SPL-02 Project Report

CharityGuard

A Decentralized Charity Donation Platform

Submitted By

Abb Muhaimeen

(BSSE Roll No: 1423, BSSE Session: 2021-2022)

Md. Tamim Hasan Toha

(BSSE Roll No: 1424, BSSE Session: 2021-2022)

Supervised By

Dr. Mohammed Shafiul Alam Khan

Professor

Institute of Information Technology

Date of Submission: 9th March, 2025

Table Of Content

List of Figures

Use Case Diagram

Swim Lane Diagram

Data Based Modeling

Class Based Modeling

Behavioral Modeling

State Transition Diagram

Sequence Diagram

Data flow Modeling

User Manual

1. Introduction

CharityGuard is a blockchain-based platform that makes charitable giving more transparent and trustworthy. It uses smart contracts on the Ethereum blockchain to ensure donations go directly to the right people, eliminating the need for middlemen (Buterin, 2014). Many traditional charities face problems such as inefficiencies, lack of transparency, and trust issues. CharityGuard solves these by allowing donors to send funds directly to recipients, ensuring donations are used properly (Nakamoto, 2008).

The platform offers key benefits like transparency, trust, efficiency, and accountability. Every transaction is recorded on the blockchain, making it easy for anyone to verify. Smart contracts automatically distribute funds based on predefined conditions, reducing administrative costs. This report covers the background of CharityGuard, challenges faced during development, possible future improvements, and key takeaways.

Transparency: All transactions are recorded on the blockchain and are publicly verifiable
Trust: Smart contracts ensure that funds are disbursed according to predefined conditions
Efficiency: Direct donor-to-recipient transfers reduce administrative overhead
Accountability: Campaign creators receive notifications about donations and milestones

2. Background of the project

CharityGuard was created to solve common problems in the charitable sector, such as fund mismanagement, delays in distribution, and high administrative costs. Blockchain technology helps by making all transactions permanent, traceable, and tamper-proof (Wood, 2015).

The platform runs on Ethereum and connects with MetaMask for secure transactions. There are three main types of users:

Donors: Can browse campaigns, donate, track contributions, and receive updates.

Fundraisers: Can create and manage campaigns, set goals, and track progress.

Charities: Have additional control over multiple recipients for a campaign.

To use the platform, users must sign up, verify their email, and connect a smart wallet. Campaigns include goals, deadlines, and recipients, ensuring secure transactions through blockchain (Gudgeon et al., 2020).

Direct Stakeholders

1. **Donors**: Individuals contributing to campaigns using cryptocurrencies or fiat payments.
2. **Charities**: Organizations creating campaigns to raise funds for their causes.
3. **Fundraisers**: Individuals or entities collaborating with charities to create and manage campaigns.
4. **Recipients**: End beneficiaries receiving the funds disbursed by the platform.

Indirect Stakeholders

1. **Developers**: Engineers maintaining and enhancing the platform's features and blockchain infrastructure.
2. **Payment Gateway Partners**: Organizations enabling fiat-to-crypto conversion and facilitating seamless payment processing (e.g., MoonPay).
3. **Email Service Providers**: Services handling communication, such as account verification and notifications.
4. **Auditors and Regulators**: External entities ensuring transparency, compliance, and legality of the platform's operations.
5. **Blockchain Network Participants**: Miners and validators involved in maintaining the blockchain that underpins the system.
6. **Data Analysts**: Individuals leveraging platform analytics for performance insights and reporting.
7. **General Public**: Individuals interested in the platform's social impact reports and success stories.

Requirements Collection

To ensure that the platform meets the expectations of its users and addresses gaps in traditional systems, requirements were gathered from stakeholders through targeted questions:

1. **Transparency Issues in Traditional Charities and Reports**
   Stakeholders highlighted the lack of verifiable transaction details and reporting mechanisms in traditional charities. This reinforced the need for blockchain-based systems to enhance transparency, allowing donors to track contributions and ensuring that fund disbursements are both verifiable and immutable.
2. **Adoption of Cryptocurrency by the General Public**
   Concerns were raised about the complexity of cryptocurrency adoption for non-technical users. Stakeholders suggested that incorporating a fiat gateway and providing educational resources could help bridge the gap and ease the transition from traditional payment systems to blockchain-based donations.
3. **Additional Suggestions**
   Stakeholders emphasized the importance of a notification system for real-time updates, detailed donation impact reports, and a decentralized governance system to empower users in decision-making. They also recommended intuitive user interfaces and robust security measures to build trust.

3. QFD

Stakeholders:

- Individual donors
- Charity foundations
- Fundraisers
- Recipients
- Corporate Recipients
- Community

Normal Requirements:

Profile and Campaign Management:

Users can create, update, and manage their profiles. Charities can create and manage campaigns with details such as titles, descriptions, categories, goals, and deadlines. Users can view and search for campaigns for easier accessibility.

History Tracking:

Donation histories are accessible for users, while charities can track all received donations.

Notifications and Updates:

Notifications are sent to users for events such as new campaigns, donation confirmations, campaign milestones, and fund disbursements, ensuring they stay informed.

Expected Requirements:

Blockchain Integration:

Donations are recorded on the blockchain. Users can connect their cryptocurrency wallets to the platform. Users can view their donation history.

Donation Processing:

Donations can be made using various cryptocurrencies. Transactions are recorded on the blockchain. Smart contracts handle automatic fund disbursements upon campaign goal completion.

Auditory Reports:

Charities can track the social impact of donations, providing detailed reports on fund usage and benefits.

Security and Data Protection:

Strong security measures, including encryption, are implemented to safeguard user data and funds, enhancing platform reliability.

Exciting Requirements:

Community Governance:

A decentralized governance system allows users to propose new features and vote on platform changes.

Incentives for Engagement:

Users earn badges and rewards for active participation.

Fiat Gateway Integration:

Non-cryptocurrency users can contribute through fiat gateways like MoonPay for traditional payments.

4. User Story

Account Management Module

The platform requires unique, verified accounts for users, categorized into Donor and Charity profiles.

1. **Donor Profile**: New donors register by providing an email, creating a secure password, and connecting their smart wallet. They confirm via an email link, completing the setup to access the dashboard.
2. **Charity Profile**: Charities apply and undergo verification by the platform's DAO. Upon approval, they will receive a temporary password and be able to set up their account.

Campaign Management Module

Charities can create, manage, and update campaigns, keeping donors engaged and informed.

1. **Creation**: Charities and Fundraisers set campaign details, goals, milestones, and initiate a smart contract for donations.
2. **Updates**: Campaign details can be updated to keep donors informed, who receive real-time notifications of any changes.
3. **Community Governance**: The Campaign owner can create polls to let the donor participate in decision-making.

Donation Processing Module

Enables secure donations using smart contracts for transparency.

1. **Initiation**: Donors select a campaign, choose donation amount and cryptocurrency, connect their wallet, and confirm, recording it on the blockchain.
2. **Processing & Disbursement**: Smart contracts verify and transfer funds to the charity upon reaching campaign goals. If goals are unmet, refunds or reallocation apply.
3. Receipt & History: Donation history, receipts, and campaign impact are visible in the donor's profile.
4. **Donor Interaction**: Donors can view detailed transactions of a campaign.
5. **Closure**: Campaigns automatically close upon reaching goals or deadlines, triggering fund disbursement or refunds if needed. Donors can view the full campaign history and donation impact.

Notification System Module

Keeps users informed of platform activities to enhance transparency.

1. **Campaign and Donation Updates**: Notifications are sent for new campaigns, donation confirmations, and campaign milestones.
2. **Disbursement and Security Alerts**: Alerts for fund disbursements and security events, such as unrecognized logins, keep users aware and engaged. Recipients will get notified about their funds when its disbursed to them.

Reporting and Analytics Module

Supports insights for donors, charities,fundraisers and recipients.

1. **Donation & Performance Reports**: Donors can download detailed donation history. Charities track campaign metrics for fundraising improvement.
2. **Social Impact Tracking**: Charities report donation impact, giving donors visibility into their contributions' real-world effects.

5. Use Case Diagram

A Use Case describes the system's behavior under various conditions as the system responds to a request from one of its stakeholders. In fact, a use case diagram is a kind of visualization of the system where an end-user has an idea of

a specific feature. It simply describes a story using corresponding actors who perform important roles in the story and make the story understandable for the users.

The first step in writing a Use Case is to define the set of “actors” that will be involved in the story. Actors are the different people or systems that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as system operators.

Primary Actor

Primary actors interact directly to achieve the required system function and derive the intended benefit from the system. They work directly with the software. They produce some information and consume some information too.

Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

Here is the use case diagram to observe the non-technical view of the System.

Actors:

Primary

- Donor
- Fundraiser
- Charity Organization

Secondary

- Recipient
- Gmail
- Crypto wallet

5.1 Level 0

Name: CharityGuard
Primary Actor: Donor, Fundraiser, Charity Organization
Secondary Actor: Recipient, Gmail, Crypto wallet

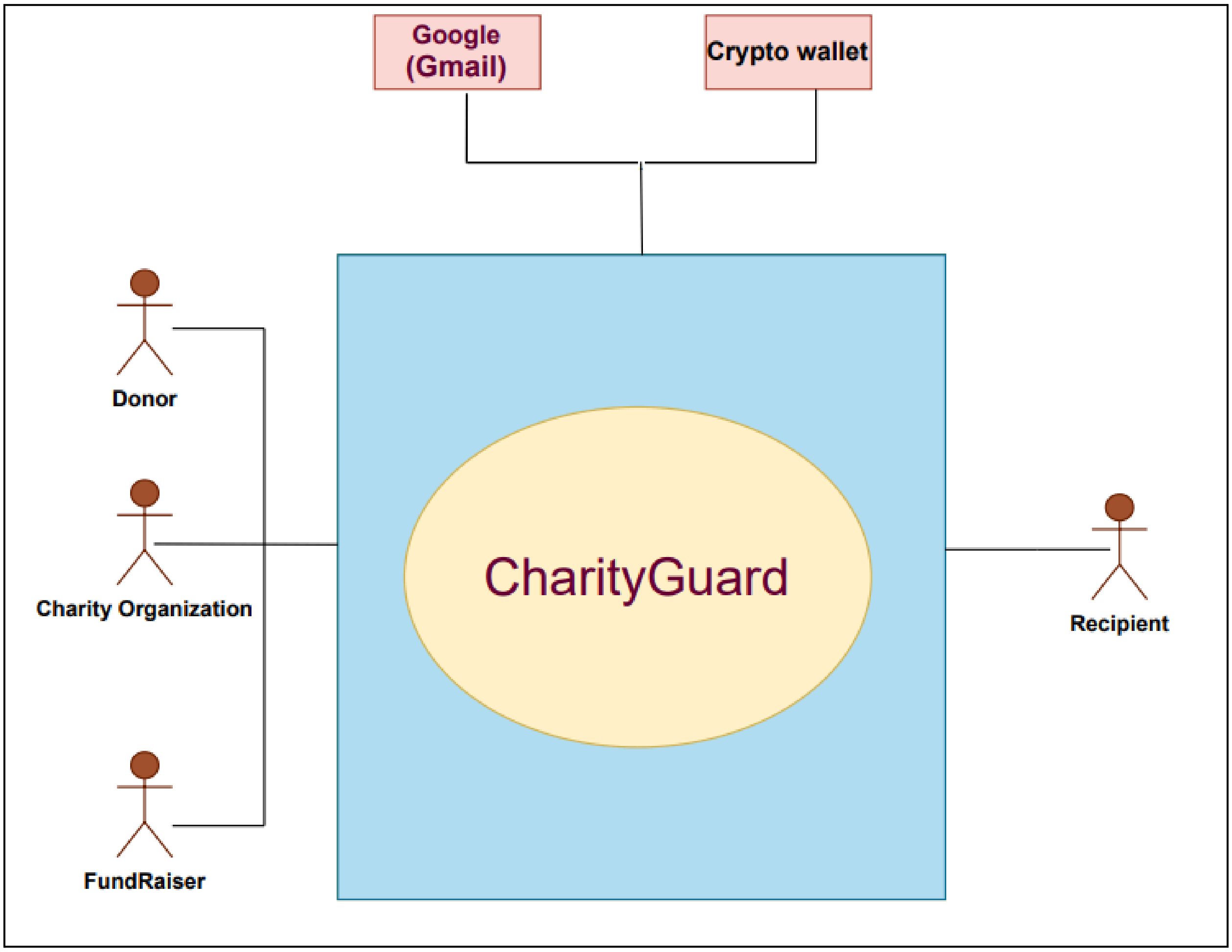


Figure 1 : CharityGuard

5.2 Level 1

Name: CharityGuard
Primary Actor: Donor, Fundraiser, Charity Organization

Secondary Actor: Recipient, Gmail, Crypto wallet

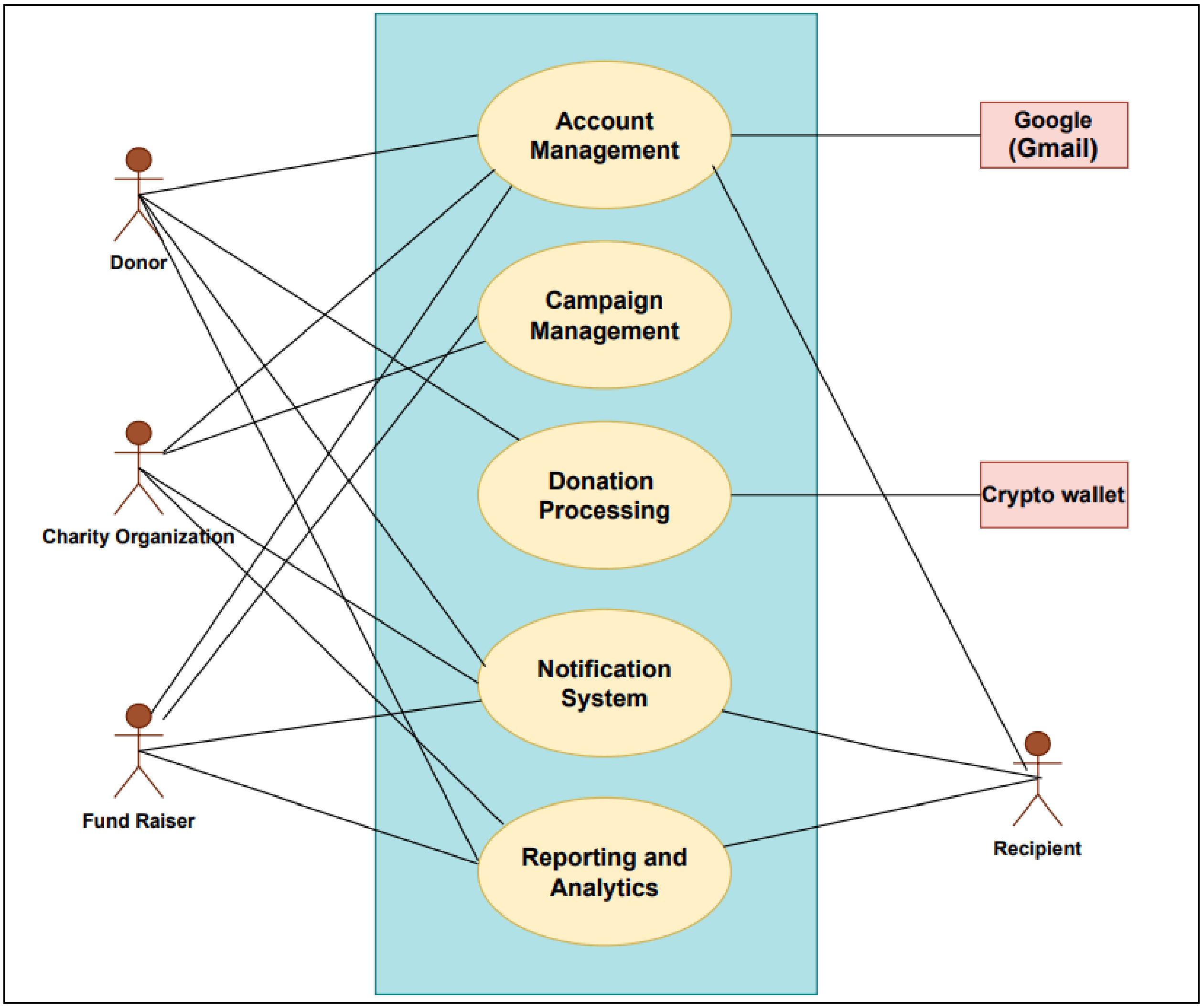


Figure 2 : CharityGuard

Notification System Module: Notifications are sent for new campaigns, donation confirmations, and campaign milestones. Alerts for fund disbursements and security events, such as unrecognized logins, keep users aware and engaged. Recipients will get notified about their funds when its disbursed to them.

5.2.1 Level 1.1

Name: Account Management
Primary Actor: Donor, Fundraiser, Charity Organization, Recipient
Secondary Actor: Gmail

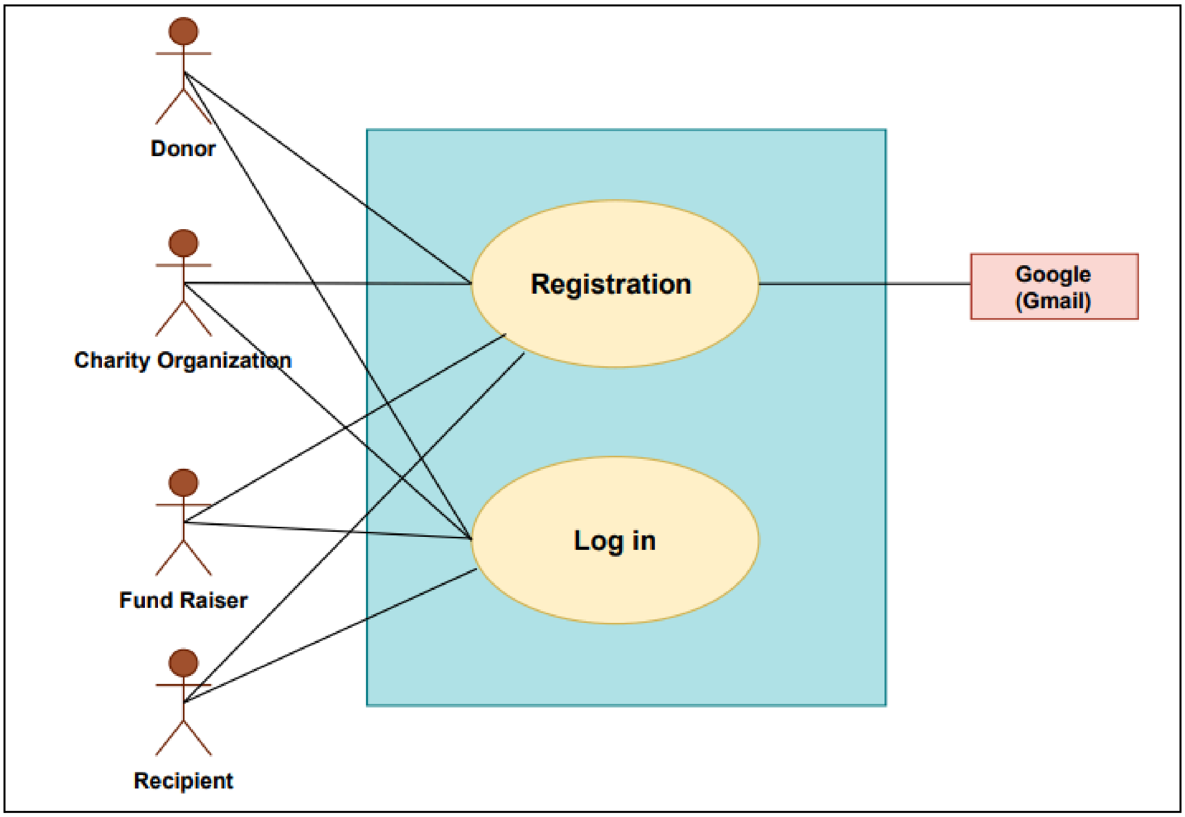


Figure 3 : Account Management

Activity 1: Users enter registration credentials to create an account.

