An Industrial Oriented Major Project Report on

CRYPTOSWAP

Submitted in Partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

CH KEERTHANA	**20BD1A056F**
D SINDHUJA	**20BD1A056J**
D SAHITYA	**20BD1A057D**
V MEGHANA	**20BD1A057T**

Under the guidance of

Mr K Ramesh
Assistant Professor, Department of CSE

KESHAV
MEMORIAL INSTITUTE
OF TECHNOLOGY
POWERED BY GENESIS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH.
Narayanaguda, Hyderabad, Telangana-29

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that this is a bonafide record of the project report titled **"Cryptoswap"** which is being presented as the Industrial Oriented Mini Project report by

1. CH KEERTHANA	**20BD1A056F**
2. D SINDHUJA	**20BD1A056J**
3. D SAHITYA	**20BD1A057D**
4. V MEGHANA	**20BD1A057T**

In partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering affiliated to the Jawaharlal Nehru Technological University Hyderabad, Hyderabad

Faculty Supervisor
(Mr K Ramesh)

Head of Department
(Mr. P. Upendar)

Submitted for Viva Voce Examination held on ______________________________

External Examiner

Vision & Mission of KMIT

Vision of KMIT

- To be the fountainhead in producing highly skilled, globally competent engineers.
- Producing quality graduates trained in the latest software technologies and related tools and striving to make India a world leader in software products and services.

Mission of KMIT

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepares students to become successful professionals.
- To establish an industry institute Interaction to make students ready for the industry.
- To provide exposure to students on the latest hardware and software tools.
- To promote research-based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises.
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students.

Vision & Mission of CSE

Vision of the CSE

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the CSE

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities

PROGRAM OUTCOMES (POs)

PO1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution ofcomplex engineering problems.

PO2. Problem Analysis: Identify formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

PO3. Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct Investigations of Complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern Tool Usage: Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The Engineer and Society: Apply reasoning informed by contextual knowledge to societal, health, safety. Legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7. Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Information Technology solutions for social upliftments.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep learning, IOT, Data Science, Full stack development, Social Networks, Cyber Security, Mobile Apps, CRM, ERP, Big Data, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting to the changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

PROJECT OUTCOMES

P1: Accurately display the current prices of cryptocurrencies.

P2: Successfully swap the amount of cryptocurrencies mentioned, in the real blockchain network.

P3: Allows successful integration between flutter and metamask.

P4: Work seamlessly over the internet.

MAPPING PROJECT OUTCOMES WITH PROGRAM OUTCOMES

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1	2	3	3	1	3	-	-	-	3	3	1	2
P2	2	3	3	2	-	-	-	-	3	3	-	3
P3	2	3	3	-	2	-	-	-	3	3	-	1
P4	3	3	3	-	2	-	-	-	3	3	1	1

1 – LOW 2 –MEDIUM 3– HIGH

PROJECT OUTCOMES MAPPING WITH PROGRAM SPECIFIC OUTCOMES

PSO	PSO1	PSO2
P1	2	3
P2	3	3
P3	3	3
P4	1	-

PROJECT OUTCOMES MAPPING WITH PROGRAM EDUCATIONAL OBJECTIVES

PEO	PEO1	PEO2	PEO3	PEO4
P1	-	1	1	2
P2	3	3	2	3
P3	1	2	2	3
P4	2	2	-	-

DECLARATION

We hereby declare that the results embodied in the dissertation entitled “Crptoswap" has been carried out by us together during the academic year 2023-24 as a partial fulfillment of the award of the B.Tech degree in Computer Science and Engineering from JNTUH. We have not submitted this report to any other university or organization for the award of any other degree.

Student Name	**Roll no.**
CH KEERTHANA	**20BD1A056F**
D SINDHUJA	**20BD1A056J**
D SAHITYA	**20BD1A057D**
V MEGHANA	**20BD1A057T**

ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work. We render our thanks to **Dr. B L Malleswari**, Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Founder & Director, **Mr. S. Nitin,** Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Ms. Deepa Ganu**, Director Academic for providing an excellent environment in the college.

We are also thankful to **Mr. P Upendar**, Head of the Department for providing us with time to make this project a success within the given schedule.

We are also thankful to our Faculty Supervisor **Mr K Ramesh**, for her/his valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation during the project work.

