BCrypty: Secure Swap Blockchain Bridge

A project submitted

in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science

by

Rayyan Waseem (2K20-BSCS-202)

Muhammad Abdullah (2K20-BSCS-236)

Supervised By

Sir Kamran Abid

DEPARTMENT OF COMPUTER SCIENCE

NFC Institute of Engineering & Technology, Multan

ACKNOWLEDGEMENTS

Starting with the name of Allah who is the most merciful and powerful of all. All admiration to Allah Almighty Lord of the entire world, the most beneficent the most merciful, owner of the Day of Judgment. We ask Allah to bestow his blessing and salutations of piece upon our noble Prophet Muhammad (PBUH). We pay our humblest gratitude to Allah to who bestowed upon his blessing which guided and helped us to complete this project report. With the thanks providence that we have come to the other corner of the knowledge and even then, the point of satiation never comes to end. We extend our heartfelt gratitude to our advisors Sir Ahtesham and Sir Abdul Qadeer for granting us the opportunity to design and develop the "BCrypty: Web Application and Mobile Application for the Computer Science Department" and for guiding us until its completion. Our sincere thanks also go to the faculty members and administrators of the student feedback system at NFC IET for their ongoing support, insightful suggestions, and crucial guidance without which this project would not have been possible.

We are grateful for the unwavering support, uplifting demeanor, and consistent inspiration provided by Prof. Dr. Akhter Ali (Vice Chancellor of NFC IET) and Dr. Naeem Aslam (Head of the Computer Science Department at NFC IET). Their continuous pursuit of excellence in all aspects of NFC IET and their selfless motivation has constantly propelled us forward.

UNDERTAKING

This is to inform you that the project entitled “**BCrypty: Secure Swap Blockchain Bridge**” is an authentic group effort done by undersigned group members. This project is a requirement for the completion of the degree “BSCS (Bachelor of Science in Computer Science)” at Computer Science department, NFC Institute of Engineering and Technology, Multan, Pakistan. This is to inform you that the project titled "**BCrypty**" is a genuine collaborative effort by the undersigned group members. It serves as a requirement for earning the **BSCS** (Bachelor of Science in Computer Science) degree at the Computer Science Department of NFC Institute of Engineering and Technology, Multan, Pakistan.

The undersigned members have fully carried out the requirements, design, and development of the project. It should be noted that this project is the result of daily collaborative effort from all group members. The project is original and has not been previously submitted to any other institution of higher learning.

ABSTRACT

BCrypty is an innovative project at the forefront of Web 3.0, introducing the Secure Swap Blockchain Bridge to facilitate secure and efficient cryptocurrency transactions. This groundbreaking solution addresses the challenges inherent in cross-chain transactions, offering users a seamless experience while ensuring the integrity and confidentiality of their assets. Through advanced cryptographic techniques and decentralized protocols, BCrypty enables secure transfers across different blockchain networks. Its user-friendly interface simplifies the process for both novice and experienced users, making cryptocurrency management accessible to all. With robust account management features, such as multi-signature wallets and customizable security settings, users have full control over their digital assets. BCrypty operates on decentralized governance principles, allowing the community to participate in decision-making processes and shaping the platform's future. By combining cutting-edge technology with user-centric design, BCrypty is revolutionizing how people transact and manage their cryptocurrency holdings. Its emphasis on security, interoperability, and usability positions BCrypty as a key player in the evolution of finance in the digital age.

FINAL YEAR PROJECT UNDERTAKING FORM

(NFC IET, Multan)

We hereby affirm that the originality and authenticity of the Final Year Project to be undertaken will be upheld. The report and/or the system that we submit after the Final Year Project will be the result of our investigations and efforts. We understand that cheating and plagiarism constitute a serious violation of the university regulations, which will not only result in a failing grade for the Final Year Project but subject us to further disciplinary actions.

Signature of Student 1:

_____

Name: Rayyan Waseem
Roll No: (2K20-BSCS-202)
Date: 21 July 2024.

Signature of Student 2:

_____

Name: Muhammad Abdullah
Roll No: (2K20-BSCS-236)
Date: 21 July 2024.

Discipline: **BSCS**
Year of Study: **2020-2024**
Area of Study: **Computer Science**
Proposed Project: Title: **BCrypty**

Sir Naeem Aslam

_____

Head of Department
Computer Science

_____

External Examiner

Sir Kamran Abid

_____

Supervisor


BCRYPTY

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

INTRODUCTION

In the realm of financial transactions, many assume that sending and receiving money is a straightforward process. However, establishing a reliable platform for such transactions requires meticulous effort and attention to detail. Success in this arena is not guaranteed, especially as users become increasingly discerning about the services they trust with their financial assets. Amidst the global landscape, characterized by the Covid-19 pandemic, the need for efficient and secure money transfer solutions has become more pronounced. From small-scale transactions to large-scale fund transfers, there's a growing demand for platforms that offer seamless and reliable money transfer services.

BCrypty introduces a revolutionary platform: a Trusted Peer-to-Peer Blockchain-Based Decentralized Autonomous Organization (DAO) Management system, accessible via web and mobile applications. In an era where banking is ubiquitous yet demands greater reliability, BCrypty stands out as the premier solution for securely and swift money transfers.

1.1 Domain (De-Fi)

Decentralized Finance, commonly referred to as De-Fi, is a novel approach to financial services that operates on decentralized networks, typically built on blockchain technology. Unlike traditional finance, which relies heavily on intermediaries such as banks, De-Fi seeks to democratize access to financial services by eliminating intermediaries and allowing direct peer-to-peer transactions.

In the De-Fi domain, various financial services, including lending, borrowing, trading, asset management, and more, are conducted through decentralized protocols and applications. These protocols and applications are typically open-source and built on blockchain platforms like Ethereum, Binance Smart Chain, or others.

De-Fi platforms offer several advantages over traditional financial systems, including:

1. Accessibility: Anyone with an internet connection and a compatible digital wallet can access De-Fi services, regardless of geographic location or socioeconomic status.

2. Transparency: Transactions on De-Fi platforms are recorded on public blockchains,

providing transparency and immutability.

3. Security: De-Fi platforms leverage the security features of blockchain technology, such as cryptographic encryption and decentralized consensus mechanisms, to protect user funds and data.

4. Interoperability: De-Fi protocols are often interoperable, meaning that users can seamlessly interact with multiple protocols and applications within the De-Fi ecosystem.

5. Programmability: Smart contracts, self-executing contracts with the terms of the agreement directly written into code, enable programmable financial transactions on De-Fi platforms, automating processes and reducing the need for intermediaries.

Overall, the De-Fi domain represents a paradigm shift in the way financial services are accessed, enabling greater financial inclusion, innovation, and autonomy for users worldwide.

1.2 Problem Statement

The major problem with the current banking platforms that we wanted to solve was:

Impediments to DeFi and Blockchain Advancement

The growth of decentralized finance (DeFi) platforms and blockchain applications is hampered by several key issues. High gas fees on the Ethereum blockchain restrict affordability and accessibility, while regulatory uncertainty around token classification limits innovation on crowdfunding platforms. Centralization risks in governance and token distribution threaten the integrity of decentralization efforts, and security vulnerabilities in smart contracts expose users to financial risks, undermining trust in DeFi systems.

1.3 Motivation

Blockchain could influence transaction processes by cutting down the processing fees.

- BCrypty's blockchain platform slashes transaction fees for money transfers, making them more affordable and accessible.

- BCrypty guarantees secure and transparent money transfers through blockchain technology, providing users with peace of mind and confidence in their transactions.

1.4 Definition of Terms

Terms used in our project are the following:

De-Fi (Decentralized Finance):

De-Fi refers to a system of financial services built on blockchain technology that eliminates intermediaries, providing decentralized access to financial products such as lending, borrowing, trading, and more.

Blockchain:

A blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without altering all subsequent blocks and the network's consensus.

1.5 Goal of our Project

- Foster financial inclusion through decentralized finance (De-Fi) solutions.
- Ensure trust and transparency in financial transactions using blockchain technology.
- Reduce transaction costs by leveraging decentralized networks.
- Drive innovation in finance by merging De-Fi, and Decentralized Governance.

CHAPTER 2: RELATED WORK

2.1 Related Work

- Integration of De-Fi and Money Transfers: This involves developing De-Fi solutions that leverage blockchain technology to streamline and secure money transfers, providing users with efficient and transparent ways to send and receive funds globally.

- Enhanced Security and Anonymity: This involves implementing robust security measures and privacy features within the platform to ensure the anonymity of users' transactions while maintaining the integrity of the blockchain network.

- Decentralized Financial Management: This involves creating tools and protocols for decentralized financial management, allowing users to manage their funds autonomously and securely without relying on centralized financial institutions.

- Tokenized Ownership and Rewards: This involves designing token models and reward systems that incentivize users to participate in the platform, offering them ownership rights and rewards in the form of tokens for their contributions and support.