Activity 2: Users enter login credentials to login to the system.

5.3 Level 1.2

Name: Campaign Management

Primary Actor: Fundraiser, Charity Organization

Secondary Actor: Donor

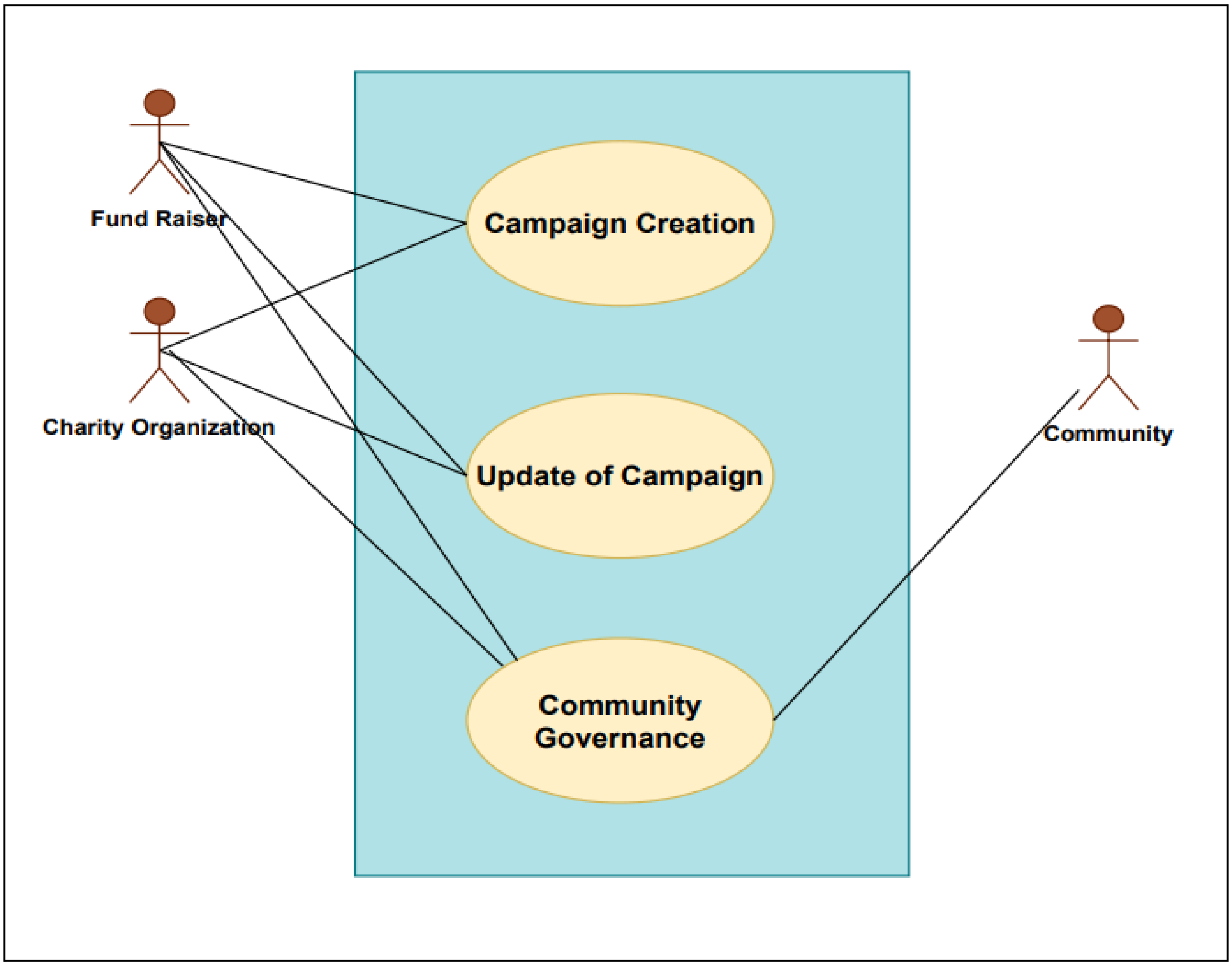


Figure 4 : Campaign Management

Activity 1: Charity and fundraiser launches new campaign.

Activity 2: Charity and fundraiser edits previously launched campaign details.

Activity 3: Donor and recipient votes in campaign polls.

5.4 Level 1.2.1

Name: Campaign creation

Secondary Actor: Fundraiser, Charity Organization

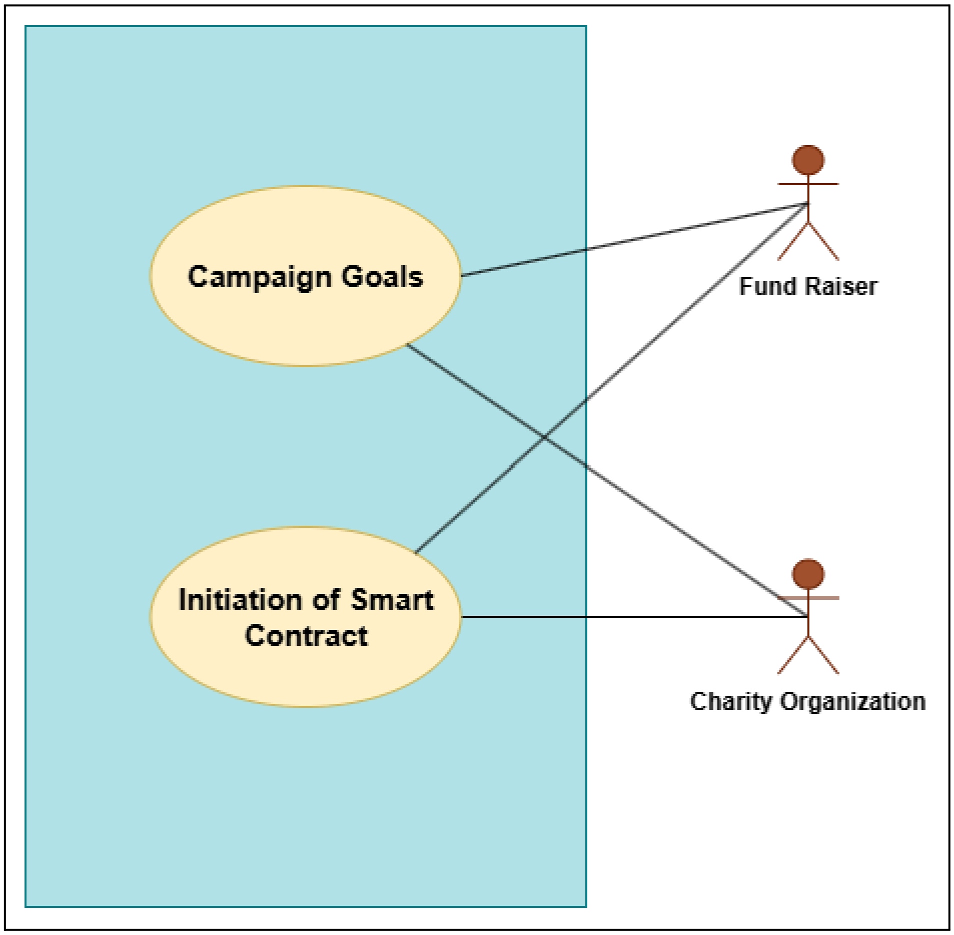


Figure 5 : Campaign creation

Activity 1: Charity and fundraiser set campaign goals and information.

Activity 2: Charity and fundraiser initiate smart contract conditions.

5.5 Level 1.3

Name: Donation Processing
Primary Actor: Donor

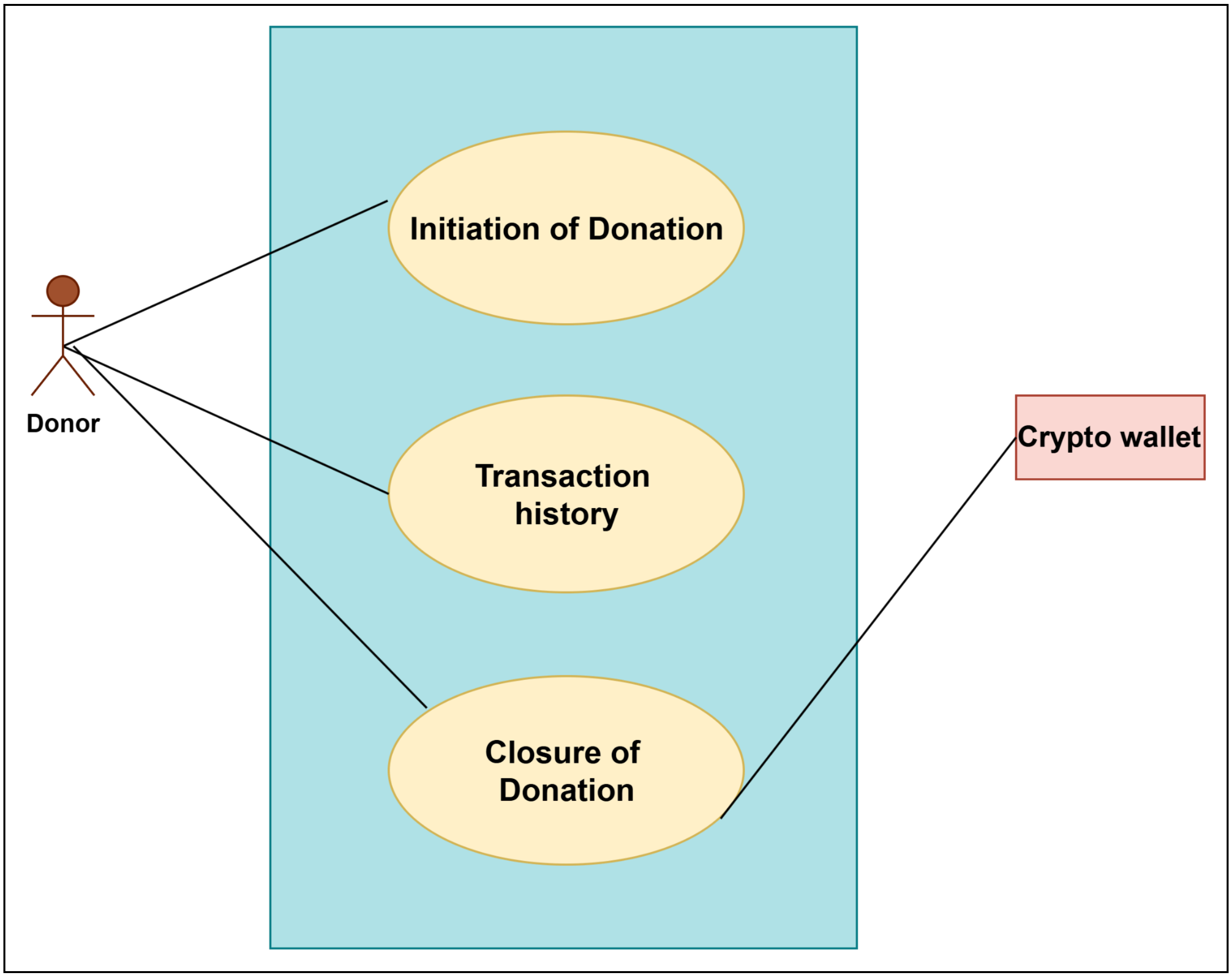


Figure 6 : Donation Processing

Activity 1: Donors select a campaign, choose donation amount and cryptocurrency, connect their wallet, and confirm, recording it on the blockchain.

Activity 2: Donors can view detailed transactions of a campaign.

Activity 3: Campaigns automatically close upon reaching goals or deadlines, triggering fund disbursement or refunds if needed.

5.6 Level 1.4

Name: Reporting and Analytics

Primary Actor: Fundraiser, Charity Organization

Secondary Actor: Donor

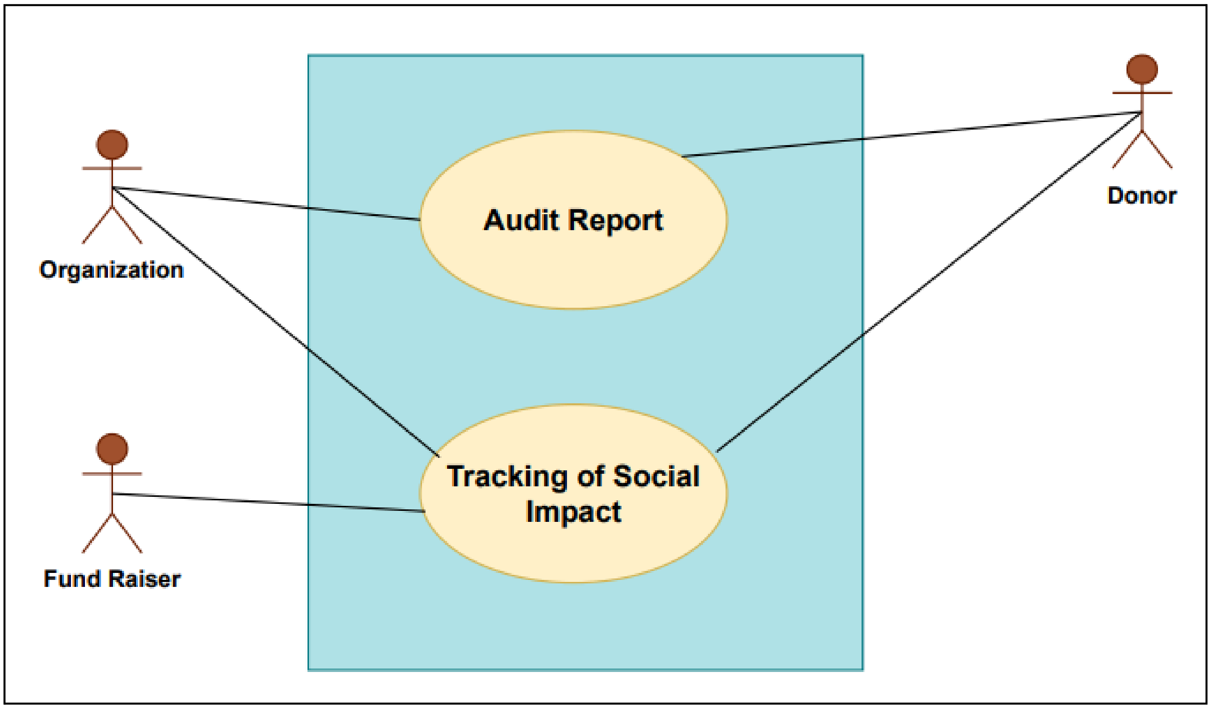


Figure 7 : Report and Analytics

Activity 1: Donors can download detailed donation history.

Activity 2: Charities can report donation impact

Activity 3: Donors can generate auditory reports.

