6.2 | Deployment Diagram | 32 |
| Figure 8.1 | Login Screen | 40 |
| Figure 8.2 | Sign up Screen | 41 |
| Figure 8.3 | Home Screen | 42 |
| Figure 8.4 | Latest News | 43 |
| Figure 8.5 | User Profile | 44 |
| Figure 8.6 | Create Wallet Screen | 45 |
| Figure 8.7 | Chat Screen | 46 |
| Figure 8.8 | Coin Graph Screen | 47 |
| Figure 8.9 | Recommended Coins Screen | 47 |
| Figure 8.10 | Prediction Chart | 48 |

# ABSTRACT

Social Swap is an all-encompassing web and mobile application designed to streamline cryptocurrency management and trading. The system employs a robust backend API, catering to user registration, authentication, wallet management, chat functionality, crypto transactions, and more. Wallet management features enable users to create and manage crypto wallets, generate unique wallet addresses, and track balances and transaction history seamlessly. Secure chat functionality allows users to communicate with each other confidentially, with messages encrypted to ensure privacy. The platform facilitates crypto transactions, enabling users to send and receive cryptocurrencies with ease, along with transparent display of transaction fees and confirmation statuses. Users can access comprehensive coin information, including prices, market cap, and volume, and view detailed graphs depicting coin price trends over time. Additionally, the application incorporates a sophisticated suggestion bot utilizing reinforcement learning for providing buy and sell suggestions.

**Keywords:** Cryptocurrency Management, Trading Platform, User Authentication, Wallet Management, Secure Chat, Crypto Transactions, Coin Information, Suggestion bot, Mobile Compatibility,

# Chapter 1

## Introduction

In this chapter, we introduce the Social Swap project, which focuses on the development of a crypto trading system. We begin by defining the relation of the project to other work in the same field, highlighting its unique contributions and innovations. Through this introductory section, we aim to grab the attention of potential readers and provide them with a comprehensive overview of the research problem, objectives, and the significance of this project within the field of crypto trading.

### 1.1 Overview

The Social Swap project aims to provide users with a comprehensive platform for managing their cryptocurrency assets securely and efficiently. This application will serve as a centralized hub for users to create, manage, and monitor their crypto wallets, engage in secure communication via built-in chat functionality, execute crypto transactions with help of suggestion bot, access real-time market data, and stay updated with the latest news in the cryptocurrency space.

### 1.2 Problem statement

The challenge lies in effective utilization of reinforcement learning algorithms in cryptocurrency trading systems to create a reliable trading platform capable of seamlessly navigating the dynamic and intricate landscape of cryptocurrency markets.

**1.3 Purpose**

The purpose of the crypto trading app, as outlined by the information provided, is to leverage reinforcement learning (RL) algorithms to develop a sophisticated trading platform that addresses the unique challenges and opportunities presented by cryptocurrency markets. This platform aims to streamline cryptocurrency management and trading by offering a comprehensive suite of features, including user registration, wallet management, secure chat functionality, crypto transactions, access to coin information and graphs, user settings customization, integration with exchanges, and access to the latest crypto news. By incorporating RL-based suggestions, the platform seeks to provide buy and sell suggestions, optimize trading strategies, and enhance overall trading performance. The overarching goal is to empower users with innovative tools and technologies that enable them to navigate the complexities of cryptocurrency markets more effectively, ultimately leading to improved financial outcomes.

### 1.4 Project Goals

These are project goals.

1. Achieve market leadership by creating a crypto trading platform that offers innovative features and superior user experience, setting new standards in the industry.
2. Ensure scalability and reliability to accommodate rapid user growth and fluctuating market conditions, maintaining uninterrupted service and maximizing trading opportunities for users.
3. Foster trader success through comprehensive educational resources, advanced market analysis tools, and collaborative features, empowering users to make informed decisions and achieve their financial goals.

#### 1.5 Project Objectives

These are project objectives:

Develop a user-friendly and intuitive crypto trading platform accessible to both experienced traders and novices, ensuring ease of navigation and usability across all features and functionalities.

Provide buy and sell suggestions by implementing RL algorithms techniques to optimize trading strategies, thereby reducing manual analysis and improving trading efficiency.

Enhance user experience and satisfaction by integrating real-time market data, advanced charting tools, and enabling users to make informed trading decisions and stay updated with market trends and developments.

#### 1.6 Project Scope

The project involves the development of an innovative cryptocurrency trading platform aimed at simplifying and enhancing the trading experience for users. This platform will offer a wide range of functionalities, including user registration and authentication, wallet management, chat functionality, secure crypto transactions, real-time display of coin information, interactive coin graphs, utilization of reinforcement learning, robust security settings, customizable user settings, seamless integration with cryptocurrency exchanges, mobile compatibility, multi-platform support, and access to the latest crypto news updates. By consolidating these features into a single platform, Social Swap aims to streamline cryptocurrency trading processes while improving efficiency, accuracy, and user satisfaction. With its user-friendly design and accessible features, Social Swap is poised to revolutionize the cryptocurrency trading landscape by providing a comprehensive solution for traders of all levels.

#### 1.7 Project History

Reinforcement learning-based crypto related model gained attraction in the 2010s [1], as traders sought more adaptive strategies. Early adoption involved rule-based bots using technical indicators. Introduction of reinforcement learning in the mid-2010s sparked experimentation. Researchers and traders explored custom algorithms to optimize trading strategies. Advancements in deep reinforcement learning algorithms, like deep Q-networks and proximal policy optimization, enabled bots to learn from historical data and adapt to market conditions. Industry players, including hedge funds, adopted reinforcement learning-based bots for improved profitability and risk management. Ongoing research and market developments continue to shape this evolving field.

#### 1.8 Project Significance

The integration of reinforcement learning within the cryptocurrency wallet project holds immense significance. By leveraging reinforcement learning, the project aims to revolutionize user trading experiences by offering advanced strategies and automation tools previously inaccessible to many. Reinforcement learning empowers users to optimize their trading activities effectively, navigating cryptocurrency markets with greater precision and adaptability. This innovative approach not only enhances the project's value proposition but also distinguishes it from traditional cryptocurrency wallets, attracting a broader user base seeking cutting-edge solutions in the dynamic landscape of digital asset trading.

It's worth noting that while the project aims to utilize reinforcement learning, major cryptocurrency exchanges have not publicly disclosed the use of reinforcement learning in their suggestion bot strategies. This highlights the innovative edge of the project, offering users access to advanced trading technologies that may not be available on mainstream platforms.

### 1.9 Proposed Solution

The proposed solution involves developing a comprehensive trading platform that leverages RL algorithms to optimize trading strategies. The platform will offer intuitive user interfaces, real-time market data analysis, customizable trading parameters, and robust risk management capabilities to empower users to navigate cryptocurrency markets effectively and achieve their financial goals.

