**BlockFund:** **Trusted Peer-to-Peer Blockchain-Based Crowdfunding**



*A project submitted*

*in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science*

**by**

**M. Hamza Sheikh** (2K19-BSCS-121) **Syed M. Umar Imtiaz Bukhari** (2K19-BSCS-111) **M. Abdul Moiz** (2k19-BSCS-119)

**Raees Ur Rehman** (2K19-BSCS-112)

**Supervised By**

### Sir Muhammad Ahtesham Noor

**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 "BlockFund: 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 “**Block Fund:** **Trusted Peer-to-Peer Blockchain-Based Crowdfunding**” 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 "Block Fund" 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

In the age of growing inflation rates and great disparity in wealth distribution, and growing scams, it is vital to build a platform that can unite people to help the causes they would like to without being worried about fraud or losing their money to fake causes.To offer an easy, one-stop solution for funding and receiving investment to your favorite causes, enhance empathy, and offer a decentralized organizational structure solution. This Web and Mobile application, “BlockFund” will provide users a good experience through Blockchain / Crowdfunding. These kinds of applications are the need of people in this modern era.Crowdfunding is when businesses, organizations or individuals [fund a business without traditional means](https://www.businessnewsdaily.com/1733-small-business-financing-options-.html) with small donations from many people. By receiving the necessary boost to cash flow, these ventures can get off the ground or launch new projects. Most of these campaigns happen via internet platforms, have set time frames for when money can be raised, and disclose specific monetary goals. Crowdfunding is a global and promptly emerging novel financing option for businesses, ideas, and projects. Crowdfunding aims at filling the existing gap of financing options for small companies with no or little track record. Different interested parties are willing that crowdfunding would be successful and prosper, including governments seeking to create jobs and foster economic growth, entrepreneurs reaching out to expand the scope of available financing options, ordinary people seeking to have the availability to invest, and intermediaries who set up as separate businesses.

**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: Signature of Student 2:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: M.Hamza Sheikh Name: Syed M.Umar Imtiaz Bukhari

Roll No: (2K19-BSCS-121) Roll No: (2k19-BSCS-111)

Date: 21 July 2022. Date: 21 July 2022.

Signature of Student 3: Signature of Student 4:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: M.Abdul Moiz Name: Raees Ur Rehman

Roll No: (2K19-BSCS-119) Roll No: (2k19-BSCS-112)

Date: 21 July 2022. Date: 21 July 2022.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Discipline: **BSCS**

Year of Study: **2019-2023**

Area of Study: **Computer Science**

Proposed Project: Title: **BlockFund**

Mr. Ahtesham Noor

Coordinator/Supervisor FYP

External Examiner

Dr. Naeem Aslam

Head, Department of Computer Science



TABLE OF CONTENTS

[ACKNOWLEDGEMENTS](#_bookmark0)

[UNDERTAKING](#_bookmark1)

[ABSTRACT](#_bookmark2)

[CHAPTER 1: INTRODUCTION 1](#_bookmark3)

* 1. [Domain 2](#_bookmark4)
  2. [Problem Statement 3](#_bookmark5)
  3. [Motivation 3](#_bookmark6)
  4. [Definition of Terms 3](#_TOC_250009)
  5. [Goal of our Project 4](#_bookmark7)

[CHAPTER 2: EXISTING SYSTEM 5](#_bookmark8)

Relative Work6

* 1. [Other websites 6](#_bookmark9)
  2. [Existing Problem](#_bookmark10)s  [7](#_bookmark10)
  3. [Solution Available in our Project 7](#_bookmark11)
  4. [System Requirements Specification 8](#_bookmark12)
     1. [User Interfaces 8](#_bookmark13)
     2. Software Interfaces And Hardware Interface 8
     3. Tech Stack 9
  5. [Use Case Diagram for Block Fund Website: 9-15](#_bookmark14)

[CHAPTER 3: SOFTWARE PROCESS MODEL 16](#_bookmark15)

* 1. [Software Process Model 17](#_TOC_250008)
  2. [Proposed Model 17-1](#_TOC_250007)9

[CHAPTER 4: PROPOSED SYSTEM 20](#_bookmark16)

4.1 [Need for the proposed system 21](#_bookmark17)

4.2 [Feature of Proposed System 22](#_TOC_250006)

4.3 [Need for Computerization 23](#_TOC_250005)

[CHAPTER 5: SYSTEM DESIGN 24](#_bookmark18)

[[Brief Description](#_bookmark18)](#_bookmark18) 25-26

* 1. Desktop App [UI/UX Design Pages 27-3](#_bookmark19)0
  2. Mobile App UI/UX Design Pages  [31-35](#_bookmark20)
  3. [DFD Diagram for](#_bookmark22) Block Fund  [36-37](#_bookmark21)

[5.4 Activity Diagram for](#_bookmark23) Block Fund  [38](#_bookmark22)

5.5 [Class Diagram for](#_bookmark24) Block Fund  [3](#_bookmark23)9

5.6 [Sequence Diagram for Block Fund](#_bookmark24) 40

5.7 [Software Architectural Design](#_bookmark25) 41

[CHAPTER 6: DEVELOPMENT](#_bookmark26) AND CODING42

* 1. [Development](#_bookmark27) And Coding43
  2. [Functional Requirements 44-4](#_bookmark28)5
  3. [Non-Functional Requirements 45-4](#_TOC_250004)6
  4. [Selection of tools and technology 4](#_TOC_250003)6
  5. [System Specification 47](#_TOC_250002)

[CHAPTER 7: SOFTWARE TESTING 48](#_bookmark29)

* 1. [Testing Process 49-](#_TOC_250001)50
  2. [Test Case Design 50-55](#_TOC_250000)
  3. Black Box Testing(Behavioral Testing) [56-57](#_bookmark32)
  4. [White Box Testing (Structural testing) 58-5](#_bookmark33)9

[CHAPTER 8: IMPLEMENTATION AND TRAINING 60](#_bookmark35)

* 1. [Implementation 61-](#_bookmark36)63
  2. [Training](#_bookmark37) 63

[Conclusion](#_bookmark38) 64

[References 65](#_bookmark39)

**LIST OF FIGURES**

**Fig 2.1:** Signup Feature for User 10

**Fig 2.2:** Login Feature for User 11

**Fig 2.3:** Allow user to Connect MetaMask 11

**Fig 2.4:** Campaign Creation feature for User 12

**Fig 2.5:** Funding Feature for User 12

**Fig 2.6:** Campaign creator to withdraw fund 13

**Fig 2.7:** Security of user data and transaction 13

**Fig 2.8:** Use Case (Admin) 14

**Fig 2.9:** Use Case (User) 15

**Fig 3.1:** Extreme Programming Model 18

**Fig 5.1:** Login/Signup Page 27

**Fig 5.2:** Home Page 27

**Fig 5.3:** Manual Page 28

**Fig 5.4:** Wallet Page 28

**Fig 5.5:** New Campaign Page 29

**Fig 5.7:** Confirmation Page 30

**Fig 5.8:** Contract Page 30

**Fig 5.9:** Mobile App Signup Screen 31

**Fig 5.10:** Mobile App Login Screen 31

**Fig 5.11:** Mobile App Forgot Password Screen 32

**Fig 5.12:** Home Screen After login 32

**Fig 5.13:** Magic Screen of Home after login 33

**Fig 5.14:** Drawer Screen before Sign up 33

**Fig 5.15:** Drawer Screen After login 34

**Fig 5.16:** Login and Sign up options 34

**Fig 5.17:** Sign up Screen with Regex 35

**Fig 5.18:** DFD Diagram 37

**Fig 5.19:** Activity Diagram 38

**Fig 5.20:** Class Diagram 39

**Fig 5.21:** Sequence Diagram 40

**Fig 5.22:** Architectural Diagram 41

**LIST OF TABLES**

**Table 2.1:** Relative Websites 6

**Table 2.2:** Hardware Requirements 8

**Table 2.3:** Software Requirements 8

**Table 5.1:** DFD Table 37

**Table 7.1:** Test Case TC001 Login to Block Fund 51

**Table 7.2:** Test Case TC002 Create Campaign 52

**Table 7.3:** Test Case TC003 Fund Campaign 53

**Table 7.4:** Test Case TC004 withdraw to metamask wallet 54

**Table 7.5:** Test Case TC005 request Funds 55

**CHAPTER 1: INTRODUCTION**

**INTRODUCTION**

Many individuals assume crowdfunding is an easy or free way of making money, but it requires a lot of effort to establish a project that backers will perceive as a valuable service. Success isn’t guaranteed, and as crowdfunding continues to gain popularity, backers have become shrewder in the projects they support. Crowdfunding is one of the most popular ways to raise funds for any project, cause, or help any individual in need. With the onset of Covid we have seen a rise in Crowdfunding activities across the globe, including small campaigns to help people get oxygen and medical help to large funds.

Our project is a Trusted Peer-to-Peer Blockchain-Based Crowdfunding and Decentralized Autonomous Organization Management **(Web and Mobile Application)**. To offer an easy, one-stop solution for funding and receiving investment to your fa

* 1. **Domain (Crowdfunding)**

It is a relatively new way of financing a project or venture, which involves collecting money from a large number of people, usually via the internet. It allows individuals, organizations, and businesses to bypass traditional funding sources such as banks, venture capital firms, and angel investors, and instead tap into the collective resources of a large pool of individuals.

In exchange for their contributions, backers or supporters are typically offered rewards or perks, such as early access to the finished product, discounts, or special recognition. In some cases, backers may receive equity in the form of shares in the company, making them part owners.

Crowdfunding campaigns are typically run on specialized platforms that provide a secure and efficient way for individuals to make contributions, and for project owners to manage their campaigns, interact with supporters, and track progress. The success of a crowdfunding campaign is dependent on several factors, including the quality of the project idea, the strength of the team behind it, the level of rewards or incentives offered, and effective marketing and outreach to reach a large audience.

* 1. **Problem Statement**

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

**Anonymity for Anti-Fraud and Security Measures:** We have seen, and continue to see a lot of crowdfunding scams happening around. There is no way to see where the funds are being used. We wanted to make the entire flow of funds transparent at every stage, so that there is no possibility of the money being misused.

* 1. **Motivation**

Blockchain could influence crowdfunding initiatives by cutting down the processing fees.

* This Project can make crowdfunding much more affordable for creators, without intermediaries.
* Investors are provided with tokens after funding a startup which gives them a form of ownership so that the fraudulent creator won’t get away easily.
  1. **Definition of Terms**

Terms used in our project are the following:

**Fundraise:** Fundraising or fund-raising is the process of seeking and gathering voluntary financial contributions by engaging individuals, businesses, charitable foundations, or governmental agencies.

**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.

**Crowdfunding**: Crowdfunding is a relatively new way of financing a project or venture, which involves collecting money from a large number of people, usually via the internet.

Smart Contracts: Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met.

* 1. **Goal of our Project**
* To offer an easy, one-stop solution for funding and receiving investment to your favorite causes, to enhance empathy, and to offer a decentralized solution for organization structure.
* To perform three main functions after logging in.
* User can invest in crowdfunding project.
* User can request funding for their projects.
* The users can make payments using our token.