Student Name	**Roll no.**
Ch Keerthana	**20BDA056F**
D Sindhuja	**20BD1A056J**
D Sahitya	**20BD1A057D**
V Meghana	**20BD1A057T**

ABSTRACT

The project is a dynamic fusion of Flutter's front-end development capabilities with the innovative world of blockchain technology. This integration creates a versatile, user-centric application that caters to a broad range of users. Flutter, as the front-end framework, provides a single codebase for multi-platform deployment, complete with a responsive and customizable user interface. The project not only embraces the hot reload feature for real-time updates but also leverages Flutter's widget-based architecture for aesthetic consistency. Meanwhile, blockchain technology enhances the application with secure and transparent interactions, real-time data, wallet management, smart contract execution, and decentralized applications. The synergy of Flutter and blockchain technology aims to offer users a seamless and intuitive experience, enabling them to interact with cryptocurrencies and blockchain networks with confidence and convenience. This project aspires to redefine user interaction with blockchain technology, presenting a gateway to a more accessible and secure digital financial landscape. User experience and adoption are also key areas of focus. The project aims to create an intuitive and user-friendly interface to drive broader adoption of cryptocurrency trading. By reducing barriers to entry and enhancing the overall user experience, Crypto Swap seeks to make cryptocurrency trading accessible to a wider audience. In summary, the Crypto Swap project is built upon a robust foundation of research and literature survey that covers a wide spectrum of topics. It offers a secure, user-centric, and decentralized solution for cryptocurrency exchange, empowering users to navigate the ever-changing cryptocurrency market with confidence and ease.

LIST OF FIGURES

CONTENTS

DESCRIPTION **PAGE**

CHAPTER - 1

CHAPTER – 2

CHAPTER - 3

CHAPTER – 4

1. INTRODUCTION

1.1 Purpose of the project:

In an era defined by technological innovation and decentralized financial systems, the fusion of Flutter's front-end development prowess with the transformative potential of blockchain technology marks a significant leap forward in the world of application development. This project introduces an advanced application that capitalizes on Flutter's capabilities to deliver a streamlined, cross-platform user interface, while seamlessly integrating blockchain technology to offer users a comprehensive and secure digital experience. The objective of this project is to leverage Flutter's versatile front-end capabilities to deliver a user-friendly and polished interface, all while seamlessly connecting to blockchain technology for real-world applications. Through this integration, users can access the benefits of blockchain, such as security, transparency, and financial inclusivity, with ease and confidence. By the end of this project, users will have at their disposal a powerful and user-centric application that simplifies their interaction with cryptocurrencies and blockchain networks, ushering in a new era of decentralized financial experiences.

The cryptocurrency market has undergone substantial growth and evolution since the inception of Bitcoin in 2009. With thousands of cryptocurrencies in existence, a flourishing decentralized finance (DeFi) ecosystem, and an expanding user base, the need for efficient and secure cryptocurrency swapping mechanisms has become increasingly evident. Traditional centralized exchanges have faced issues related to security breaches, custody risks, and counterparty reliance. To address these concerns, decentralized solutions have gained prominence. Metamask, an Ethereum-based wallet, plays a pivotal role in DeFi by granting users control over their digital assets and facilitating interactions with DeFi applications. Leveraging Metamask's capabilities, Crypto Swap aims to provide a secure and user-friendly environment for peer-to-peer cryptocurrency swaps. Furthermore, the cryptocurrency market is renowned for its volatility, making real-time price information a necessity for informed decision-making. Crypto Swap responds to this need by offering up-to-the-minute pricing data, particularly for leading cryptocurrencies like Bitcoin and Ethereum.

based on blockchain technology, provided an alternative to traditional financial systems.

1. **Decentralized Finance (DeFi):**

The concept of DeFi emerged as a decentralized, blockchain-based financial ecosystem that seeks to replace traditional intermediaries, such as banks and financial institutions, with smart contracts and decentralized applications (dApps).

2. **Proliferation of Cryptocurrency Exchanges:**

As cryptocurrencies gained popularity, a multitude of cryptocurrency exchanges emerged to facilitate the buying, selling, and trading of digital assets. These exchanges came in various forms, including centralized and decentralized platforms.

3. **Security and Custody Concerns:**

Centralized exchanges faced security challenges, with several high-profile hacks and breaches, raising concerns about the safety of users' funds. The need for secure and non-custodial solutions became apparent.