### 1.10 Project Scheduling

Figure 1.1: Project Timeline

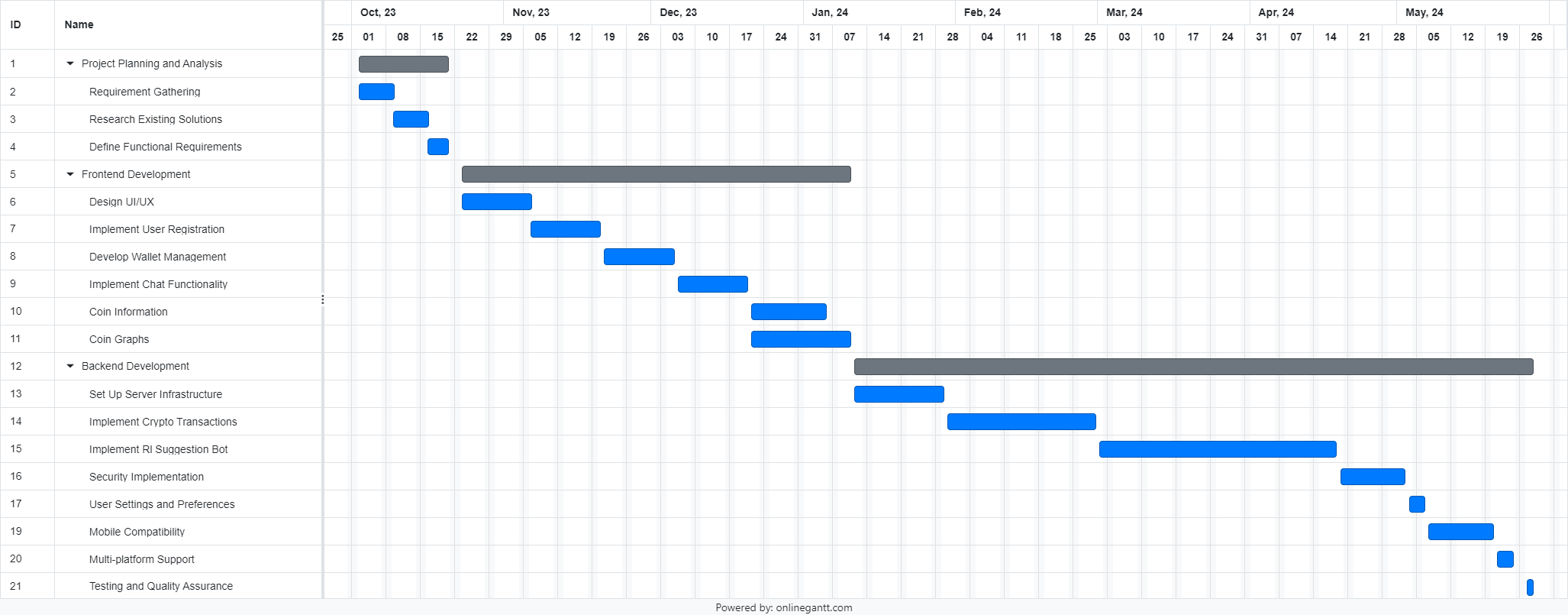


Figure 1.1 Project Scheduling

### 1.11 Risks and Risk Mitigation

***1.11.1 Technology Risks***

There is a risk of encountering technological challenges during the development and implementation of the trading platform, including compatibility issues, scalability limitations, and security vulnerabilities. A comprehensive technical evaluation of all components and technologies used in the platform to ensure compatibility with existing systems and infrastructure and implement robust security measures to protect against cyber threats and vulnerabilities. Regular testing and quality assurance procedures will be conducted to identify and address any technical issues early in the development process.

***1.11.2 Financial Risks***

Financial risks such as budget overruns, funding shortages, and unexpected expenses may impact the project's progress and viability. To mitigate these risks, a detailed budget plan, accurately estimating project costs and resource requirements. Regular financial monitoring and reporting will be implemented to track expenditures and ensure adherence to the budget. Additionally, contingency funds will be set aside to cover unexpected expenses or budgetary fluctuations, providing financial stability and flexibility throughout the project lifecycle.

# Chapter 2

## Theoretical background

The following chapter is an insightful literature review of the objectives defined above in chapter 1. This chapter will give proper insight into what kind of work has been done over period in each objective study. Each objective has its frameworks, methodologies, and study approaches which make an in-depth mindset and flow to this synopsis of research work and innovation to future impacts. Every aspect of the study will be clarified and organized in this chapter's literature composition with help of research points of view.

### 2.1 Literature review

Cryptocurrency trading has witnessed a surge in interest due to the potential for high returns and the proliferation of digital assets. In recent years, researchers and practitioners have explored the application of reinforcement learning (RL) techniques to develop automated suggestion bots capable of making intelligent decisions in cryptocurrency markets.

### *2.1.1 Overview of existing research*

In the realm of cryptocurrency trading using reinforcement learning (RL), significant researchers have made notable contributions to the field. Mnih *et al*. [2], introduced Deep Q-Networks (DQN), a pivotal deep RL algorithm, which laid the groundwork for its application across various domains. Their pioneering work demonstrated the potential of deep RL techniques, particularly in complex decision-making scenarios. Haarnoja *et al*. [3], extended RL to continuous action spaces with Soft Actor-Critic (SAC), showcasing its efficacy in navigating intricate market environments. Their research underscored the adaptability and versatility of RL algorithms, particularly in addressing real-world challenges. Additionally, Liu *et al* [4] explored RL algorithms for developing adaptive suggestion bots, highlighting their ability to outperform traditional strategies in profitability and risk management. These findings underscore the promising potential of RL-based suggestions bot in enhancing trading strategies and navigating dynamic cryptocurrency markets. However, despite these advancements, there are still notable gaps in the previous research that warrant further exploration. These include addressing challenges such as market volatility, data scarcity, and regulatory uncertainties. Additionally, improving the robustness and reliability of RL-based suggestions bot in real-world trading environments is imperative. Furthermore, exploring innovative approaches to enhance the performance and adaptability of RL algorithms in cryptocurrency markets is crucial for advancing the field.

### *2.2* Systematic literature review (SLR)

*Table 2.1: Systematic literature review (SLR)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Title** | **Methodology** | **Objectives** | **Result** | **Limitations** | **Conclusion** |
| 1 | Human-level control through deep reinforcement learning [2] | Utilized Deep Q-Network (DQN) architecture with convolutional neural networks (CNNs), experience replay, and target networks for training. | Demonstrate DQN's efficacy in achieving human-level control on Atari 2600 games, refining network architecture, and training methodology for improved stability and performance. | Achieved human-level performance on Atari games, surpassing previous approaches in stability, sample efficiency, and final performance. | Computational requirements, sensitivity to hyperparameters, challenges in handling partially observable environments, and limited generalization beyond Atari games. | Deep reinforcement learning, particularly DQN, shows promise in achieving human-level control; refinement of methods and further research required to address limitations and broaden applicability. |
| 2 | Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor [3] | Introduced Soft Actor-Critic (SAC), utilizing off-policy maximum entropy reinforcement learning with stochastic actor representations. | Develop a robust algorithm for continuous control tasks, emphasizing high sample efficiency and stability. | SAC achieves state-of-the-art performance on continuous control benchmarks, demonstrating superior sample efficiency and robustness. | Applicability may be restricted to continuous control tasks, and computational demands could hinder real-world deployment. | SAC represents a significant advancement in deep reinforcement learning, offering efficient exploration and stable learning in continuous control domains, while further research is needed to address scalability and applicability concerns. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | Improving the cryptocurrency price prediction performance based on reinforcement learning [5] | The researchers used reinforcement learning (RL) for predicting cryptocurrency prices. They developed a model that combines RL with a Long Short-Term Memory (LSTM) network, a type of recurrent neural network that can learn and remember over long sequences of information. | The main objective of the study was to improve the prediction accuracy of cryptocurrency prices, which is a challenging task due to the highly volatile and unpredictable nature of the cryptocurrency market. | The results demonstrated that the RL-LSTM model significantly improved the prediction performance compared to traditional methods. It was able to accurately predict the upward and downward trends of cryptocurrency prices. | While the study achieved promising results, it did not mention specific limitations. However, it can be inferred that the model's effectiveness may be limited by factors such as sudden market changes or the availability and quality of historical data. | The study concluded that reinforcement learning can be effectively used to improve the prediction performance of cryptocurrency prices. This suggests that RL could be a valuable tool for investors and traders in the cryptocurrency market. |
| 4 | The recurrent reinforcement learning crypto agent [6] | The study used a recurrent reinforcement learning model for cryptocurrency trading. The model, known as the Recurrent Reinforcement Learning (RRL) agent, was applied to the cryptocurrency market. | The primary goal of the research was to apply machine learning, specifically reinforcement learning, to cryptocurrency trading and examine its effectiveness. | The study found that the RRL agent was able to generate profitable trading strategies for the Bitcoin market. It outperformed a simple buy-and-hold strategy and showed the ability to adapt to market changes. | The study did not mention specific limitations. However, it can be inferred that the model's performance could be affected by factors such as market volatility. | The study concluded that recurrent reinforcement learning can be effectively used in cryptocurrency trading. This suggests that RRL could be a promising approach for developing automated trading systems in the cryptocurrency market. |