6. Swim Lane Diagram

6.1 Level 1.1

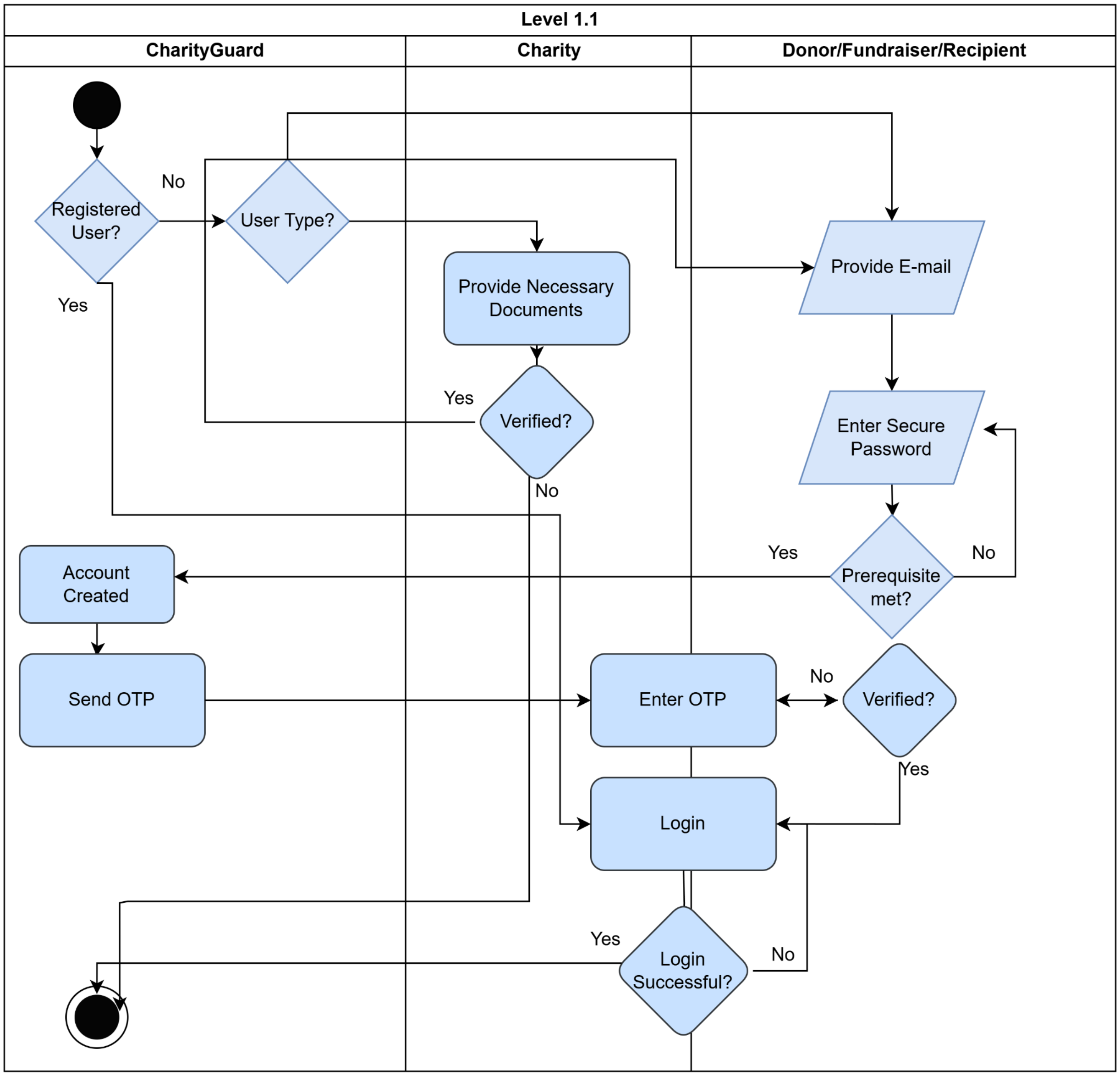


Figure 8 : Account Management

6.2 Level 1.2

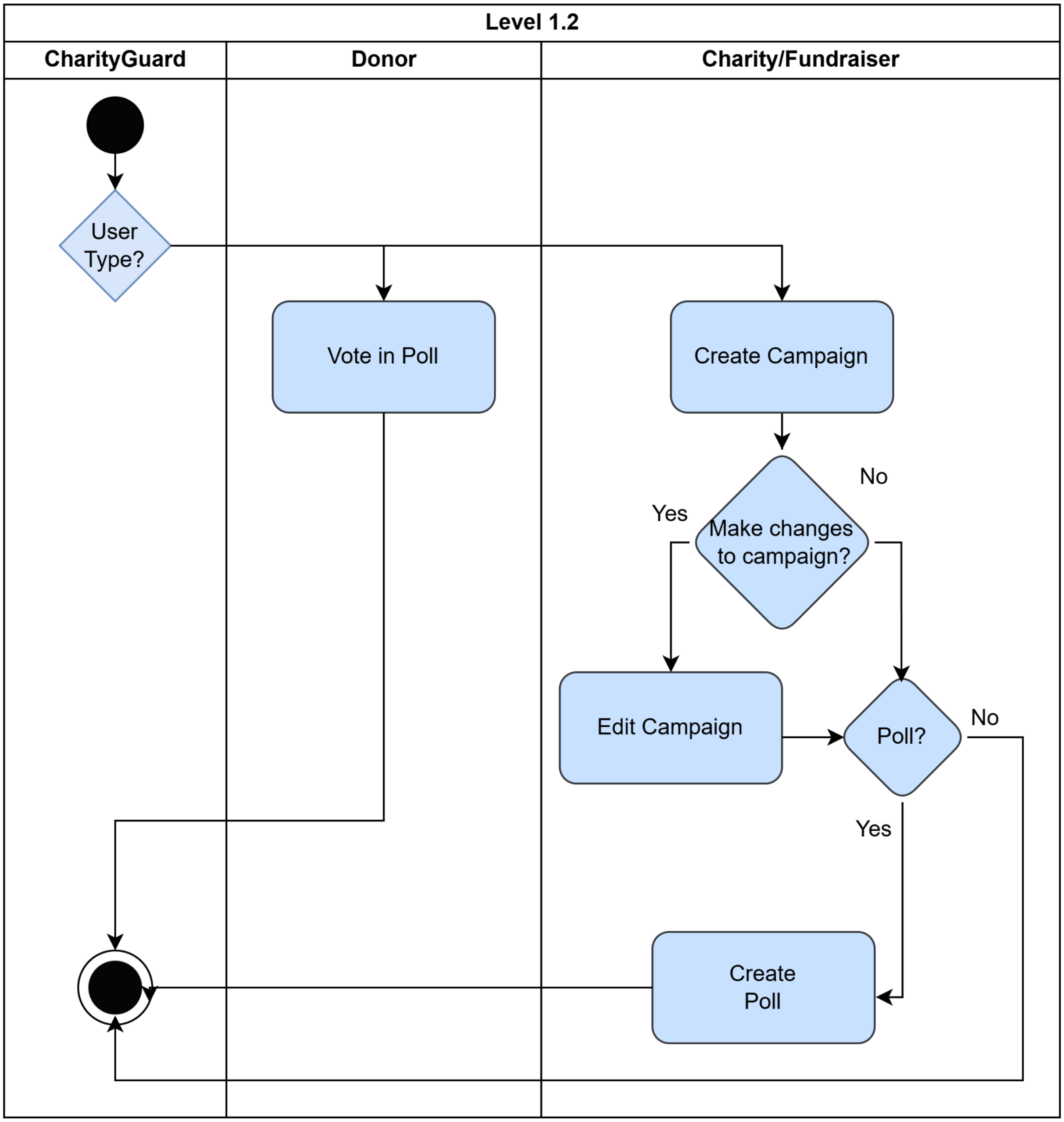


Figure 9 : Campaign Management

6.3 Level 1.2.1

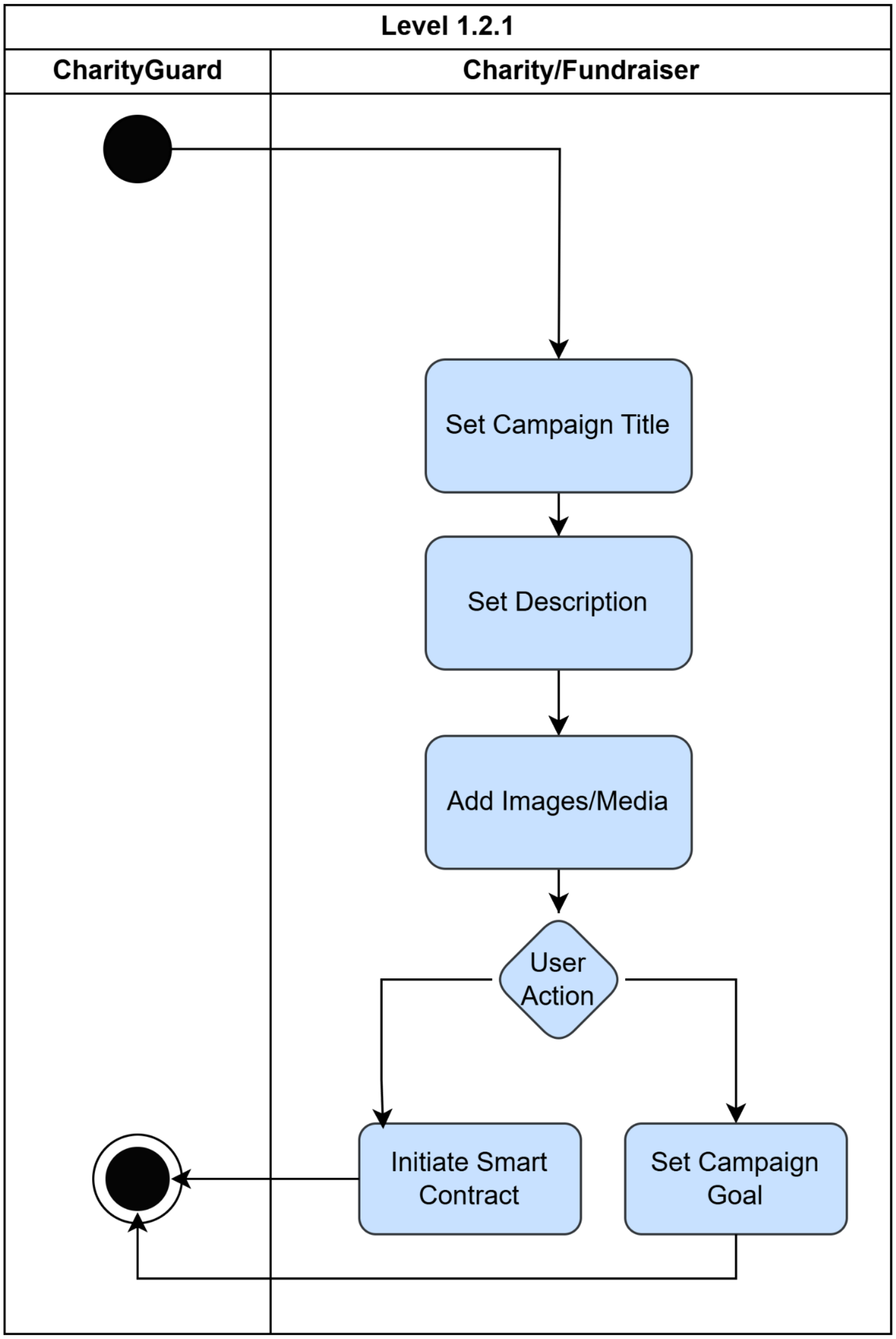


Figure 10 : Campaign Creation

6.4 Level 1.3

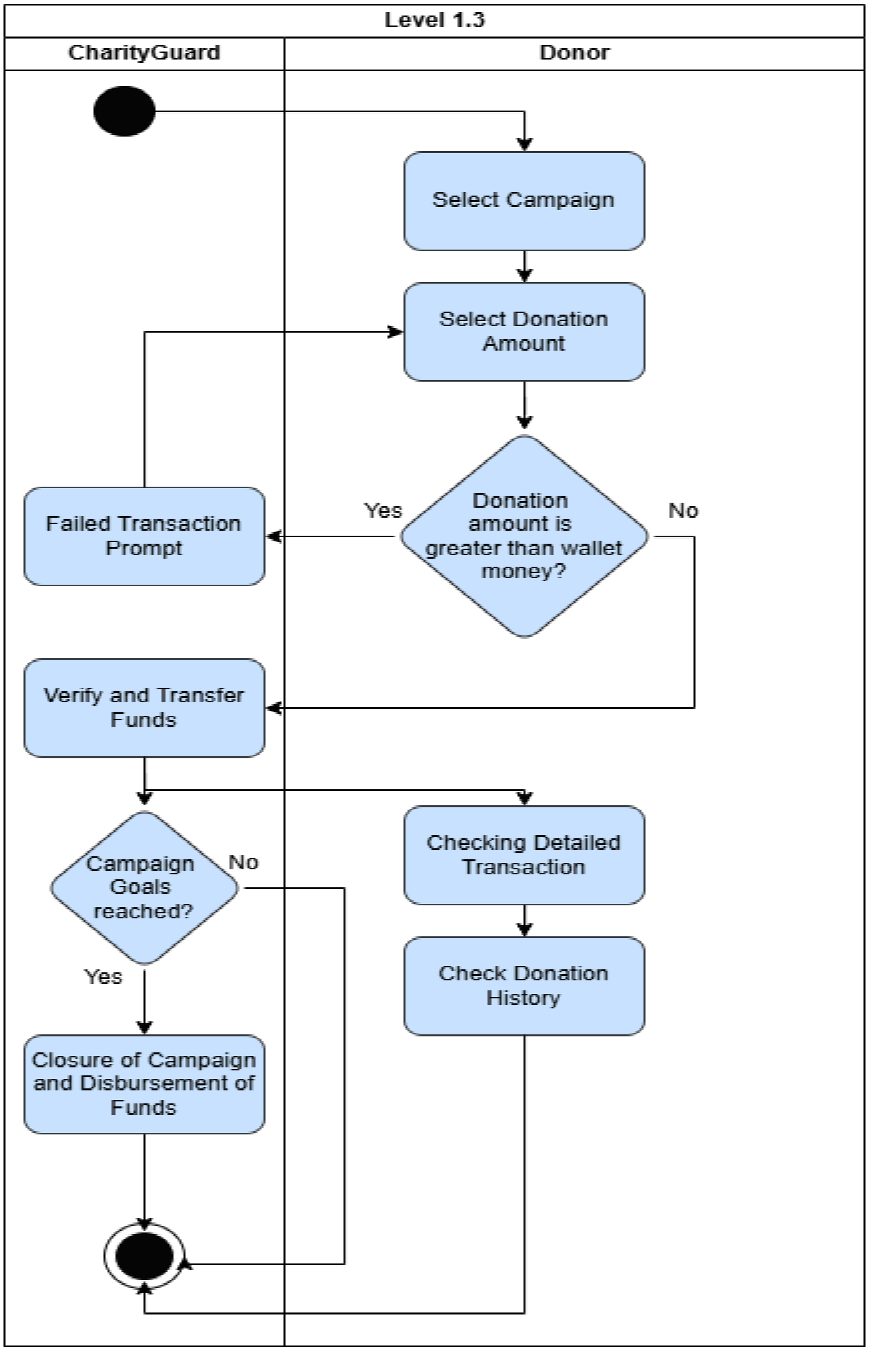

Level 1.3
CharityGuard
Donor
Select Campaign
Select Donation Amount
Donation amount is greater than wallet money?
Yes
No
Failed Transaction Prompt
Verify and Transfer Funds
Campaign Goals reached?
No
Yes
Closure of Campaign and Disbursement of Funds
Checking Detailed Transaction
Check Donation History

Figure 11 : Donation Processing

6.5 Level 1.5

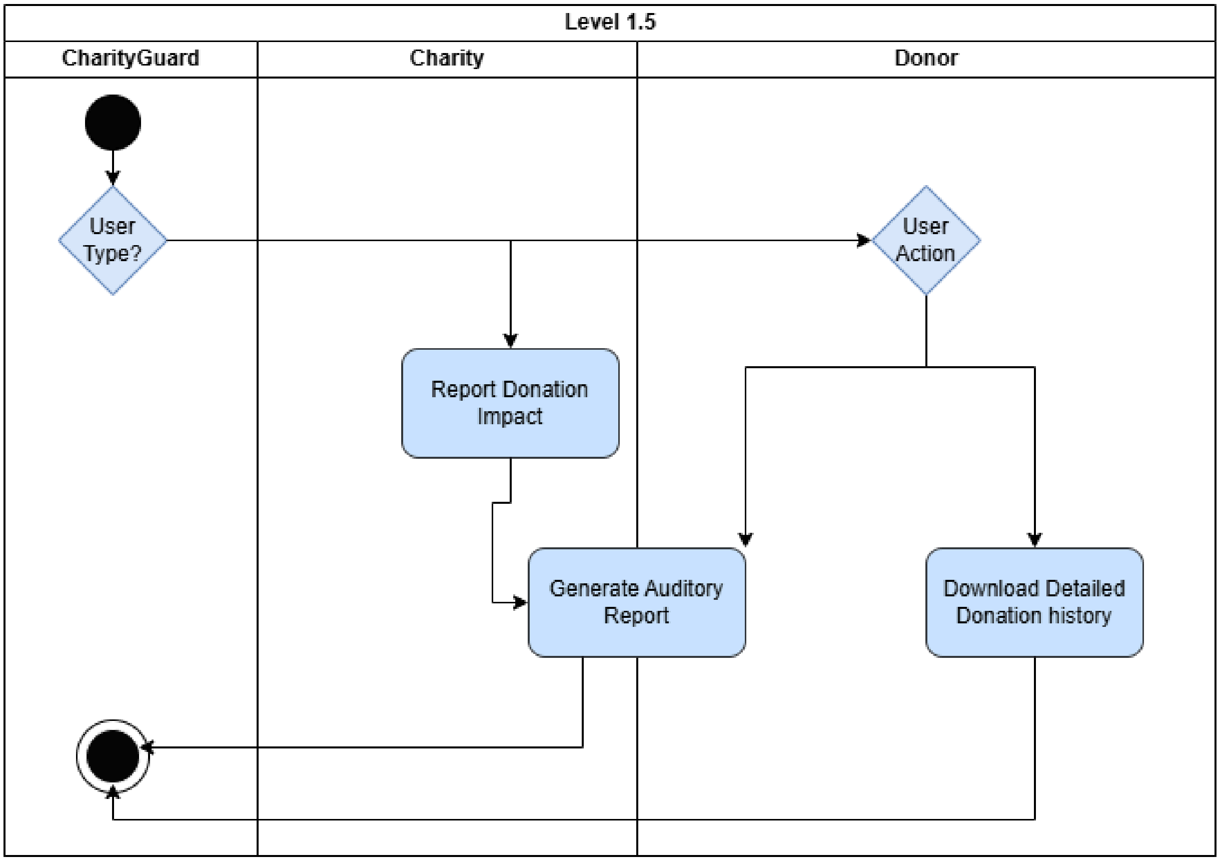


Figure 12 : Report & Analytics

7. Data Based Modeling

7.1 Probable Data Object Identification

SL No.	Noun	P/S	Attributes
1.	**User**	S	4,5,6
2.	**Donor**	S	4,5,6
3.	**Charity**	S	4,5,6
4.	**Email**	S	✅
5.	**Smart Wallet**	S	✅
6.	**Password**	S	✅
7.	**Campaign**	S	10,11,12,17
8.	**Amount**	S	✅
9.	**Fundraiser**	S	4,5,6
10.	**Goal**	S	✅
11.	**Milestone**	S	✅

12.	**Smart Contract**	S	✅
13.	**Transaction**	S	8, 14,15,16
14.	**Transaction Hash**	S	✅
15.	**Currency type**	S	✅
16.	**Donation Time**		✅
17.	**Poll**	S	40,41
18.	**Notification**	S	19,20
19.	**Notification Type**	S	✅
20.	**Notification details**	S	✅
21.	**Recipient**	S	4,5,6
22.	**Report**	S	23
23.	**Social Impact**	S	✅
24.	**Account**	P	

25.	**Profile**	P	
26.	**DAO (Decentralized Autonomous Organization)**	P	
27.	**Dashboard**	P	
28.	**Donation**	P	
29.	**Cryptocurrency**	P	
30.	**Blockchain**	P	
31.	**Receipt**	P	
32.	**Donation History**	P	
33.	**Owner**	P	
34.	**Alert**	P	
35.	**Disbursement**	P	
36.	**Security Event**	P	

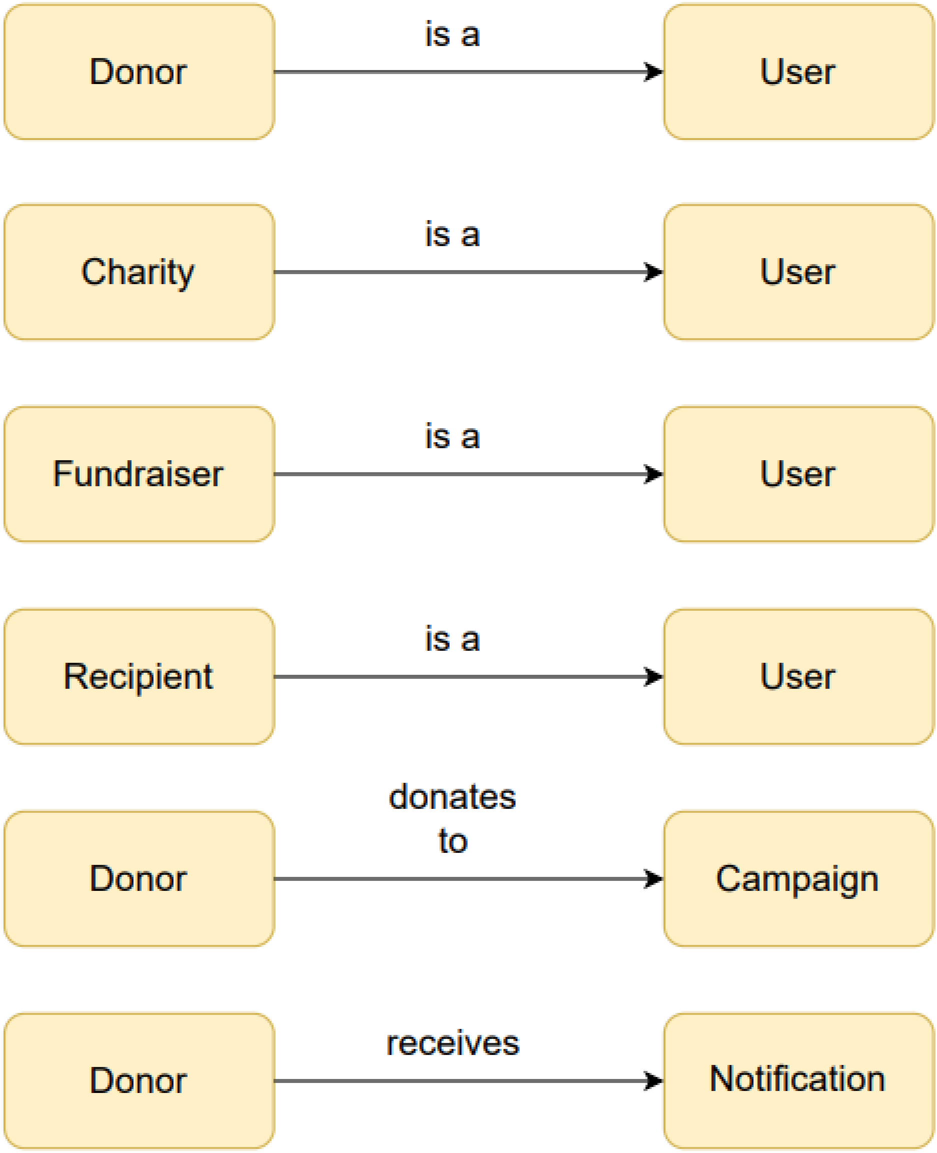
37.	**Insight**	P	
38.	**Metric**	P	
39.	**Performance Report**	P	
40.	**Poll counter**	S	✅
41.	**Poll ID**	S	✅