Other Websites

Coinbase	coinbase.com
Uniswap	uniswap.org
MakerDAO	makerdao.com
Compound Finance	compound.finance
Indiegogo	indiegogo.com

Table 2.1: Relative Websites

Existing Problems

- High gas fees on Ethereum blockchain hinder affordability and accessibility for users of platforms like Compound Finance and Uniswap.
- Regulatory uncertainty regarding token classification limits innovation and project diversity on crowdfunding platforms such as Kickstarter and Indiegogo.
- Centralization risks in governance and token distribution pose challenges to decentralization efforts on platforms like MakerDAO and Compound Finance.
- Security vulnerabilities in smart contracts and protocols undermine trust and expose users to financial risks on De-Fi platforms like Uniswap and Compound Finance.

Solution Available in our Project

- Integrate layer 2 scaling solutions to alleviate high gas fees and enhance transaction throughput for users.
- Engage legal professionals to navigate regulatory challenges and ensure compliance with evolving legal frameworks.
- Establish decentralized governance structures, such as DAOs, to foster transparency, inclusivity, and community-driven decision-making.
- Implement advanced security measures, including regular audits and multi-signature authentication, to fortify the platform against potential security threats.
- Develop user-friendly interfaces and educational resources to empower users with the knowledge and tools needed to navigate the platform effectively and securely.

2.2 System Description

System Requirements Specification

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may include a set of use cases that describe user interactions that the software must provide.

System Hardware & Software Requirements

For users to access our Web App, the following hardware specifications should be sufficient:

Hardware	Description
Processor	Dual-core or higher
RAM	4 GB or higher
Storage	128 GB or higher; SSD preferred
Network	High-speed internet connection

Table 2.2: Hardware Requirements

It is also important for users to have a device that is secure and capable of running a blockchain wallet or client, such as a laptop or desktop computer, to securely store and manage their digital assets. Mobile devices, such as smartphones and tablets, can also be used but may have limitations in terms of security and functionality.

Software	Description
Operating System	Windows
Server/web technology	Web3.js, Solidity, Metamask
Languages	JavaScript, React, Vite JS
Tools	WebStorm, Canva, Draw.io

Table 2.3: Software Requirements

Tech Stack

In order to achieve the solution, we have chosen a tech stack that is

- Optimized for speed
- Efficient
- Secure

Following are the list of all tools and technologies we have used for this Application:

- Web3.JS
- Vite.JS
- ReactJS
- MetaMask
- Solidity
- Hardhat

Project Objectives

Requirement #: R1 **Requirement Type:** Functional
Description: The platform shall allow users to connect their MetaMask wallet for secure transactions.
Rationale: To provide a secure and convenient way for users to perform blockchain transactions on the platform.
Originator: uTecho
Fit Criterion: Users can connect their MetaMask wallet and perform transactions without any issues.
Customer Satisfactions: 7 **Customer Dissatisfactions:** 2
Priority: 3 **Dependencies:** **Conflicts:**

Supporting Materials: None
History: Created on 19 September, 2023.

utecho

Fig 2.1: Allow user to Connect MetaMask

Requirement #: R2 **Requirement Type:** Non-functional
Description: The platform shall allow users to see their transactions history of connected account
Rationale: To provide a convenient way for users to check their recent transactions and verify them on the platform.
Originator: uTecho
Fit Criterion: Users can check transaction history without needing to navigate through Metamask.
Customer Satisfactions: 8 **Customer Dissatisfactions:** 2
Priority: 5 **Dependencies: R1** **Conflicts:**

Supporting Materials: None
History: Created on 1 October, 2023.

utecho

Fig 2.2: Allow user to see Transaction History

Requirement #: R3 **Requirement Type:** Non-functional
Description: The platform shall feature a wallet card to provide a quick overview of the connected wallet and the currency selected.
Rationale: To provide a convenient way for users to know about the account that's currently in use by the platform.
Originator: uTecho
Fit Criterion: Users can easily identify the account currently in use and appreciate the readily available information.
Customer Satisfactions: 9 **Customer Dissatisfactions:** 2
Priority: 7 **Dependencies: R1** **Conflicts:**

Supporting Materials: None
History: Created on 25 October 2023.

utecho

Fig 2.3: Allow user to see quick info about connected account

Requirement #: R4 **Requirement Type:** Functional
Description: The platform shall feature a very strict no KYC policy to ensure no data of users is collected.
Rationale: To provide users with reassurance that their personal data is not collected and hence secure.
Originator: uTecho
Fit Criterion: Users are comfortable in utilizing our platform with no worries.
Customer Satisfactions: 10 **Customer Dissatisfactions:** 2
Priority: 2 **Dependencies:** **Conflicts:**

Supporting Materials: None
History: Created on 15 November 2023.

utecho

Fig 2.4: Feature a very strict no KYC policy

Requirement #: R5 **Requirement Type:** Functional
Description: The platform shall have adequate security measures for the security of their transactions and user data.
Rationale: To provide users with reassurance that all their personal information and transactions will remain safe and can't be tampered with.
Originator: uTecho
Fit Criterion: Users are reassured about the security of the platform.
Customer Satisfactions: 11 **Customer Dissatisfactions:** 0
Priority: 1 **Dependencies: R4, R1** **Conflicts:**

Supporting Materials: None
History: Created on 9 December 2023.

utecho

Fig 2.5: Security of user data and transactions

Use Case Diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

- **User Interaction with Web App:** Users access the platform's web application to perform various actions related to transferring and receiving money over crypto. The web app provides a user-friendly interface for seamless interaction with the platform.
- **Connect Blockchain Wallet:** Users have the option to connect their blockchain wallets to the platform. This connection allows the platform to access their wallet information and facilitates the initiation of transfers or payments directly from their wallet.
- **Initiate Transfers:** Once users have connected their blockchain wallets, they can initiate transfers or payments through the platform. Users input the recipient's wallet address and the amount to be transferred, and the platform processes the transaction securely and efficiently.
- **Transaction Verification:** Transactions initiated through the platform undergo verification to ensure their integrity. This verification process is carried out externally through online EVM (Ethereum Virtual Machine) machines or similar mechanisms. By verifying transactions, the platform enhances security and builds trust among users regarding the reliability of their transactions.

Use case diagrams are used to describe tasks or use cases that a subject, the system should, or can perform with the help of one or more actors.

A use case diagram contains four components.

Boundary: The boundary defines the system concerning the world.
Actor: Actors usually are individuals who according to their roles perform tasks.
Use cases: Use cases are tasks or roles performed by the actors in the system.
Relationships: A relationship is a link between and among the actors and the use cases.