### 2.3 Critical Analysis

The theoretical background chapter delves into the realm of cryptocurrency suggestions bot utilizing reinforcement learning (RL) techniques, exploring the contributions of key researchers such as Mnih *et al*. [2], Haarnoja *et al*. [3], and Liu *et al* [4]. Mnih *et al*. laid the groundwork with Deep Q-Networks (DQN), while Haarnoja et al. extended RL to continuous action spaces through Soft Actor-Critic (SAC), and Liu *et al*. explored adaptive trading strategies. These researchers employed various methodologies, training DQN on raw pixel inputs, utilizing SAC for navigating complex market environments, and dynamically adjusting trading strategies based on market conditions. Their findings underscore the effectiveness of RL-based suggestions bot in outperforming traditional strategies. However, gaps in previous research persist, including challenges related to market volatility, data scarcity, and regulatory uncertainties, necessitating further investigation. The systematic literature reviews conducted shed light on the current state of RL-based cryptocurrency suggestions bot, emphasizing the need for empirical studies to validate effectiveness and address limitations, while offering valuable insights into the future directions of research and development in this domain.

**PTO**

# Chapter 3

## System Requirements

This chapter will discuss the research to achieve our objectives and to answer the questions that have arisen during our study. In-depth dive into methodological aspects along with performance metrics and dataset will be understood with intense discussions.

### 3.1 Functional Requirements

#### *3.1.1 User Management*

The system should allow users to register, authenticate, and manage their accounts securely, details such as name, email, gender, and password.

***3.1.2 Wallet management***

Users should be able to create and manage multiple cryptocurrency wallets, generate unique wallet addresses, and view their wallet balances and transaction history.

#### *3.1.3 Chat Functionality*

Users should have the capability to securely communicate with each other within the application, sending and receiving text messages, emojis, and attachments.

***3.1.4 Crypto Transactions***

Users should be able to send and receive cryptocurrencies to/from other users, specifying the amount and recipient address for each transaction, with transaction fees and confirmation status displayed.

***3.1.5 Coin Graphs***

The system should provide graphical representations of coin prices over time, offering various options for different timeframes and displaying relevant indicators to users.

***3.1.6 Suggestions bot***

The application should feature a suggestions bot utilizing reinforcement learning to provide buy and sell suggestions. Users should be able to enable/disable the bot's suggestion feature and monitor its performance.

#### *3.1.7 News Updates*

The system should provide users with access to the latest news and updates from the cryptocurrency world, including articles, blog posts, and announcements.

### 3.2 Non-Functional Requirements

#### *3.2.1 Secure Communication*

The system should ensure secure communication between the frontend and backend using HTTPS to protect user data and prevent unauthorized access.

#### *3.2.2 Cross Platform Compatibility*

The front-end web and mobile apps should work on multiple platforms, including desktop and mobile devices, ensuring a consistent user experience.

#### *3.2.3 Responsive Design*

The front-end web and mobile apps should be responsive and adaptable to different screen sizes and orientations.

#### *3.2.4 User Interface*

The front-end apps should have an intuitive and user-friendly interface with appropriate validation and error handling to provide a smooth user experience.

### 3.3 Use Case Diagram

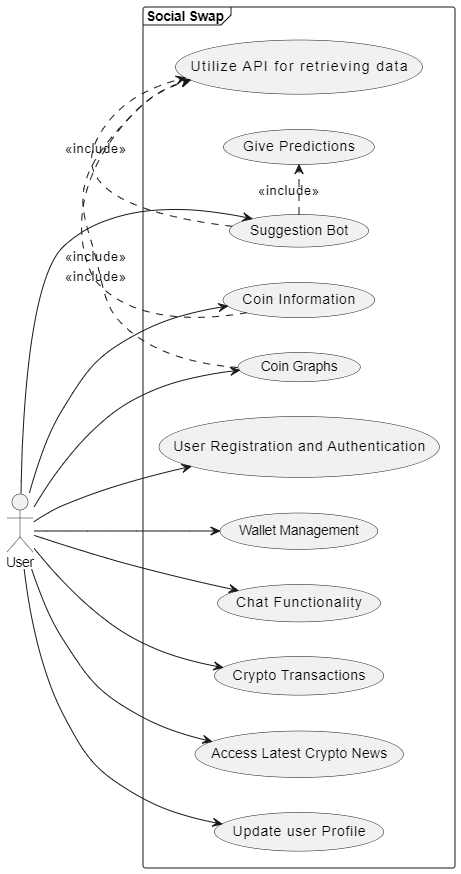


Figure 3.1 Use Case Diagram

**3.4 Use case Description.**

#### *3.4.1 Use Case of User*

Table 3.1: Description of User actor’s Register use case

|  |  |  |
| --- | --- | --- |
| **Use Case Name** |  | User Registration and Authentication |
| **Actors** |  | User |
| **Preconditions** |  | The user must access the registration page of the cryptocurrency trading application. |
| **Normal Flow** | **Descriptions** | The user navigates to the registration page and enters their desired username, email, and password.  The user submits the registration form.  The system validates the entered information, ensuring the username and email are not already in use and that the password meets security requirements.  Upon successful validation, the system creates a new user account with the provided details.  The system sends a verification email to the user's email address.  The user verifies their email by following the instructions in the email.  Once verified, the system confirms the successful registration and logs the user into the application. |
| **Postconditions** | The user has successfully registered an account and is logged into the cryptocurrency trading application. |
| **Alternative flow and exceptions** | | If the entered username or email already exists in the system, the user is notified and prompted to provide different information.  If there are validation errors in the entered information (e.g., invalid email format), the user is notified and prompted to correct the errors.  If the verification email is not received or the user fails to verify within a specified time, the registration process may be considered incomplete. |
| **Nonfunctional requirements** | | The registration process should be user-friendly, secure, and efficient.  The system should handle concurrent registration attempts and protect against potential security risks such as account enumeration or brute-force attacks. |

#### 

#### *3.4.2 Use Case of Wallet Management*

Table 3.2: Description of User Wallet Management use case

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | | Wallet Management |
| **Actors** | | User |
| **Preconditions** | | The user must be logged into the cryptocurrency trading application. |
| **Normal Flow** | **Descriptions** | The user accesses the wallet management section of the application.  The user creates a new wallet or selects an existing wallet.  The user can add or remove cryptocurrencies from the wallet.  The user can view the balances and transaction history of the selected wallet. |
| **Postconditions** | The user has successfully managed their cryptocurrency wallets. |
| **Alternative flow and exceptions** | | If there are errors in adding or removing cryptocurrencies from the wallet, the user is notified and prompted to try again.  If the selected wallet does not exist or cannot be accessed, the user is notified and prompted to select a valid wallet. |
| **Nonfunctional requirements** | | The wallet management process should be intuitive, secure, and responsive.  The system should accurately display real-time balances and transaction history. |