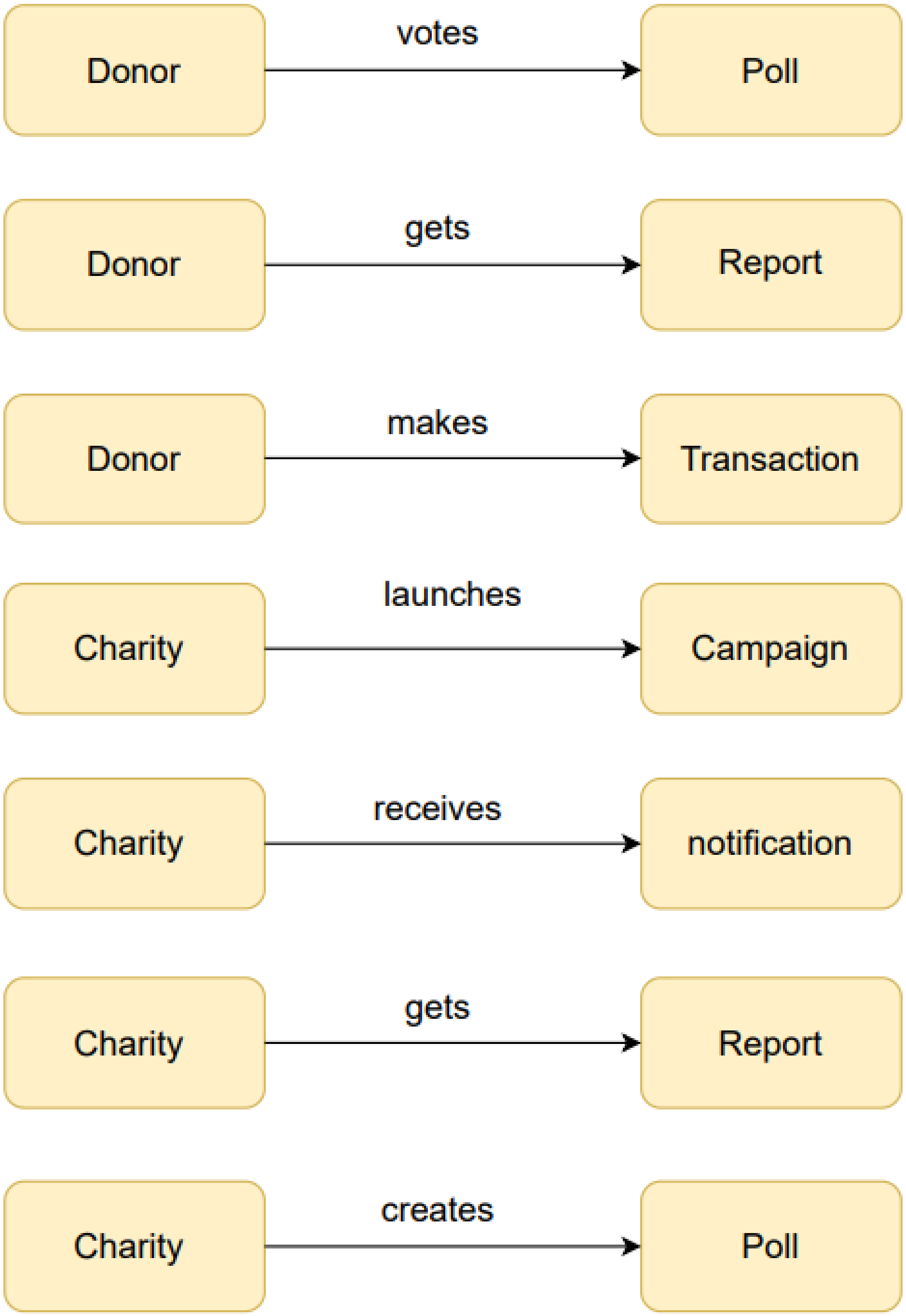
7.2 Data Objects:

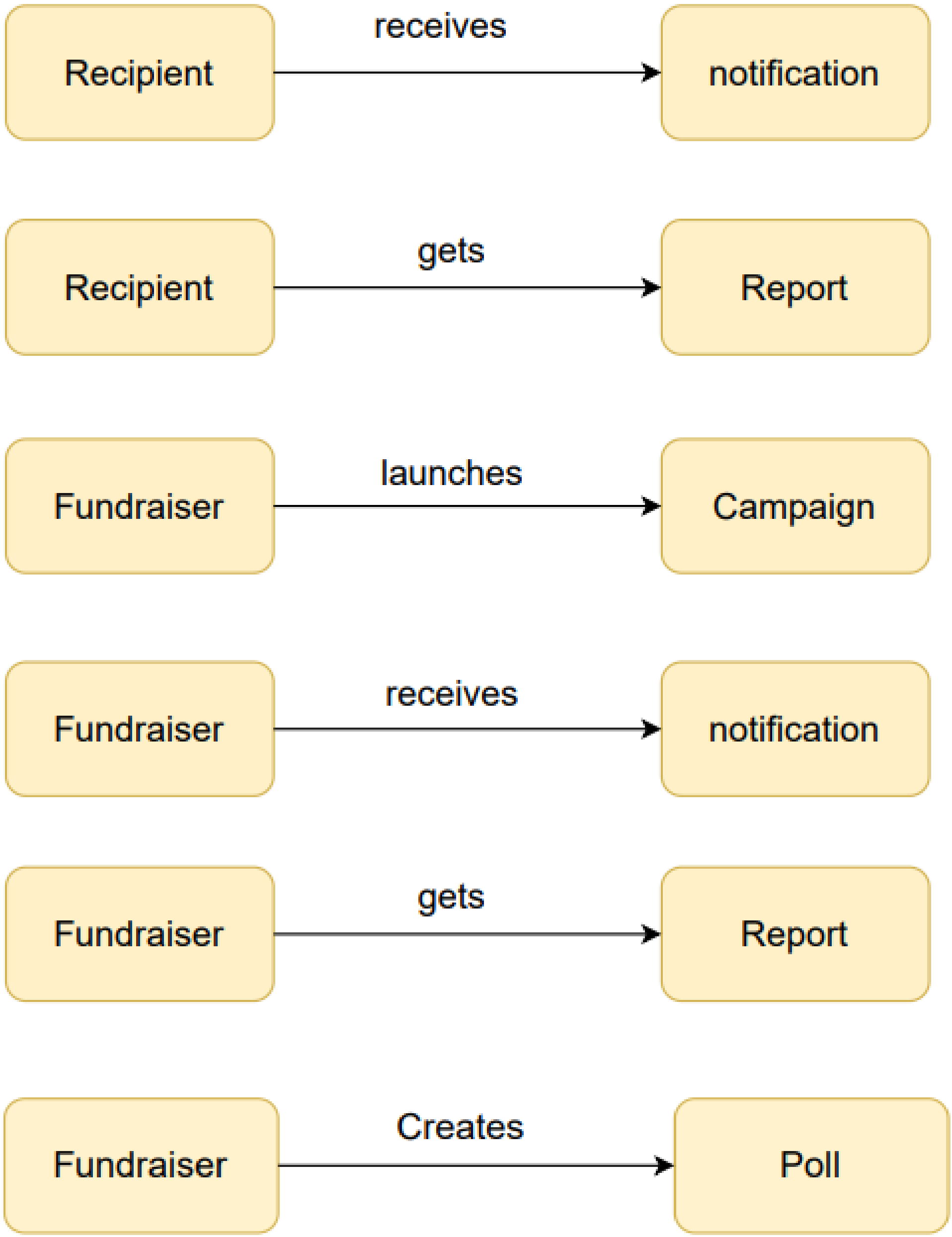
1.	**User**	4,5,6
2.	**Donor**	4,5,6
3.	**Charity**	4,5,6
4.	**Fundraiser**	4,5,6

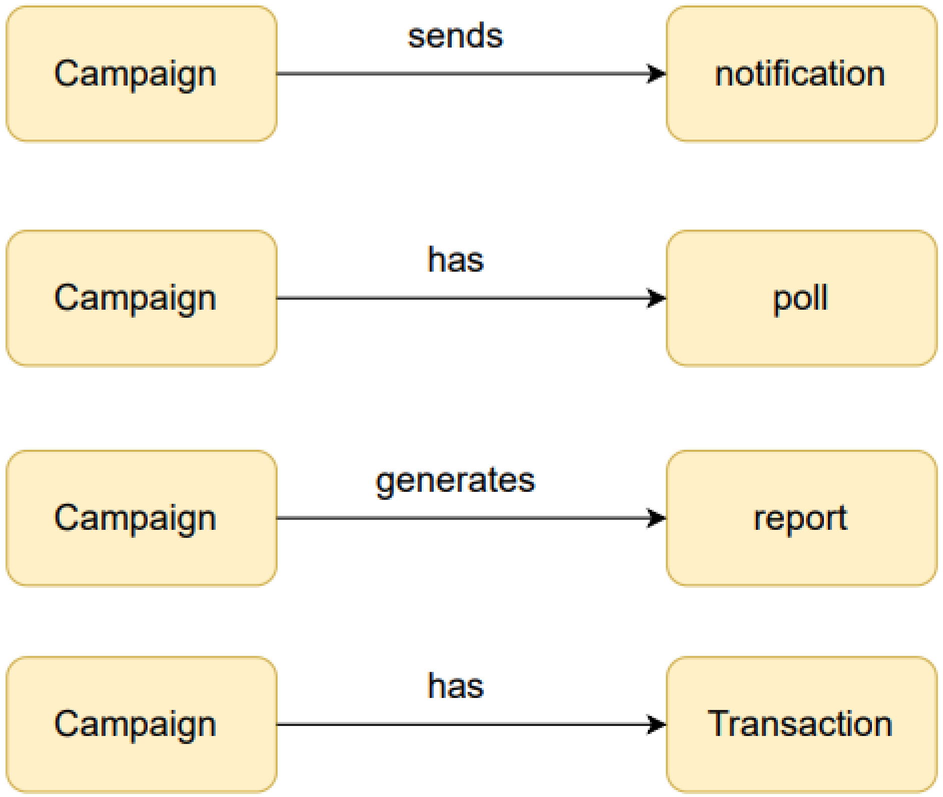
5.	**Recipient**	4,5,6
6.	**Campaign**	10,11,12,17
7.	**Transaction**	8, 14,15,16
8.	**Poll**	40,41
9.	**Notification**	19,20
10.	**Report**	23

7.3 Relationship Between Data Objects

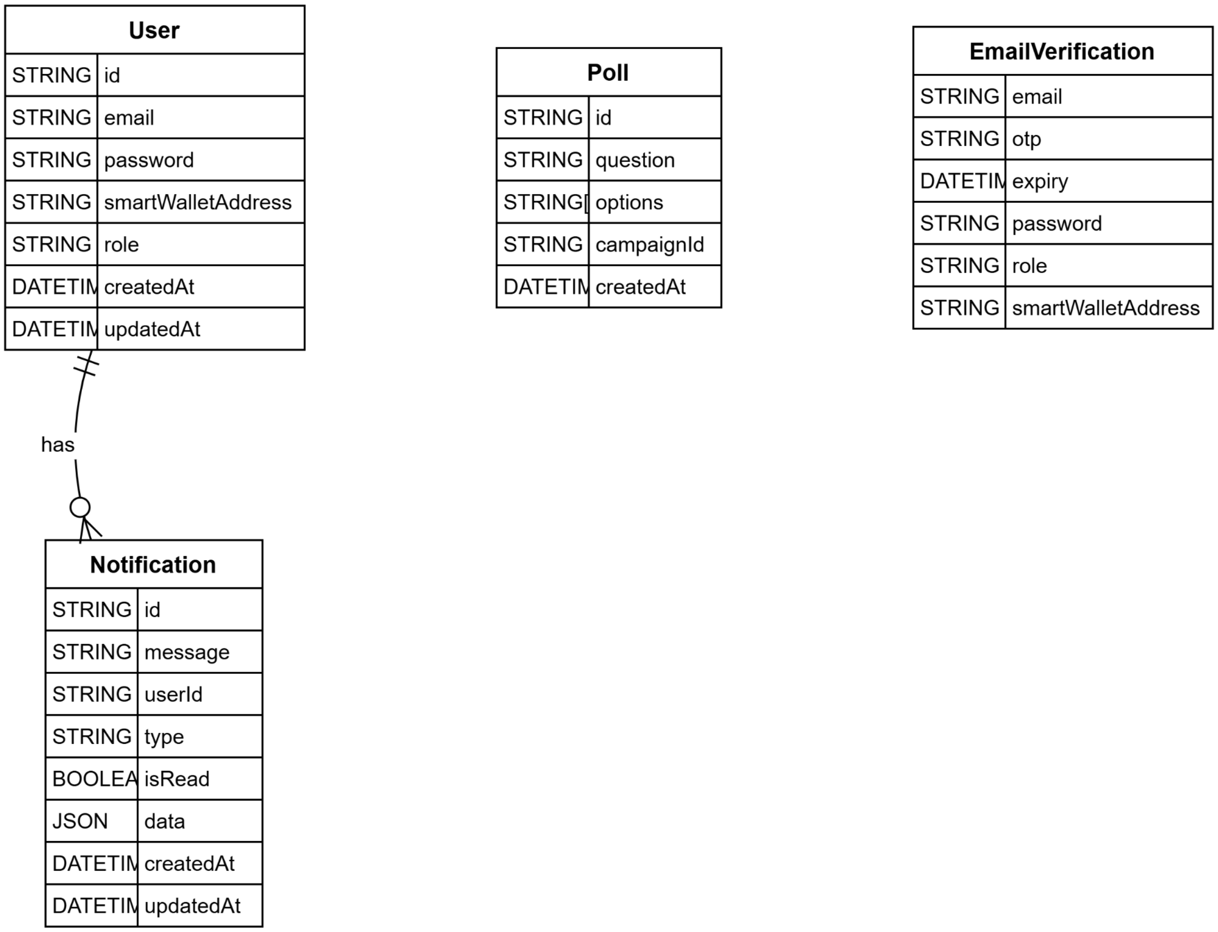



Donor
votes
Poll
Donor
gets
Report
Donor
makes
Transaction
Charity
launches
Campaign
Charity
receives
notification
Charity
gets
Report
Charity
creates
Poll


Recipient
receives
notification
Recipient
gets
Report
Fundraiser
launches
Campaign
Fundraiser
receives
notification
Fundraiser
gets
Report
Fundraiser
Creates
Poll


Campaign
sends
notification
Campaign
has
poll
Campaign
generates
report
Campaign
has
Transaction

7.4 Database Diagram



User	
STRING	id
STRING	email
STRING	password
STRING	smartWalletAddress
STRING	role
DATETIM	createdAt
DATETIM	updatedAt

Poll	
STRING	id
STRING	question
STRING[	options
STRING	campaignId
DATETIM	createdAt

EmailVerification	
STRING	email
STRING	otp
DATETIM	expiry
STRING	password
STRING	role
STRING	smartWalletAddress

Notification	
STRING	id
STRING	message
STRING	userId
STRING	type
BOOLEA	isRead
JSON	data
DATETIM	createdAt
DATETIM	updatedAt

Figure 13 : Database Diagram

7.5 SCHEMA DIAGRAM:

Entity	Attributes	DataType
Users	User_ID	INT (Primary Key)
	Email	VARCHAR(255)
	Password	VARCHAR(255)
	Smart_Wallet	VARCHAR(255)
	role	enum
	Is_Verified	bool
	createdAt	DateTime
	UpdatedAt	DateTime
Polls	Poll_ID	String(Primary Key)
	question	String
	options	String
	Created_At	DateTime
	Campaign_ID	String
Notifications	Notification_ID	String(Primary Key)
	message	String
	User_ID	String
	type	String
	isRead	boolean

Entity	Attributes	DataType
	data	Json
	createdAt	DateTime
	updatedAt	DateTime
EmailVerification	email	String
	otp	String
	password	String
	role	String
	smartWalletAddres	String
	expiry	DateTime

Blockchain

Campaigns	Campaign_ID	uint256
	title	String
	goal	uint256
	raised	uint256
	isActive	bool
	image	string
	deadline	uint256
	Payable owner	address
	description	String

Donor	donorAddress	address
	amount	uint256
	txHash	bytes32
Recipients	recipientAddress	address
	amount	uint256
	txHash	bytes32

8. Class Based Modeling

Class Based Modeling Concept

Class-based modeling identifies classes, attributes and relationships that the system will use. It represents the object. The system manipulates the operations.

The elements of the class based model consist of classes and object, attributes, operations, class – responsibility - collaborator (CRC) models.

Classes are determined using underlining each noun or noun clause and entering it into the simple table. Attributes are the set of data objects that are defining a complete class within the context of the problem. The operations define the behavior of an object.

8.1 Noun List:

Sl no.	Noun
1.	**User**
2.	**Donor**
3.	**Charity**
4.	**Email**
5.	**Smart Wallet**
6.	**Password**
7.	**Campaign**

Sl no.	Noun
8.	**Amount**
9.	**Fundraiser**
10.	**Goal**
11.	**Milestone**
12.	**Smart Contract**
13.	**Transaction**
14.	**Transaction Hash**
15.	**Currency type**
16.	**Donation Time**
17.	**Poll**
18.	**Poll counter**
18.	**Notification**
19.	**Notification Type**
20.	**Notification details**
21.	**Recipient**
22.	**Report**
23.	**Social Impact**
24.	**Account**
25.	**Profile**

Sl no.	Noun
26.	DAO (Decentralized Autonomous Organization)
27.	Dashboard
28.	Donation
29.	Cryptocurrency
30.	Blockchain
31.	Receipt
32.	Donation History
33.	Owner
34.	Alert
35.	Disbursement
36.	Security Event
37.	Insight
38.	Metric
39.	Performance Report

8.2 General Classification

Candidate classes were then characterized in seven general classes. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Noun	Criteria
User	External Entity, Role,organizational units
Donor	External Entity, Role,things
Charity	External Entity, Role,things
Email	Things,external entity
Smart Wallet	Things,external entity
Password	Things,
Campaign	Things,events, organizational units
Amount	Things
Fundraiser	External Entity, Role
Goal	Thing
Milestone	Thing
Smart Contract	Thing
Transaction	Things,event, structure

Noun	Criteria
Transaction Hash	Thing
Currency Type	Thing
Donation Time	Event
Poll	Event, things, organizational units
Poll Counter	Event
Notification	Event, things, organizational units
Notification Type	Structure
Notification Details	Structure
Recipient	External Entity, Role
Report	Thing, Organizational unit, event
Social Impact	Thing
Account	Structure
Smart Contract	External Entity, Organizational Unit,
Profile	Structure
DAO	External Entity, Organizational Unit
Dashboard	Place
Donation	Thing, Event
Cryptocurrency	Thing
Blockchain	Place
Receipt	Thing
Donation History	Structure

Noun	Criteria
Owner	Role
Alert	Structure
Disbursement	Event
Security Event	Event
Insight	Thing
Metric	Thing
Performance Report	Thing

8.3 Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three

characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Noun	Criteria
User	1-3

Noun	Criteria
Donor	1-5
Charity	1-5
Fundraiser	1-5
Authentication	2,6
Smart Wallet	1,2,6
Recipient	1
Campaign	1-5
Transaction	1,2
Blockchain	1,5,6
Report	1,5,6
Notification	1-5

8.4 Class Card:

Users	
Attributes	**Methods**

`-userID` (String):
`-username` (String):
`-email` (String):
`-password` (String):
`-role` (String)
`-dateCreated`
`-walletConnected` (Boolean)
	`+viewProfile()`:

Users	
Responsibility	**Collaborators**
View Profile	
View Notification	Notifications
View Dashboard	

AuthService	
Attributes	**Methods**

	+register()
	+login()

Responsibility	**Collaborators**
Register	User
Login	User
Logout	User
Verify Login	

Campaign	
Attributes	**Methods**

Campaign	

-campaignID (String)
-title (String)
-description (String)
-category (String): Type of campaign
-goalAmount (Decimal)
-currentAmount (Decimal)
-deadline (Date): End date of the campaign.
-ownerID (String):
-pollDetails(String)
	+connect()
	+getUserCampaign()
	+createCampaign()
	+fetchCampaigns()
	+fetchDonors()
	+fetchRecipients()
	+donate()

Responsibility	**Collaborators**
Create Campaign	Notification
Update Campaign	
Manage Polls	
View Campaign Donations	Blockchain