Use Case: User

• Actors: User

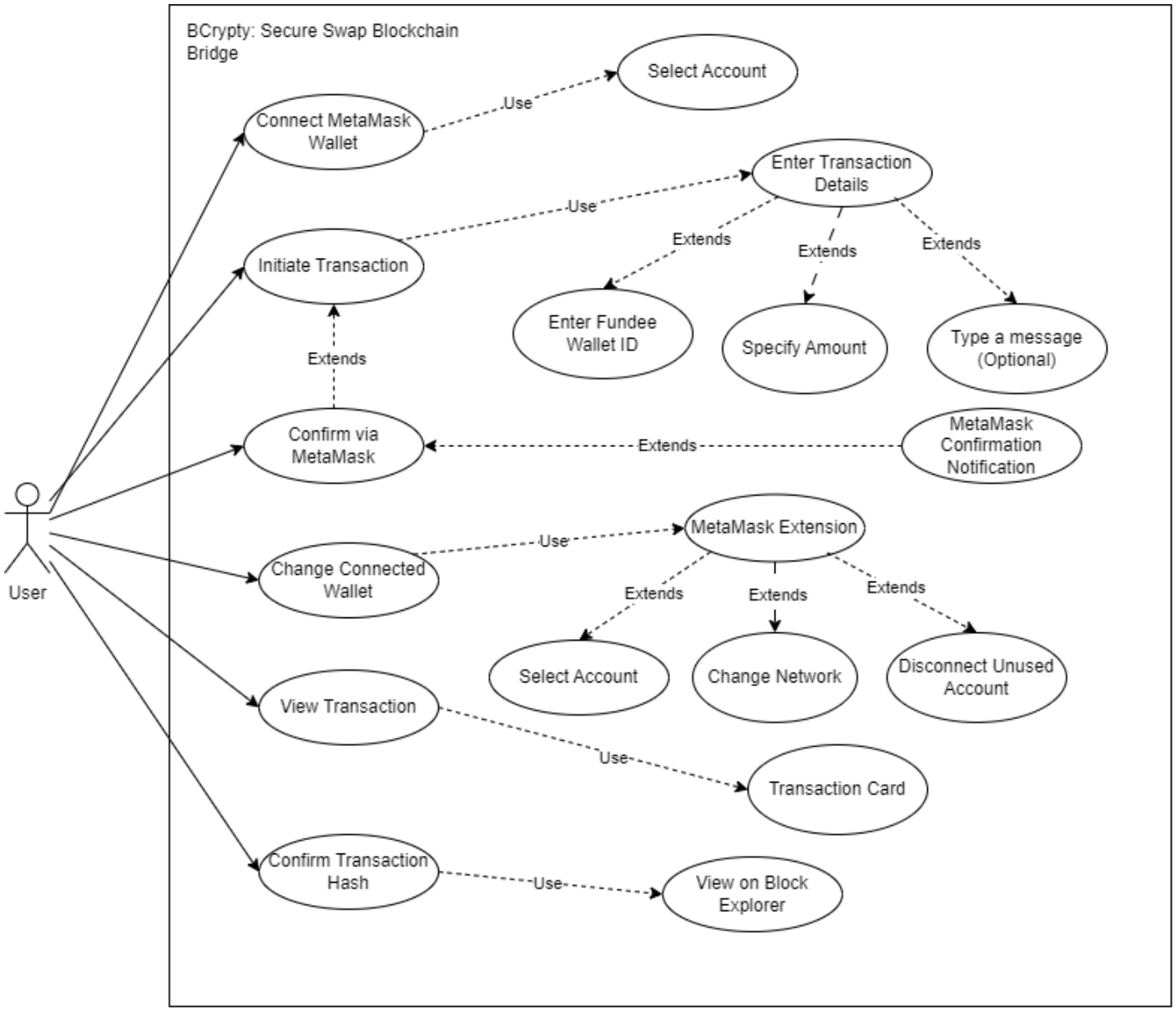

Fig 2.6: Use Case (User)

Activity Diagram

The activity diagram used to describe flow of activity through a series of actions. Activity diagram is an important diagram to describe the system. The activity described as an action or operation of the system.

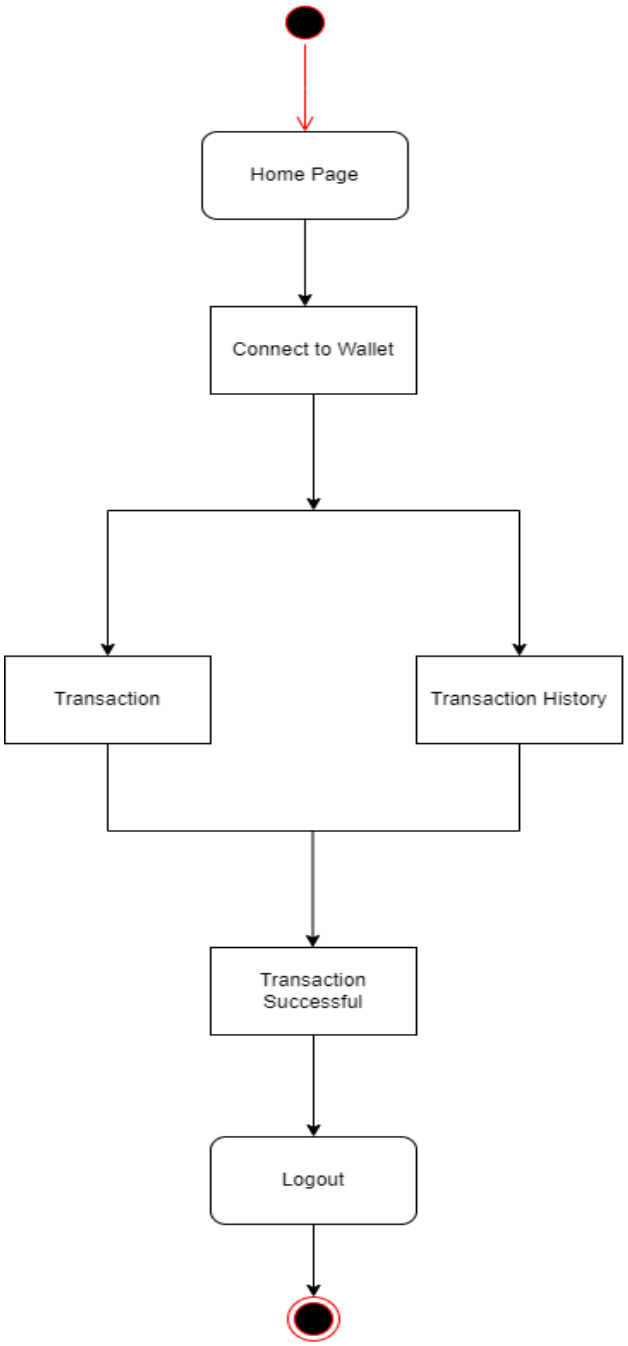

Fig 2.7: Activity Diagram

Sequence Diagram

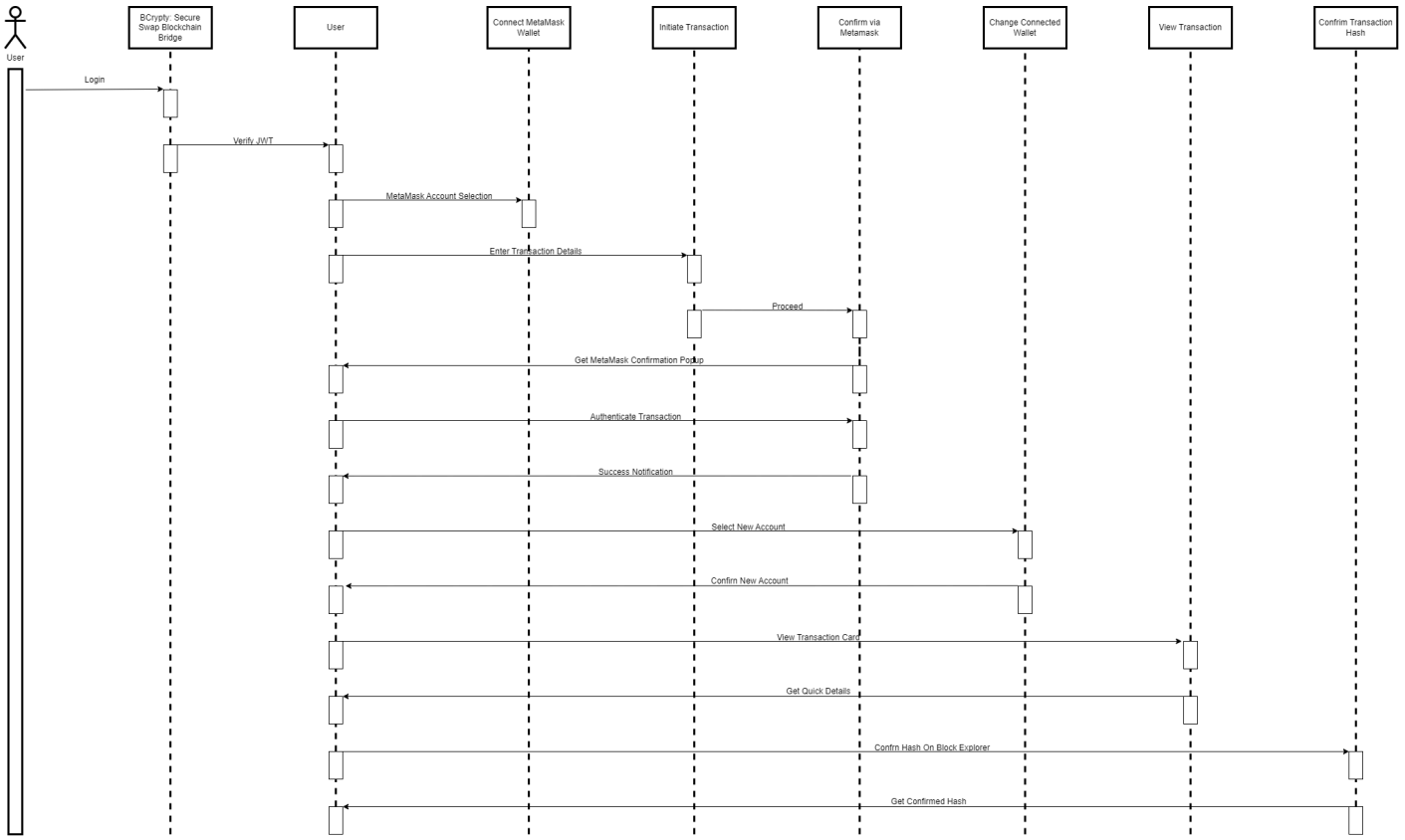

Fig 2.8: Sequence Diagram

Software Architectural Design

Software architecture discusses the vital structures of the software system and the rules of formulating such structures and software systems. Each structure consists of software elements, relationships among those elements, and properties of elements and their relationships.