#### *3.4.3 Use Case of Crypto Transactions*

*Table 3.3: Description of User* *Crypto Transactions use case*

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | | Crypto Transactions |
| **Actors** | | User |
| **Preconditions** | | The user must be logged into the cryptocurrency trading application. |
| **Normal Flow** | **Descriptions** | The user initiates a cryptocurrency transaction.  The user selects the cryptocurrency to send, enters the recipient's address, and specifies the amount.  The system validates the transaction details and the user's account balance.  Upon successful validation, the system processes the transaction.  The recipient receives the cryptocurrency in their external wallet. |
| **Postconditions** | The cryptocurrency transaction is successfully completed. |
| **Alternative flow and exceptions** | | If the transaction details are invalid or incomplete, the user is notified and prompted to provide correct information.  If the user's account balance is insufficient to complete the transaction, the user is notified and prompted to add funds. |
| **Nonfunctional requirements** | | The transaction process should be fast, secure, and reliable.  The system should provide real-time updates on transaction status. |

***3.4.4 Use Case of Suggestion Bot***

*Table 3.4: Description of User* *Suggestion Bot use case*

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | | Suggestion Bot |
| **Actors** | | User |
| **Preconditions** | | The user must be logged into the cryptocurrency trading application. |
| **Normal Flow** | **Descriptions** | The user accesses the suggestion bot section of the application.  The user configures trading strategies, risk management rules, and other parameters.  The suggestion bot monitors market conditions and give predictions according to the configured parameters. |
| **Postconditions** | The suggestion bot is successfully predicting trades according to the user's preferences. |
| **Alternative flow and exceptions** | | If there are errors in configuring the suggestion bot to give predictions, the user is notified and prompted to correct the settings. |
| **Nonfunctional requirements** | | The suggestion bot should be reliable, responsive.  The system should provide real-time updates on the bot's performance. |

#### 

#### *3.4.5 Use Case of Chat Functionality*

*Table 3.5: Description of User Chat Functionality use case*

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | | Chat Functionality |
| **Actors** | | User |
| **Preconditions** | | The user must be logged into the cryptocurrency trading application. |
| **Normal Flow** | **Descriptions** | The user accesses the chat functionality within the application.  The user selects a contact or enters a chat room to initiate a conversation.  The user sends text messages, emojis, or attachments to communicate with other users securely.  The messages are encrypted to ensure privacy and security.  The user receives messages from other users and can view the chat history. |
| **Postconditions** | The user successfully communicates with other users through the chat functionality. |
| **Alternative flow and exceptions** | | If there are errors in sending or receiving messages, the user is notified and prompted to retry.  If the chat room is full or unavailable, the user is informed and prompted to try again later. |
| **Nonfunctional requirements** | | The chat functionality should be responsive, providing real-time communication between users.  Messages should be encrypted to ensure confidentiality and security.  The system should support multimedia attachments such as images and files for richer communication. |

***3.4.6 Use Case of Latest News Updates***

*Table 3.6: Description of User* *Latest News Updates use case*

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | | Latest News Updates |
| **Actors** | | User |
| **Preconditions** | | The user must be logged into the cryptocurrency trading application. |
| **Normal Flow** | **Descriptions** | The user accesses the news section within the application.  The user browses through the latest articles, blog posts, and announcements related to cryptocurrencies and blockchain technology.  The user selects articles based on their interests and preferences.  The user reads the selected articles to stay informed about market trends, regulatory developments, and technological advancements.  The news feed is regularly updated to ensure users have access to the most current information. |
| **Postconditions** | The user successfully reads the latest news updates relevant to cryptocurrency trading. |
| **Alternative flow and exceptions** | | If there are errors in loading or displaying news articles, the user is notified and prompted to refresh the page.  If the news feed is temporarily unavailable, the user is informed and prompted to try again later. |
| **Nonfunctional requirements** | | The news section should provide a user-friendly interface for browsing and reading articles.  Articles should be categorized and searchable to facilitate navigation and discovery.  The system should fetch and display news updates in real-time to keep users informed about the latest developments. |

***3.4.7 Use Case of Profile Update***

*Table 3.7: Description of User Profile Update Use Case*

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | | Profile Update |
| **Actors** | | User |
| **Preconditions** | | The user must be logged into the cryptocurrency trading application. |
| **Normal Flow** | **Descriptions** | The user navigates to the profile section of the application.  The user selects the option to edit their profile.  The user updates their profile information.  The user saves the updated profile information.  The system validates the updated information and saves it to the database.  The user receives a confirmation that their profile has been successfully updated |
| **Postconditions** | The user's profile information is successfully updated and saved in the system. |
| **Alternative flow and exceptions** | | If there are errors in the updated information (e.g., invalid email format), the user is prompted to correct them. |
| **Nonfunctional requirements** | | If the profile update fails due to a system error, the user is notified and asked to try again later.  The profile update process should be user-friendly and responsive.  The system should ensure data validation and security during the update process. |

**PTO**

# Chapter 4

## Methodology

In this chapter, a high level of objective concerns of this research has been explained thoroughly. This synopsis research work is formulated to grasp the attention of potential drawbacks that was noted. This chapter will explain the objective norms and associations to this work along with objective shaping into full fledge experimental algorithmic outcome.

### 4.1 Project Planning

The Social Swap project involves developing a web and mobile application to predict the profit or loss of cryptocurrencies. The project, executed over 6 months using an Agile methodology, starts with initial user research to gather insights and define project goals and requirements, followed by the creation of a prioritized Product Backlog with user stories and features. Design and prototyping involve developing wireframes and mockups for key features. Development is organized into multiple sprints (2-4 weeks each), focusing on implementing user authentication, data collection, and machine learning algorithms for prediction. Each sprint includes planning, development, continuous testing (unit, integration, usability, security, and performance), and review phases make necessary improvements. Deployment and release planning are conducted in stages, preparing a release plan with user documentation and support materials. This Agile approach ensures a flexible, responsive development process that continuously adapts to user needs and project goals, delivering a high-quality application within the planned timeframe.

### 4.2 Methodology for Software Development

Methodologies are structured approaches that help teams efficiently navigate the process of software development, ensuring that goals and requirements are met effectively.



Figure 4.1 Agile Methodology

### *4.2.1 Existing Methodologies*

In the realm of software development, various methodologies are employed to guide teams through the development process. These include:

* Agile
* Iterative Waterfall
* Scrum
* Extreme Programming
* Rapid Application Development
* Spiral
* Prototyping.

### *4.2.2 Selected Methodology*

For our project, we've opted for Agile Software Development. Agile is chosen because it's adaptable, allowing us to make iterative changes to our codebase as needed throughout the development process.

Agile Software Development prioritizes adaptability, teamwork, and customer satisfaction. It's grounded in the Agile Manifesto, which values individuals and interactions, functional software, customer collaboration, and responsiveness to change.

**Pros:**

* **Adaptability:** Agile allows us to respond quickly to changes in requirements, ensuring that the product remains aligned with customer needs.
* **Customer Collaboration:** Regular interaction with customers ensures that the final product meets their expectations.
* **Iterative Approach:** Incremental delivery of functional software allows for early feedback and continuous improvement.
* **Transparency:** Regular demonstrations provide stakeholders with visibility into project progress and upcoming features.

**Cons:**

* **Dependency on Customer Involvement:** Agile relies heavily on customer involvement, which can be challenging to maintain, especially if customers are busy or unavailable.
* **Complexity:** Managing multiple iterations and frequent changes can add complexity to the development process, requiring effective project management and communication.
* **Resource Intensive:** Agile requires a dedicated and collaborative team, which may not be feasible for all projects or organizations.

### 4.3 Reason for Selected Methodology

Selecting the Agile software development methodology for our "Social Swap" project allows us to enhance flexibility, collaboration, and product quality. Agile’s iterative process enables us to adapt quickly to changes and incorporate continuous feedback, ensuring that the final product meets customer needs. This approach ensures that the app's functionalities, user interface, and features align closely with user requirements and expectations.

By fostering close collaboration among cross-functional teams and involving stakeholders regularly, Agile improves communication and transparency. Additionally, the focus on delivering incremental improvements and continuous testing helps bring high-quality features to market faster, ultimately increasing customer satisfaction and project success.