Notification	
Attributes	**Methods**
-notificationID (String)	+save()

-userID (String)
-type (String)
-details (String)
-timestamp (DateTime)
	+markAsRead()
	+markAllAsRead()
	+getUserNotifications()
	+getUnreadCount()

Responsibility	**Collaborators**
Send Notification	User, Donor, Charity
Notify Campaign Updates	Campaign

Report	
Attributes	**Methods**

-campaignID (String)	+fetchDonors()
	+fetchRecipients()
	+generatePDF()

Responsibility	**Collaborators**
Generate Report	Campaign
Display Campaign Performance	Campaign

8.5 CRC Diagram:

Diagram ID: 1
Name: User

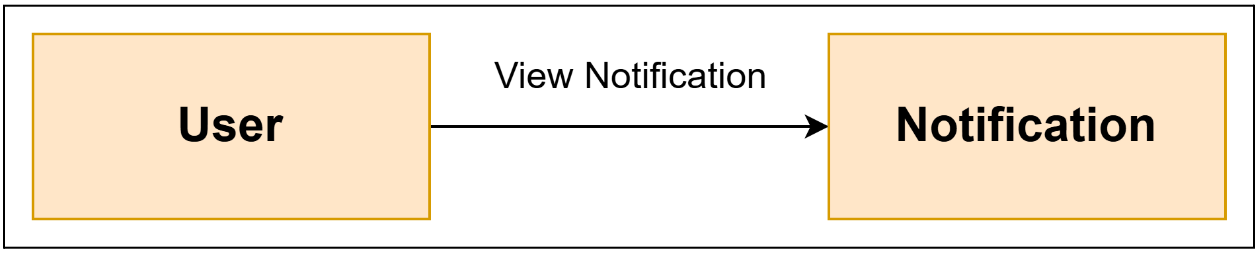


Figure 14 : CRC (User)

Diagram ID: 2
Name: Charity

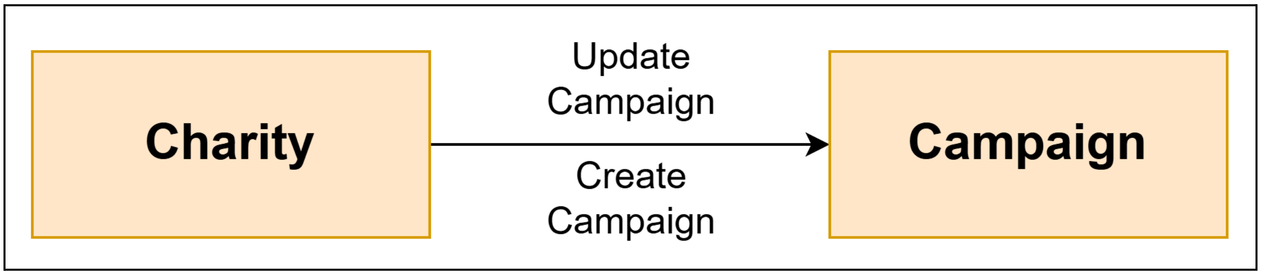


Figure 15 : CRC (Charity)

Diagram ID: 3

Name: Campaign

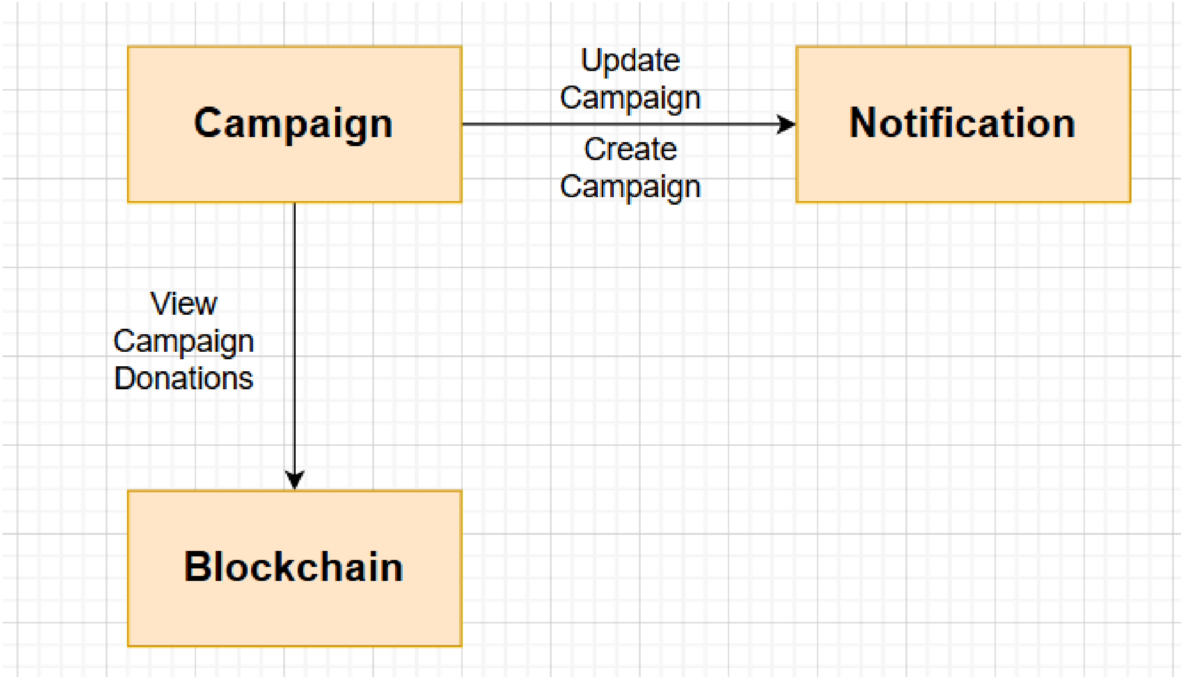


Figure 16 : CRC (Campaign)

Diagram ID: 4

Name: Fundraiser

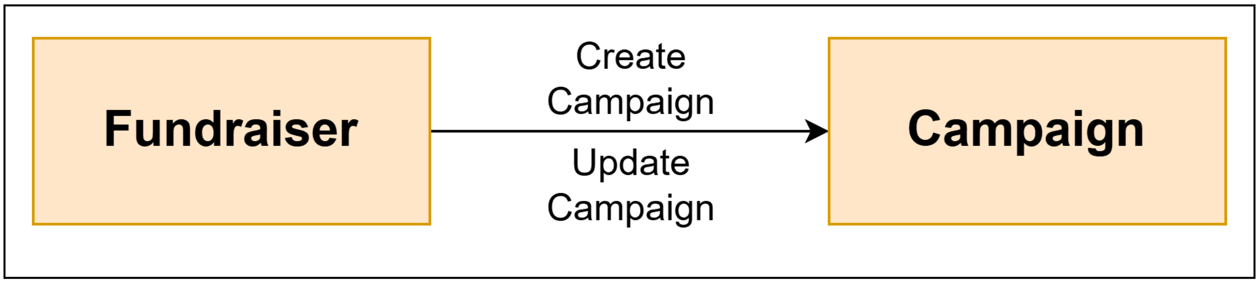


Figure 17 : CRC (Fundraiser)

Diagram ID: 5

Name: Notification

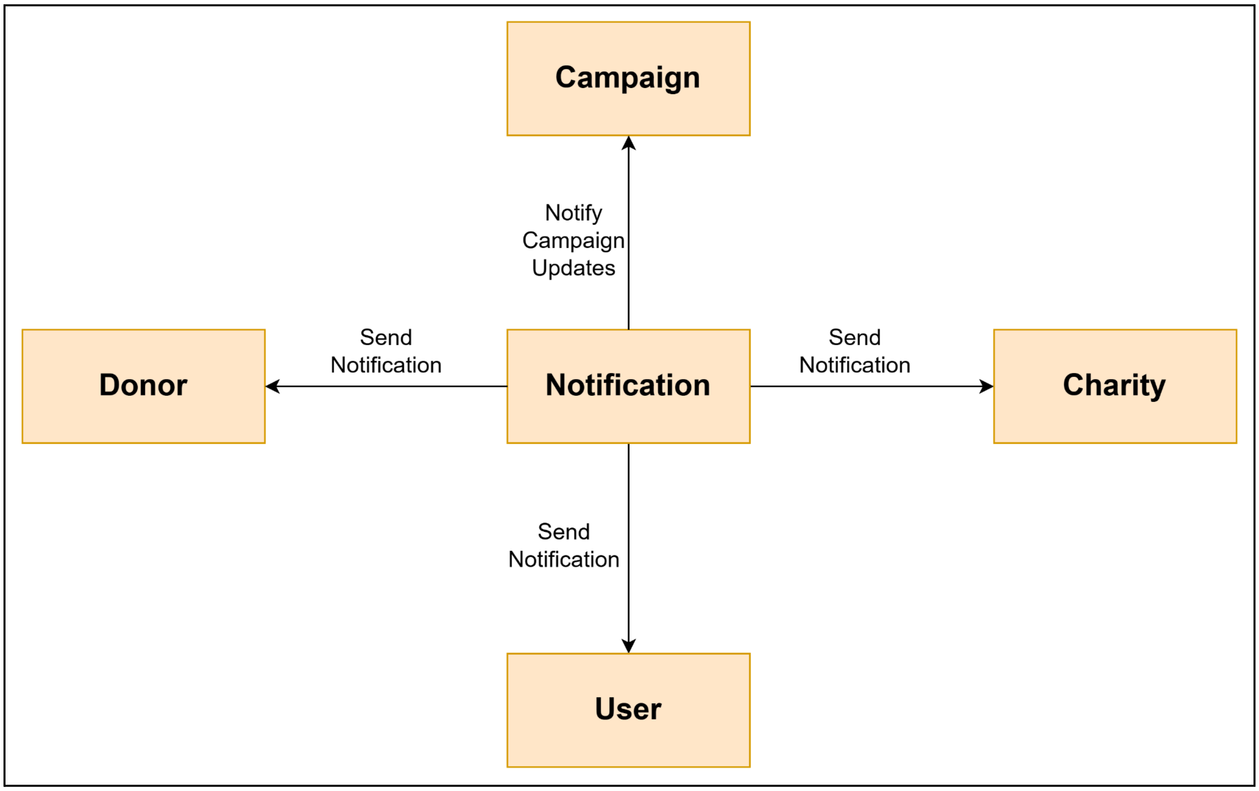


Figure 18 : CRC (Notification)

Diagram ID: 6
Name: Report

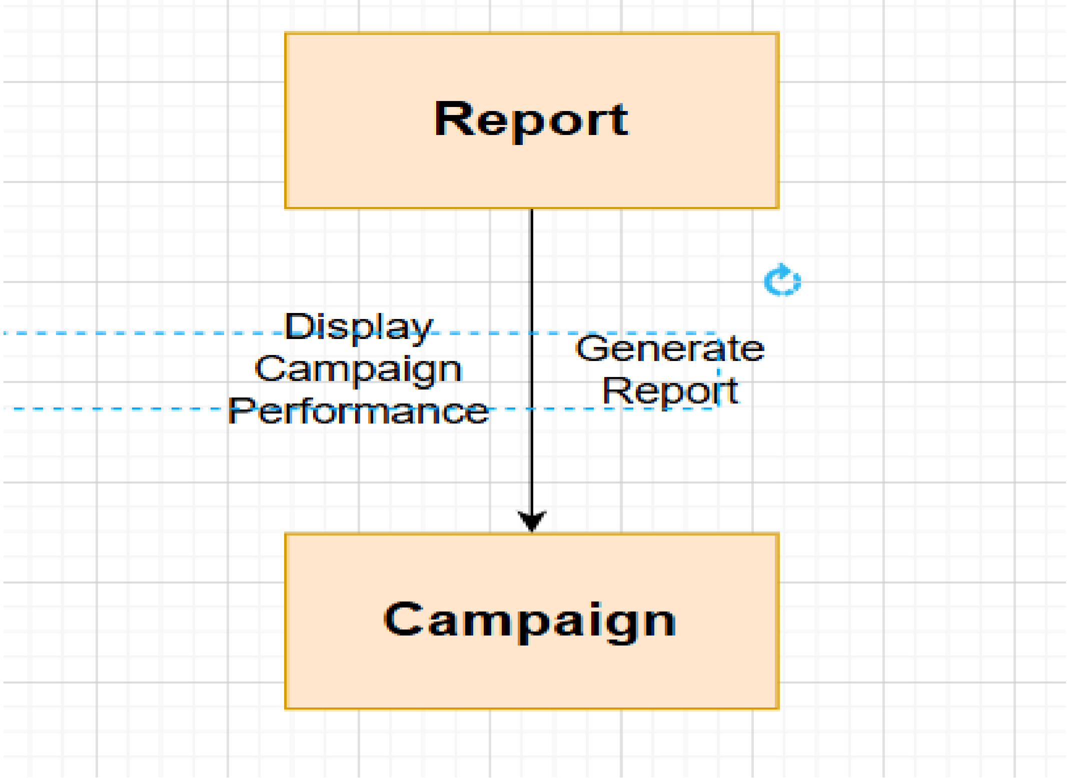


Figure 19 : CRC (Report)

Diagram ID: 7
Name: AuthService

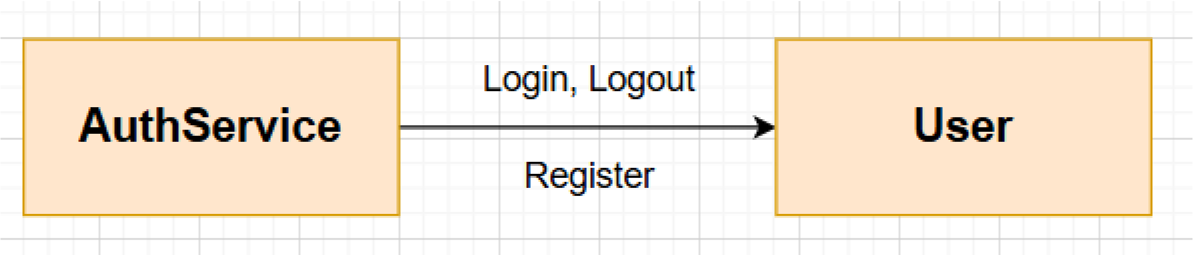


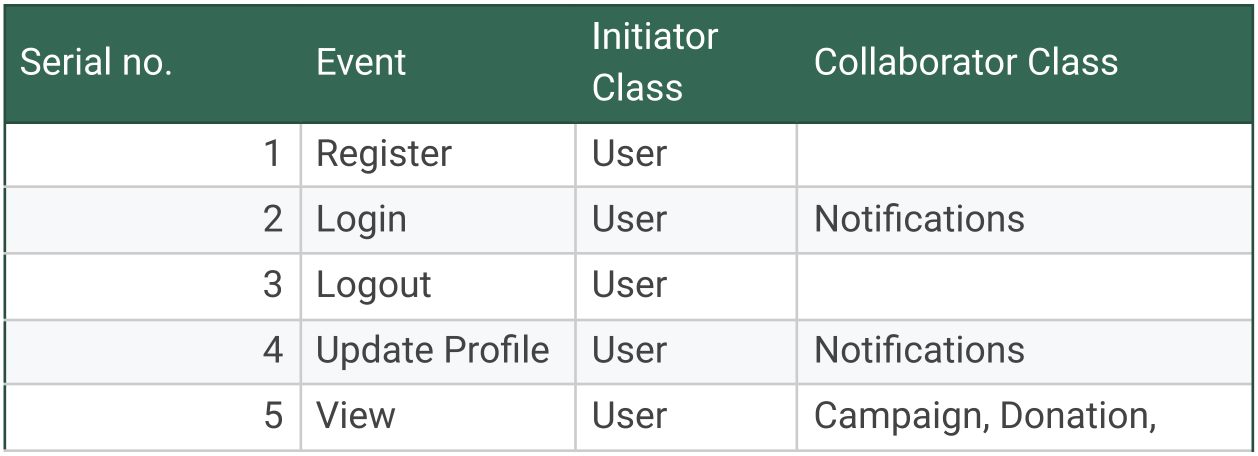
Figure 20 : CRC (Authentication)

9. Behavioral Modeling

Behavioral modeling is the process of predicting future behavior using current and relevant data on consumer and company spending. There are two distinct characterizations of states that must be taken into account in behavioral modeling. First, the state of each class as the system executes its task, and then the system's state as seen from the outside while the system executes its task.

9.1 State Transition Diagrams

A UML state diagram, which depicts the active states for each class and the occasions (triggers) that induce changes between these active states, is part of a behavioral model. Here we've listed the most important events that trigger the change of states.



Serial no.	Event	Initiator Class	Collaborator Class
1	Register	User	
2	Login	User	Notifications
3	Logout	User	
4	Update Profile	User	Notifications
5	View	User	Campaign, Donation,

	Dashboard		Notification
6	Create Campaign	Campaign	Smart Contract
7	Update Campaign	Campaign	Notification
8	Set Poll Details	Campaign	Notification
9	Create Poll	Campaign	Donor
10	View Campaign	Donor	Campaign
11	Donate to Campaign	Donor	Campaign, Smart Wallet, Smart Contract
12	View Donation History	Donor	Donation
13	Generate Report	Report	Campaign, Donation
14	Notify Users	Notifications	User
15	Process Donation	Smart Wallet	Campaign, Smart Contract
16	Disburse Funds	Smart Contract	Campaign
17	Track Social Impact	Report	Campaign
18	Record Transaction	Blockchain	Smart Contract
19	Validate Authentication	Authenticator	User
20	Handle Failed	Authentica	Notification

	Login	tor	
21	Initiate Refund	Smart Contract	Campaign, Donation
22	Notify Fund Disbursement	Notifications	Smart Contract, Recipient
23	Track Campaign Performance	Report	Campaign