4. **Role of Metamask:**

Metamask, an Ethereum-based wallet, played a critical role in the DeFi ecosystem by allowing users to manage their digital assets securely. Its wallet and browser extension provided a user-friendly gateway to interact with DeFi applications.

5. **Real-Time Price Information:**

In the volatile cryptocurrency market, access to real-time price data for major cryptocurrencies like Bitcoin and Ethereum became essential. Users needed to make timely and informed trading decisions.

The background for the project stems from the evolving landscape of mobile application development and the ever-growing significance of blockchain technology:

Flutter's Emergence: Flutter, an open-source UI framework developed by Google, has gained prominence for its ability to streamline the development of cross-platform mobile applications. Its single codebase approach, widget-based architecture, and hot reload feature have made it a preferred choice for developers.

Blockchain Revolution: Blockchain technology has transformed various industries, offering secure, decentralized, and transparent solutions. It underpins cryptocurrencies like Bitcoin and Ethereum and enables smart contracts, decentralized applications (dApps), and secure digital asset management.

User-Centric Approach: User experience is at the forefront of application design. Flutter's capacity for creating attractive and responsive user interfaces aligns with the growing demand for user-centric applications.

1.2 Problem with Existing System:

In the world of cryptocurrency trading and decentralized finance, the Crypto Swap project confronts the challenges of security breaches, user trust, complex user experiences, the absence of real-time price data, regulatory uncertainties, intermediary reliance, privacy and data security concerns, and the need for user-friendly access to DeFi. The project aims to resolve these issues by providing a secure and intuitive non-custodial platform that utilizes Metamask wallet accounts, while also delivering real-time pricing information for leading cryptocurrencies, empowering users with control and trust in their cryptocurrency interactions. The Crypto Swap project aims to address these multifaceted issues by providing a secure, intuitive, and transparent platform that leverages Metamask wallet accounts for cryptocurrency swapping. It also endeavors to offer real-time pricing information for major cryptocurrencies, thereby enhancing user decision-making capabilities.

1.3 Proposed System

In doing so, the project aspires to empower users to confidently and conveniently navigate the world of cryptocurrency trading and DeFi, all while championing decentralization and trustless transactions.

1. **Enhance Security:**

Ensure a secure and trustless environment for cryptocurrency swapping, reducing vulnerabilities and risks associated with centralized exchanges.

2. **Simplify User Experience:**

Create an intuitive and user-friendly interface to streamline cryptocurrency trading, making it accessible to both experienced traders and newcomers.

3. **Provide Real-Time Data:**

Offer users access to up-to-the-minute pricing data for major cryptocurrencies, particularly Bitcoin and Ethereum, to enable informed decision-making.

4. **Enhance Privacy and Data Security:**

Implement robust privacy measures to safeguard user information and assets in a data-sensitive environment.

5. **Empower Users:**

Ultimately, empower users to confidently engage with cryptocurrencies, DeFi, and blockchain networks in a secure, transparent, and user-centric manner.

1.4 Scope of the Project:

The project's scope encompasses several key aspects and functionalities:

1. **User-Friendly Interface:**

The project will develop a user-friendly and intuitive interface for cryptocurrency swapping, catering to both novice and experienced traders.

2. **Secure Wallet Integration:**

Metamask wallet accounts will be integrated to provide users with a secure and non-custodial solution for managing their digital assets.

3. **Blockchain Interaction:**

Users will be able to interact with blockchain networks securely, conduct peer-to-peer transactions, and access decentralized applications (dApps).

4. **Smart Contract Support:**

The project will facilitate smart contract execution, allowing users to engage with decentralized applications and automated agreements.

5. **Compliance and Regulatory Alignment:**

The platform will navigate evolving cryptocurrency regulations to ensure legal compliance and user protection.

6. **Enhanced Security Measures:**

Robust security protocols will be implemented to protect user information and assets, addressing privacy and data security concerns.

7. **Non-Custodial Solutions:**

Users will have control over their digital assets without relying on third-party intermediaries, aligning with the principles of trustless transactions.

Limitations:

1. Dependency on External Factors:

The project relies on external data sources for real-time cryptocurrency pricing, and the accuracy and availability of this data are subject to external factors.

2. Regulatory Uncertainty:

Evolving cryptocurrency and DeFi regulations can impact the project's operations and may require continuous adaptation to remain compliant.

3. Security Risks:

While the project aims to provide a secure environment, it may still be susceptible to unforeseen security vulnerabilities and attacks, necessitating ongoing security enhancements.