#### *4.3.1 Quick Response to change*

Agile fosters a culture that allows teams to respond swiftly to constantly shifting priorities and requirements. This adaptability is particularly useful in the rapidly evolving technology sector and the dynamic cryptocurrency market.

#### *4.3.2 Regular Demonstration*

Agile techniques emphasize regular demonstrations of project progress. Stakeholders can clearly see the project’s status, upcoming challenges, and new features due to this transparency. This includes showcasing features like user signup, cryptocurrency trading, AI/ML-based profit and loss predictions, chat functionalities, profile management, and crypto wallet transactions

#### *4.3.3 Agile as Mindset*

Agile represents a shift in culture that values adaptability, collaboration, and client satisfaction. It empowers team members and promotes a cooperative and positive work atmosphere, essential for developing a complex application like "Social Swap" that integrates AI/ML models and secure crypto transactions.

#### *4.3.4 Creating Tangible Value*

Agile places a high priority on creating tangible value as soon as possible in a project. For "Social Swap," this means delivering early iterations of core features such as basic trading functionalities, user authentication, and AI/ML prediction models. Customers benefit from the early delivery of promised advantages and opportunities for prompt feedback and modifications.

By selecting Agile for the "Social Swap" project, we ensure a development process that is both responsive to change and focused on delivering high-quality, user-centric features. This approach not only enhances the overall efficiency and effectiveness of the development team but also significantly boosts customer satisfaction by delivering a product that closely aligns with user expectations and market demands.

# Chapter 5

## System Architecture

This chapter focuses on the system architecture, the conceptual model that defines the structure, behavior, and more views of a system.

### 5.1 Architecture

For the Social Swap project, we have opted for the Microservices Architecture, a modern system design pattern widely adopted in software development. Microservices architecture breaks down an application into a collection of small, independently deployable services, each responsible for a specific business function.

### *5.1.1 Microservices*

Microservices embody distinct components of the application, each dedicated to a specific business function like user authentication, cryptocurrency trading, AI/ML prediction, chat functionality, profile management, and transaction processing. Developed, deployed, and scaled independently, these microservices grant agility and flexibility in both development and maintenance endeavors. Intercommunication among microservices occurs through well-defined APIs, fostering loose coupling and facilitating seamless integration and interoperability across the system.

### *5.1.2 Microservices in Action*

1. The **User Authentication Service** handles user registration, login, and authentication processes, ensuring secure access to the application.
2. The **Cryptocurrency Trading Service** enables users to buy, sell, and trade cryptocurrencies, providing real-time market data and transaction processing capabilities.
3. The **AI/ML Prediction Service** utilizes machine learning algorithms to predict the profit and loss of cryptocurrencies, providing valuable insights to users.
4. The **Chat Service** facilitates communication between users, allowing them to interact, share information, and discuss trading strategies.
5. The **Profile Management Service** enables users to view and update their profiles, manage account settings, and customize preferences.
6. The **Transaction Service** manages cryptocurrency wallet creation and transaction processing, ensuring secure and reliable fund transfers.

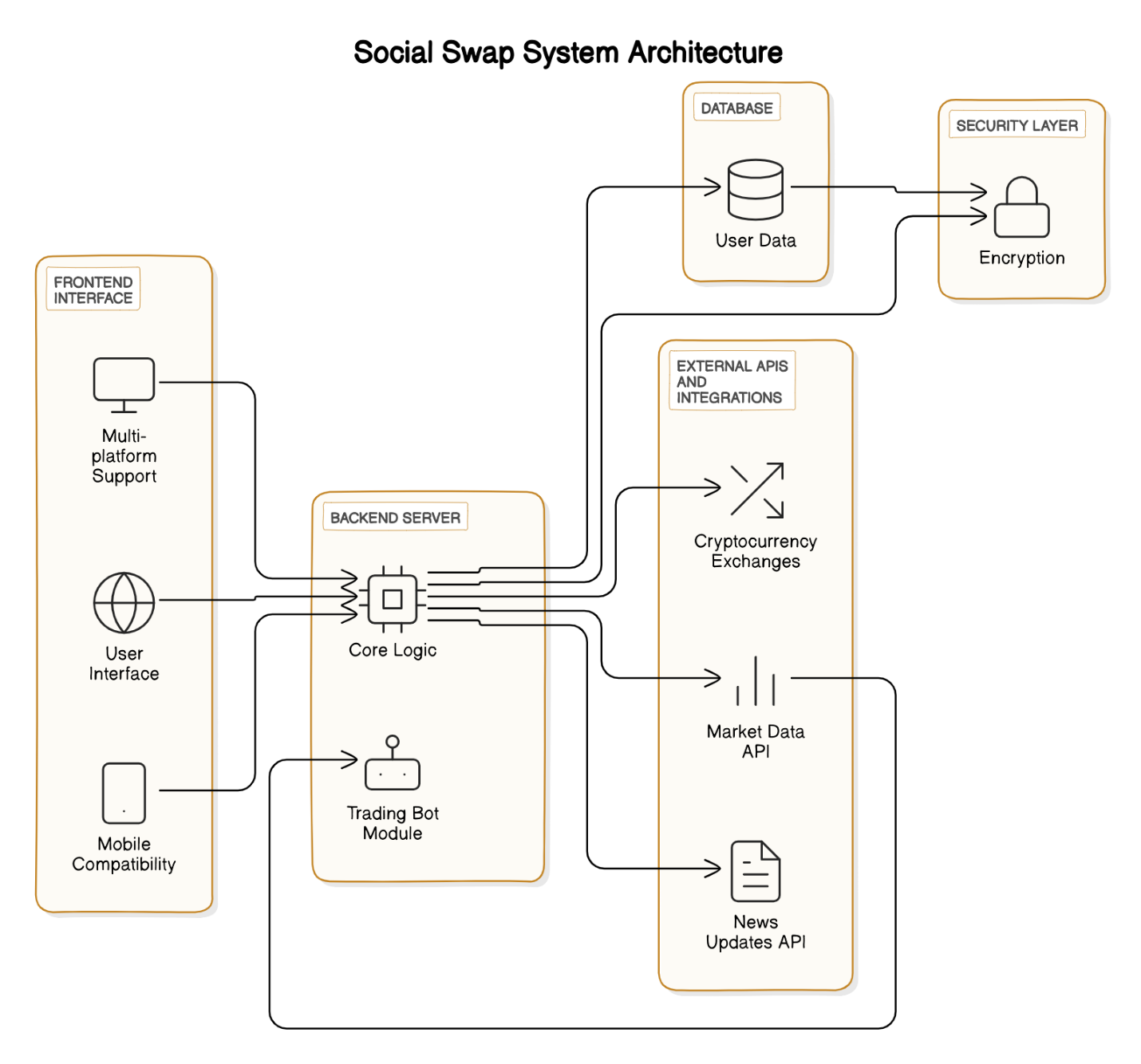


Figure 5.1 Architecture Diagram of Social Swap

### 5.2 Activity Diagram

This Activity Diagram depicts the architecture with three main components: frontend, backend, and database. The frontend allows users to perform actions such as crypto trading, chatting, and updating their profile, while the backend processes these actions and interacts with the database to update records and perform trades on the blockchain. The database stores user records, wallet information, and trades records.

A diagram of a company

Description automatically generated

Figure 5.2 Activity Diagram of Social Swap

**5.3 Sequence Diagram**A white background with black lines

Description automatically generated

Figure 5.3 Sequence Diagram

# Chapter 6

## System Implementation

In this chapter, we're exploring how Social Swap functions. We detail the steps taken to transform Social Swap's concepts into a working system, encompassing everything from its creation to its implementation in real-world scenarios.