**CHAPTER 2: EXISTING SYSTEM**

**RELATIVE WORK**

* Development of decentralized crowdfunding platforms: This involves creating blockchain-based platforms that allow individuals to contribute funds to a cause or project they support and provide project owners with a secure and efficient way to manage their campaigns and interact with supporters.
* Integration of blockchain and crowdfunding: This involves exploring ways to leverage the benefits of blockchain technology, such as security, Anonymity, and decentralization, to enhance the crowdfunding process and improve the experience for both backers and project owners.
* Decentralized governance and decision-making: This involve researching and developing new consensus mechanisms and voting systems that allow members to make decisions in a decentralized and secure manner.
* Token economics: This involves exploring the design and implementation of token models and reward systems that incentivize members

# 2.1 Other websites

|  |  |
| --- | --- |
| **Kickstarter** | https://www.kickstarter.com/ |
| **GoFundMe** | https://www.gofundme.com/ |
| **Start engine** | https://www.startengine.com/ |
| **Fundable** | https://www.fundable.com/ |
| **Mighty Cause** | https://www.mightycause.com/ |

**Table 2.1:** Relative Websites

**2.2 Existing Problems**

* Usually, crowdfunding apps have limited security measures, so investors risk their money by supporting start-ups.
* Crowdfunding apps don’t protect intellectual property. It’s possible that a bigger company steals an idea from a start-up, and implements it faster and even in better quality.
* High fees also make a standard crowdfunding landscape less amazing.
* Grossly unequal distribution of power and revenue that is typical in many businesses

**2.3 Solution Available in our Project**

* Blockchain could influence crowdfunding initiatives by cutting down the processing fees.
* It doesn’t need to engage intermediaries or third parties in financial transactions, so it can make crowdfunding much more affordable for creators.
* Blockchain-based crowdfunding models could enable creators to start raising funds not on one but on many platforms creating profitable combinations.
* Smart contracts can transfer funds to a startup only for a designated milestone, so there’s no room for fraud in this type of fundraising.
* Investors are provided with tokens after funding a startup which gives them a form of ownership so that the fraudulent creator won’t get away easily.

**2.4 System Requirements Specification**

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

**2.4.1 USER INTERFACES**

* Admin Interfaces
* User Interfaces

**2.4.2 SYSTEM HARDWARE AND SOFTWARE REQUIREMENTS**

For users to access our Web and Mob 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, Web3.dart, Metamask |
| Languages | JavaScript, React, Next JS, Flutter, Dart |
| Tools | VS code, Adobe XD, MS Projects |

**Table 2.3:** Software Interface

**2.4.3 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:

**[2.5 Use Case Diagram](https://meeraacademy.com/use-case-diagram-for-online-shopping/)**

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.

* In **Blockfund**: **Trusted Peer-to-Peer Blockchain-Based Crowdfunding**, an admin can log in with a username and password. The admin has access to the all projects. Additionally, the admin can manage these projects.
* The admin can view all the projects that have been approved, filter them by category, and view investor transactions from this platform. Moreover, the admin has the authority to accept or reject the project proposal. The list of all rejected projects can also be viewed, along with information about the project and the user attached to it.
* In transactions, a list of approved projects is displayed along with details about whether or not they tampered with the crowdsourcing process. The admin can view the list of users and their general information.
* For users to access information, they have to register their accounts and log in using a username and password.The user has access to the list of projects for which they want funding. They can view transactions by investors from this platform with a status of whether the transaction is manipulated and apply filters by category, project details, and more. Invest the money; if even one transaction is manipulated, investing will not be permitted
* The users can perform three main functions after logging in. They can invest in crowdfunding projects, request funding. The users can make payments using our token.

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.

**Requirement #**: R1 **Requirement Type:** Functional

**Description:** The platform shall provide a signup feature for users to create an account.

**Rationale:** To allow new users to register and access the platform's features.

**Originator:** GoFundMe

**Fit Criterion:** Users can successfully sign up and create an account.

**Customer Satisfaction:** 9 **Customer Dissatisfaction:** 2

**Priority:** 10 **Dependencies:** **Conflicts:**

**Supporting Materials:** None

**History:** Created at 9 March, 2022.

Volere:

Copyright ©Atlantic Systems Guild

**Fig 2.1:** Signup Feature for User

**Fig 2.2:** Login Feature for user

**Requirement #**: R3 **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:** GoFundMe

**Fit Criterion:** Users can connect their MetaMask wallet and perform transactions without any issues.

**Customer Satisfaction:** 8 **Customer Dissatisfaction:** 4

**Priority:** 7 **Dependencies:** **Conflicts:**

**Supporting Materials:** None

**History:** Created at 16 March, 2022.

Volere:

Copyright ©Atlantic Systems Guild

**Requirement #**: R2 **Requirement Type:** Functional

**Description:** The platform shall provide a login feature for users to access their accounts.

**Rationale:** To allow registered users to log in and access the platform's features.

**Originator:** GoFundMe

**Fit Criterion:** Users can successfully log in to their accounts.

**Customer Satisfaction:** 9 **Customer Dissatisfaction:** 1

**Priority:** 9 **Dependencies:** R1 **Conflicts:**

**Supporting Materials:** None

**History:** Created at 11 March, 2022.

Volere:

Copyright ©Atlantic Systems Guild

**Fig 2.3:** Allow user to Connect MetaMask

**Requirement #**: R4 **Requirement Type:** Functional

**Description:** The platform shall provide a campaign creation feature for users who have logged in/signup.

**Rationale:** To allow users to create campaigns for crowdfunding dining events on the blockchain-based platform.

**Originator:** GoFundMe

**Fit Criterion:** Users who have logged in/signup can access the campaign creation feature.

**Customer Satisfaction:** 10 **Customer Dissatisfaction:** 1

**Priority:** 9 **Dependencies: R2,R1** **Conflicts:**

**Supporting Materials:** None

**History:** Created at 19 March, 2022.

Volere:

Copyright ©Atlantic Systems Guild

**Fig 2.4:** Campaign Creation feature for User

**Requirement #**: R5 **Requirement Type:** Functional

**Description:** The platform shall provide a funding feature for users to contribute to a campaign.

**Rationale:** To enable users to support crowdfunding dining campaigns by making financial contributions.

**Originator:** GoFundMe

**Fit Criterion:** Users can successfully make financial contributions to a campaign.

**Customer Satisfaction:** 9 **Customer Dissatisfaction:** 3

**Priority:** 8 **Dependencies: R4** **Conflicts:**

**Supporting Materials:** None

**History:** Created at 2 April, 2022.

Volere:

Copyright ©Atlantic Systems Guild

**Fig 2.5:** Funding Feature for User

**Requirement #**: R6 **Requirement Type:** Functional

**Description:** The platform shall allow campaign creators to withdraw funds from their campaigns.

**Rationale:** To provide campaign creators with the ability to access and utilize the funds raised.

**Originator:** GoFundMe

**Fit Criterion:** Campaign creators can successfully withdraw funds from their campaigns.

**Customer Satisfaction:** 9 **Customer Dissatisfaction:** 2

**Priority:** 8 **Dependencies: R4** **Conflicts:**

**Supporting Materials:** None

**History:** Created at 27 March, 2022.

Volere:

Copyright ©Atlantic Systems Guild

**Fig 2.6:** Campaign creator to withdraw fund

**Requirement #**: R7 **Requirement Type:** Non-Functional

**Description:** The platform shall ensure the security of user data and transactions by leveraging appropriate security measures provided by the Firebase platform.

**Rationale:** To maintain the confidentiality, integrity, and availability of user information and funds using Firebase's built-in security features.

**Originator:** Firebase

**Fit Criterion:** The platform effectively utilizes Firebase's security features, including authentication, data encryption, and access control, to protect user data and transactions.

**Customer Satisfaction:** 9 **Customer Dissatisfaction:** 2

**Priority:** 9 **Dependencies:**  **Conflicts:**

**Supporting Materials:** None

**History:** Created at 8 April, 2022.

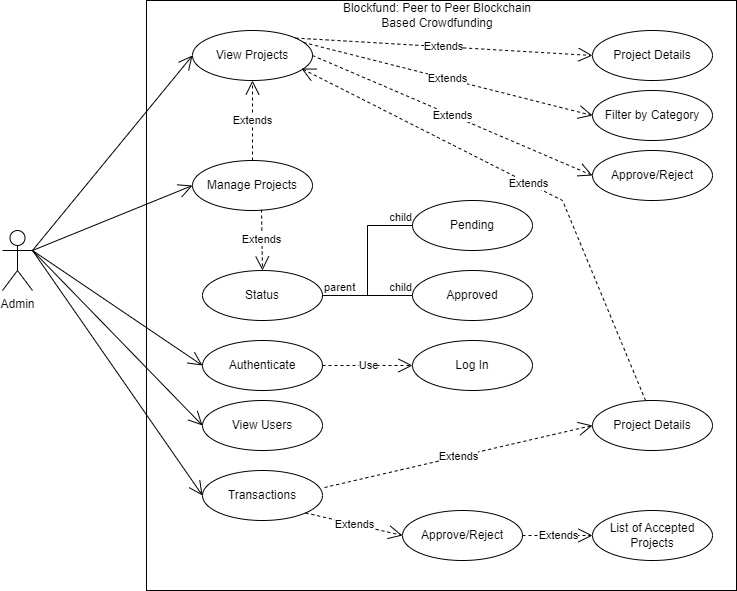
Volere:

Copyright ©Atlantic Systems Guild

**Fig 2.7:** Security of user data and transaction

Use Case: Admin

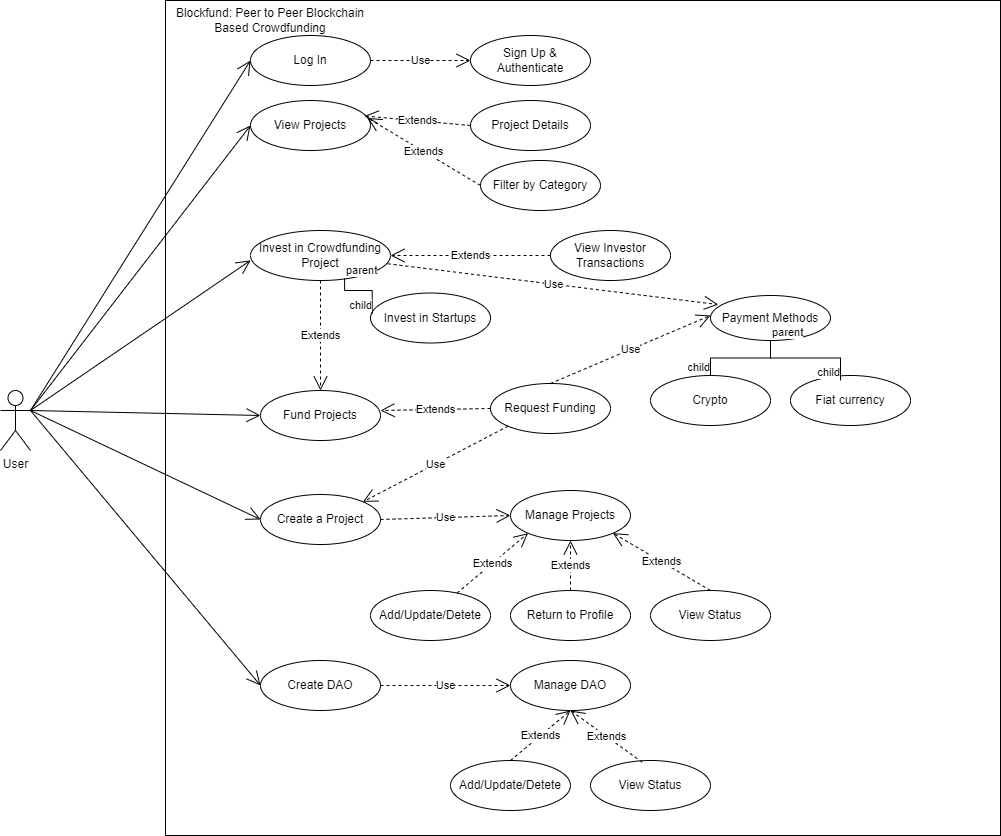
• Actors: Admin



**Fig 2.8:** Use Case (Admin)

Use Case: User

• Actors: User



**Fig 2.9:** Use Case (User)

**CHAPTER 3: SOFTWARE PROCESS MODEL**

* 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**

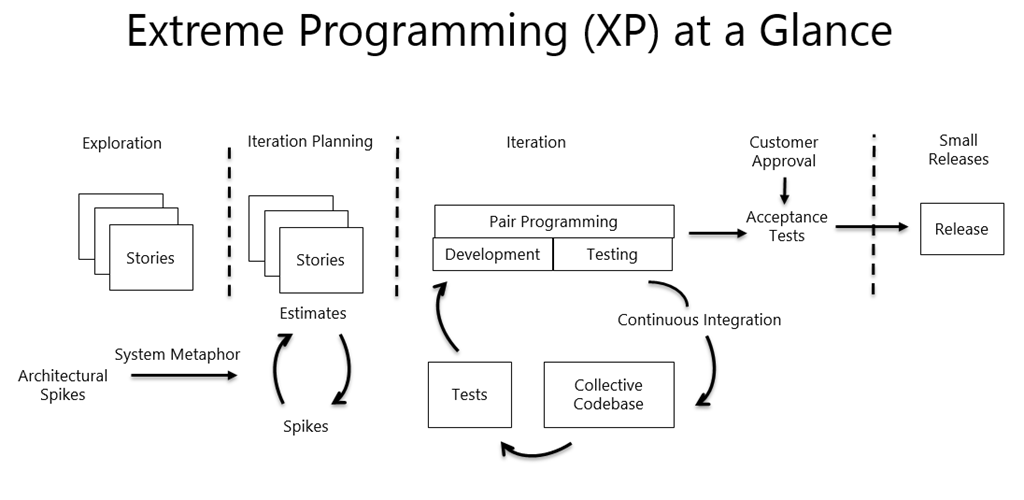
Choosing the right software process model depends on the specific needs and constraints of the project, such as the size, complexity, budget, and risk tolerance.

**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**

* 1. **Need for the proposed system**

The need for the proposed system, BlockFund, is driven by several challenges in traditional crowdfunding platforms:

* **High Fees and Intermediaries:** Traditional crowdfunding platforms may charge high fees, including transaction fees, platform fees, and processing fees, which can significantly reduce the amount of funds received by fundraisers. Moreover, these platforms may involve multiple intermediaries, such as banks or payment processors, adding complexity and cost to the fundraising process.
* **Limited Access and Control:** Traditional crowdfunding platforms may have restrictions on who can participate, including geographical limitations, platform-specific regulations, and approval processes. Additionally, fundraisers may have limited control over their campaigns, as the platform may impose restrictions on campaign content or fundraising goals.
* **Lack of Security and Trust:** Traditional crowdfunding platforms may be vulnerable to fraud, identity theft, and other security breaches, which can undermine the trust of donors. Moreover, the centralized nature of these platforms can expose them to single points of failure, making them susceptible to data breaches or downtime.
* **Lengthy Payment Processing Times:** Traditional crowdfunding platforms may have delays in payment processing, with fundraisers waiting for extended periods to receive their funds. This can cause inconvenience and financial strain for fundraisers who may need immediate access to funds for their campaigns.
* **Lack of Global Reach:** Traditional crowdfunding platforms may have limitations in reaching a global audience, with restrictions on cross-border transactions, currency conversions, and language barriers. This can hinder the potential of fundraisers to access a wider donor base and raise funds internationally.

In summary, the need for BlockFund arises from the limitations of traditional crowdfunding platforms, including lack of Anonymity, high fees and intermediaries, limited access and control, lack of security and trust, lengthy payment processing times, and lack of global reach. BlockFund aims to address these challenges by leveraging blockchain technology to create a transparent, secure, and efficient peer-to-peer crowdfunding ecosystem that empowers fundraisers and donors alike.

* 1. **Feature of Proposed System**

The future of BlockFund, the proposed peer-to-peer blockchain-based crowdfunding system, holds promising potential for transforming the crowdfunding landscape. Here are some possible scenarios for the future of BlockFund:

* **Increased Anonymity:** BlockFund's use of blockchain technology provides a transparent and immutable ledger of all transactions, ensuring that fundraisers and donors can have increased visibility into how funds are being utilized. This Anonymity can enhance trust and confidence in the crowdfunding process, attracting more fundraisers and donors to the platform.
* **Lower Fees and Reduced Intermediaries:** With BlockFund's direct peer-to-peer approach, fundraisers can avoid intermediaries, such as banks or payment processors, and reduce fees associated with traditional crowdfunding platforms. This can result in more funds being received by fundraisers, enabling them to achieve their fundraising goals more effectively.
* **Enhanced Fundraiser Control:** BlockFund can empower fundraisers with greater control over their campaigns, allowing them to customize campaign content, set their own fundraising goals, and manage their campaigns independently. This can enable fundraisers to create more personalized and impactful fundraising campaigns that resonate with donors.
* **Improved Security and Trust:** BlockFund's use of blockchain technology can enhance security and trust in the crowdfunding process. The immutability and Anonymity of blockchain transactions can reduce the risks of fraud, identity theft, and other security breaches, providing a more secure fundraising environment for both fundraisers and donors.
* **Faster Payment Processing:** BlockFund's blockchain-based system can potentially streamline payment processing, eliminating delays associated with traditional crowdfunding platforms. This can provide fundraisers with faster access to funds, enabling them to implement their campaigns more efficiently.
* **Global Reach:** BlockFund's decentralized nature and use of blockchain technology can enable cross-border crowdfunding, allowing fundraisers to access a global donor base without the limitations of geographical restrictions, currency conversions, or language barriers. This can expand fundraising opportunities for projects with international appeal.

In conclusion, the future of BlockFund looks promising, with potential for increased Anonymity, lower fees, enhanced fundraiser control, improved security and trust, faster payment processing, and global reach. By leveraging the benefits of blockchain technology, BlockFund has the potential to revolutionize the crowdfunding landscape and provide a trusted and efficient platform for fundraisers and donors.

* 1. **Need for Computerization**
* Efficiency and Automation
* Scalability and Accessibility
* Data Management and Analysis
* Anonymity and Security
* User Experience and Convenience
* Compliance and Reporting

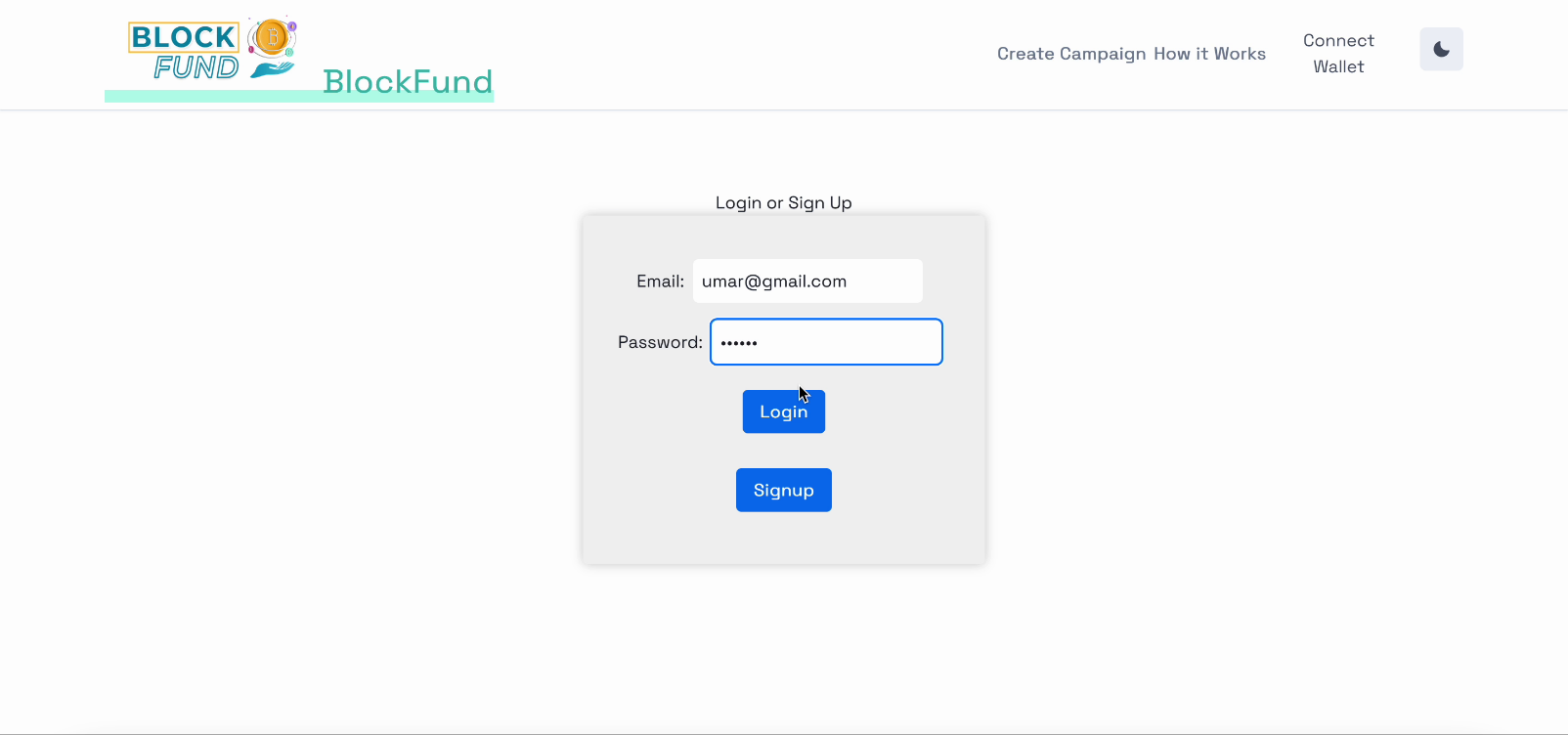
**CHAPTER 5: SYSTEM DESIGN**

**Brief Description**

The system design of BlockFund, a trusted peer-to-peer blockchain-based crowdfunding platform, involves several components and functionalities that work together to provide a seamless and secure crowdfunding experience. Here are some key aspects of the system design:

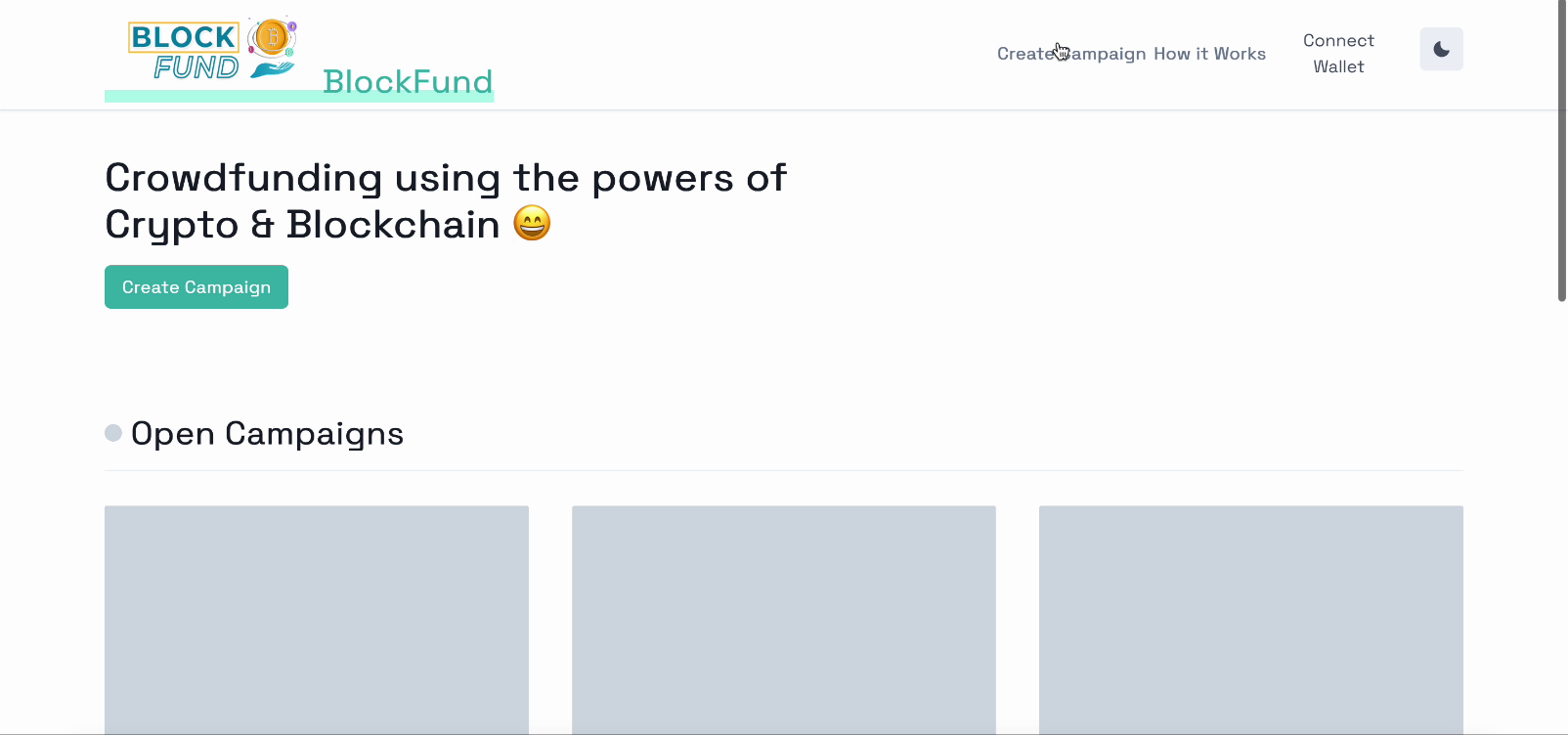
* **Blockchain Technology:** BlockFund leverages blockchain technology to provide Anonymity, immutability, and security to the crowdfunding process. The platform utilizes a decentralized blockchain network to record all transactions and campaign data, ensuring that they are securely stored and cannot be altered or tampered with.
* **User Registration and Authentication:** BlockFund requires users to register and create accounts to participate in the crowdfunding process. The system design includes user registration and authentication processes to ensure that only legitimate users can create and manage campaigns, make donations, and track their contributions. Authentication mechanisms such as username/password, two-factor authentication, or biometric authentication can be implemented to enhance security.
* **Campaign Creation and Management:** BlockFund allows fundraisers to create and manage crowdfunding campaigns. The system design includes features for fundraisers to create campaign pages, set fundraising goals, upload campaign details, images, and videos, and manage campaign updates. Fundraisers can also track the progress of their campaigns and communicate with donors through the platform.
* **Donation Processing and Payment Integration:** BlockFund includes functionality for donors to make contributions to crowdfunding campaigns. The system design integrates with payment gateways or cryptocurrency wallets to facilitate secure and seamless donation processing. Donors can select their preferred payment method, and the system ensures that transactions are recorded on the blockchain and reflected in the campaign progress.
* **Anonymity and Accountability:** BlockFund's system design emphasizes Anonymity and accountability. All transactions and campaign data are recorded on the blockchain, creating a transparent and immutable ledger that can be audited by stakeholders. This ensures that funds are used as intended, and campaign progress can be verified by donors and fundraisers alike.
* **Communication and Notifications:** BlockFund includes communication features such as messaging and notifications to facilitate communication between fundraisers and donors. Fundraisers can provide updates, express gratitude, and respond to donor inquiries through the platform. Donors can receive notifications about campaign updates, donation confirmations, and other relevant information.
* **Reporting and Analytics:** BlockFund's system design includes reporting and analytics capabilities to provide insights into campaign performance, donor behavior, and other relevant metrics. This information can be used for data-driven decision-making, platform optimization, and compliance with reporting requirements.
* **Security and Privacy:** BlockFund's system design incorporates robust security measures to protect user data, transactions, and the integrity of the platform. This includes encryption of data, authentication mechanisms, regular security audits, and adherence to relevant privacy regulations.
* **Mobile and Desktop App Interfaces:** BlockFund is designed as both a mobile and desktop app, with user-friendly interfaces that provide a seamless experience for both fundraisers and donors. The system design ensures that the platform is responsive, accessible, and optimized for different devices and screen sizes.
  1. **Desktop App UI/UX Design Pages**