4. Metamask Integration:

Dependency on Metamask wallet accounts means that any issues or updates related to Metamask may affect the project's functionality.

5. **User Experience Variability:**

The user experience may vary depending on the users' familiarity with cryptocurrency and blockchain technology, potentially creating a learning curve for newcomers.

6. **Cryptocurrency Market Volatility:**

The project provides real-time pricing data, but it cannot control or mitigate the inherent volatility of the cryptocurrency market, which can result in financial risks for users.

7. **Data Privacy:**

Despite robust security measures, no system is entirely immune to privacy breaches, and the project may face privacy concerns and potential data breaches.

8. **User Adoption Challenges:**

Encouraging users to transition from traditional centralized exchanges to a decentralized and non-custodial platform may pose adoption challenges.

9. **Scalability:**

As the user base grows, the project may encounter scalability issues that require ongoing optimization to maintain performance.

10. **Emerging Technologies:**

The project's technology stack may become outdated as new blockchain and cryptocurrency technologies emerge, necessitating updates and adaptations.

11. **Resource Limitations:**

The project's success and scalability may be limited by available resources, both in terms of development and financial capabilities.

It's essential for the Crypto Swap project to acknowledge and address these limitations to mitigate risks and continue improving the platform's functionality, security, and user experience.

1.5 Organization of the document

Here's a step-by-step organization for the development of the Crypto Swap project:

1. **Project Initiation:**
   - Define the project's objectives, goals, and scope.
   - Assemble the project team, designate roles, and establish communication channels.
   - Develop a project plan, including timelines, milestones, and deliverables.
2. **Research and Analysis:**
   - Conduct a thorough analysis of the cryptocurrency and DeFi landscape.
   - Identify the specific challenges and opportunities in the market.
   - Research Metamask wallet integration and real-time pricing data sources.

3. **User Interface Design:**
   - Create wireframes and mockups for the platform's user interface.
   - Incorporate user feedback to refine the design for a user-friendly experience.
   - Develop the visual design and branding elements.
4. **Architecture and Technology Selection:**
   - Choose the technology stack for the project, including front-end and back-end components.
   - Design the technical architecture, outlining how Metamask and blockchain integration will work.
   - Ensure scalability and security considerations are integrated into the architecture.
5. **Development:**
   - Begin the development process, starting with the creation of a secure and intuitive user interface.
   - Implement the Metamask integration, allowing users to connect their wallets securely.
   - Develop the functionality for real-time pricing data retrieval and display.
6. **Security Implementation:**
   - Implement robust security protocols to protect user data and assets.
   - Conduct security testing to identify and address vulnerabilities.
   - Ensure compliance with industry best practices for secure cryptocurrency handling.
7. **Regulatory Compliance:**
   - Stay updated on evolving cryptocurrency regulations and compliance requirements.
   - Implement necessary measures to meet legal standards while ensuring user privacy and security.
8. **Testing and Quality Assurance:**
   - Conduct extensive testing to verify the functionality, security, and user experience.
   - Perform unit testing, integration testing, and user acceptance testing.
   - Address and resolve any identified issues or bugs.
9. **User Testing:**
   - Recruit a group of users to participate in beta testing.
   - Collect user feedback and make necessary adjustments to the platform based on their input.
   - Ensure the platform is user-centric and intuitive.

10. Deployment:

- Prepare the platform for production deployment.
- Set up hosting, server infrastructure, and necessary environment configurations.
- Perform final testing in the live environment.

2. LITERATURE SURVEY

1. Smart Contracts and Ethereum (2015):

The advent of Ethereum in 2015, co-founded by Vitalik Buterin, brought about a transformative shift in the cryptocurrency realm by introducing the groundbreaking concept of smart contracts. Unlike traditional contracts, these self-executing agreements are coded directly into the blockchain, automating and enforcing contractual terms. Ethereum's innovation extended beyond mere cryptocurrency, enabling the creation of decentralized applications (dApps) through these smart contracts. Vital to this evolution was the Ethereum Virtual Machine (EVM), providing a standardized runtime environment. The platform's flexibility also facilitated tokenization, giving rise to Initial Coin Offerings (ICOs) and a surge in blockchain-based fundraising. Ethereum's impact transcended financial applications, inspiring developers to explore diverse use cases.