### 6.1 System Tools and Technologies

In process of developing Social Swap following technologies and tools are used:

##### 6.1.1 System Tools

System tools are essential components utilized by developers to streamline the development process, enhance productivity, and ensure the successful implementation of software applications.

###### **Backend Tools**

* **Visual Studio Code (VS Code):**

1. **Description:** VS Code is a lightweight, yet powerful source code editor developed by Microsoft, equipped with features like syntax highlighting, code completion, debugging support, and version control integration.
2. **Role:** As the primary integrated development environment (IDE), VS Code empowers developers in writing and managing backend code for the Social Swap application. It facilitates efficient coding practices and collaboration among team members, particularly for Python Django framework development.

* **Firebase:**

1. **Description:** Firebase, offered by Google, provides a comprehensive platform for mobile and web app development, encompassing real-time database, authentication, cloud storage, hosting, and more.
2. **Role:** Firebase's real-time database feature forms a cornerstone of the Social Swap backend, enabling seamless storage and synchronization of user data. Its authentication services bolster application security, ensuring safe user authentication and authorization processes.

###### **Frontend Tools**

* **Emulators:**

1. **Description:** Emulators simulate mobile devices or operating systems on a computer, facilitating the testing of frontend mobile app UIs without physical devices.
2. **Role:** We have used android emulators to test the frontend mobile app UI developed with Flutter. They provide a convenient means to validate app functionality, performance, and responsiveness across various device configurations and screen sizes.

##### 6.1.2 System Technologies

System technologies comprise the underlying frameworks and libraries utilized in the development process to implement core functionalities and features of software applications.

###### **Backend Technologies**

* **Python (TensorFlow):**

1. **Description:** Python, a versatile programming language, coupled with TensorFlow, an open-source machine learning framework, serves as the primary technology stack for developing and training machine learning models.
2. **Role:** Python with TensorFlow empowers developers in data preprocessing, feature engineering, model training, evaluation, and deployment for accurate predictions in cryptocurrency profit and loss analysis within the Social Swap application.

###### **Frontend Technologies**

* **Flutter:**

1. **Description:** Flutter, a UI software development kit (SDK) by Google, facilitates the creation of natively compiled applications across multiple platforms.
2. **Role:** Flutter is instrumental in crafting the mobile app UI for Social Swap, enabling the development of visually appealing and responsive user interfaces across iOS and Android platforms.
3. **Dependencies:** 
   * cupertino\_icons: ^1.0.2
   * firebase\_core: ^2.25.4
   * cloud\_functions: ^4.6.0
   * firebase\_auth: ^4.17.5
   * cloud\_firestore: ^4.15.5
   * random\_string: ^2.3.1
   * shared\_preferences: ^2.2.2
   * intl: ^0.18.0
   * firebase\_storage: ^11.6.0
   * image\_picker: ^1.0.7
   * file\_picker: ^6.1.1
   * readmore: ^2.2.0
   * url\_launcher: ^6.2.4
   * http: ^0.13.6
   * chart\_sparkline: ^1.0.15+1
   * syncfusion\_flutter\_charts: ^24.2.4+3
   * flutter\_spinkit: ^5.2.0
   * get: ^4.6.6
4. **Testing Technologies**

As part of the system implementation of Social Swap, we have used Selenium for unit testing. Selenium is a robust framework for testing web applications, and we specifically used Selenium WebDriver version 4.21.0.

For scripting the test cases, we used Python bindings for Selenium WebDriver. The Python bindings offered the necessary functions to interact with Selenium WebDriver in a more Pythonic way, which fits in well with our Python-based backend.

To facilitate the browser-based testing, we used ChromeDriver version 114.0.5735.90. ChromeDriver is a standalone server that implements WebDriver's wire protocol, providing necessary support for the HTML5 features of modern web applications. It allowed us to automate Google Chrome for the testing process, enabling us to test our application in a real-world browser environment.

### 6.2 Class Diagram

### A diagram of a computer Description automatically generated

Figure 6.1 UML Class Diagram

### 6.3 Deployment Diagram

**A screenshot of a computer

Description automatically generated**

Figure 6.2 Deployment Diagram

# Chapter 7

## System Testing

### 7.1 System Testing

System testing for the Social Swap crypto trading app involves a comprehensive evaluation of the entire application, ensuring that all its components and functionalities, such as account creation, fund transfer, crypto trading, and withdrawal processes, integrate seamlessly and operate as intended. The aim is to verify that Social Swap meets specified requirements, including functional aspects like accurate trade execution and non-functional aspects such as fast response times, reliability of services, and user-friendly interface, in a real-world trading environment. Unlike unit or integration testing, which focuses on individual features or modules, system testing evaluates Social Swap as a whole, ensuring that all the different components work together cohesively. It typically follows all the individual features have been tested and integrated, and precedes acceptance testing, where the app is tested in a production-like environment.

##### 7.1.1 Black Box Testing

Black-box testing is a technique where the system's functionality is evaluated without considering its internal structure or implementation details. Black box testing for the Social Swap crypto trading app involves assessing the app's functionality without delving into its internal structure or implementation specifics. Testers interact with the app purely from a user's standpoint, focusing on inputs such as trading commands, outputs like trade execution confirmations, and observed behavior including responsiveness and accuracy of trade executions. Adherence to the app's specifications and requirements is crucial, aiming to identify defects or issues in functionality without relying on knowledge of the internal code or design. This comprehensive evaluation maintains independence from the app's internal workings, providing an unbiased and objective assessment of its user-facing functionality.

### 7.2 Test Cases

##### 7.2.1 User Registration Test Cases

Table 7.1 Test Case for User Registration

|  |  |
| --- | --- |
| Component Name | Authentication Module |
| Module Name | User Registration |
| Condition being tested | Validating the functionality of user registration. |
| Expected Result | Upon successful registration, the user should be added to the system database and provided access to the application. |
| Success Scenarios | Upon successful user registration, they are redirected to the home page |
| Failure Scenarios | * User attempts to register with an existing email or username then system displays an error message indicating that the email or username is already in use. * User submits incomplete registration information then system prompts the user to fill in all required fields. |
| Test Result (Pass/ Fail) | Passed |

##### 7.2.2 User Login Test Case

Table 7.2 Test Case for User Login

|  |  |
| --- | --- |
| Component Name | Authentication Module |
| Module Name | User Login |
| Condition being tested | Validating the functionality of user login. |
| Expected Result | Upon successful login, the user should be granted access to the application dashboard. |
| Success Scenarios | User enters valid login credentials then user is successfully logged in and redirected to the dashboard. |
| Failure Scenarios | User enters incorrect username or password then system displays an error message indicating invalid credentials. |
| Test Result (Pass/ Fail) | Passed |

##### 7.2.3 AI/ML Prediction Test Cases

Table 7.3 Test Case for AI/ML Model Prediction

|  |  |
| --- | --- |
| Component Name | Prediction Module |
| Module Name | AI/ML Prediction |
| Condition being tested | Validating the accuracy and functionality of the AI/ML model for predicting cryptocurrency profit and loss. |
| Expected Result | The model should provide accurate predictions based on historical data and relevant factors. |
| Success Scenarios | The model successfully processes the data and generates predictions for profit and loss. |
| Failure Scenarios | The model produces unreliable predictions or errors due to insufficient or incorrect input data. |
| Test Result (Pass/ Fail) | Passed |

##### 7.2.4 Chat Functionality Test Cases

Table 7.4 Test Case for Chat Functionality

|  |  |
| --- | --- |
| Component Name | Chat Module |
| Module Name | Chat Functionality |
| Condition being tested | Validating the functionality of the chat feature for users to communicate with each other. |
| Expected Result | Users should be able to send and receive messages seamlessly within the application. |
| Success Scenarios | The message is successfully delivered to the recipient, and both users can view the conversation history. |
| Failure Scenarios | System displays an error message indicating the message could not be sent and prompts the user to retry. |
| Test Result (Pass/ Fail) | Passed |