### Login / Sign up Page



**Fig 5.1:** Login / Sign up Page

### Home Page

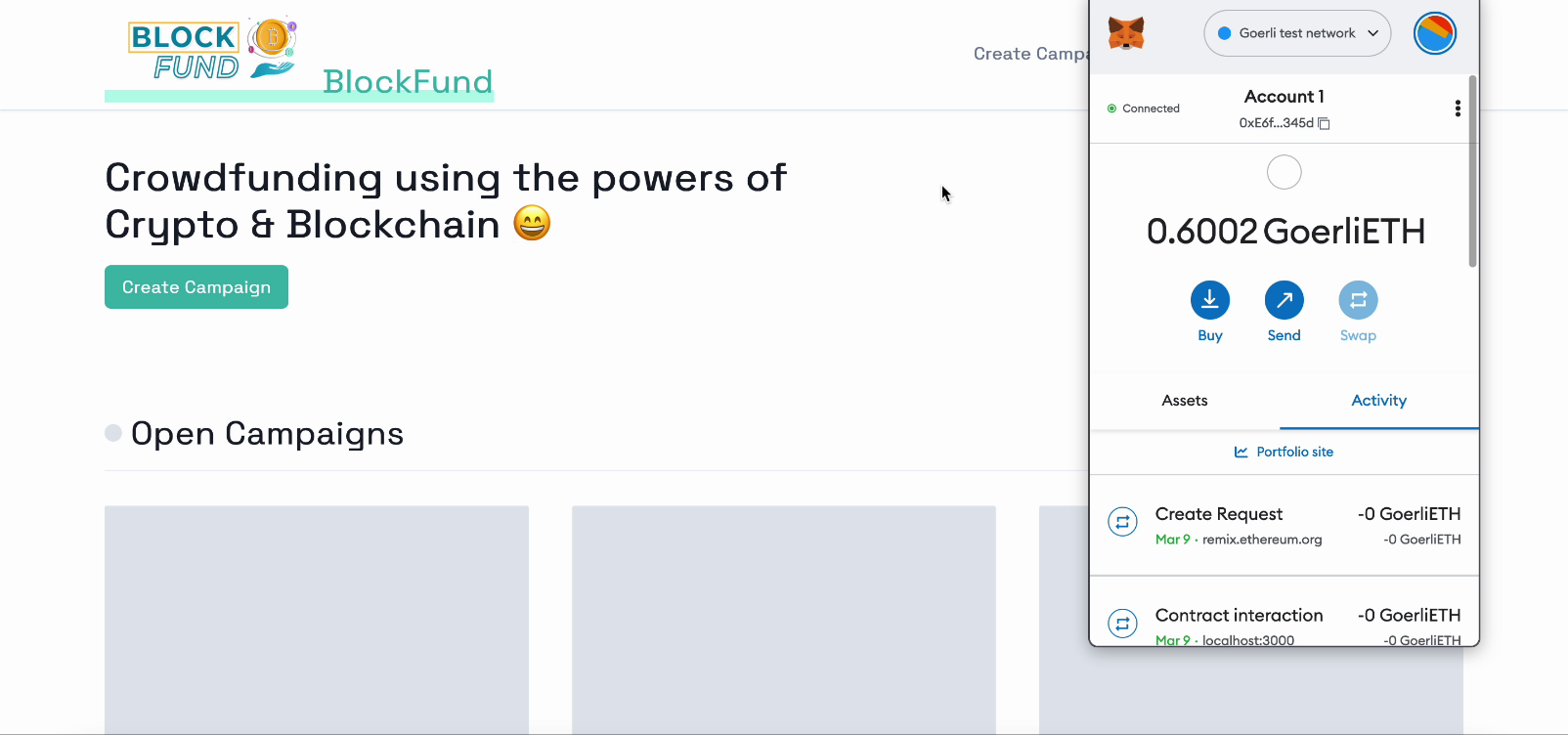


**Fig 5.2:** Home Page

### Manual Page

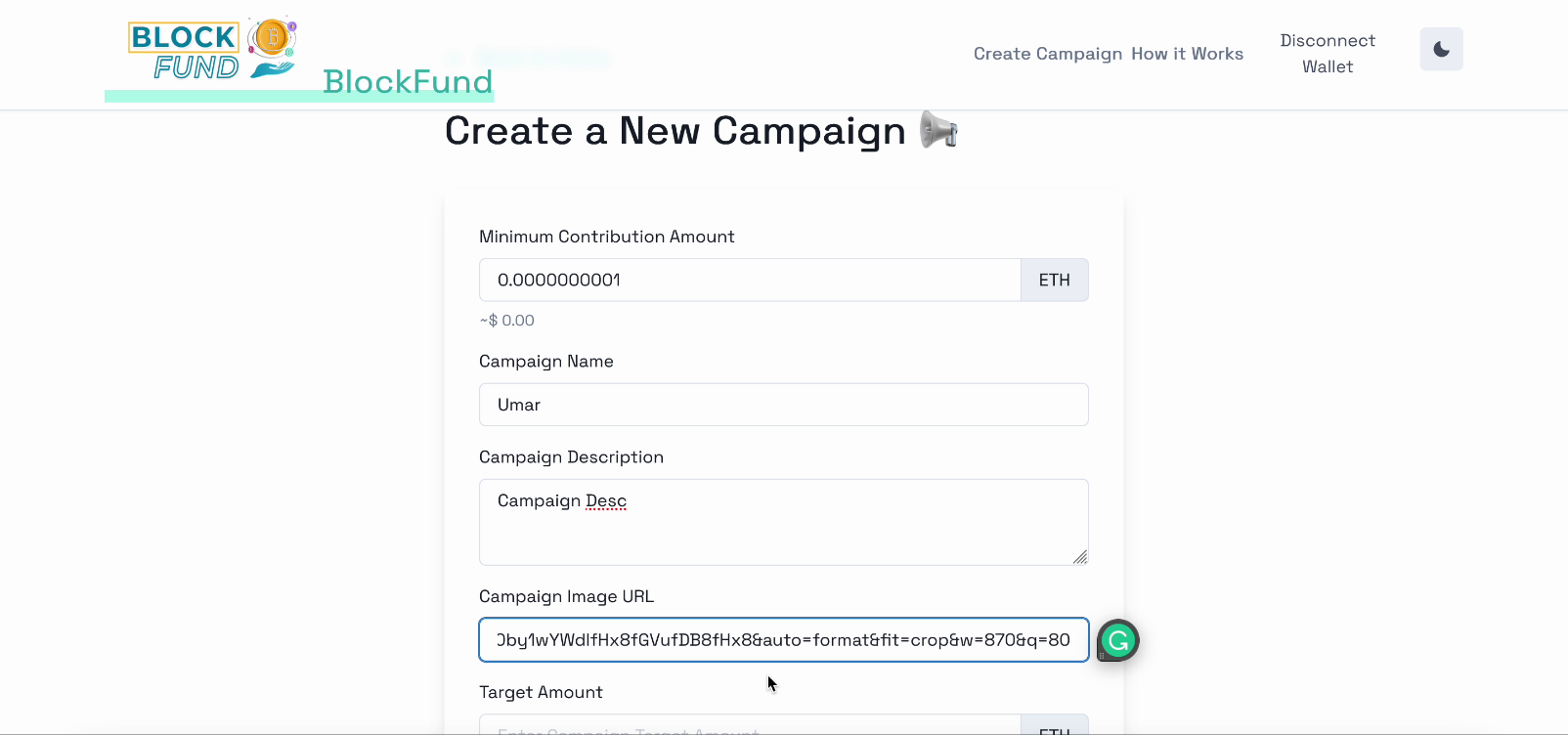
**Fig 5.3:** Manual Page

### Wallet Page



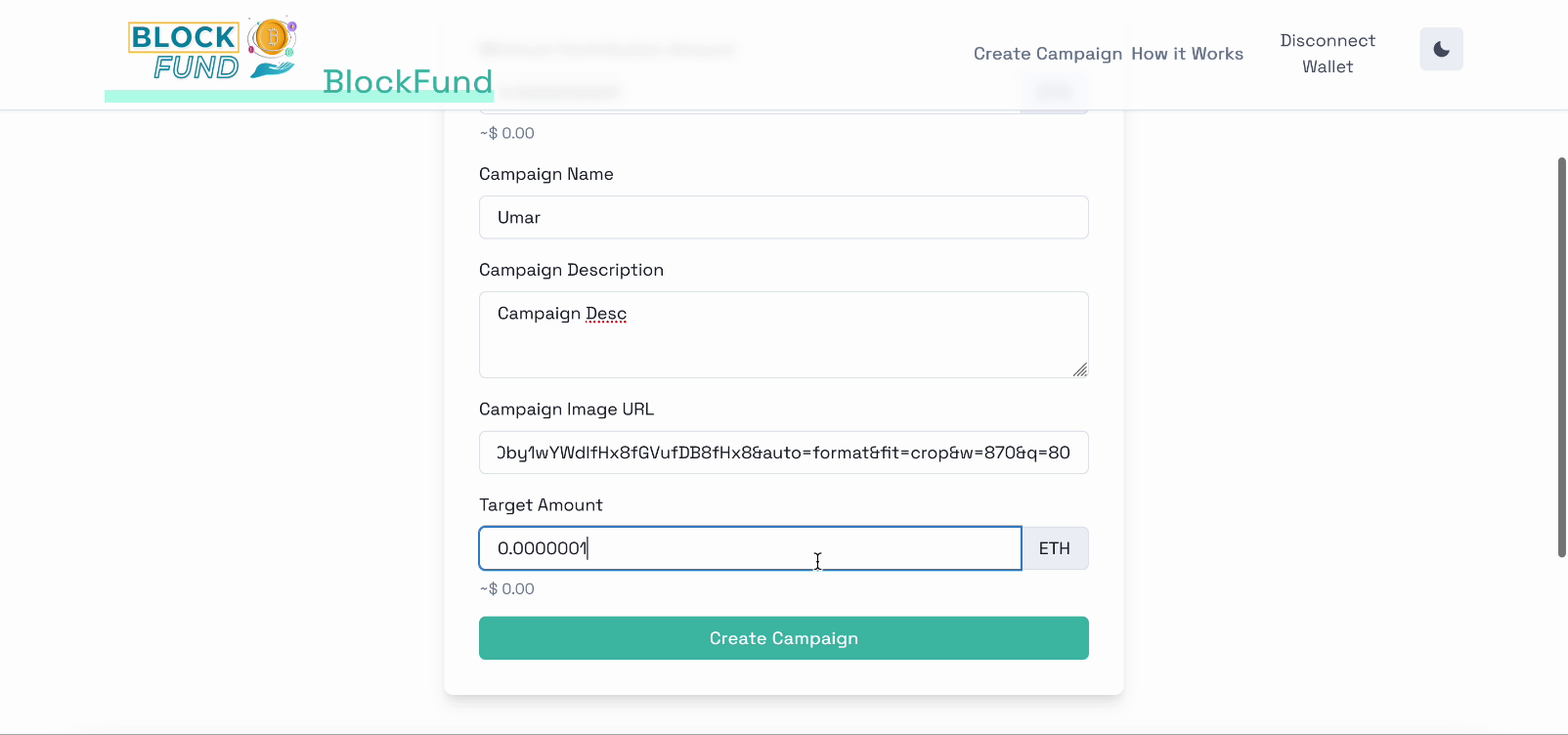
**Fig 5.4:** Wallet Page

### New Campaign Page



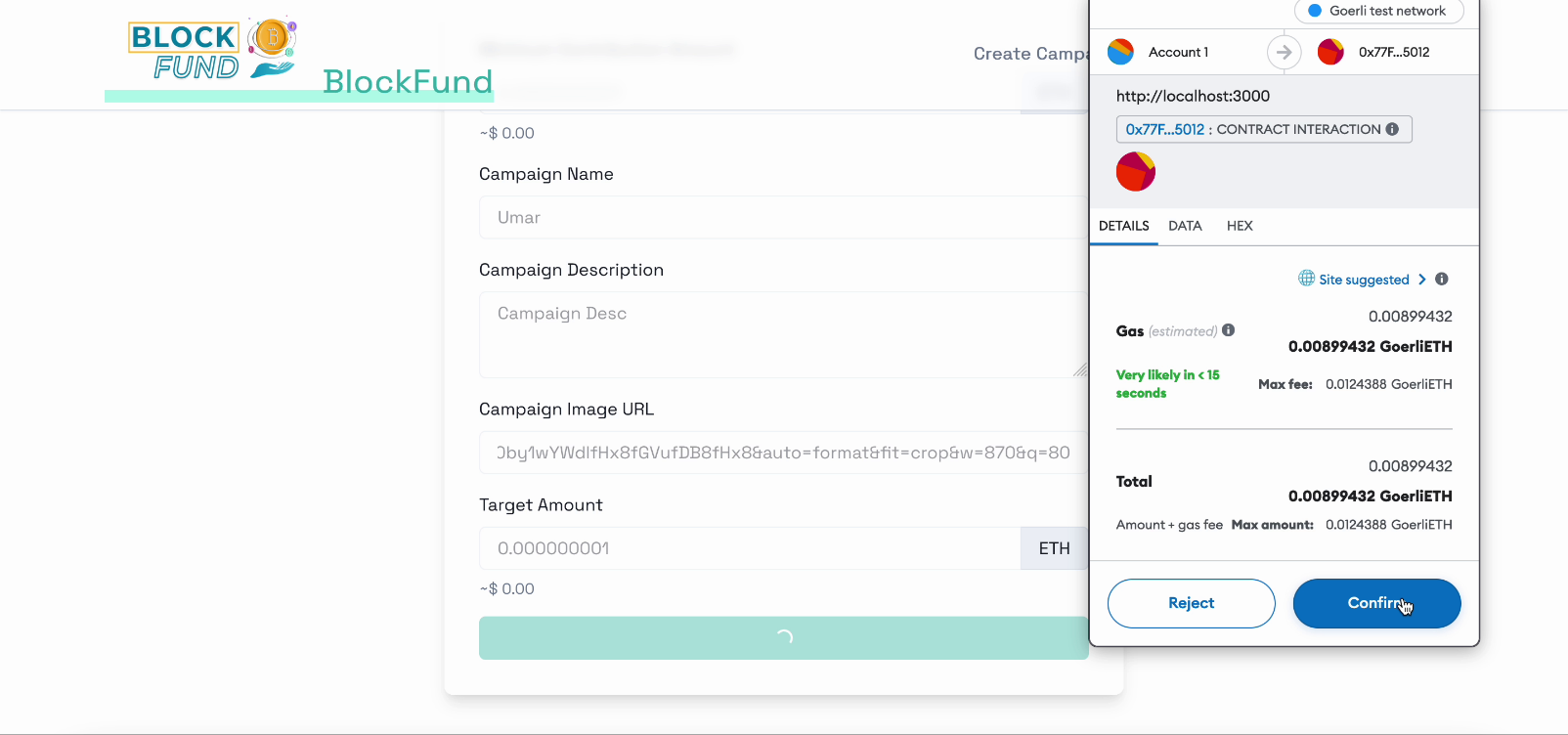
**Fig 5.5:** New Campaign Page

### New Campaign Page



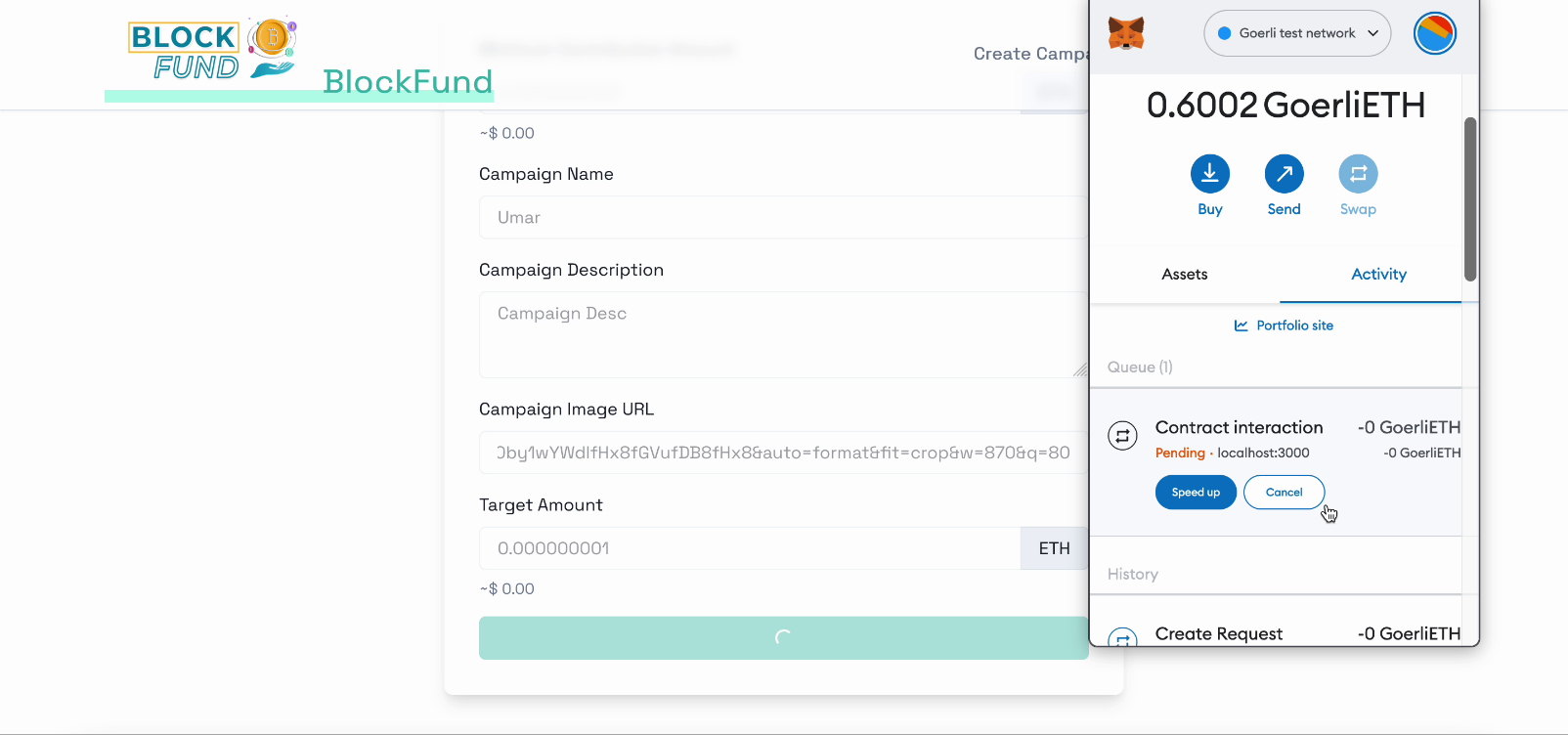
**Fig 5.6:** New Campaign Page

### Confirmation Page



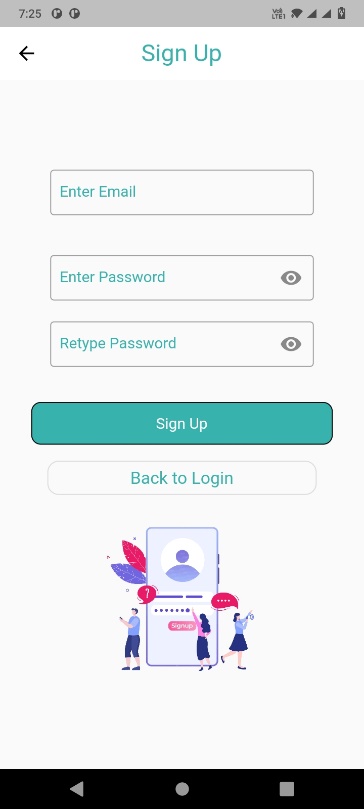
**Fig 5.7:** Confirmation Page

### Contract Page

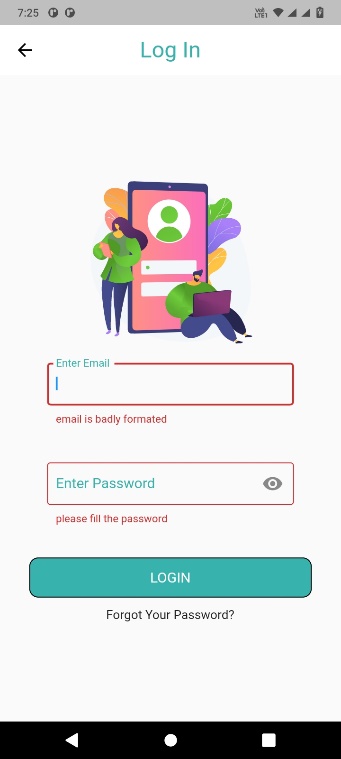


**Fig 5.8:** Contract Page

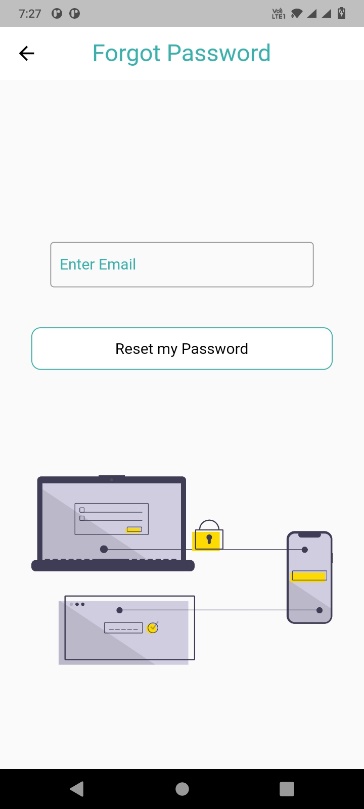
* 1. **Mobile App UI/UX Designs**

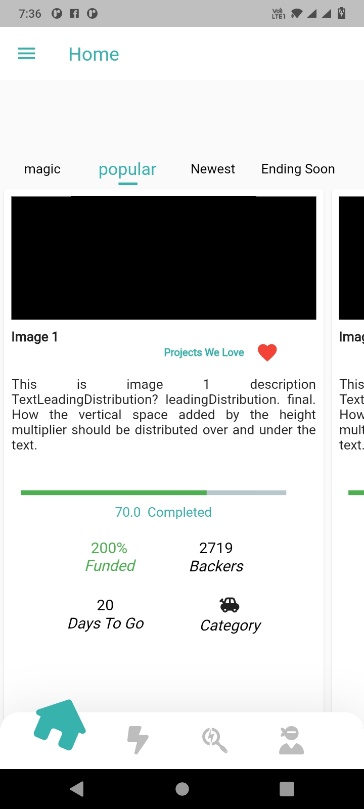
****

**Fig 5.9:** Sign up Screen

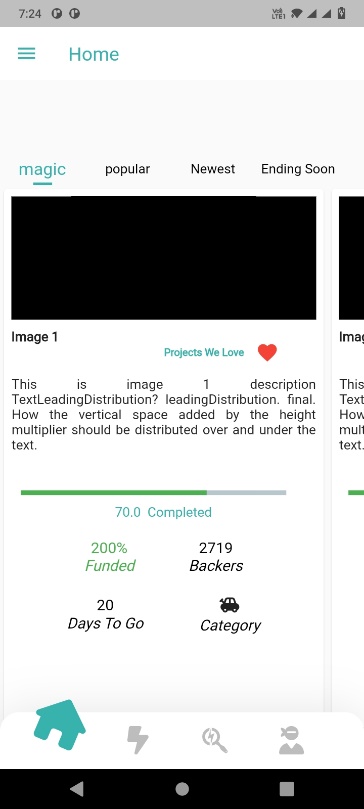
****

**Fig 5.10:** Login Screen

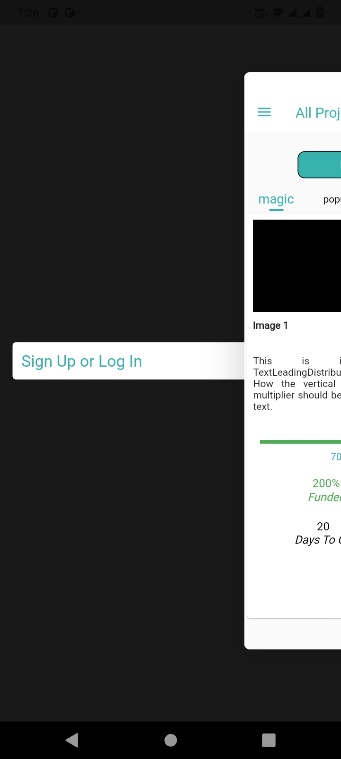
****

**Fig 5.11:** Forgot Password Screen

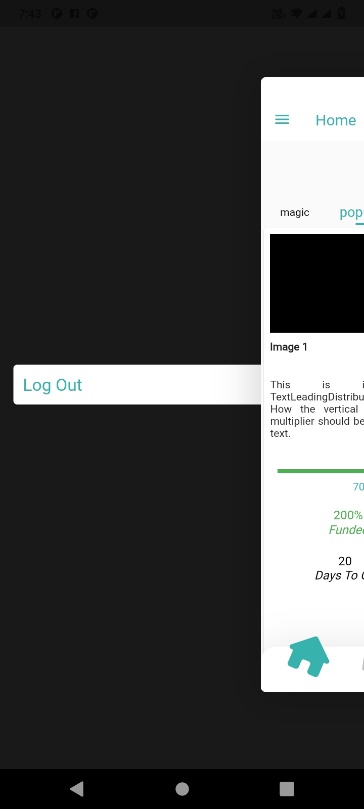
**Fig 5.12:** Home Screen after login

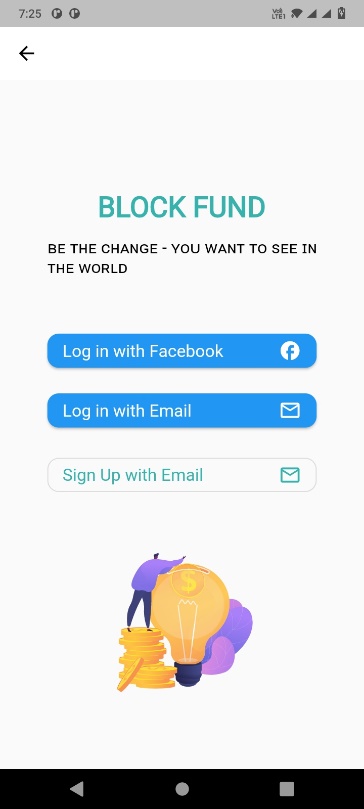
****

**Fig 5.13:** Magic Screen of home after login

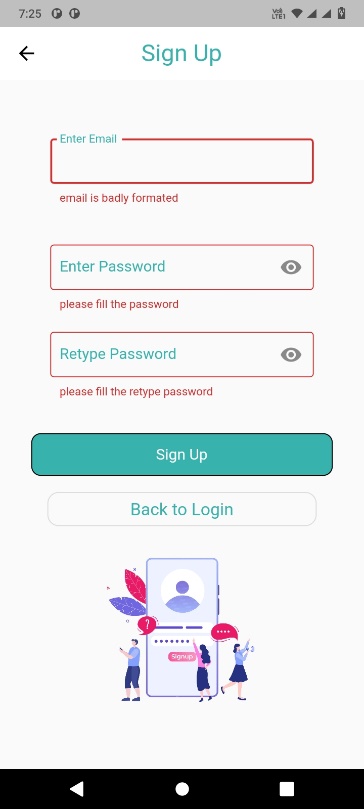
****

**Fig 5.14:** Drawer Screen before sign up

****

**Fig 5.15:** Drawer Screen after login

**Fig 5.16:** Log in and sign up options

****

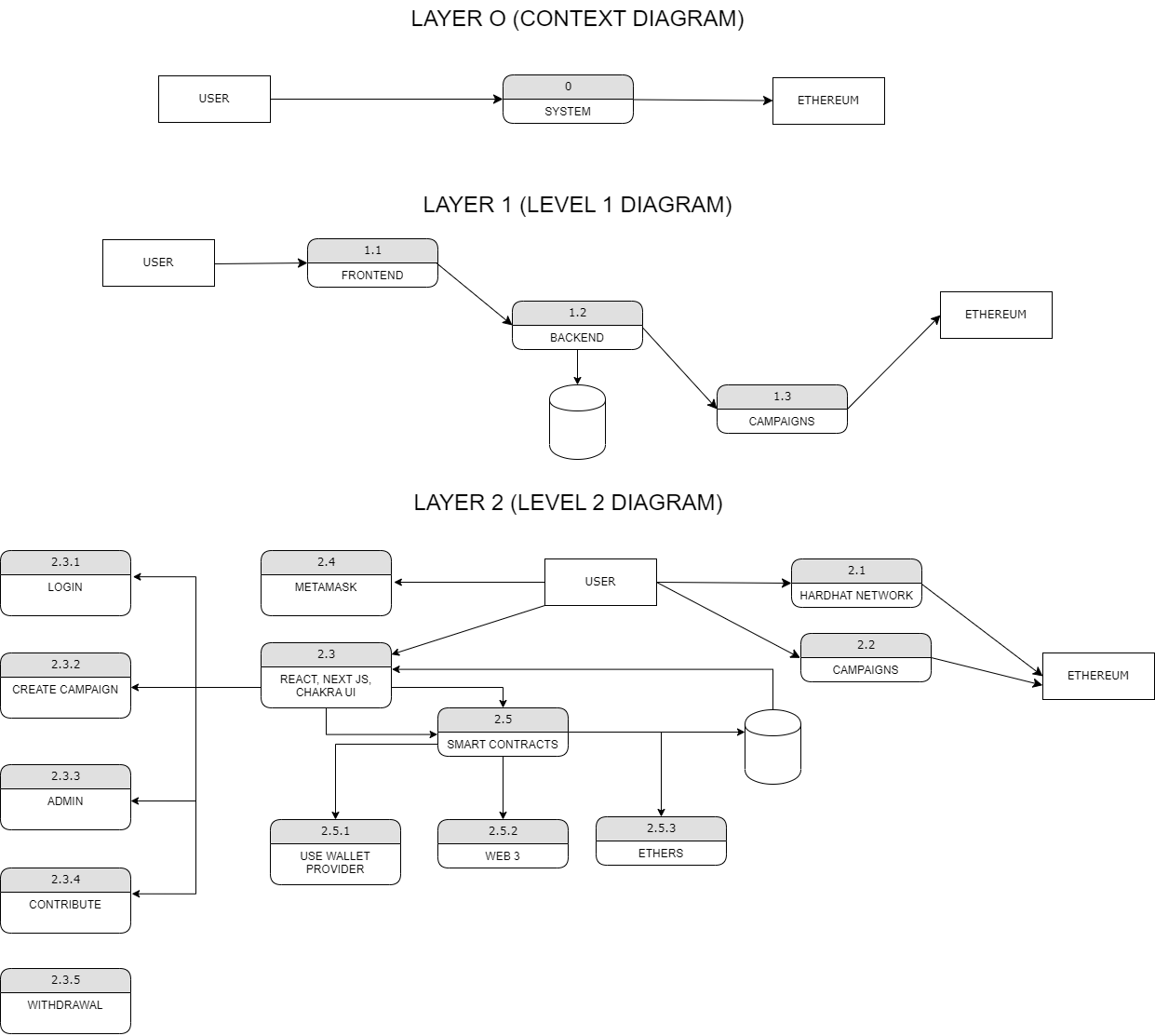
**Fig 5.17:** Sign up Screen with regex

* 1. **DFD Diagram for Block Fund**

A data flow diagram is a graphical view of how data is processed in a system in terms of input and output. The Data flow diagram (DFD) contains some symbol for drawing the data flow diagram.

|  |
| --- |
| **Level 0 DFD:** |
| * A level 0 DFD represents the highest level of abstraction and provides a broad overview of the system. It typically consists of a single process symbol representing the entire system. |
| * The level 0 DFD shows the external entities that interact with the system and the data flows between these entities and the system. |
| * The data flows in a level 0 DFD represent high-level information exchanges between different components of the system. |