Despite challenges like scalability issues, Ethereum remains at the forefront of blockchain development, undergoing upgrades to address concerns and maintain its influential role in shaping decentralized technologies. Its success has not only spurred further innovation in the blockchain space but has also influenced the creation of various other blockchain platforms, solidifying Ethereum's enduring significance in the evolution of decentralized systems.

2. Cross-Chain Atomic Swaps (2017 and onwards):

Cross-Chain Atomic Swaps, introduced in 2017 and onwards, revolutionize blockchain technology by enabling the direct exchange of different cryptocurrencies across diverse blockchains. These swaps, facilitated by smart contracts, are atomic, ensuring either the complete execution or none at all. Overcoming the interoperability challenge in the cryptocurrency landscape, Cross-Chain Atomic Swaps operate in a trustless and decentralized manner, eliminating the need for intermediaries. This innovation allows users to exchange assets directly, enhancing security and reducing counterparty risk. Despite initial challenges, ongoing developments and the integration of protocols like the Lightning Network demonstrate the growing interest and potential of Cross-Chain Atomic Swaps, contributing to the vision of a decentralized and interconnected blockchain ecosystem.

3. Security Insights - Unraveling SegWit Attack Vectors in Atomic Cross-Chain Swaps (2019):

Théberge and O'Connor's 2019 research paper, "Atomic Cross-Chain Swaps: A SegWit Attack Vector," significantly advances the academic understanding of security in atomic swaps, focusing on Segregated Witness (SegWit) implications. The paper identifies potential attack vectors related to SegWit, shedding light on security challenges in cross-chain swaps. Offering crucial insights for developers and security experts, it informs the design of robust protocols, enhancing the overall security of atomic swaps. The findings contribute to ongoing efforts to fortify blockchain security as atomic swaps become integral, influencing industry best practices and shaping standards. The collaborative nature of their research underscores the community-driven approach to addressing security concerns in emerging blockchain technologies.

3. SYSTEM REQUIREMENT SPECIFICATIONS

3.1 Introduction to SRS

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

Problem Requirement/Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

3.2 Role of SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

3.3 Requirement Specification Document:

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. A software document is primarily prepared for a project, software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

The purpose of SRS (Software Requirement Specification) document is to describe the external behavior of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirement of the application or software. The complete software requirements for the system are captured by the SRS. This section introduces the requirement specification document for Word Building Game using Alexa which enlists functional as well as non-functional requirements.

3.4 Functional Requirement Specification

The System after careful analysis has been identified to be present with the following modules. A functional requirement defines a function of a system or its components. Functional requirement may be calculation, technical details, data manipulation and processing and other specific functionality that defines what a system is supposed to accomplish the functional requirement specification documents the operation and activities that a system able to perform. Functional requirements include functions performed by specific screens, outlines of work flows performed by the system, and other business compliance requirements the system must meet. This project has four modules.

3.5 Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use. The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance. In this project, Data publisher (or data holder, who collects data from record owner ex. Alice and bob) and data miner or the public, called the data recipient and record owners like patients and doctors.

Modifiability

Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).

Portability

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

3.6 Non Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting System Non-functional requirements place restrictions on the product being developed, the development process, and specify external constraints that the product must meet. Examples of NFR include safety, security, usability, reliability and performance Requirements. Project management issues (costs, time, and schedule) are often considered as non-functional requirements.

Reliability

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

Security

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time) to break into the system. Do not discuss solutions (e.g. passwords) in a requirements document

Usability

Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

Legal

There may be legal issues involving privacy of information, intellectual property rights, export of restricted technologies, etc.

3.7 Hardware Requirements

Mobile:

A modern mobile with an android or IOS system is the primary hardware requirement for development. Ensure it meets the recommended system requirements for software tools.

Processor and RAM:

A multicore processor and at least 6GB of RAM are recommended to handle development tasks effectively and run your application locally.

Internet Connection

A stable and fast internet connection is necessary for updating code, installing packages, and deploying crypto swap application. A reliable internet connection ensures efficient development.

3.8 Software Requirements

Code Editor (VS Code):

Visual Studio Code (VS Code) is a free, open-source code editor with a wide range of extensions, making it a popular choice for web development. It provides a seamless coding experience for creating our application.

Version Control (Git):

Git is an essential tool for version control and collaborative development. It allows you to track changes in your codebase, collaborate with other developers, and manage your project's code repository on platforms like GitHub.