##### 7.2.5 Profile Update Test Cases

Table 7.5 Test Case for Profile Update

|  |  |
| --- | --- |
| Component Name | User Profile Module |
| Module Name | Profile Update |
| Condition being tested | Validating the functionality of user profile updates. |
| Expected Result | Upon successful profile update, the changes should be reflected in the user's profile information. |
| Success Scenarios | User successfully updates their profile information, the system saves the changes, and the updated information is visible on the user's profile. |
| Failure Scenarios | • User attempts to save changes without filling in all required fields, then the system prompts the user to fill in all required fields.  • User attempts to save changes with invalid input, then the system displays an error message indicating the invalid input. |
| Test Result (Pass/ Fail) | Passed |

##### 7.2.6 Latest News Update Test Cases

Table 7.6 Test Case for Latest News Update

|  |  |
| --- | --- |
| Component Name | News Module |
| Module Name | Latest News Updates |
| Condition being tested | Validating the functionality of the Latest News Updates feature. |
| Expected Result | Users should be able to select and read the latest news articles related to cryptocurrencies and blockchain technology. The news feed should be regularly updated with the most current information. |
| Success Scenarios | User successfully accesses the news section, browses through the latest articles, selects articles based on their interests, and reads the selected articles. The system updates the news feed regularly. |
| Failure Scenarios | • If there are errors in loading or displaying news articles, the system notifies the user and prompts to refresh the page.  • If the news feed is temporarily unavailable, the system informs the user and prompts to try again later. |
| Test Result (Pass/ Fail) | Passed |

##### 7.2.7 Wallet Test Cases

Table 7.7 Test Case for Wallet

|  |  |
| --- | --- |
| Component Name | Wallet Module |
| Module Name | Wallet Management |
| Condition being tested | Validating the functionality of the Wallet Management feature. |
| Expected Result | User should be able to create a new wallet. View the balances and transaction history of the selected wallet. |
| Success Scenarios | User successfully accesses the wallet management section, creates a new wallet or selects an existing one, adds or removes cryptocurrencies, and views the balances and transaction history. The system accurately displays real-time balances and transaction history. |
| Failure Scenarios | • If there are errors in adding or removing cryptocurrencies from the wallet, the system notifies the user and prompts them to try again.  • If the selected wallet does not exist or cannot be accessed, the system notifies the user and prompts to select a valid wallet. |
| Test Result (Pass/ Fail) | Passed |

# Chapter 8

## Application Prototype

Before the final version of Social Swap is developed, a prototype of this cryptocurrency trading application serves as a preliminary model, created to visualize its functionality and user interface. This prototype acts as a tangible representation of the proposed solution, allowing stakeholders to assess its potential effectiveness, feasibility, and appeal.

By prototyping, developers can incorporate user feedback and adjust the direction of Social Swap, supporting an iterative refinement of the application's features and design. This includes elements such as the trading interface, wallet management, chat functionality, and the suggestion bot.

By utilizing a prototype, the team can identify and rectify issues early in the development cycle. This proactive approach ultimately results in a final product that is more polished, user-friendly, and tailored to the needs of cryptocurrency traders and enthusiasts. The prototype will serve as a crucial tool for ensuring that Social Swap offers a unique and seamless experience for its users in the dynamic world of cryptocurrency trading.

The following are the prototype screenshots of the Social Swap application, each accompanied by a concise description of the specific functionality they represent. These images provide a tangible representation of the user interface and various operations, offering a comprehensive understanding of the application's workflow and user experience. Please note that these visuals are part of a preliminary version of the application, subject to modifications and enhancements as we progress through the development process, incorporating user feedback and iterative design improvements.

### 8.1 Login In

The login screen for the Social Swap project features a clean, user-friendly interface with fields for email and password, a prominent "Login" button, and options for social media login. It includes a "Forgot Password?" link and a sign-up prompt for new users. The design ensures security and responsiveness across devices.

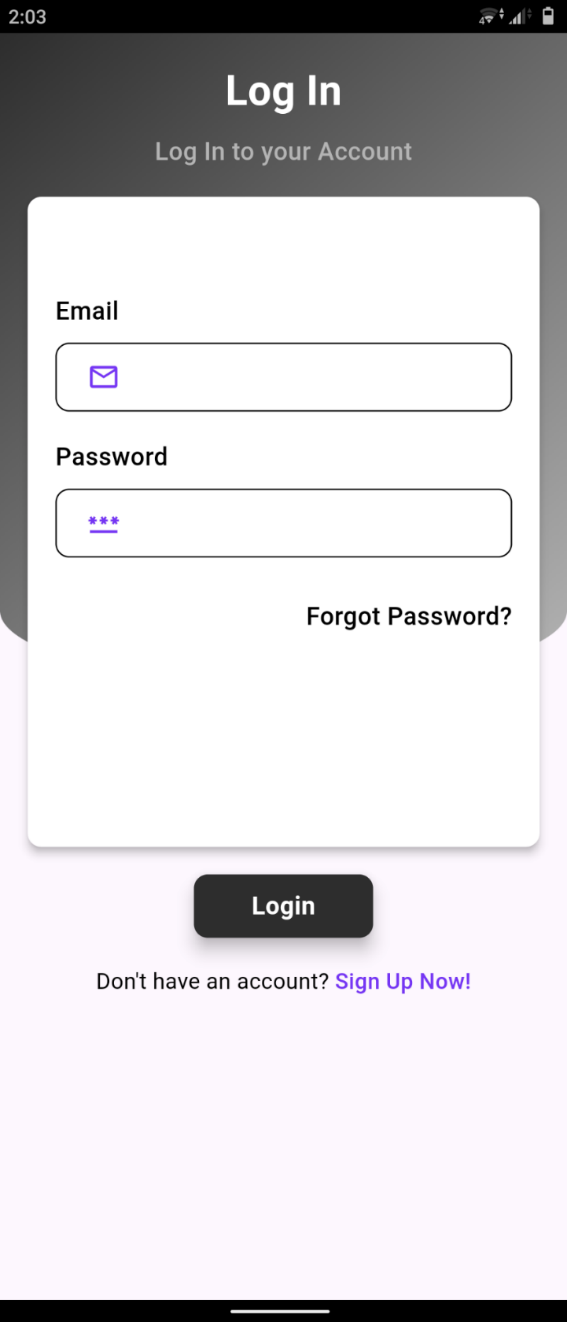


Figure 8.1 Login Screen

### 8.2 Sign Up

The sign-up screen for the Social Swap project offers a straightforward, inviting interface with fields for user information, a prominent "Sign Up" button, and social media registration options. The design is clean, secure, and optimized for various devices.