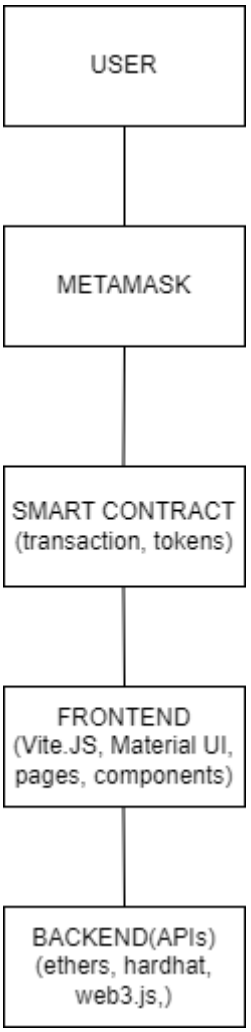

Fig 2.9: Architectural Diagram

CHAPTER 3: SOFTWARE PROCESS MODEL

3.1 Software Process Model

A software process model refers to a framework that defines the steps and activities involved in developing a software system. It is a systematic approach to software development that provides a roadmap for the entire software development life cycle.
There are several popular software process models, including:

1. **Waterfall model**
2. **Agile model**
3. **Spiral model**
4. **V-Model**
5. **Incremental model**

Choosing the right software process model depends on the specific needs and constraints of the project, such as the size, complexity, budget, and risk tolerance.

3.2 Proposed Model

The Extreme Programming Model (XP) is a lightweight agile software development methodology based on simplicity, communication, feedback, and courage principles. XP is built upon values, principles, and practices, and its goal is to allow small to mid-sized teams to produce high-quality software and adapt to evolving and changing requirements. What sets XP apart from the other agile methodologies is that XP emphasizes the technical aspects of software development. Extreme programming is precise about how engineers work since following engineering practices allows teams to deliver high-quality code sustainably.
Extreme programming is, in a nutshell, about good practices taken to an extreme. Since pair-programming is good, let's do it all of the time. Since testing early is good, let's test before the production code is even written.

The model is shown in the figure below:

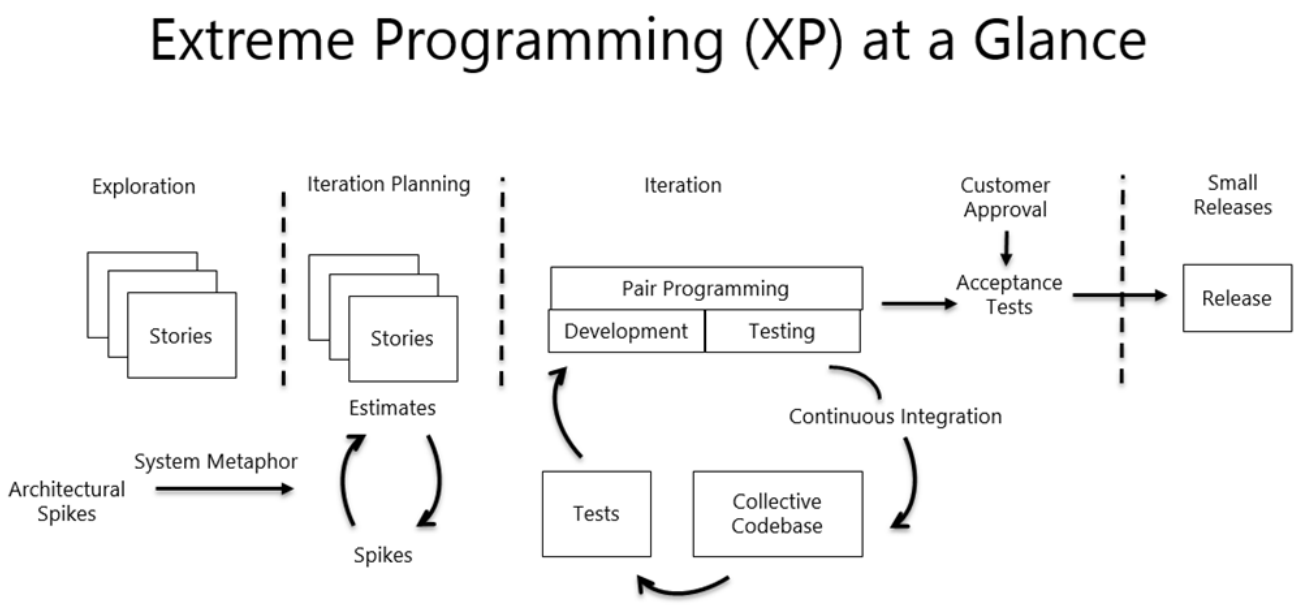

Fig 3.1: The Extreme Programming Model

Advantages of the Extreme Programming Model:

- The greatest advantage of Extreme Programming is that this methodology allows software development companies to **save costs and time** required for project realization. XP eliminates unproductive activities to reduce costs and frustration of everyone involved. It allows developers to focus on coding.
- One of the major advantages of Extreme Programming is that it **reduces the risks** related to programming or related to project failure. At the end XP ensures that the client gets exactly what he wants.
- **Simplicity** is one more advantage of Extreme Programming projects. The developers who prefer to use this methodology create extremely simple code that can be improved at any moment.
- The basic advantage of XP is that the whole **process is visible and accountable**. The developers make concrete commitments about what they will accomplish, show concrete progress.

- **Constant feedback**; demonstrate the software early and often, listen carefully and make any changes needed. Sprints help the team to move in the right direction.
- This approach creates **working software faster**. Regular testing at the development stage ensures detection of all bugs, and the use of customer approved validation tests to determine the successful completion of a coding block ensures implementation of only what the customer wants and nothing more.

CHAPTER 4: PROPOSED SYSTEM

4.1 Need for the proposed system

The need for BCrypty's proposed system for sending and receiving money over the blockchain arises from several shortcomings in traditional financial systems:

1. **High Fees and Intermediaries:** Traditional financial systems often involve high transaction fees and multiple intermediaries, such as banks or payment processors, which can significantly reduce the amount of funds received by senders and recipients. BCrypty aims to eliminate these intermediaries and associated fees, making money transfers more affordable and efficient.

2. **Limited Access and Control:** Traditional financial systems may impose restrictions on who can participate in money transfers, such as geographical limitations or platform-specific regulations. BCrypty seeks to provide universal access to its platform, allowing users worldwide to send and receive funds with ease and without arbitrary restrictions.

3. **Lack of Security and Trust:** Traditional financial systems are susceptible to various security breaches, including fraud, identity theft, and data breaches, which can erode trust among users. By leveraging blockchain technology, BCrypty ensures transparency, immutability, and enhanced security in money transfers, thereby fostering trust and confidence among users.

4. **Lengthy Payment Processing Times:** Traditional financial systems may suffer from delays in payment processing, with senders and recipients waiting for extended periods to access their funds. BCrypty's blockchain-based platform enables near-instantaneous transactions, providing senders and recipients with timely access to their funds without unnecessary delays.

5. **Lack of Global Reach:** Traditional financial systems may face limitations in reaching a global audience, such as restrictions on cross-border transactions or language barriers. BCrypty's platform transcends geographical boundaries, enabling users from diverse backgrounds to send and receive funds seamlessly, thus unlocking the potential for global financial inclusion and accessibility.

In summary, BCrypty's proposed system for sending and receiving money over the blockchain addresses the limitations of traditional financial systems by offering a transparent, secure, and efficient solution that empowers users to conduct peer-to-peer transactions with ease and confidence.

4.2 Feature of Proposed System

The future of BCrypty, the proposed peer-to-peer blockchain-based system, holds promising potential for transforming the banking landscape. Here are some of BCrypty's system

1. **Decentralized Transactions:** BCrypty facilitates peer-to-peer transactions without the need for intermediaries, ensuring direct and secure transfers of funds between senders and recipients.

2. **Transparent Ledger:** Utilizing blockchain technology, BCrypty maintains a transparent and immutable ledger of all transactions, providing users with a verifiable record of their financial activities.

3. **Low Transaction Fees:** BCrypty minimizes transaction costs by leveraging blockchain's efficiency, allowing users to transfer funds at a fraction of the cost associated with traditional financial systems.

4. **Enhanced Security:** BCrypty employs advanced cryptographic techniques and decentralized consensus mechanisms to ensure the security and integrity of transactions, protecting users' funds from unauthorized access or tampering.

5. **Global Accessibility:** BCrypty transcends geographical boundaries, enabling users from around the world to send and receive funds seamlessly, thus promoting financial inclusion and accessibility on a global scale.

In conclusion BCrypty's system for sending and receiving money over the blockchain offers a revolutionary approach to financial transactions, providing users with decentralized, transparent, and efficient solutions. By leveraging blockchain technology, BCrypty addresses the limitations of traditional financial systems, offering features such as decentralized

transactions, transparent ledger, low transaction fees, enhanced security, and global accessibility. With BCrypty, users can enjoy seamless peer-to-peer transactions, empowered by the transparency, security, and accessibility inherent in blockchain technology.