Smart Contracts and Solidity:

Smart Contracts are self-executing contracts with predefined rules and automated actions, often deployed on blockchain networks. And solidity is a high-level programming language used to write these smart contracts for Ethereum and other compatible blockchains. Smart contracts play a major role in the backend.

Metamask (mobile application):

An external metamask mobile application is important because to get the metamask wallet accounts and the token accounts, we launch that application from our crypto swap application in a very efficient way.

Flutter:

Flutter is the frontend for our mobile application. It provides an effortless and enhancive user interface for the users.

4. SYSTEM DESIGN

4.1 Introduction to UML

The unified modeling language (UML) is a general-purpose visual modeling language that is intended to provide a standard way to visualize the design of a system. UML provides a standard notation for many types of diagrams which can be roughly divided into three main groups: behavior diagrams, interaction diagrams, and structure diagrams. The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design.

4.2 UML Diagrams

UML diagrams illustrate the quantifiable aspects of a system that can be described visually, such as relationships, behavior, structure, and functionality. In a UML diagram, the diagram elements visually represent the classifiers in a system or application. These classifiers are the diagrammatic representation of a source element. UML diagrams provide views of source elements; however, diagram elements do not have semantic value.

4.2.1 Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior. To clarify a bit in details, dynamic behavior means the behavior of the system when it is running/operating. So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So, use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So, to model the entire system numbers of use case diagrams are used. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Show the interacting among the requirements are actors.Show the interacting among the requirements are actors.

So, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. In brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.