| **Level 1 DFD:** |
| * A level 1 DFD provides a more detailed view of the system by breaking down the main process of the level 0 DFD into subprocesses. |
| * The level 1 DFD expands the main process symbol of the level 0 DFD into multiple processes, each representing a specific task or operation. |
| * It shows the data inputs and outputs for each process, as well as the data flows between the processes. |
| **Level 2 DFD:** |
| * A level 2 DFD further decomposes the processes of the level 1 DFD into more detailed subprocesses. |
| * It breaks down the processes from level 1 into smaller, more manageable tasks. |
| * The level 2 DFD provides a more comprehensive understanding of the system by showing the internal processes and data flows within each subprocess. |

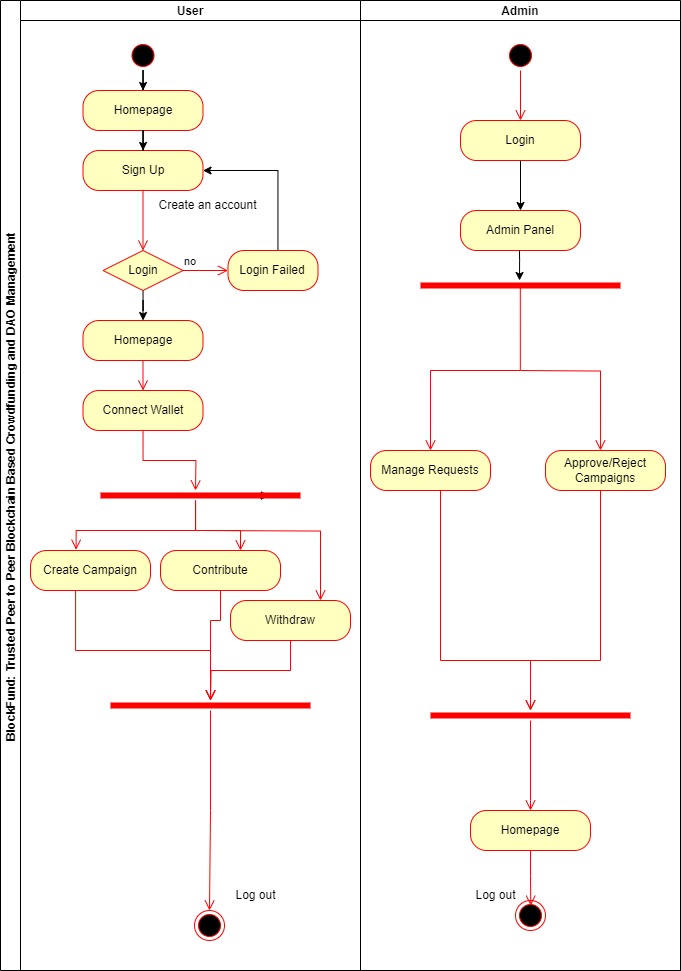
**Table 5.1:** DFD



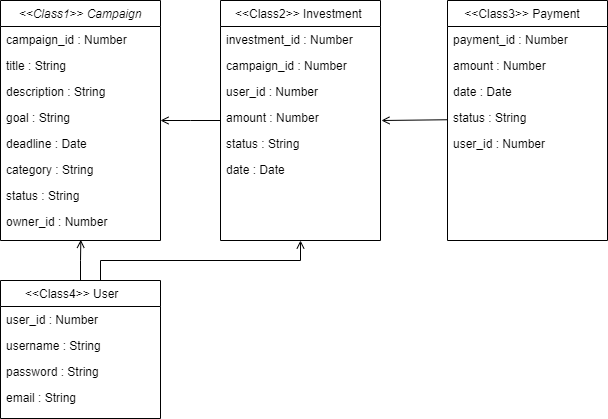
**Fig 5.18:** DFD Diagram

**5.4 Activity Diagram for Block Fund**

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.

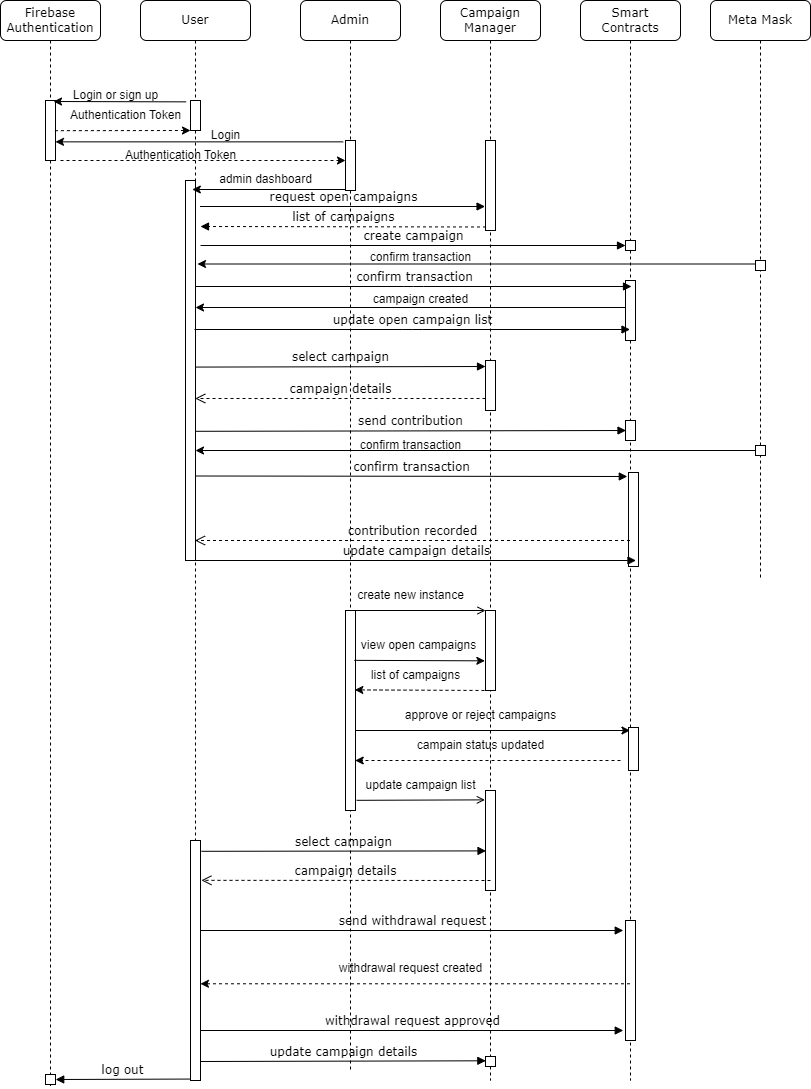


**Fig 5.19:** Activity Diagram

**5.5 Class Diagram for Block Fund**

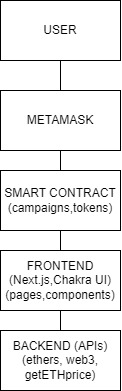
**Fig 5.20:** Class Diagram

**5.6 Sequence Diagram for Block Fund**

****

**Fig 5.21:** Sequence Diagram

**5.7 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 5.22:** Architectural Diagram

**CHAPTER 6: DEVELOPMENTAND CODING**

* 1. **Development And Coding**

In the context of the BlockFund project, coding would involve the development of software code for the implementation of the desktop and mobile app. This would include writing code in various programming languages and using relevant frameworks and libraries to build the different components and features of the app.

The coding process for the BlockFund app would likely involve the following steps:

* **Front-end Development:** This involves designing and implementing the user interface (UI) of the app, including screens, forms, buttons, and other user interactions. Front-end technologies such as HTML, CSS, and JavaScript would be used to create the visual elements and user experience of the app.