4.3 Need for Computerization

- Efficiency and Automation
- Scalability and Accessibility
- Data Management and Analysis
- Anonymity and Security
- User Experience and Convenience
- Compliance and Reporting

CHAPTER 5: SYSTEM DESIGN

Description

BCrypty introduces a groundbreaking web-based platform designed to facilitate seamless and secure money transfers over the blockchain. Here's a glimpse into its system design:

- **Blockchain-Powered Security:** BCrypty harnesses the power of blockchain technology to ensure unparalleled security and integrity in all money transfer transactions. Utilizing a decentralized network, transactions are securely recorded and immutable, safeguarding user funds from unauthorized access or tampering.

- **User-Centric Authentication:** Prioritizing user security, BCrypty implements robust authentication measures, requiring user registration for initiating money transfers. The platform offers a seamless and intuitive registration process, ensuring a user-friendly experience for all users.

- **Efficient Money Transfer Management:** Users experience seamless money transfer management through BCrypty's intuitive web interface, empowering them to initiate, track, and manage transactions with ease. Real-time updates and transaction history provide users with valuable insights into their financial activities.

- **Streamlined Payment Processing:** BCrypty seamlessly integrates with blockchain networks and cryptocurrency wallets, facilitating swift and hassle-free money transfers for users. Transactions are promptly recorded on the blockchain, ensuring transparency and accountability.

- **Anonymity and Privacy:** BCrypty emphasizes user anonymity and privacy, with all money transfer transactions securely recorded on the blockchain. This ensures user privacy while maintaining transparency and accountability in financial transactions.

- **Effective Communication:** BCrypty facilitates effective communication between users through its web-based platform, enabling users to exchange messages and notifications regarding money transfer activities.

- **Insightful Transaction Analytics:** BCrypty offers comprehensive transaction analytics, providing users with valuable insights into their financial activities. These analytics empower

users to make informed decisions and optimize their money transfer strategies.

- **Rigorous Security Measures:** BCrypty upholds stringent security standards, employing encryption and regular security audits to safeguard user data and transactions on its web-based platform.

- **Seamless Web Experience:** BCrypty ensures a seamless and intuitive web experience for users, with a user-friendly interface optimized for desktop and mobile browsers. The platform offers responsive design and cross-browser compatibility to cater to diverse user preferences