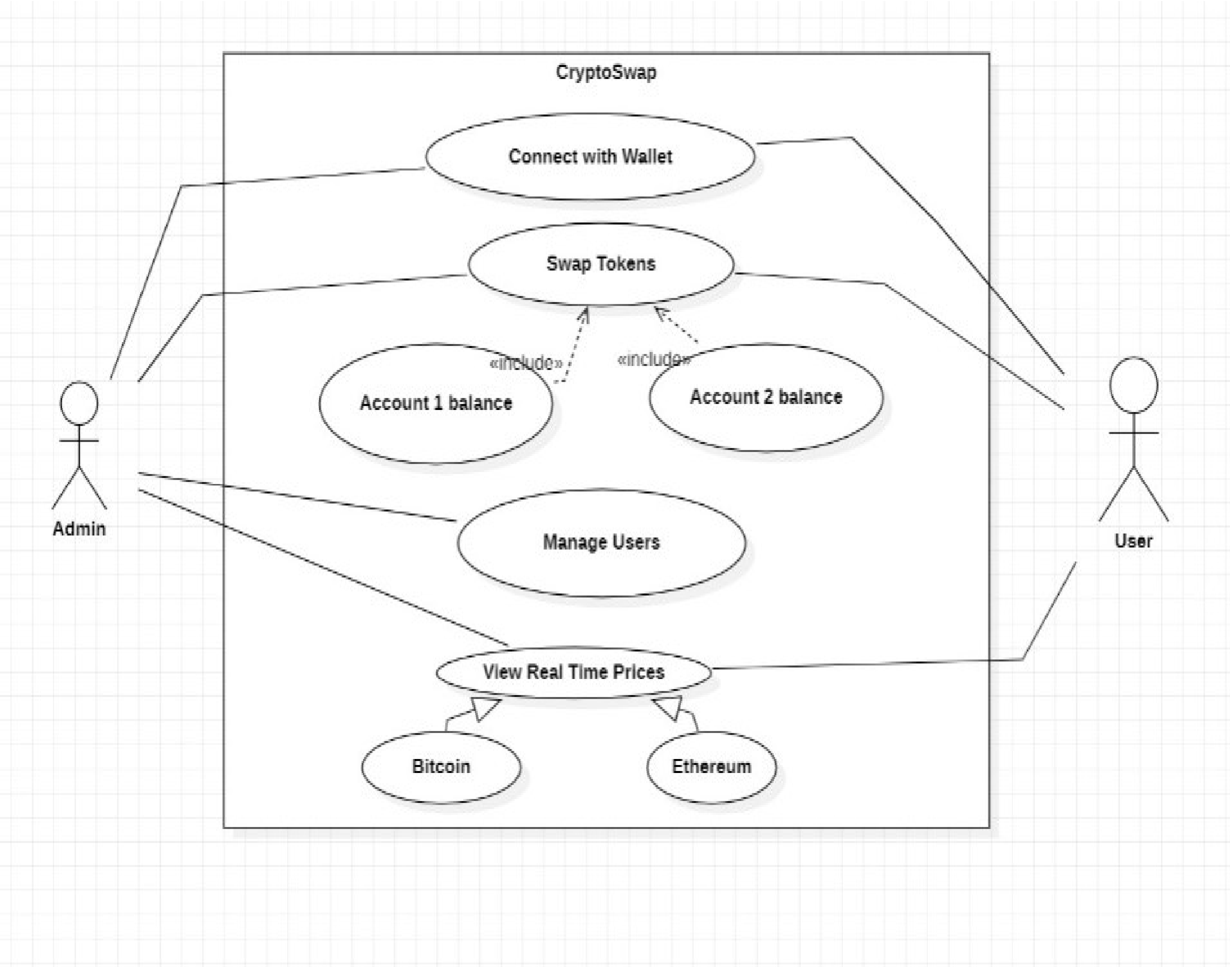

Fig 4.2.1 UML Use Case Diagram

4.2.2 State-Chart Diagram

A state chart UML diagram is a type of UML diagram that specifically represents the dynamic behaviour of a system using states, transitions, and events.

Here's a brief overview:

- **States:** Represent different conditions or modes that an object or system can be in.
- **Transitions:** Arrows between states, showing the events or conditions that trigger a change from one state to another.
- **Events:** Triggers for state transitions.
- **Actions:** Activities or behaviors associated with transitions.
- **Initial State:** Represents the starting point of the system.
- **Final State:** Indicates the end of the system's lifecycle.

The Statechart UML diagram is particularly useful for modeling complex systems with dynamic behavior, such as the various states a process goes through or the modes of operation for a software component. It enhances understanding, communication, and design of systems that involve state changes over time.