9.1.1 Diagram ID: 1

Class Name: User

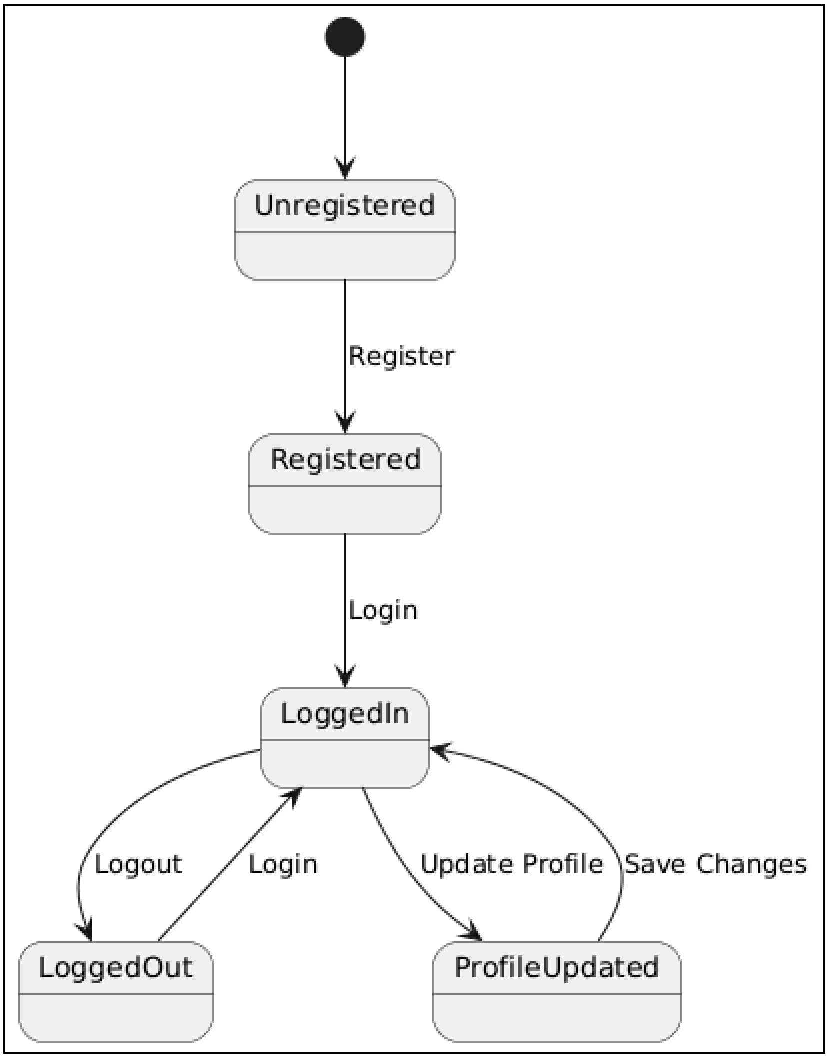


Figure 21 : User State Transition

9.1.2 Diagram ID: 2

Class Name: Donor

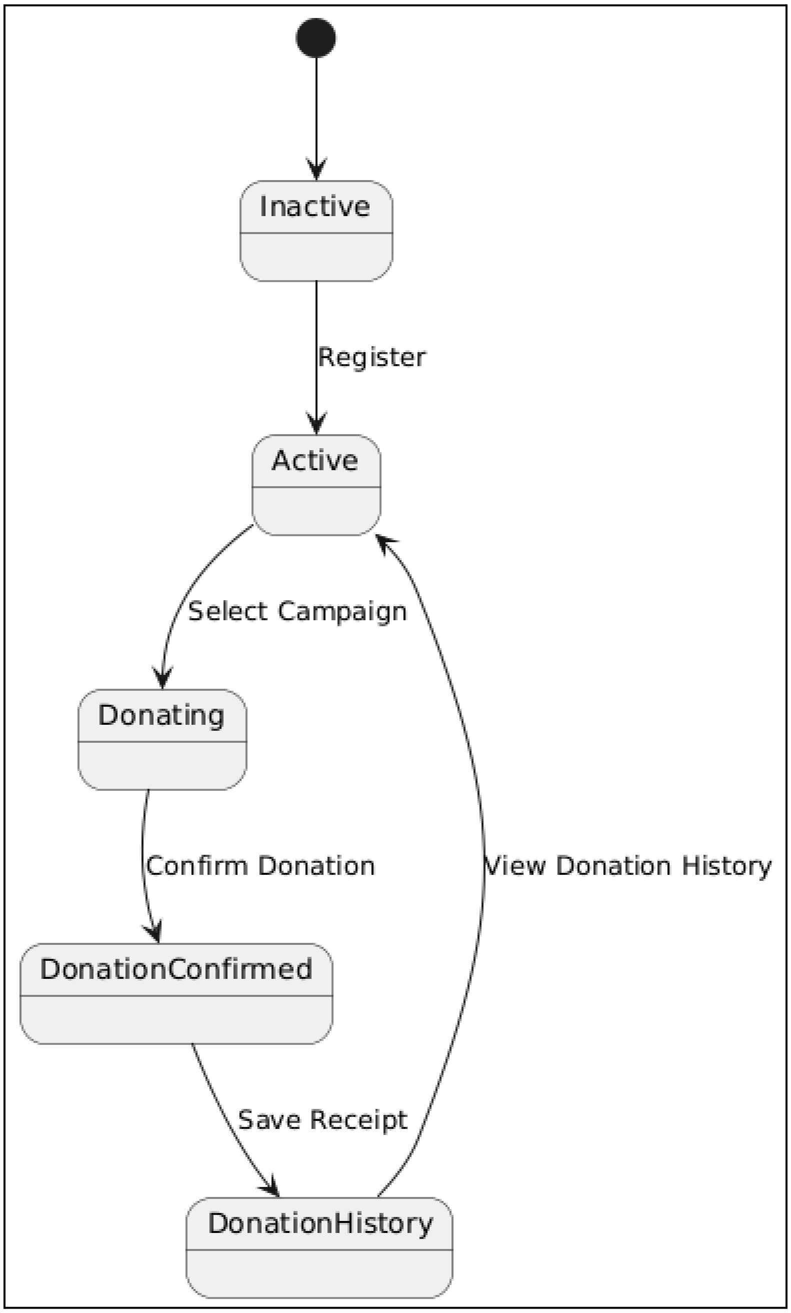


Figure 22 : Donor State Transition

9.1.3 Diagram ID: 3

Class Name: Charity

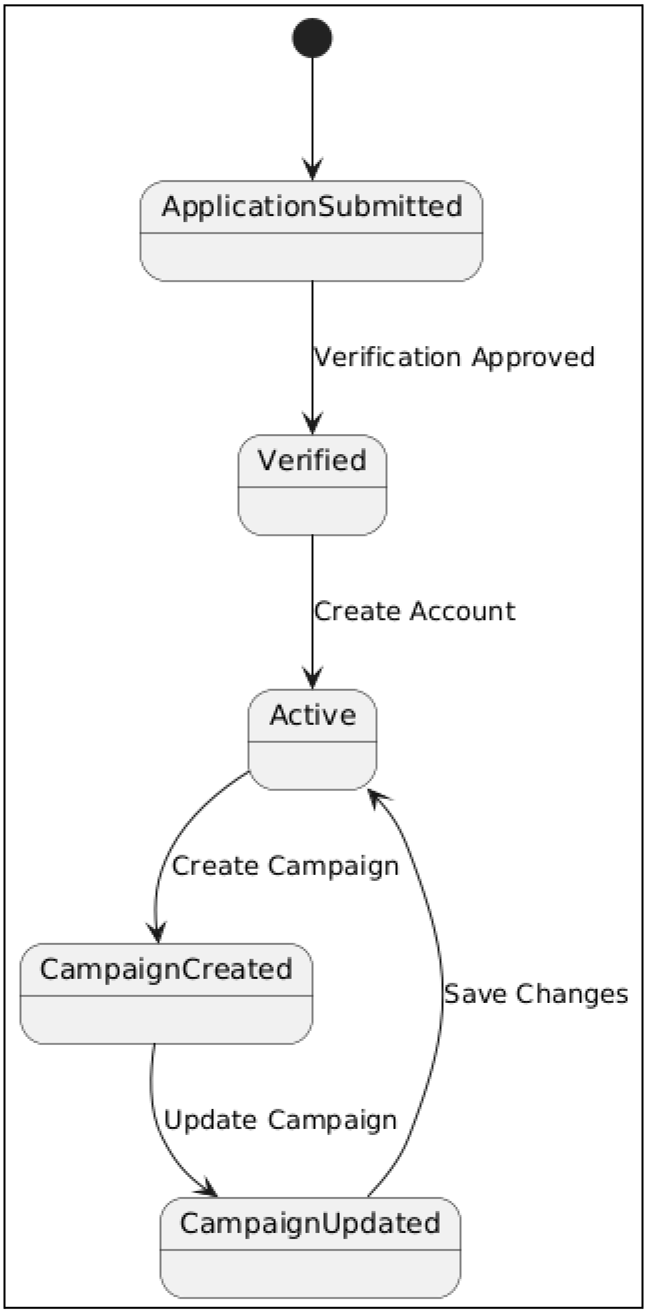


Figure 23 : Charity State Transition

9.1.4 Diagram ID: 4

Class Name: Campaign

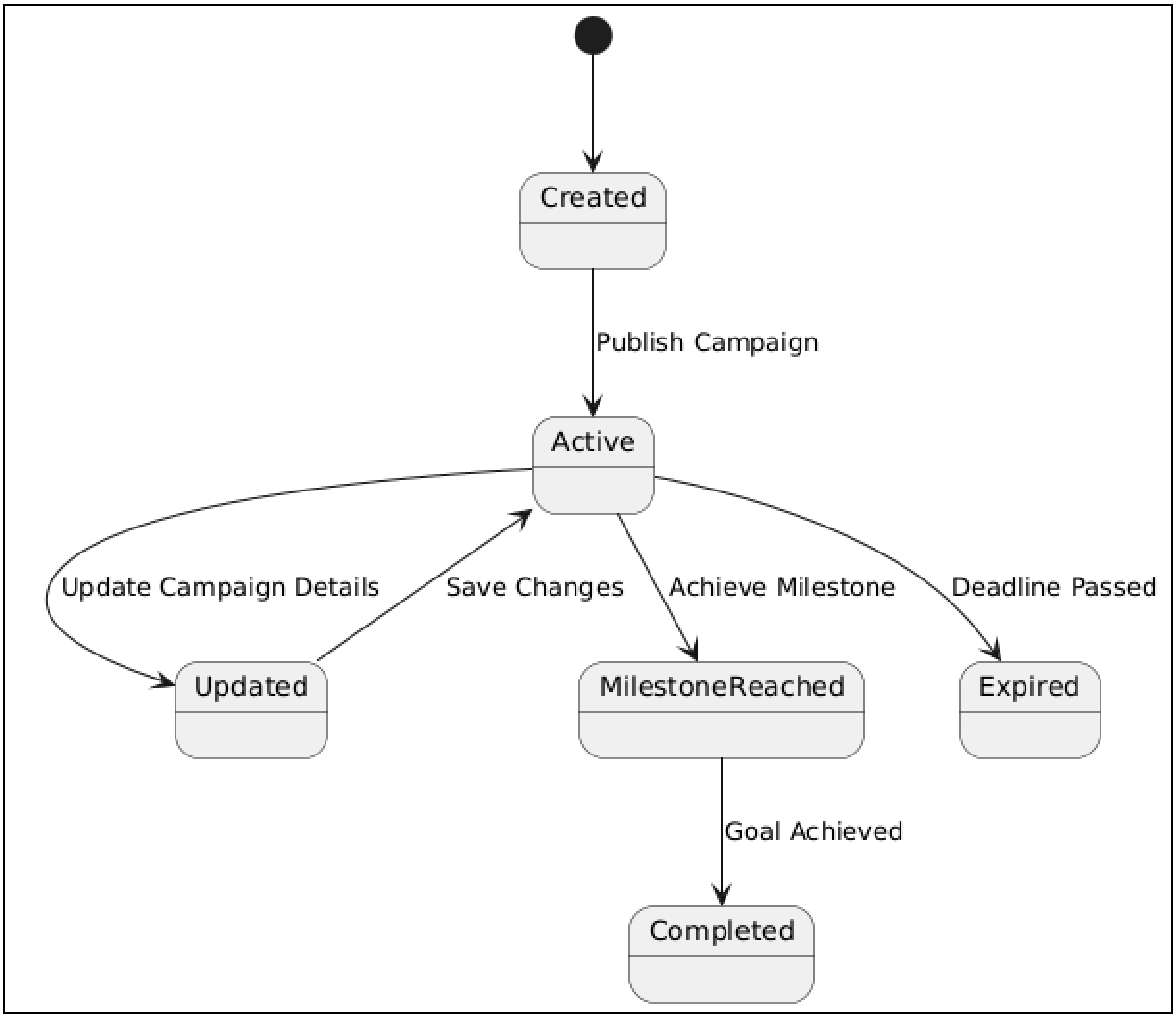


Figure 24 : Campaign State Transition

9.1.5 Diagram ID: 5

Class Name: Transaction

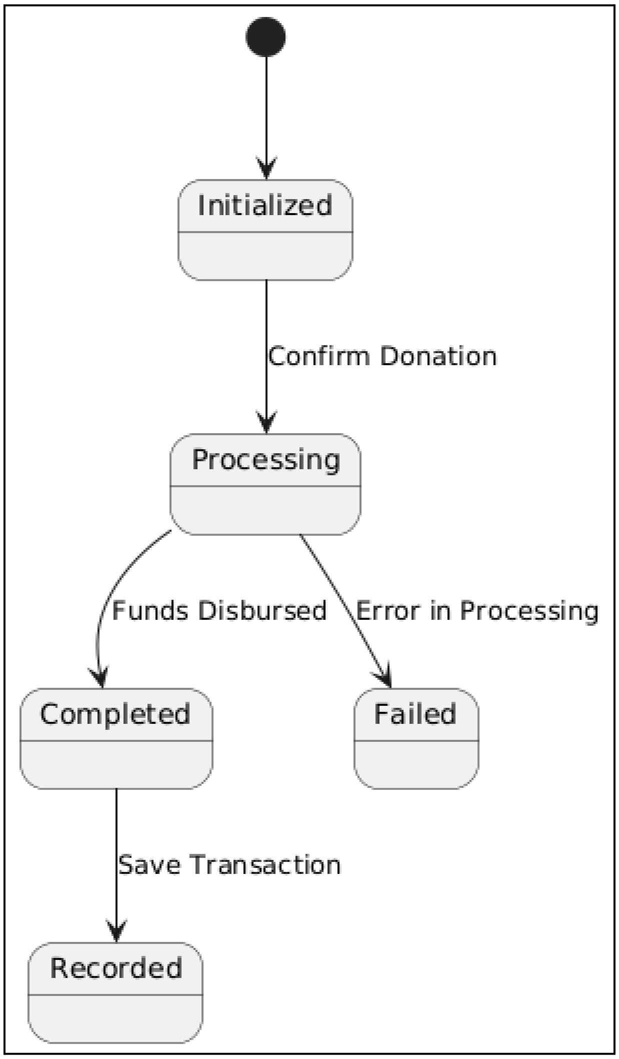


Figure 25 : Transaction State Transition

9.1.6 Diagram ID: 6

Class Name: Notification

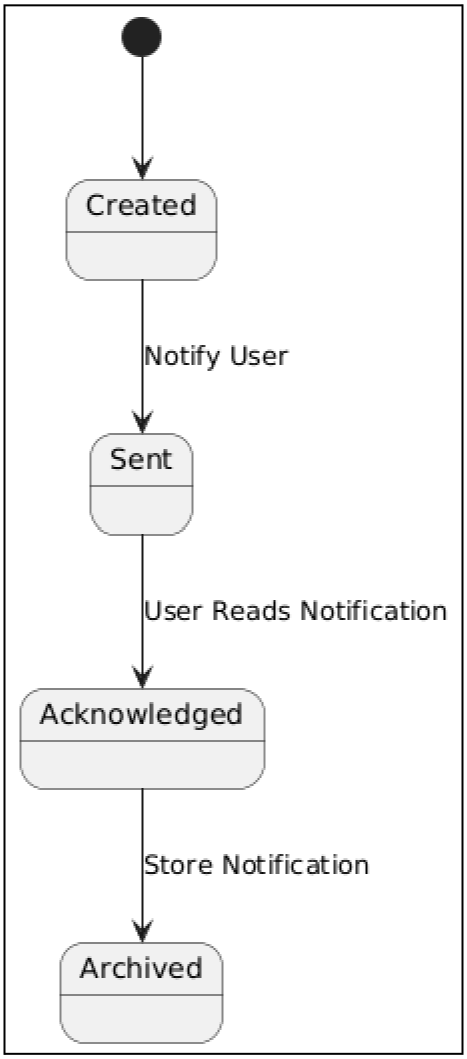
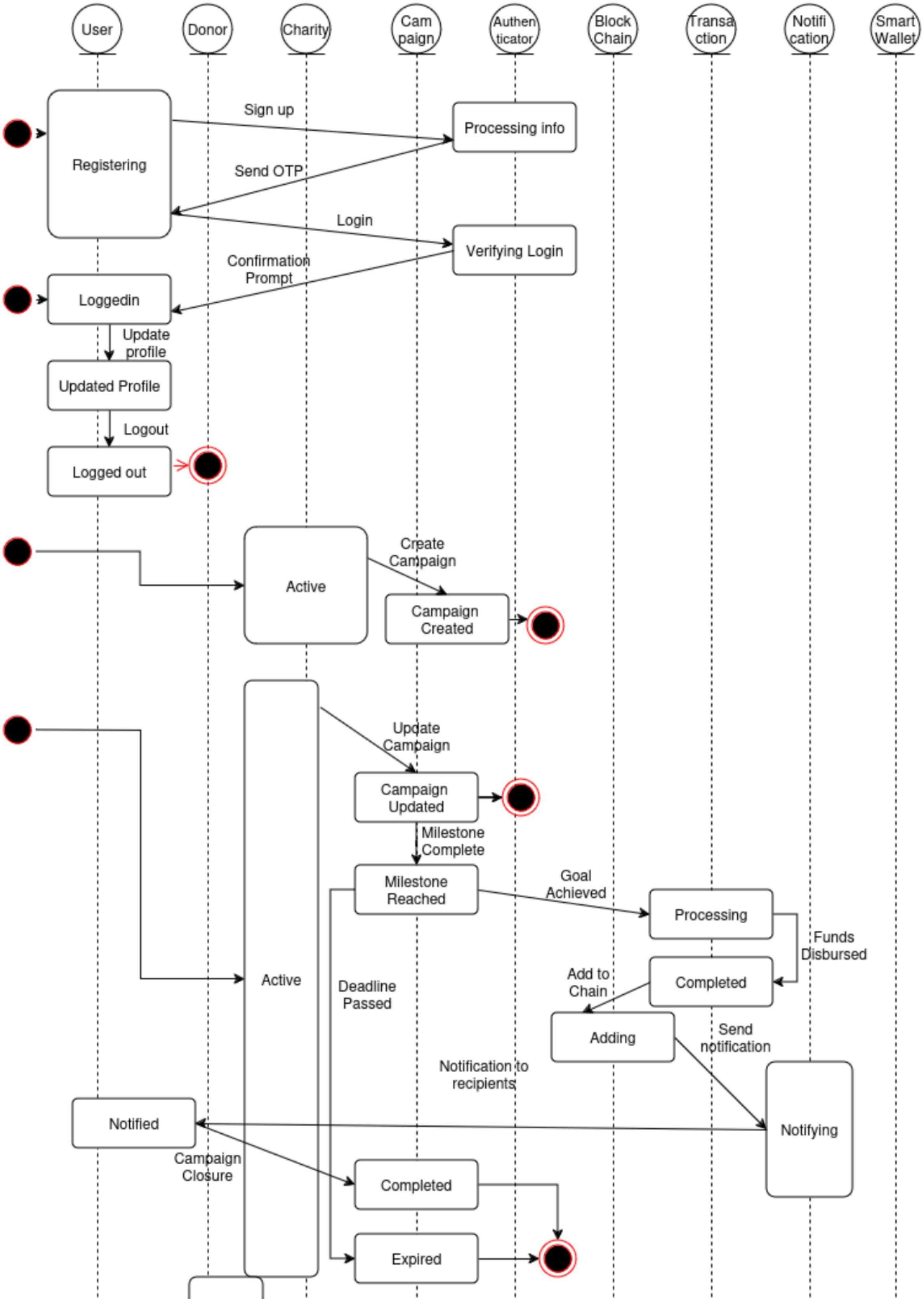


Figure 26 : Notification State Transition

9.2 Sequence Diagram:


User
Donor
Charity
Cam paign
Authen ticator
Block Chain
Transa ction
Notifi cation
Smart Wallet
Registering
Sign up
Processing info
Send OTP
Login
Verifying Login
Confirmation Prompt
Loggedin
Update profile
Updated Profile
Logout
Logged out
Active
Create Campaign
Campaign Created
Update Campaign
Campaign Updated
Milestone Complete
Milestone Reached
Goal Achieved
Processing
Funds Disbursed
Completed
Add to Chain
Adding
Send notification
Deadline Passed
Notification to recipients
Notified
Notifying
Campaign Closure
Completed
Expired