* **Back-end Development:** This involves building the server-side components of the app that handle data storage, processing, and communication with external APIs or blockchain networks. Back-end technologies such as server-side programming languages (e.g., Python, Node.js, or PHP), databases (e.g., MySQL, MongoDB), and web frameworks (e.g., Express, Flask) may be used to implement the logic and functionality of the app.
* **Blockchain Integration:** Since BlockFund is based on blockchain technology, coding would involve integrating with the relevant blockchain platform (e.g., Ethereum) to perform transactions, create and manage smart contracts, and interact with the blockchain network.
* **Security Measures:** Coding for the BlockFund app would also involve implementing security measures to protect user data, funds, and transactions. This may include encryption, authentication, and other security best practices to ensure the confidentiality, integrity, and availability of the app and its data.
* **Testing and Debugging:** Throughout the coding process, thorough testing and debugging would be carried out to identify and fix any potential bugs or vulnerabilities in the app. This would involve writing test cases, conducting testing, and resolving issues to ensure the quality and reliability of the app.
* **Documentation:** Proper documentation of the code would also be an important aspect of the coding process, including comments in the code, user manuals, and technical documentation to help with future maintenance, updates, and troubleshooting.
* Overall, coding for the BlockFund app would involve the development of software code using relevant programming languages, frameworks, and libraries to implement the front-end and back-end components, integrate with the blockchain platform, implement security measures, conduct testing and debugging, and properly document the code for future reference.
  1. **Functional Requirements**

**User authentication and authorization:**

* Secure registration and login system for users and administrators.
* Role-based access control to manage user permissions.