5.1 Desktop App UI/UX Design Pages

1. Home Page

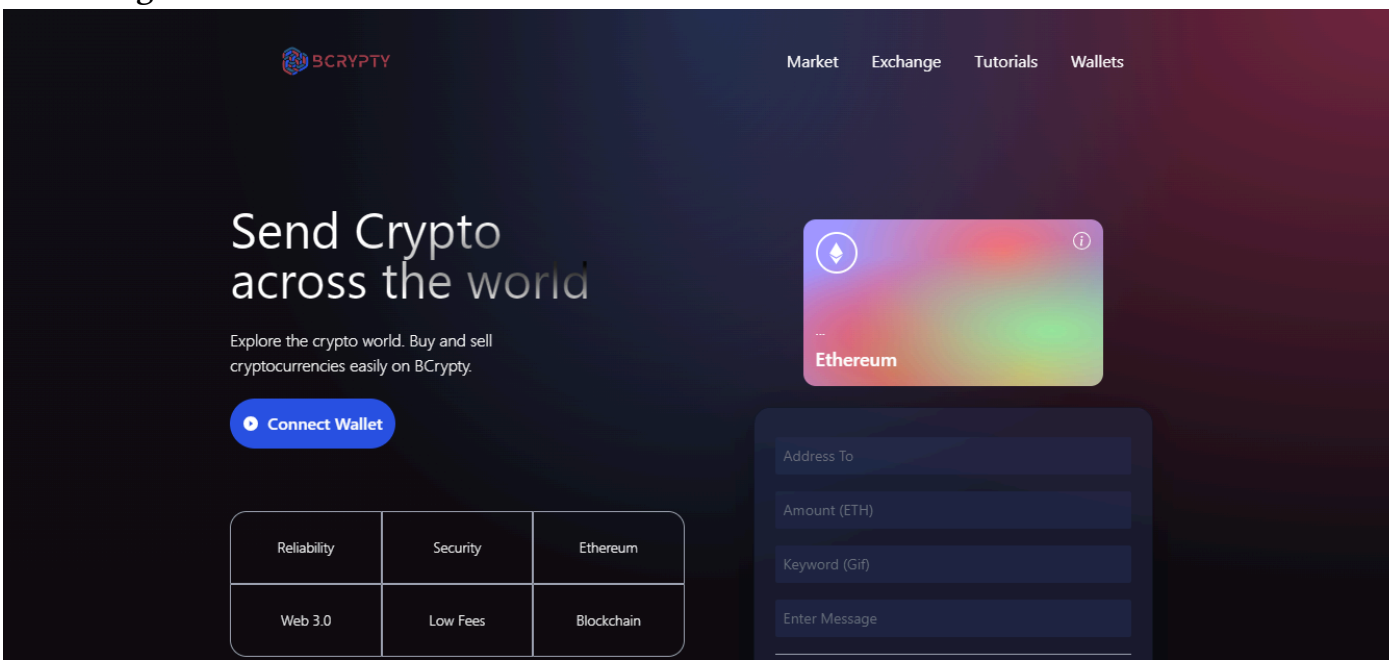

Fig 5.1: Home Page

2. Wallet Connection

Fig 5.2: Wallet Connection

3. Start Transaction Page

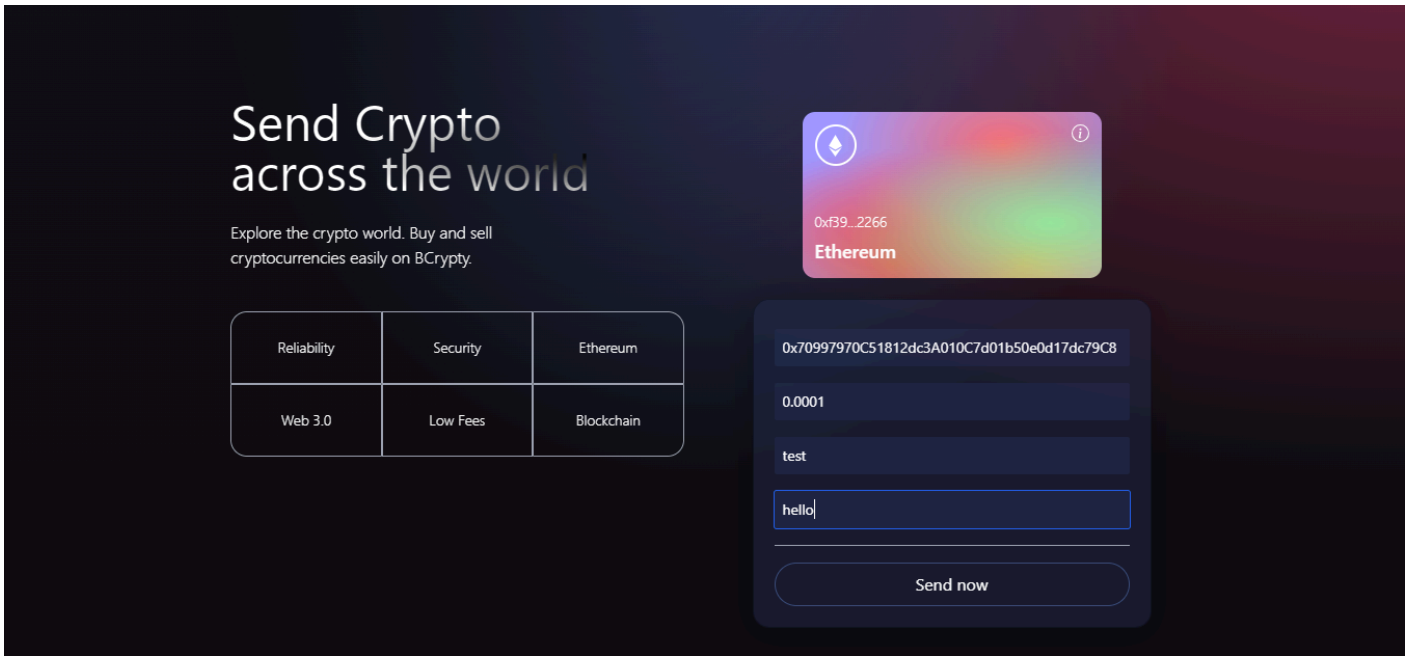

Fig 5.3: Start Transaction Page

4. Transaction Confirmation Page

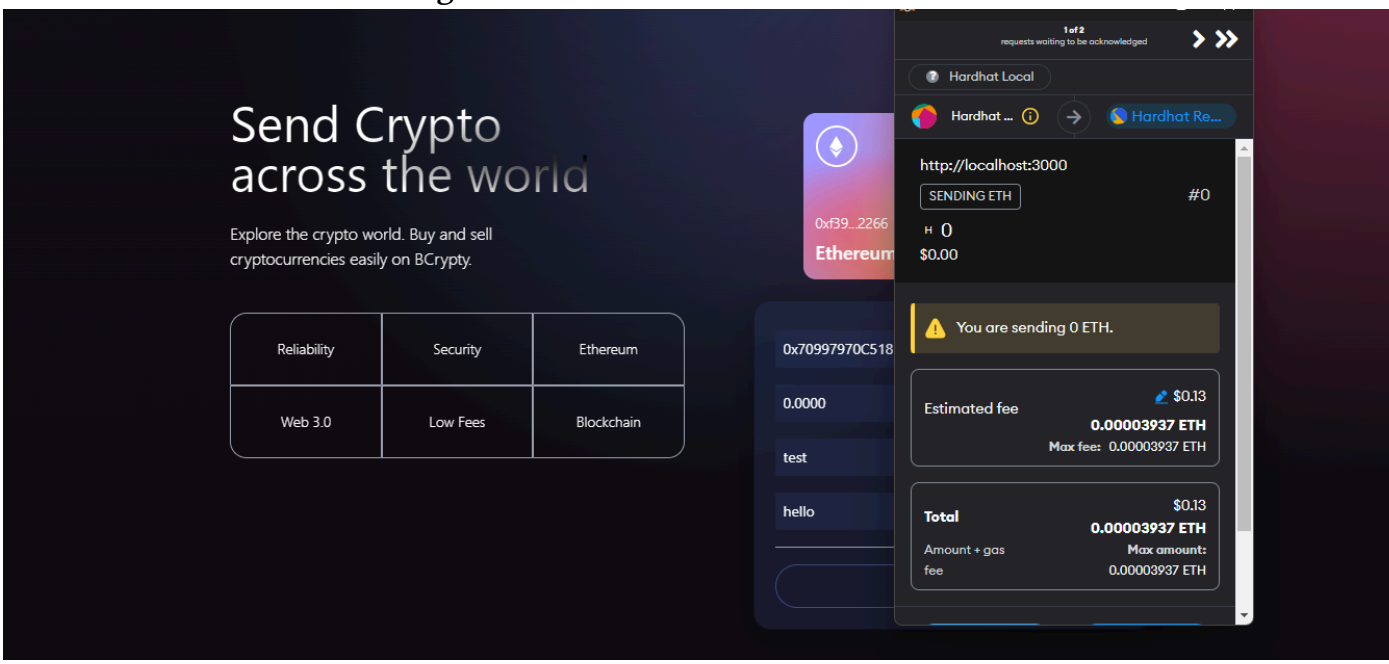

Fig 5.4: Transaction Confirmation Page

5. Transactions Card

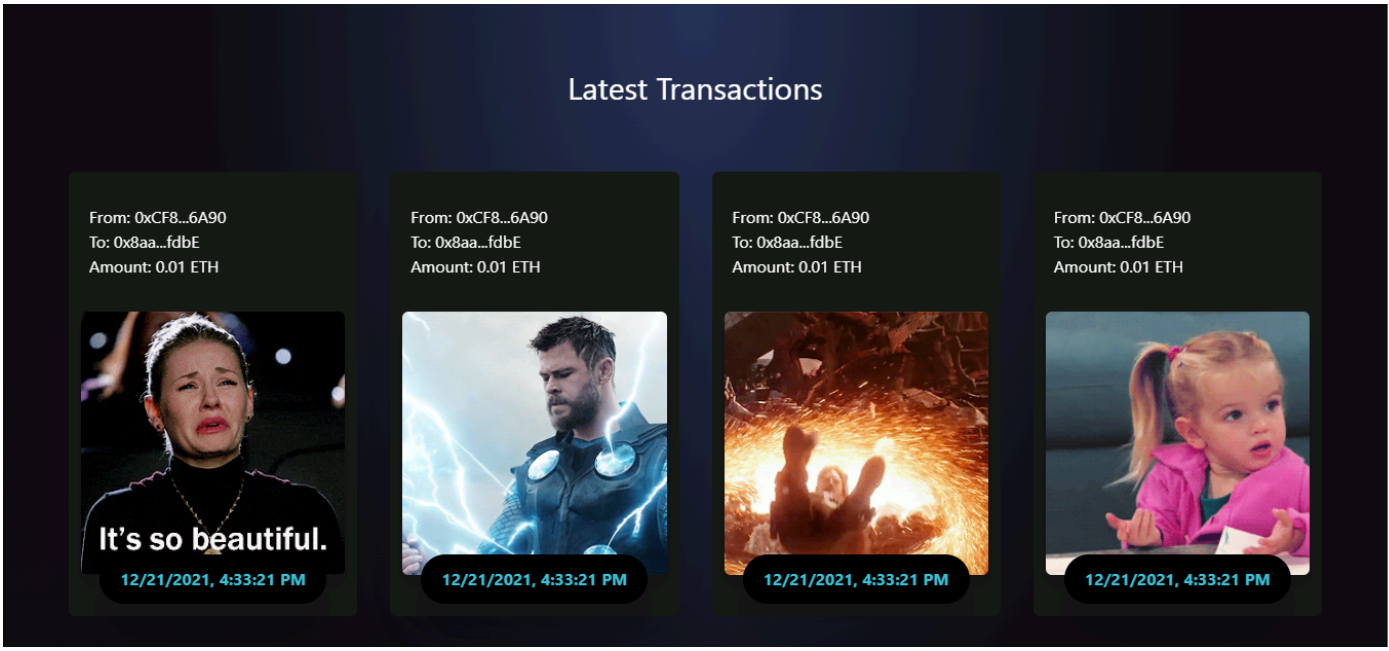

Fig 5.5: Transactions Card.

CHAPTER 6: DEVELOPMENT & CODING

6.1 Development And Coding

The development process for BCrypty would follow a systematic approach to ensure the successful implementation of the desktop and mobile app. Here's an outline of the development process using the technologies you mentioned:

- **Project Planning and Requirements Gathering:**
  - Define project goals, features, and requirements.
  - Identify target users and their needs.
  - Plan the development timeline and resource allocation.
- **Front-end Development:**
  - Set up the development environment using React with ViteJS for fast development.
  - Design the user interface (UI) components and layout based on wireframes and design mockups.
  - Implement responsive and intuitive UI features using React components.
  - Integrate Metamask for Ethereum wallet functionality.

- **Back-end Development:**
  - Set up the server-side environment using Express.js for Node.js web application framework.
  - Choose MongoDB as the database for storing user data and transaction information.
  - Develop RESTful APIs for handling user authentication, data storage, and interaction with the blockchain.
  - Implement Web3.js library to interact with the Ethereum blockchain network for cryptocurrency transactions.
- **Blockchain Integration:**
  - Develop smart contracts using Solidity language for defining crowdfunding rules and transactions on the Ethereum blockchain.
  - Use Hardhat for Ethereum development environment and testing smart contracts.
  - Integrate smart contracts with the back-end APIs to facilitate crowdfunding transactions securely.
- **Security Measures:**

- Implement encryption for securing user data and transactions.
- Ensure proper authentication mechanisms to prevent unauthorized access.
- Conduct security audits and penetration testing to identify and fix vulnerabilities.

- **Testing and Debugging:**
  - Write unit tests and integration tests for front-end and back-end components.
  - Test the app's functionality across different devices and browsers.
  - Conduct end-to-end testing to simulate real-world usage scenarios.
  - Debug and fix any issues or bugs encountered during testing.
- **Documentation:**
  - Document the codebase with comments and documentation to aid in future maintenance and updates.
  - Create user manuals and technical documentation for developers and users.
- **Deployment and Maintenance:**
  - Deploy the app to production servers or cloud platforms.
  - Monitor app performance and user feedback for continuous improvement.
  - Provide ongoing maintenance and support to address any issues or updates.

6.2 Functional Requirements

- **User Authentication and Authorization:**
- Secure registration and login system for users and administrators.
- Role-based access control to manage user permissions, ensuring only authorized actions are performed.
- **Transaction Management:**
- Allow users to initiate and receive cryptocurrency transactions securely over the blockchain.
- Implement encryption and authentication mechanisms to protect user data and transactions.
- Provide real-time transaction tracking and history for users to monitor their activity.
- **User Interface (UI) and User Experience (UX):**
- Design a responsive and intuitive UI for desktop devices.
- Ensure seamless navigation and user-friendly interactions to enhance the overall user experience.
- Utilize Material UI components for consistent styling and visual appeal.
- **Search and Filtering:**

- Implement search functionality for users to find specific transactions or account information.
- Allow users to filter transactions by various criteria, such as date, amount, or transaction type.