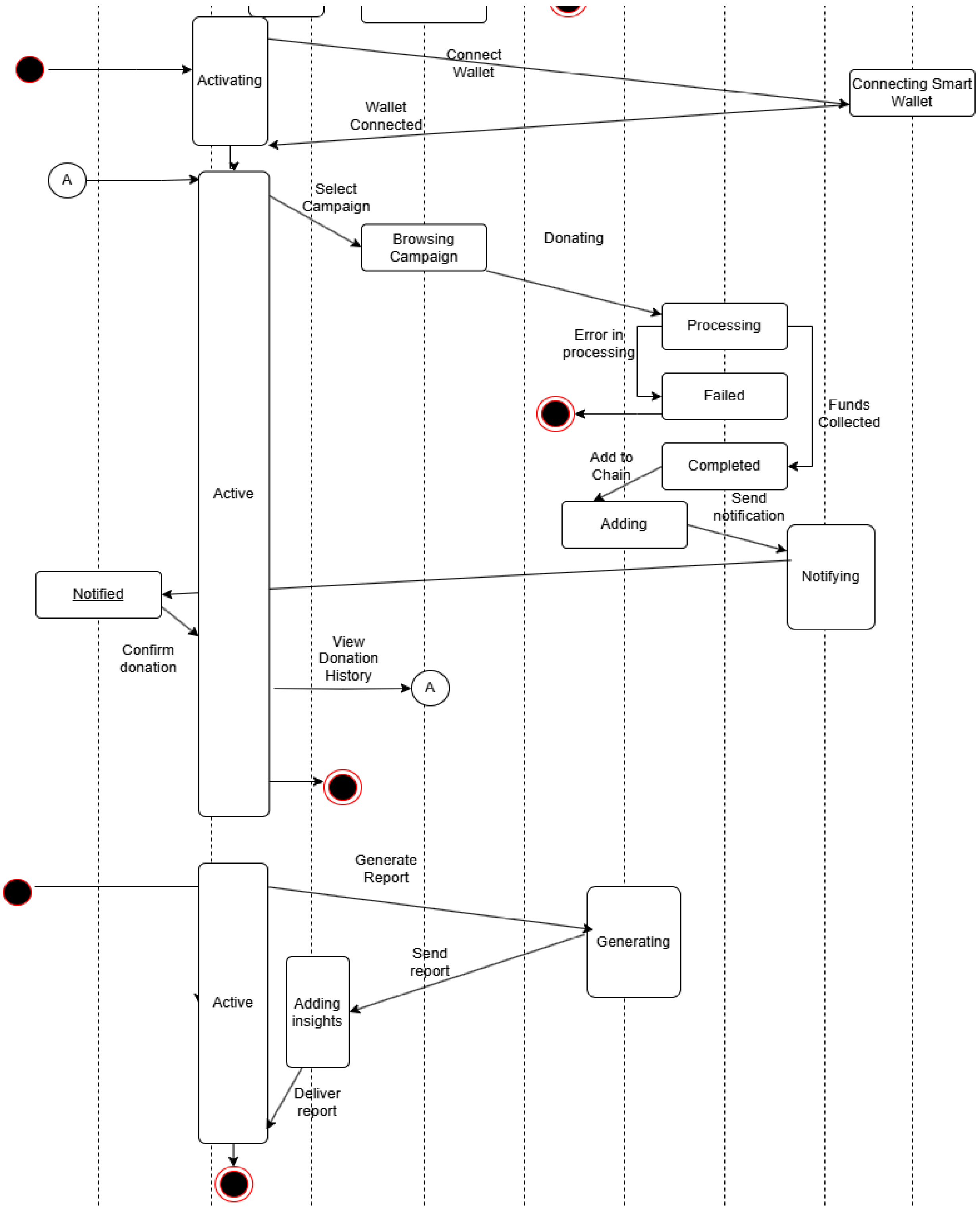


Figure 27 : Sequence Diagram

10. Data flow Modeling

10.1 Level 0

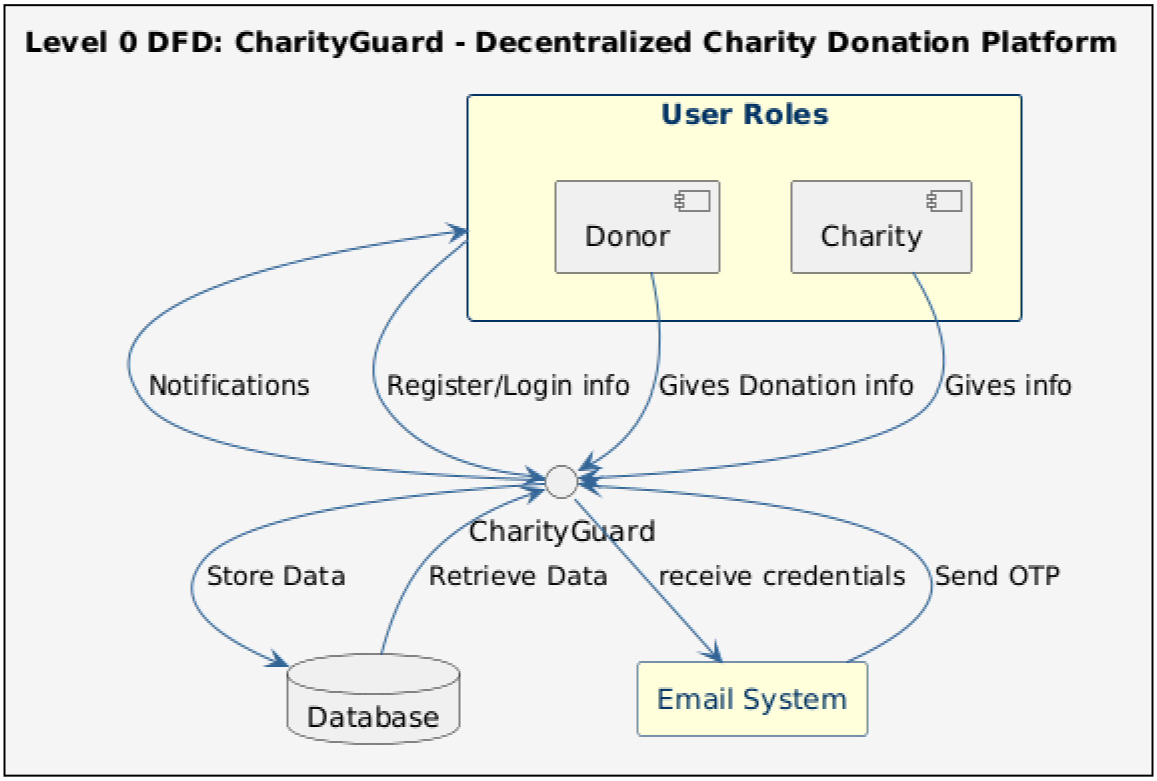


Figure 28 : Data Flow Diagram

10.2 Level 1

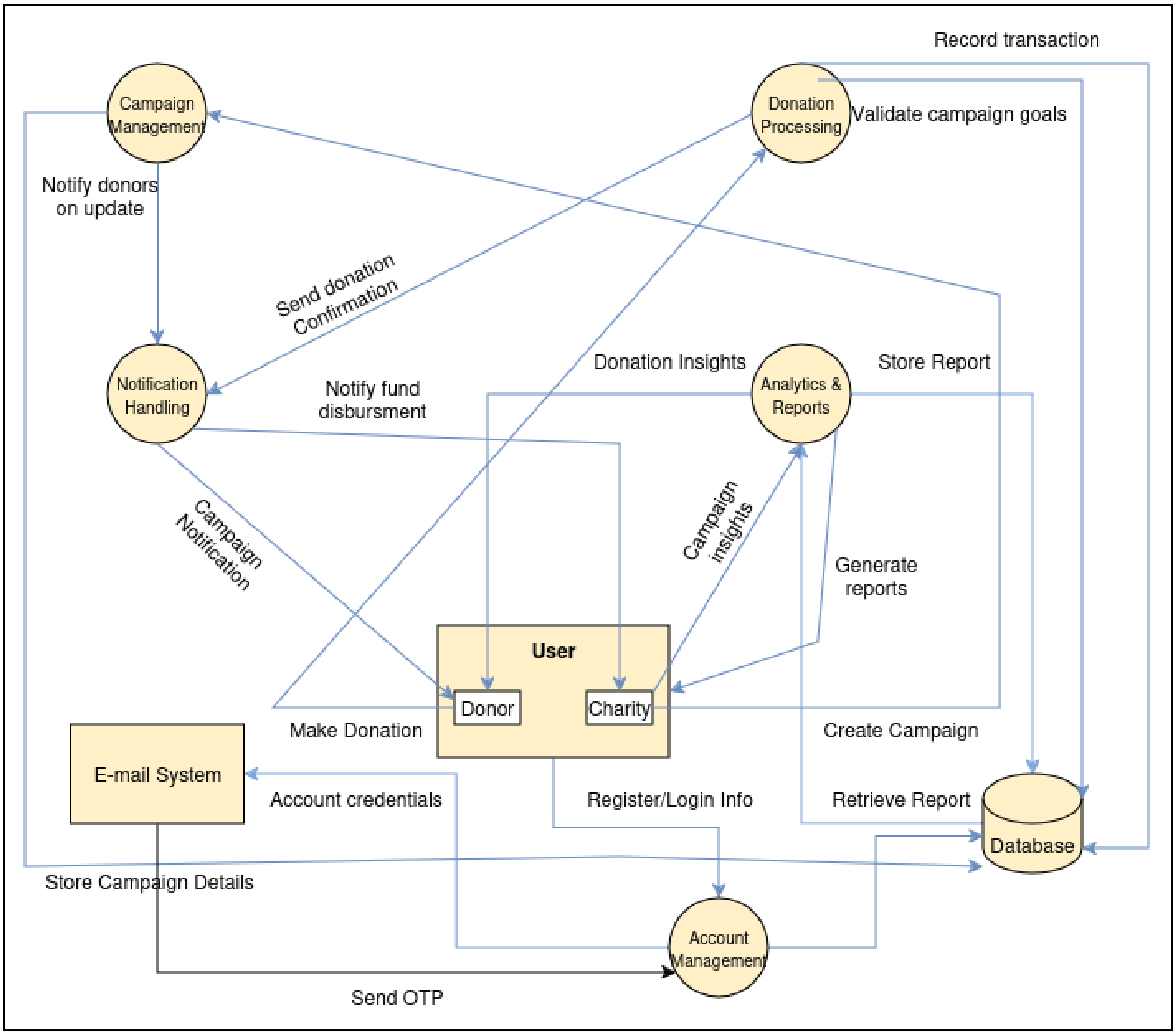


Figure 29 : Data Flow Diagram

11. User Manual

11.1 System Requirements

- Modern web browser (Chrome, Firefox, Edge, Safari)
- MetaMask wallet extension installed
- Ethereum for transactions (can be on mainnet or testnet)

11.2 User Roles

CharityGuard supports three primary user roles, each with specific permissions:

1. **Donor:**

    - Browse campaigns
    - Make donations to campaigns
    - Track donation history
    - Receive notifications about campaigns they've supported
    - Access report about a specific campaign

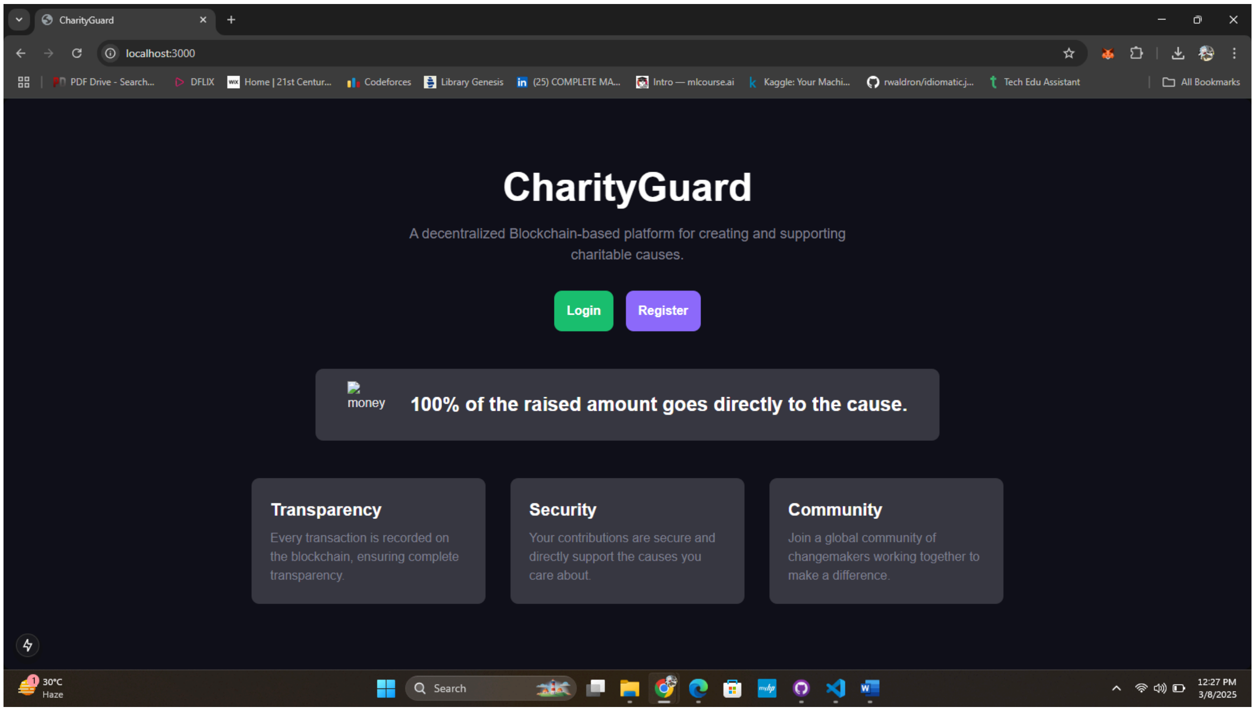
2. **Fundraiser:**

    - All Donor capabilities
    - Create and manage fundraising campaigns
    - Set campaign goals and deadlines
    - Receive notifications about donations to their campaigns
    - Access analytics about campaign performance

3. **Charity:**

    - All Fundraiser capabilities
    - Can also add multiple recipients for a campaign

11.3 Landing page:



CharityGuard

A decentralized Blockchain-based platform for creating and supporting charitable causes.

Login Register

money 100% of the raised amount goes directly to the cause.

Transparency

Every transaction is recorded on the blockchain, ensuring complete transparency.

Security

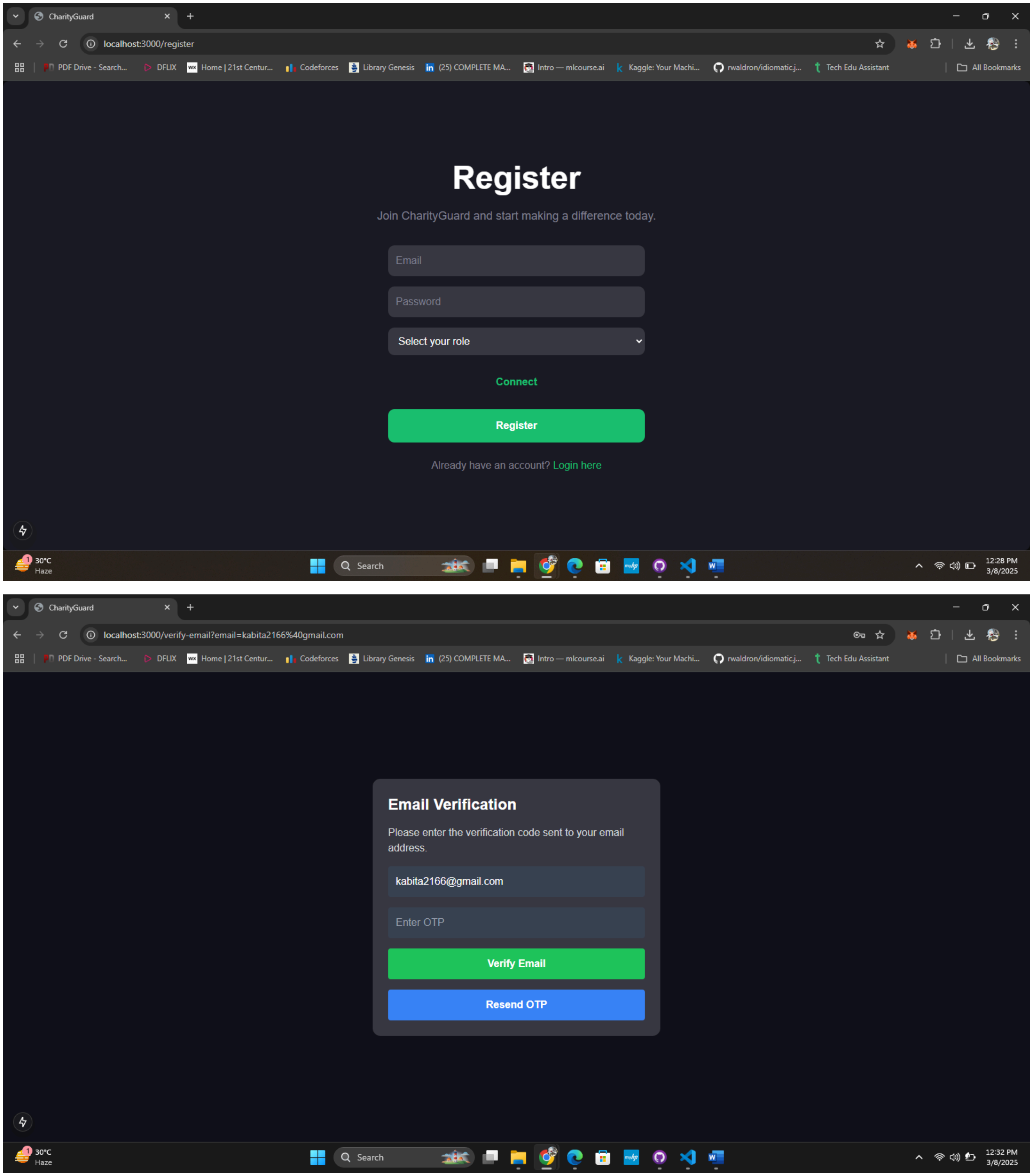
Your contributions are secure and directly support the causes you care about.

Community

Join a global community of changemakers working together to make a difference.