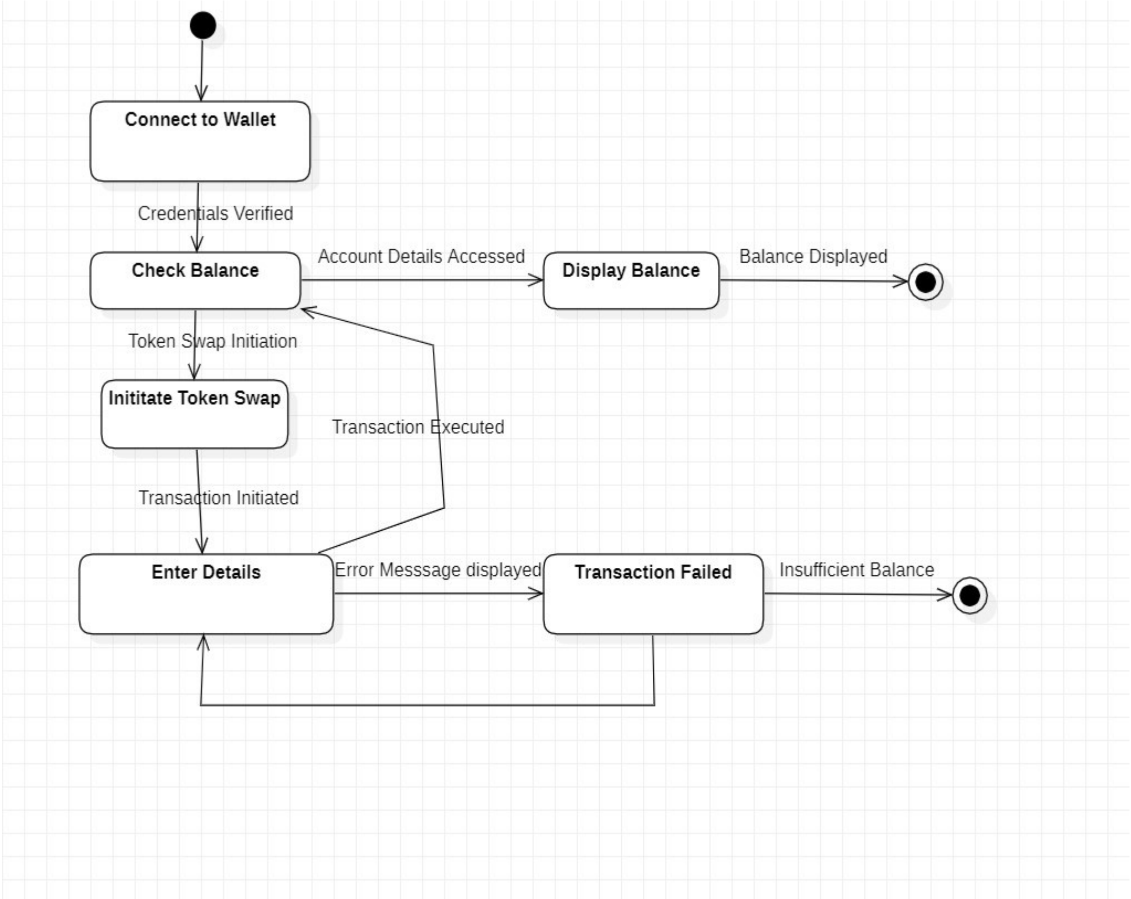

Fig 4.2.2 UML State Chart Diagram

4.2.3 Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of Sequence Diagram

- To model high-level interaction among active objects within a system.
- To model interaction among objects inside a collaboration realizing a use case.
- It either models generic interactions or some certain instances of interaction.

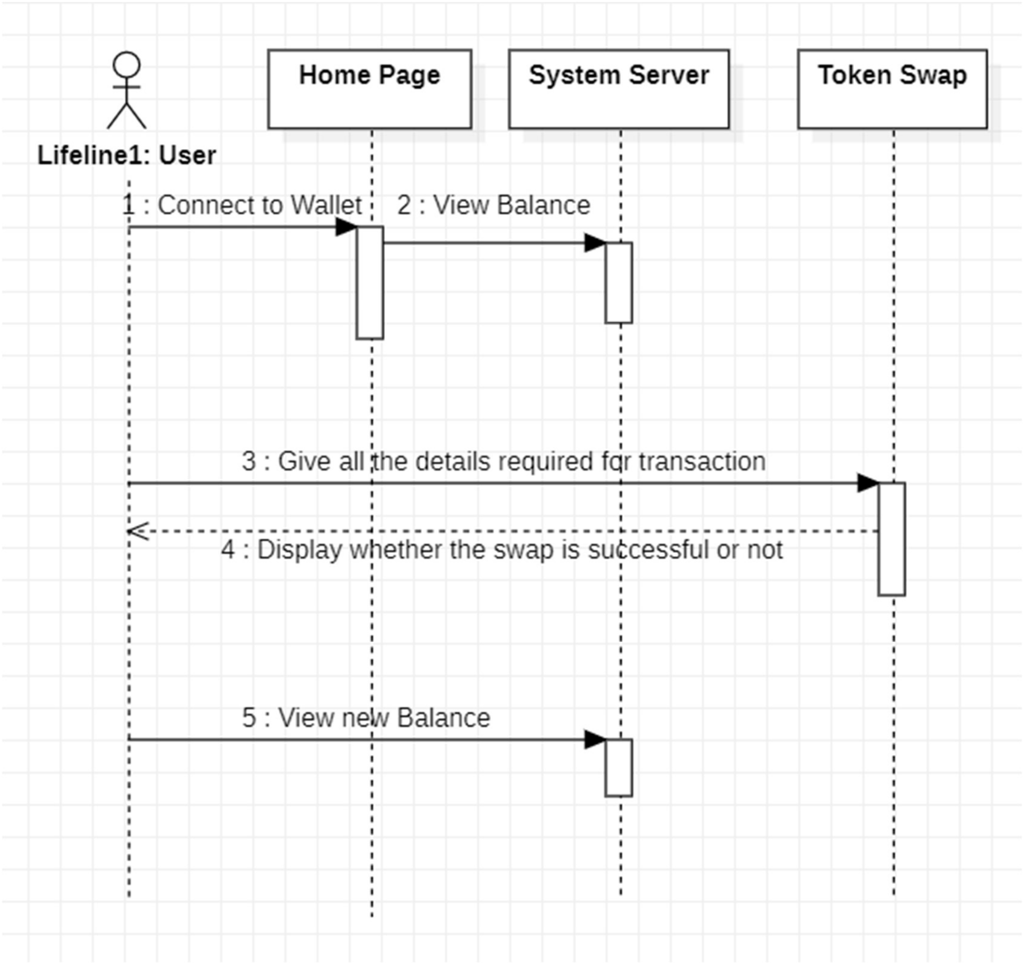

Fig 4.2.3 UML Sequence Diagram

4.2 Technologies Used

Front-End:

- Flutter

Back-End:

- BlockChain
- Solidity