- **Notifications and Alerts:**

- Utilize the Metamask extension for notifications and alerts related to transaction confirmations and account activities.
- Notify users of successful transactions, account updates, or security-related events.

- **Security Measures:**

- Implement robust security measures, including encryption, authentication, and authorization, to protect user accounts and transactions.
- Conduct regular security audits and updates to address potential vulnerabilities and ensure the platform's integrity.

- **Reporting and Analytics:**

- Provide users with access to transaction history and account statements for auditing and reporting purposes.
- Generate analytics on transaction volumes, trends, and patterns to help users make informed financial decisions.

- **Integration with Blockchain Technology:**

- Integrate with blockchain networks like Ethereum to facilitate secure and transparent transactions.
- Utilize smart contracts for executing and enforcing transaction logic, ensuring trust and reliability in the platform.

- **Compliance with Regulations:**

- Ensure compliance with relevant financial regulations and data protection laws to safeguard user rights and privacy.
- Implement measures for KYC (Know Your Customer) and AML (Anti-Money Laundering) compliance, where applicable.

- **Error Handling and Recovery:**

- Implement robust error handling mechanisms to gracefully manage and recover from unexpected errors or system failures.
- Provide users with informative error messages and prompts to help troubleshoot and resolve issues efficiently.

6.3 Non-Functional Requirements

Performance:

- Optimize loading times for the web application.
- Minimize latency for transactions and interactions with the smart contract.

Scalability:

- Design the system to accommodate a growing number of users.
- Ensure the application can handle increased traffic and activity.

Security:

- Implement secure coding practices and regular security audits.
- Protect user data and transactions with strong encryption.

Maintainability:

- Write clean, modular, and well-documented code.
- Follow best practices for version control and code reviews.

Compatibility:

- Ensure compatibility with major web browsers and devices.
- Test the application across various screen resolutions and operating systems.

Accessibility:

- Design the application to be accessible for users with disabilities.
- Follow accessibility guidelines, such as the Web Content Accessibility Guidelines (WCAG).

Usability:

- Prioritize ease of use and intuitiveness in the user interface design.
- Conduct user testing to identify and resolve usability issues.

Reliability:

- Implement robust error handling and recovery mechanisms.

- Monitor the application's uptime and ensure quick response to any issues.

6.4 Selection of tools and technology

- Hardware

Hardware	Description
Processor	Dual-core or higher
RAM	4 GB or higher
Storage	128 GB or higher; SSD preferred
Network	High-speed internet connection

- Software

Software	Description
Operating System	Windows
Server/web technology	Web3.js, Solidity, Metamask
Languages	JavaScript, React, Vite JS
Tools	WebStorm, Canva, Draw.io

Graphical User Interface

- Welcome Section (With wallet card)
- Services Section
- Transactions Section
- Footer & Information

6.5 System Specification

Requirement Analysis

The BCrypty System Requirements provides high-level functional requirements.

The functions of the system are discussed below.

User can directly connect their metamask wallet and transfer funds or they can register as a user

- **Documentation:** Comprehensive user, installation, and administration documentation along with help resources.
- **Licensing and Legal Concerns:** Ensuring compliance with relevant licensing and legal regulations pertaining to cryptocurrency transactions and blockchain technology.
- **Packaging:** Efficient packaging of the application for easy deployment of web application.
- **Standards:** Adherence to technical, safety, and quality standards in blockchain technology implementation
- **Operational Concerns:** Defined error handling procedures and backup frequency to ensure smooth operation of the system
- **Application-specific Domain Rules:** Detailed information on the handling of cryptocurrency transactions, including the entire cycle of sending and receiving money over the blockchain network

CHAPTER 7: SOFTWARE TESTING

Software Testing

Software testing is a critical process in the development life cycle of a software application. It involves systematically evaluating and verifying the functionality, performance, security, and other aspects of the software to ensure that it meets the intended requirements and functions as expected. Software testing helps identify defects, errors, and vulnerabilities in the software and ensures that the software is reliable, robust, and of high quality. It helps in detecting and fixing issues early in the development process, which reduces the risk of costly and time-consuming fixes in the later stages of the software development life cycle.

There are various types of software testing, including:

- **Unit Testing:** This type of testing involves testing individual units or components of the software in isolation to ensure that they function correctly.
- **Integration Testing:** This type of testing involves testing the integration and interaction between different components or modules of the software to ensure that they work together as expected.
- **System Testing**: This type of testing involves testing the entire system as a whole to ensure that it meets the specified requirements and functions as intended in the target environment.
- **Acceptance Testing:** This type of testing involves testing the software against the business requirements or user acceptance criteria to ensure that it meets the intended purpose and is ready for deployment.
- **Performance Testing:** This type of testing involves testing the performance and scalability of the software under different load conditions to ensure that it performs well and meets the performance requirements.
- **Security Testing:** This type of testing involves testing the software for vulnerabilities and weaknesses in its security features to ensure that it is secure and protects against potential threats.
- **Usability Testing:** This type of testing involves testing the software's user interface, ease of use, and overall user experience to ensure that it is user-friendly and meets the needs of the intended users.
- **Regression Testing:** This type of testing involves retesting previously tested functionalities to ensure that changes or fixes in the software do not introduce new issues or impact existing functionalities.

- **Automated Testing:** This involves using automated tools and scripts to execute tests and verify the software's functionality, performance, and other aspects, which helps in speeding up the testing process and ensuring consistent results.

  Software testing is an essential part of the software development process and helps ensure that the software is of high quality, reliable, and meets the intended requirements. It helps in reducing the risk of software failures, enhancing user satisfaction, and improving the overall performance and security of the software.

7.1. Testing Process

All the interfaces and functionality of the app was being tested at every stage. Initially the interfaces were tested at the time each of them was being created. Further we designed the test cases to test the app at each and every possible point and scenario. Those test cases are explained in the next step.

During the process we identify bugs that we find and resolve them so that it meets all the technical requirements. Different methodologies are used for this purpose, whatever the methodology is, the main aim is to deploy an error free system.

7.2. Test Case Design

Test case design is the process of creating a set of test cases or test scenarios that will be used to verify the functionality and behavior of a software application. It involves identifying the input conditions, expected outputs, and the preconditions and post-conditions for each test case. The goal of test case design is to ensure that the software application meets the intended requirements, functions correctly, and is free from defects, errors, and vulnerabilities. It involves designing test cases that cover different scenarios and conditions to ensure comprehensive testing of the software.

There are various techniques for test case design, such as black-box testing, white-box testing etc. These techniques help in deriving the test cases and ensure that the testing effort is effective and efficient.

Test Cases

Test cases are formulated to guarantee that all the functionalities of the system have been

checked at least one-time during testing and that all logical circumstances have been tested.

Test Case	TC001
Test Case Description	Trying to connect MetaMask Wallet
Pre-Condition	User must have Metamask Extension installed in Browser
Test Case Name	**Connect Wallet**
Steps	1. Tap On Connect Wallet Button 2. Select Account from MetaMask Wallet 3. Click "Connect" button **Expected Result:** **Alert**: MetaMask Wallet Connects and Connect Wallet button disappears **Actual Result:** Wallet Connection Successful **Status:** Passed

Table 7.1: Test Case TC001 Connect Wallet

Test Case No:	TC002
Test Case Description	Transfer funds to other wallet
Pre-Condition	MetaMask wallet must be connected and must have sufficient funds
Test Case Name	**Transfer Funds**
Steps	1. Navigate to "Wallet" section 2. Enter all required account details and about 3. Click "Send Now" button **Expected Result:** - MetaMask confirmation Alert: Confirm Transfer **Actual Result:** - Transfer Successful **Status:** Passed

Table 7.2: Test Case TC002 Transfer Funds

Test Case No:	TC003
Test Case Description	See Transaction in Transactions Section
Pre-Condition	User must have done some previous transactions
Test Case Name	**Test Transaction Card Function**
Steps	1. Navigate to Transaction Section 2. See All Transactions With Timestamps **Expected Result:** **Alert**: User Sees their Transaction in the list **Actual Result:** To be filled after test execution **Status:** Passed

Table 7.3: Test Case TC003 Test Transaction Card Function

7.3. Black Box Testing (behavioral testing)

The main objective of black-box testing is to validate the software's functionality, usability, and compatibility with the expected requirements, without being influenced by the internal implementation. Black-box testing can be performed at different levels of testing, including unit testing, integration testing, system testing, and acceptance testing. It typically involves creating test cases based on the software's functional specifications or requirements, and then executing those test cases to verify if the software behaves as expected.

Advantages of black-box testing include:

- Independence from internal implementation
- Simulates end-user perspective
- Encourages comprehensive testing
- Facilitates collaboration

Test Case ID: BC-TC001

Test Case Title: Connect MetaMask Wallet Account

Test Objective: To verify if a user can connect MetaMask Wallet using the BCrypty app.