**Campaign creation:**

* Allow users to create and submit new campaigns for fundraising.
* Define campaign goals, descriptions, deadlines, and other relevant information.

**Campaign management:**

* Allow administrators to approve or reject submitted campaigns.
* Enable administrators to monitor and manage active campaigns.
* Provide users with the ability to edit or delete their own campaigns.

**Donation functionality:**

* Allow users to donate to approved campaigns.
* Implement secure payment processing with a cryptocurrency like Ethereum.
* Display progress towards the campaign's fundraising goal.

**User interface (UI) and user experience (UX):**

* Design a responsive UI for both desktop and mobile devices.
* Implement a clear, user-friendly navigation system.
* Integrate Chakra UI components for consistent styling.

**Search and filtering:**

* Implement search functionality for users to find specific campaigns.
* Allow users to filter campaigns by various criteria, such as deadline, goal, or category.

**Notifications and alerts:**

* Notify users of successful donations or changes to their campaigns.
* Alert administrators of new campaign submissions or other critical events.

**Reporting and analytics:**

* Generate reports on campaign performance for administrators.
* Provide analytics on user engagement, donation patterns, and overall platform performance.
  1. **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 and campaigns.
* 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.
  1. **Selection of tools and technology**
     + Hardware
     + Software

#### Graphical User Interface

* + - Login/Signup Page
    - Home Page
    - Manual Page
    - Wallet Page
    - New Campaign
    - Contract Page
  1. **System Specification**

#### Requirement Analysis

The Block Fund System Requirements provides high-level of functional requirements.

The functions of the system are discussed below.

Admin can directly login this system but the user person used to register to login this system:

* Admin.
* User

#### Software Requirement Specification

Elements of the DSM Supplementary Specification:

* Documentation (user, installation, administration) and help.
* Licensing and other legal concerns.
* Packaging.
* Standards (technical, safety, quality).
* Physical environment concerns (for example, heat or vibration).
* Operational concerns (for example, how do errors get handled, or how often should backups be done?).
* Application-specific domain rules information in domains of interest (for example, what is the entire cycle of credit payment handling?)

**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.

* 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.

* 1. **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 login using Invalid Credentials |
| Pre-Condition | User must have access to Login Feature |
| Test Case Name | Login to BlockFund |
| Steps | 1. Enter Username  2. Enter Password  3. Click "Login" button  **Expected Result:**  **Alert**: Wrong username or Invalid Password!  **Actual Result:**  Wrong username or Invalid Password!  **Status:** Passed |

**Table 7.1:** Test Case TC001 login to block Fund

| Test Case No: | TC002 |
| --- | --- |
| Test Case Description | Creating a campaign with complete information |
| Pre-Condition | User must be logged in and has access to the "Create Campaign" feature |
| Test Case Name | **Create Campaign** |
| Steps | 1. Navigate to "Create Campaign" section  2. Enter all required campaign details  3. Click "Create Campaign" button  **Expected Result:**  - Alert: Campaign successfully created!  **Actual Result:**  - To be filled after test execution  **Status:** Passed |

**Table 7.2:** Test Case TC002 Create Campaign

| Test Case No: | TC003 |
| --- | --- |
| Test Case Description | Funding a campaign with valid amount |
| Pre-Condition | User must be logged in and has sufficient balance in their wallet |
| Test Case Name | **Fund Campaign** |
| Steps | 1. Select a campaign  2. Enter valid funding amount  3. Click "Fund Campaign" button  **Expected Result:**  **Alert**: Funding successful!  **Actual Result:**  To be filled after test execution  **Status:** Passed |

**Table 7.3:** Test Case TC003 Fund Campaign

| Test Case No: | TC004 |
| --- | --- |
| Test Case Description | Withdrawing funds to Metamask wallet with valid details |
| Pre-Condition | User must be logged in and has sufficient balance in their campaign wallet |
| Test Case Name | **Withdraw to Metamask Wallet** |
| Steps | 1. Click "Withdraw Funds" option  2. Enter valid Metamask address  3. Enter valid withdrawal amount  4. Confirm the transaction  **Expected Result:**  **Alert:** Withdrawal successful!  **Actual Result:**  To be filled after test execution  **Status:** Passed |

**Table 7.4:** Test Case TC004 withdraw to Metamask Wallet

| Test Case No: | TC005 |
| --- | --- |
| Test Case Description | Requesting funds with valid details |
| Pre-Condition | User must be logged in and has sufficient balance in their campaign wallet |
| Test Case Name | **Request Funds** |
| Steps | 1. Click "Request Funds" option  2. Enter valid reason for requesting funds  3. Enter valid amount  4. Send the request  **Expected Result:**  **Alert**: Request for funds sent successfully!  **Actual Result**:  To be filled after test execution  **Status:** Passed |

**Table 7.5:** Test Case TC005 request Funds

# 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: BF-TC001**

**Test Case Title: Create a New Campaign**

**Test Objective:** To verify if a user can successfully create a new crowdfunding campaign / Login using the BlockFund app.

**Test Steps:**

* Launch the BlockFund app.
* Log in with valid credentials or register a new account.
* Navigate to the "Create Campaign" feature.
* Fill in the required details for the campaign, such as campaign name, description, funding goal, campaign duration, and any additional information.
* Upload an image or video for the campaign (optional).
* Choose a blockchain-based crowdfunding option, such as Ethereum or Bitcoin, for fundraising.
* Click on the "Create Campaign" button to submit the campaign details.
* Verify if the campaign is successfully created and displayed in the app's campaign list.
* Verify if the campaign details, such as name, description, funding goal, campaign duration, and fundraising option, are correctly displayed.
* Logout from the app.

**Test Data:**

* Valid campaign details, such as campaign name: "Save the Oceans", description: "A crowdfunding campaign to raise awareness about ocean conservation", funding goal: $10,000, campaign duration: 30 days.
* Valid blockchain-based crowdfunding option, such as Ethereum.

**Expected Result:**

* The campaign is successfully created and displayed in the app's campaign list.
* The campaign details, such as name, description, funding goal, campaign duration, and fundraising option, are correctly displayed.
* The app allows the user to upload an image or video for the campaign (optional).

# 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: BF-TC002**

**Test Case Title: Verify Campaign Creation with Invalid Funding Goal**

**Test Objective:** To verify if the system correctly handles invalid input for the funding goal during campaign creation.

**Test Steps:**

* Launch the BlockFund app.
* Log in with valid credentials or register a new account.
* Navigate to the "Create Campaign" feature.
* Fill in the campaign details, such as campaign name, description, campaign duration, and select a valid fundraising option.
* Enter an invalid funding goal, such as a negative value, zero, or a non-numeric value, in the funding goal field.
* Click on the "Create Campaign" button to submit the campaign details.
* Verify if the system displays an error message indicating that the funding goal is invalid.
* Verify if the campaign is not created and remains on the campaign creation page for further corrections.
* Retry with valid funding goal values and verify if the campaign is created successfully.
* Logout from the app.

**Test Data:**

* Campaign details: Campaign name: "Test Campaign", description: "A crowdfunding campaign for testing purposes", campaign duration: 30 days.
* Invalid funding goal values: -1000, 0, "abc".

**Expected Result:**

* The system displays an error message indicating that the funding goal is invalid for all invalid funding goal values.
* The campaign is not created and remains on the campaign creation page for further corrections.
* The system allows the user to retry with valid funding goal values and creates the campaign successfully.

**CHAPTER 8: IMPLEMENTATION AND TRAINING**

* 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 HTML, CSS, and JavaScript for desktop and mobile devices.
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 BlockFund could involve market research, user testing, regulatory compliance, marketing, and launch strategies. The app could be marketed to potential campaign creators and backers, with campaigns promoted through social media, online advertising, and partnerships with relevant influencers or organizations. The app would need to comply with relevant crowdfunding regulations and maintain security measures to protect user data and transactions. Continuous monitoring, updates, and improvements would be necessary to ensure the app remains competitive in the market and meets user needs.

**Project Overview**

**Project Name:** *BlockFund: Trusted Peer-to-Peer Blockchain-Based Crowdfunding*

**Objective:** To develop a desktop and mobile app for peer-to-peer crowdfunding using blockchain technology, providing a trusted and transparent platform for campaign creators and backers.

**Scope:** The app will allow campaign creators to create and manage crowdfunding campaigns, set funding goals, and offer incentives to backers. Backers can browse and support campaigns using cryptocurrency transactions on the blockchain. The app will include features such as user authentication, campaign creation and management, transaction tracking, and smart contracts for secure and transparent crowdfunding.

**Key Features:**

* **User authentication:** Users will be able to create accounts, login, and manage their profiles.
* **Campaign creation and management:** Campaign creators can create campaigns, set funding goals, upload campaign details, and manage campaigns.
* **Campaign browsing:** Backers can browse and search for campaigns based on various criteria such as category, funding status, and popularity.
* **Cryptocurrency transactions:** Backers can support campaigns by making cryptocurrency transactions on the blockchain.
* **Smart contracts:** Smart contracts will be used to ensure transparent and secure crowdfunding, with funds released to campaign creators only when funding goals are met.
* **Notifications:** Users will receive notifications about campaign updates, funding milestones, and other relevant information.

**Target Users:** Campaign creators, backers, and supporters interested in participating in peer-to-peer crowdfunding using blockchain technology.

**Technologies Used:** Blockchain, technologies (NextJS, JavaScript, Web3.JS, Flutter, Dart, Solidity, MetaMask, 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 crowdfunding using blockchain technology.

**Project Timeline:** The project timeline will depend on the scope and complexity of the app, and may include 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.

* 1. **Training**

As a peer-to-peer crowdfunding app based on blockchain technology, BlockFund does not require any traditional training data or machine learning models for its operation. However, the development team responsible for building the BlockFund app will need to have expertise Blockchain, technologies (NextJS, JavaScript, Web3.JS, Flutter, Dart, Solidity, MetaMask, 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 BlockFund app.

# Conclusion

In conclusion, the BlockFund project aims to develop a trusted peer-to-peer crowdfunding app based on blockchain technology. The app will provide a transparent and secure platform for campaign creators and backers, allowing them to participate in crowdfunding campaigns using cryptocurrency transactions on the blockchain.

The project overview includes key features such as user authentication, campaign creation and management, campaign browsing, cryptocurrency transactions, and smart contracts for secure and transparent crowdfunding. The expected outcome is a fully functional, secure, and user-friendly desktop and mobile app that meets the requirements and goals of the project.

The project will require a skilled development team with expertise in blockchain, front-end and back-end technologies, security measures, and testing. Training and updates on the latest technologies and best practices will be necessary to ensure a successful implementation of the app.

Overall, the BlockFund project has the potential to revolutionize the crowdfunding industry by leveraging the Anonymity and security of blockchain technology, providing a trusted platform for crowdfunding campaigns, and empowering both campaign creators and backers to participate in a secure and transparent crowdfunding ecosystem.

# References

* <https://krazytech.com/projects/sample-software-requirements>
* [http://softwaretestingfundamentals.com](http://softwaretestingfundamentals.com/)
* [https://devblogs.microsoft.com](https://devblogs.microsoft.com/)
* [https://www.lucidchart.com](https://www.lucidchart.com/)

[http://web.uettaxila.edu.pk/CMS/AUT2011/seSCbs/tutorial/Electronic\_Book(UML%](http://web.uettaxila.edu.pk/CMS/AUT2011/seSCbs/tutorial/Electronic_Book(UML%2024%20Hours).pdf)

[2024%20Hours).pdf](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\_](http://www.visualparadigm.com/support/documents/vpuserguide/3563/3564/85375_)

drawingentit.html