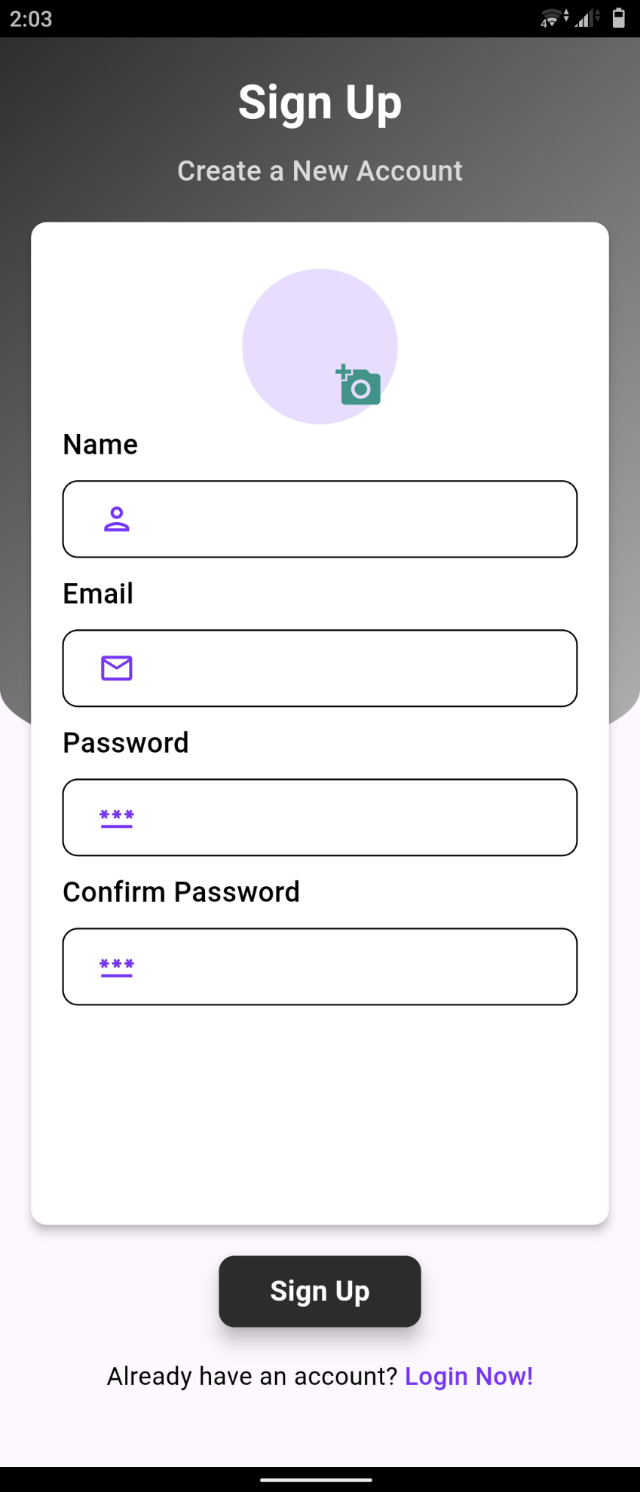


Figure 8.2 Sign Up Screen

### 8.3 Portfolio

The portfolio screen for the Social Swap project displays a comprehensive overview of the user's cryptocurrency assets. It includes detailed sections for current holdings, and total portfolio value. The interface is designed to be intuitive and informative, featuring interactive elements and real-time data updates, ensuring a seamless and engaging user experience across all devices.

A screenshot of a phone

Description automatically generated

Figure 8.3 Home Screen

### 8.4 News Feed

The news feed screen in the Social Swap project presents a dynamic, scrolling interface showcasing the latest updates and. The design ensures a visually appealing and responsive experience across devices.

A screenshot of a phone

Description automatically generated

Figure 8.4 Latest News

### 8.5 User Profile

The user profile screen in the Social Swap project features a detailed and customizable interface showcasing personal information, profile picture, and bio. The design is user-friendly and visually appealing, allowing for easy navigation and profile updates, ensuring an engaging and personalized user experience across devices.

A screenshot of a phone

Description automatically generated

Figure 8.5 User Profile

### 8.6 Create Wallet

The create wallet screen in the Social Swap project offers a straightforward and secure interface for users to set up a new cryptocurrency wallet. The design emphasizes security with clear instructions and visual indicators, ensuring a seamless and user-friendly process across all devices.

A screenshot of a computer

Description automatically generated

Figure 8.6 Create Wallet Screen

### 8.7 Chats

The chats screen in the Social Swap project features a clean, intuitive interface for real-time messaging. It includes a list of users who have sent or received messages from the current user. The design ensures seamless navigation, with contact lists, chat history, and notification indicators, providing an engaging and responsive user experience across devices.

A screenshot of a phone

Description automatically generated

Figure 8.7 Chat Screen

### 8.8 Market

The market screen in the Social Swap project provides users with real-time access to cryptocurrency market data and trends. It includes interactive graphs, price charts, and detailed coin information. The interface is designed for ease of use. This ensures users can stay informed and make informed trading decisions, fostering an engaging and responsive user experience across devices.

### 8.9 Predictions

Figure 8.9 Recommended Coins Screen

Figure 8.8 Coin Graph Screen

### A prediction model generates buy and sell signals on a graph, indicating optimal times to enter (buy) or exit (sell) positions based on market analysis and historical data patterns.

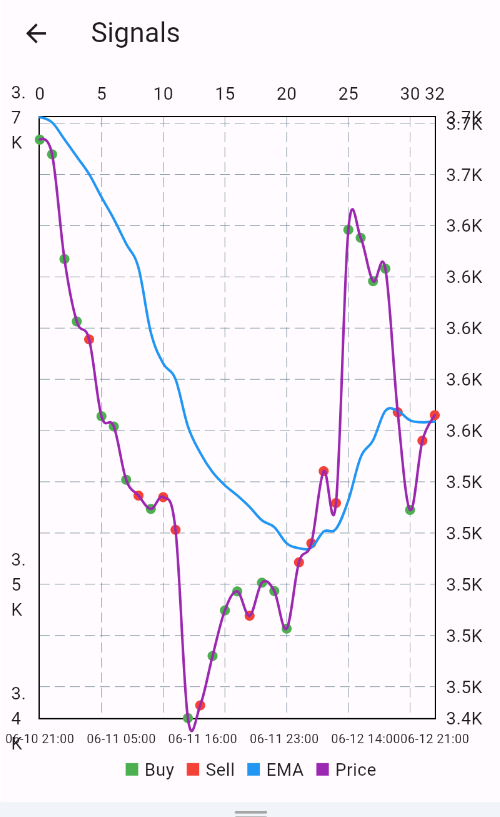


Figure 8.10Prediction Chart

# Conclusion

This project has successfully developed the Social Swap crypto trading app, providing a comprehensive and user-friendly platform for cryptocurrency trading. The system incorporates a robust backend implemented by utilizing advanced AI/ML models for accurate trade predictions and a secure and efficient wallet management system for seamless transactions.

The evaluation of the app's performance through rigorous system, black box, and unit testing demonstrates its effectiveness in providing a seamless crypto trading experience. While the system has certain limitations, such as its reliance on specific models for trade predictions and the absence of certain advanced features, future work can address these limitations and further enhance the system's capabilities. By exploring advanced AI/ML models, incorporating real-time market updates, and improving user interface and experience, the system can become more robust and reliable in facilitating cryptocurrency trading.

Overall, this project contributes to the ongoing efforts to make cryptocurrency trading more accessible and user-friendly. The Social Swap application provides a valuable tool for users to engage in cryptocurrency trading, manage their digital assets, and make informed trading decisions in the dynamic crypto market.

# References

|  |  |
| --- | --- |
| [1] | J. M. a. M. Saffell, "Reinforcement Learning for Optimized Trade Execution," 2010. |
| [2] | M. e. al, "Human-level control through deep reinforcement learning," 2015. |
| [3] | H. e. al, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," 2018. |
| [4] | L. e. al, "Exploring Reinforcement Learning Algorithms for Developing Adaptive Trading Bots," 2019. |
| [5] | Y. B. Z Shahbazi, "Improving the cryptocurrency price prediction performance based on reinforcement learning," 2021. |
| [6] | N. F. P. B. G Borrageiro, "The recurrent reinforcement learning crypto agent," 2022. |
| [7] | F. A.-S. Ali Al-Ameer, "A Methodology for Securities and Cryptocurrency Trading Using Exploratory Data Analysis and Artificial Intelligence," p. 4, 2021. |
| [8] | Management Engineering Department, College of Business, Korea Advanced Institute of Science and Technology, Seoul, Korea, "Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory," 2023. |
| [9] | H. Andey, "Deep Reinforcement Learning for Trading Cryptocurrencies," 2022. [Online]. |