Test Steps:

- Launch the BCrypty app.
- Log in with valid credentials or register a new account.(Optionally)
- Navigate to the "Connect Wallet" Button and click it.
- Choose a wallet account to connect to App and click connect
- Verify if the account is connected properly.
- Verify if the account address shows in the wallet card.
- Logout from the app.

Expected Result:

- The MetaMask wallet is successfully connected to the BCrypty app.
- The user's wallet address is displayed in the wallet card, indicating a successful connection.

- Upon logging out from the app and logging back in, the connected MetaMask wallet remains connected without any issues.
- No changes or disruptions occur in the user's connected MetaMask wallet due to the connection with the BCrypty app.
- The user can seamlessly initiate and receive cryptocurrency transactions using the connected MetaMask wallet within the BCrypty app.

7.4. White Box Testing (structural testing)

White-box testing, also known as structural testing or glass-box testing, is a software testing technique that focuses on testing the internal structure and implementation details of the software system. In white-box testing, the tester has knowledge of the internal code, structure, and logic of the software being tested, and uses this knowledge to design test cases that specifically target the internal components and functionality of the system. White-box testing involves testing the software at the source code level, and it typically includes testing various paths, conditions, and branches within the code to ensure that all possible scenarios are tested. This type of testing is usually performed by developers or testers who have a strong understanding of the programming language, software architecture, and design patterns used in the software system.

White-box testing techniques include statement coverage, branch coverage, path coverage, and condition coverage, among others. These techniques aim to achieve high code coverage, which means that a large portion of the code is tested to ensure that it is functioning correctly.

Test Case ID: BC-TC002

Test Case Title: Verify funds transfer to other account

Test Objective: To verify if the system correctly handles funds Transfer to another account,

Test Steps:

- Launch the BCrypty app.
- Log in with valid credentials.
- Navigate to the "Transfer Funds" feature.
- Enter the recipient's wallet address and the amount to be transferred.
- Click on the "Transfer" button to initiate the transfer.
- Verify if a confirmation popup from MetaMask appears, asking the user to confirm the transaction.
- Confirm the transaction in MetaMask.
- Verify if a confirmation message is displayed within the BCrypty app, confirming the successful transfer.

- Logout from the app.

Test Data:

- Recipient's wallet address: [0x7019d3Cc0a363c30d0D31bF55cC6351c78f09A50]
- Amount to be transferred: [0.0001 ETH]

Expected Result:

- A confirmation popup from MetaMask appears, asking the user to confirm the transaction.
- Upon confirming the transaction in MetaMask, a confirmation message is displayed within the BCrypty app, confirming the successful transfer of funds to the recipient's account.
- The transferred amount is reflected in the recipient's account balance.
- The system handles the funds transfer securely and efficiently, without any errors or discrepancies.

CHAPTER 8: IMPLEMENTATION & TRAINING

8.1 Implementation

1. **Project Planning:** Define project scope, objectives, and requirements. Create a project timeline and allocate resources.
2. **Technology Selection:** Choose appropriate technologies, frameworks, and databases for development.
3. **System Design:** Create a detailed system design, including app architecture, components, and user interfaces.
4. **Front-End Development:** Implement user interfaces using ReactJS, ViteJS for web.
5. **Back-End Development:** Develop server-side components using server-side programming languages, frameworks, and databases.
6. **Blockchain Integration:** Integrate blockchain technology for crowdfunding transactions and smart contracts.
7. **Security Implementation:** Implement security measures to protect against common vulnerabilities.
8. **Testing:** Conduct thorough testing for functionality, usability, performance, and security.
9. **Deployment**: Deploy the app to a production environment, configure server settings, and set up infrastructure.
10. **Maintenance and Support:** Provide ongoing maintenance and support for bug fixes, updates, and enhancements.

Real-life implementation of BCrypty would involve meticulous market research, user-centric testing, and stringent regulatory compliance efforts. The platform's focus on providing a seamless system for peer-to-peer money transfers over blockchain technology requires strategic marketing initiatives targeting potential users seeking reliable and secure payment solutions. Prioritizing user data protection, BCrypty would adhere to relevant financial regulations while implementing robust security measures to safeguard transactions and user information. Continuous monitoring and iterative improvements would ensure the platform remains competitive and aligned with the evolving needs of its user base..

Project Overview

Project Name: *BCrypty: Secure Swap Blockchain Bridge*

Objective: To create a comprehensive web application, BCrypty, facilitating peer-to-peer transactions leveraging blockchain technology. BCrypty aims to establish a secure and transparent platform, enabling seamless money transfers between users while upholding user anonymity and transaction integrity.

Scope: Develop BCrypty, a web application designed to facilitate secure peer-to-peer cryptocurrency transactions utilizing blockchain technology. BCrypty's primary goal is to provide users with a transparent and reliable platform for sending and receiving cryptocurrency. Users will have the ability to seamlessly browse, support, and execute transactions for various initiatives on the blockchain. Key objectives include implementing robust authentication, transaction tracking, and smart contracts to ensure security and transparency in all transactions. BCrypty aims to enhance trust and accessibility in cryptocurrency transactions by offering a user-friendly and secure platform for users to conduct their financial activities.

8.2 Key Features

- **User authentication:** Users will be able to login with metamask and optionally create an account to manage their profiles.
- **Cryptocurrency transactions:** Making cryptocurrency transactions on the blockchain
- **Smart contracts:** Smart contracts will be used to ensure transparent and secure transfers, with funds released to sender and they can view transaction details on the blockchain
- **Notifications:** Users will see metamask notification pop up with confirmation message

Target Users: BCrypty serves individuals seeking efficient blockchain-based money transfers, catering to cryptocurrency enthusiasts and tech-savvy users without fundraising feature

Technologies Used: Blockchain, technologies (Vite.JS, JavaScript, Web3.JS ,Solidity, MetaMask, Hardhat, VSCode)), security measures (e.g., encryption, authentication), and testing tools.

Expected Outcome: A fully functional, secure, and user-friendly desktop and mobile app that provides a trusted platform for peer-to-peer money transfers using blockchain technology.

Project Timeline: The project timeline will depend on the scope and complexity of the app, encompassing phases such as planning, design, development, testing, deployment, and maintenance.

Resources: The project will require a team of skilled developers, designers, testers, and project managers, as well as appropriate hardware, software, and infrastructure resources.

8.2 Training

As a peer-to-peer crowdfunding app based on blockchain technology, BCrypty does not require any traditional training data or machine learning models for its operation. However, the development team responsible for building the BCrypty app will need to have expertise Blockchain, technologies (Vite.JS, JavaScript, Web3.JS,Solidity, MetaMask, Hardhat, VSCode), security measures (e.g., encryption, authentication), and testing tools. The team will need to be proficient in designing and developing smart contracts using blockchain programming languages, implementing secure transaction processing using cryptocurrency, and ensuring proper authentication and encryption measures to protect user data and funds. Additionally, the team will need to thoroughly test the app to identify and fix any potential bugs or vulnerabilities before deployment. It's important for the team to have a good understanding of the crowdfunding industry, market trends, and user expectations to create a successful and user-friendly app. Regular training and updates on the latest technologies and best practices in blockchain, web development, and security will be necessary to ensure a high-quality and secure implementation of the BCrypty app.

CHAPTER 9: CONCLUSION

Conclusion

In conclusion, BCrypty stands at the forefront of financial innovation, offering a groundbreaking platform for seamless and secure transactions via blockchain technology. With a steadfast commitment to user privacy, transparency, and efficiency, BCrypty redefines the way individuals send and receive money, transcending traditional limitations and fostering a new era of financial empowerment.

Through its robust infrastructure and cutting-edge features, BCrypty ensures that every transaction is executed with utmost security and reliability. By harnessing the power of blockchain, BCrypty eliminates intermediaries, minimizes transaction costs, and accelerates settlement times, thus providing users with unparalleled control over their financial transactions.

BCrypty's user-friendly interface and intuitive design make it accessible to users of all levels, while its adherence to the highest standards of security guarantees peace of mind for every participant. As the landscape of finance continues to evolve, BCrypty remains steadfast in its mission to revolutionize the way money is transferred and received, paving the way for a future where financial transactions are borderless, transparent, and accessible to all.

References

- https://krazytech.com/projects/sample-software-requirements
- http://softwaretestingfundamentals.com
- https://devblogs.microsoft.com
- https://www.lucidchart.com
- http://web.uettaxila.edu.pk/CMS/AUT2011/seSCbs/tutorial/Electronic_Book(UML%2024%20Hours).pdf
- https://www.guru99.com/test-case.html
- https://www.visualparadigm.com/support/documents/vpuserguide/3563/3564/85375_drawingentit.html