Figure: 30 (Landing page)

11.4 Registration and Authentication


localhost:3000/register

Register

Join CharityGuard and start making a difference today.

localhost:3000/verify-email?email=kabita2166%40gmail.com

Email Verification

Please enter the verification code sent to your email address.

kabita2166@gmail.com

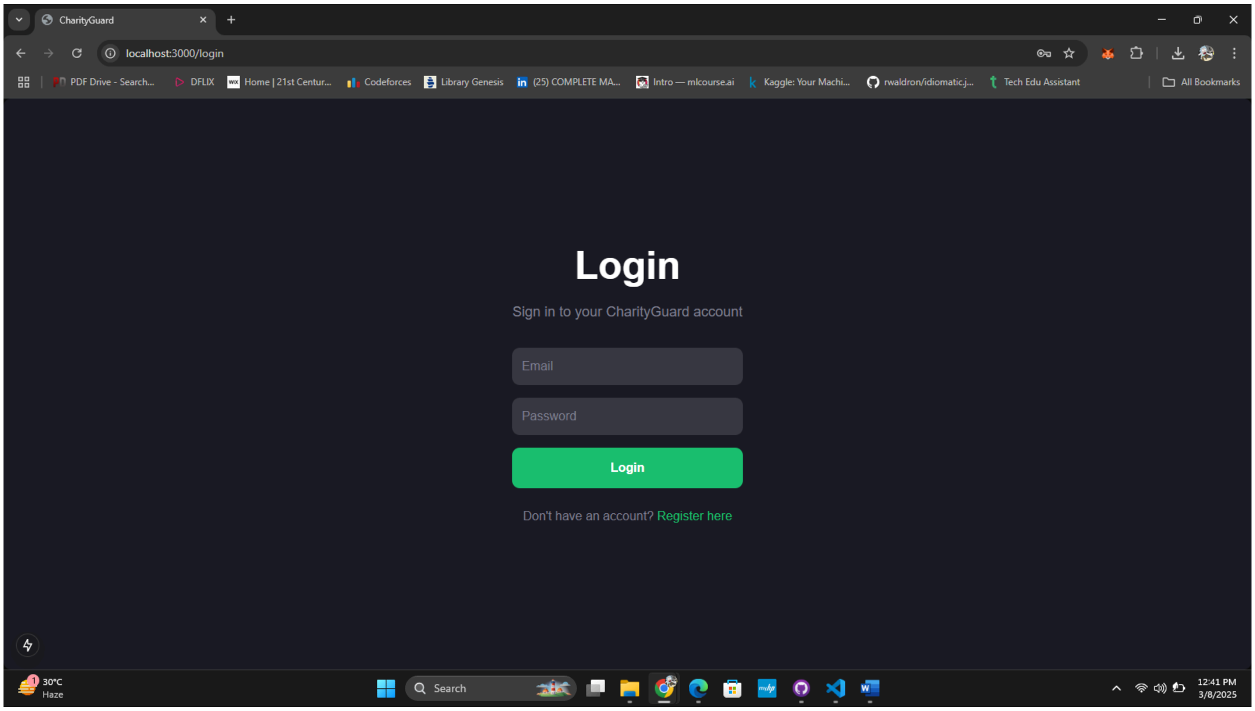
Figure: 31 (Registration and Authentication)

1. **Sign Up:**

- Navigate to the landing page and click "Sign Up"
- Enter your email address and create a password
- Enter your Smart Wallet Address
- Select your user role (Donor, Charity, or Fundraiser)

- Verify your email through the verification code sent to your inbox

2. **Login:**



Login

Sign in to your CharityGuard account

Don't have an account? Register here

Figure: 32 (Login)

- Enter your email and password
- The system will automatically connect to your previously registered account

3. **Wallet Connection:**

- If your wallet is not connected, click "Connect Wallet" in the navigation bar
- Follow the MetaMask prompts to authorize the connection

11.5 Homepage:

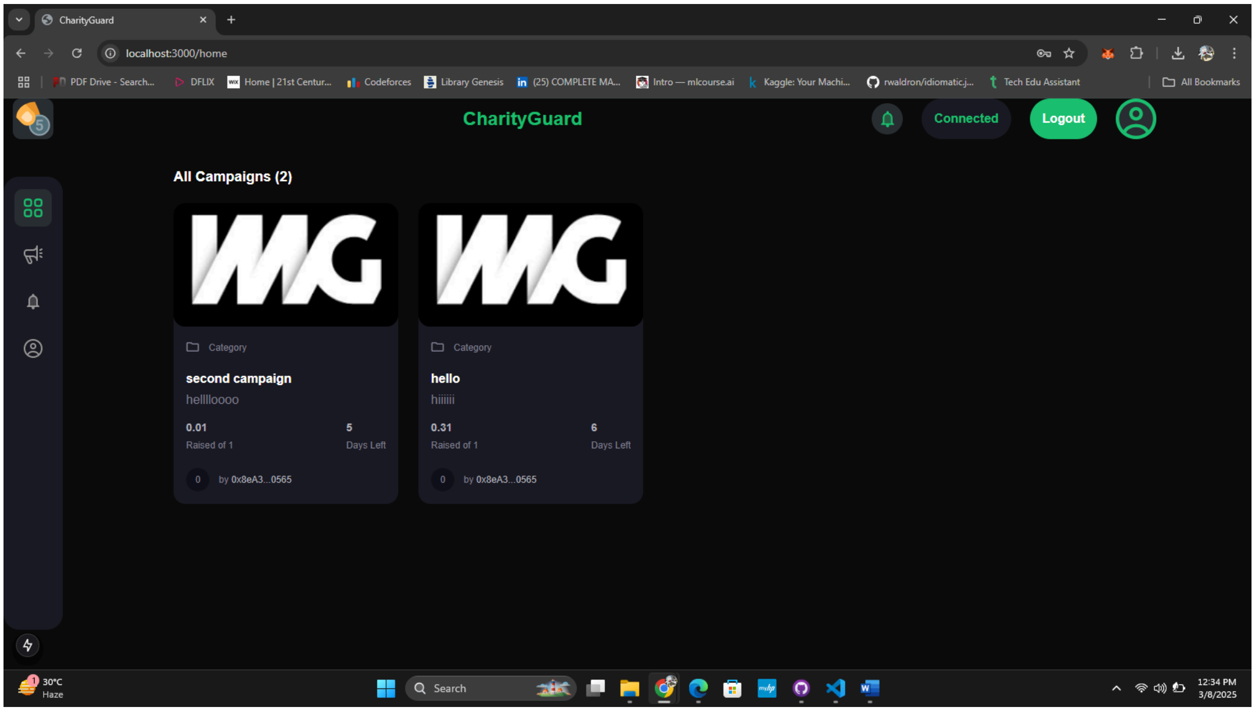
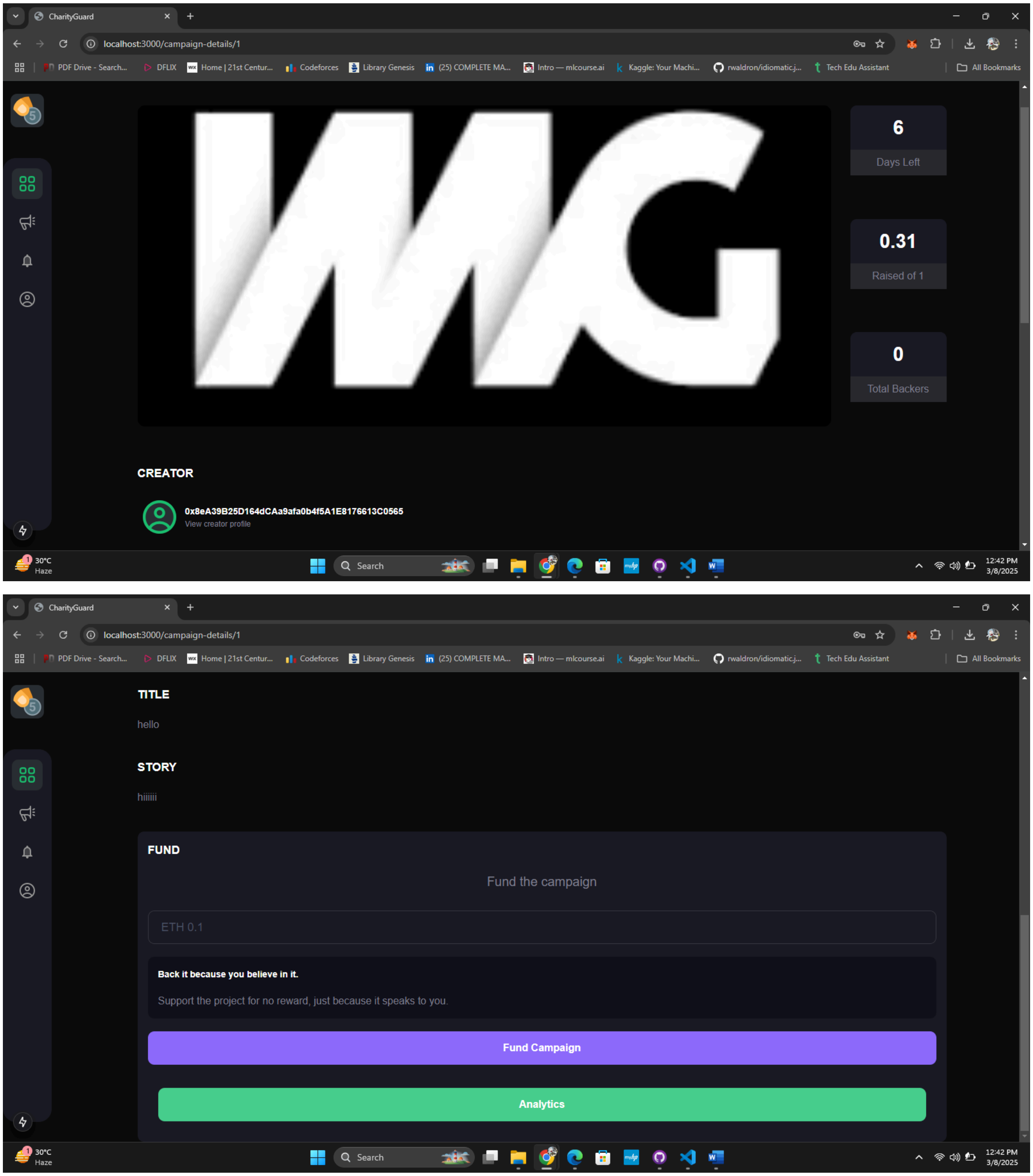


Figure: 33 (Homepage)

Visit the homepage to see featured campaigns

11.6 Campaign Details:



CREATOR

0x8eA39B25D164dCAa9afa0b4f5A1E8176613C0565

View creator profile

TITLE

hello

STORY

hiiiii

FUND

Fund the campaign

ETH 0.1

Back it because you believe in it.

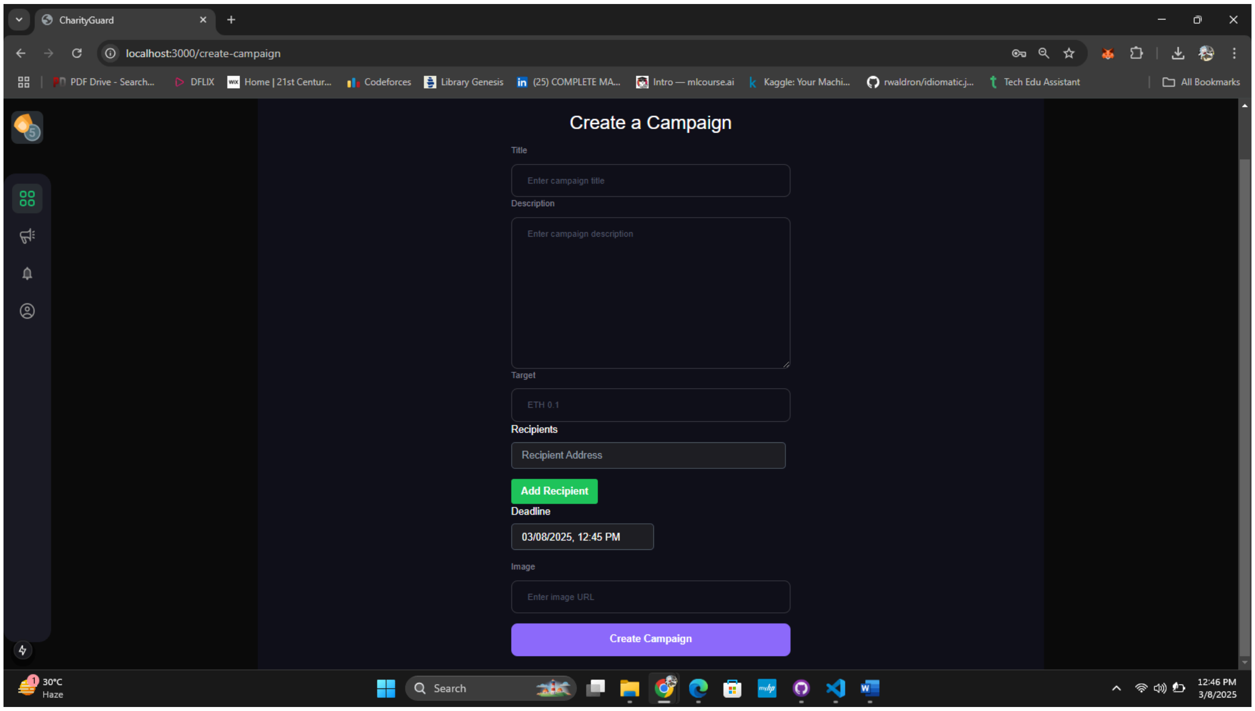
Support the project for no reward, just because it speaks to you.

Figure: 34 (Campaign Details)

- Click on any campaign to view detailed information
- See the campaign goal, current funding, deadline, and description
- View campaign creator information which will redirect to the user-profile page
- View analytics of the campaign which will redirect to campaign-report page

11.7 Campaign Management

Campaign creation:



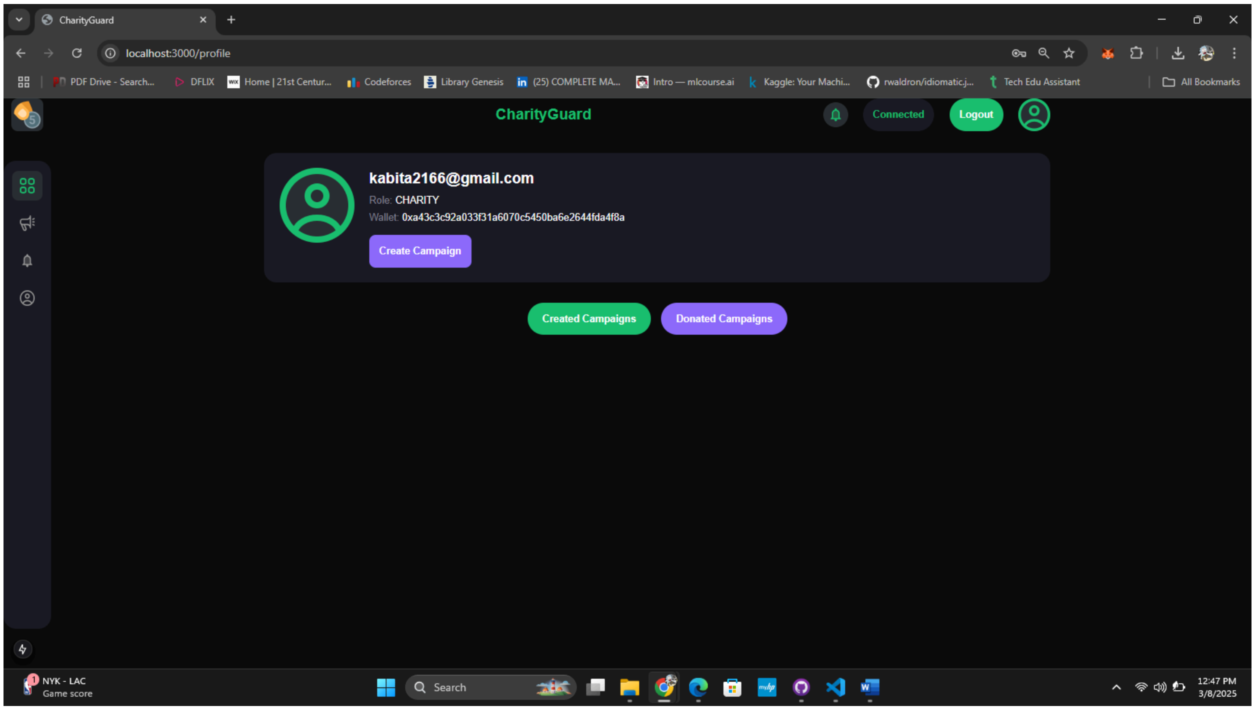
Deadline

03/08/2025, 12:45 PM

Figure: 35 (Campaign Creation)

- Click "Create Campaign" in the navigation bar
- Fill in the required details:
- Title: A concise name for your campaign
- Description: Detailed information about the purpose and goals
- Funding Goal: The amount of ETH you aim to raise
- Deadline: The date by which you hope to reach your goal
- Recipients(only for charity): Add wallet addresses for intended beneficiaries with allocation
- Submit your campaign for creation on the blockchain
- Once confirmed, you'll receive a notification that your campaign is live

Managing Existing Campaigns:



kabita2166@gmail.com

Role: CHARITY

Wallet: 0xa43c3c92a033f31a6070c5450ba6e2644fda4f8a

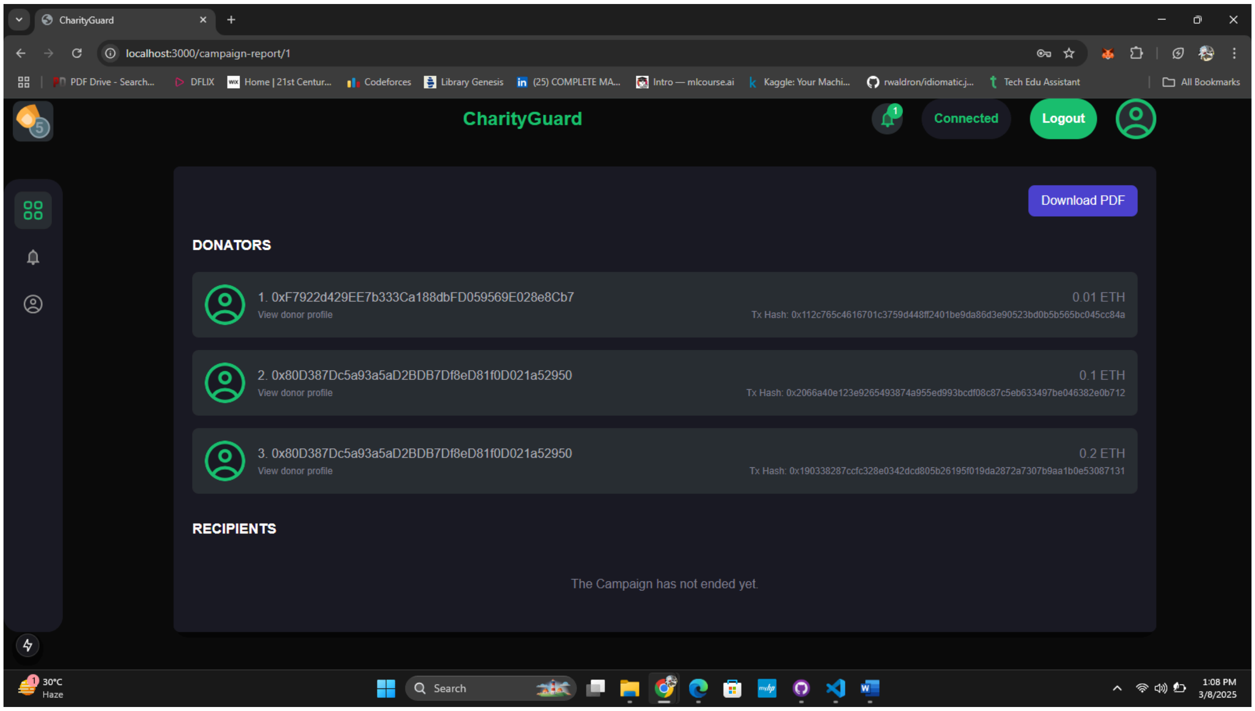
Figure: 36 (Profile)

- Navigate to "My Profile" and select "Created Campaigns" or, “Donated Campaigns”
- View your active and completed campaigns
- Monitor donation progress in real-time
- Receive notifications when donations are made or milestones are reached

Donation Process:

- Find a campaign you wish to support
- Click "Donate" button on the campaign page
- Enter the donation amount in ETH
- Confirm the transaction in your MetaMask wallet
- Wait for blockchain confirmation
- Receive a notification confirming your donation
- Tracking Donations
- Visit "My Profile" to see your donation history
- View campaigns you've supported with donation amounts
- Check the progress of campaigns you've donated to
- Receive notifications about campaign milestones and updates

11.8 Campaign Report Page:



DONATORS

1. 0xF7922d429EE7b333Ca188dbFD059569E028e8Cb7 View donor profile 0.01 ETH Tx Hash: 0x112c765c4616701c3759d448ff2401be9da86d3e90523bd0b5b565bc045cc84a

2. 0x80D387Dc5a93a5aD2BDB7Df8eD81f0D021a52950 View donor profile 0.1 ETH Tx Hash: 0x2066a40e123e9265493874a955ed993bcdf08c87c5eb633497be046382e0b712

3. 0x80D387Dc5a93a5aD2BDB7Df8eD81f0D021a52950 View donor profile 0.2 ETH Tx Hash: 0x190338287ccfc328e0342dcd805b26195f019da2872a7307b9aa1b0e53087131

RECIPIENTS

The Campaign has not ended yet.

Figure: 37 (Campaign Report Page)

11.9 Notification System

CharityGuard keeps users informed through a comprehensive notification system:

Types of Notifications

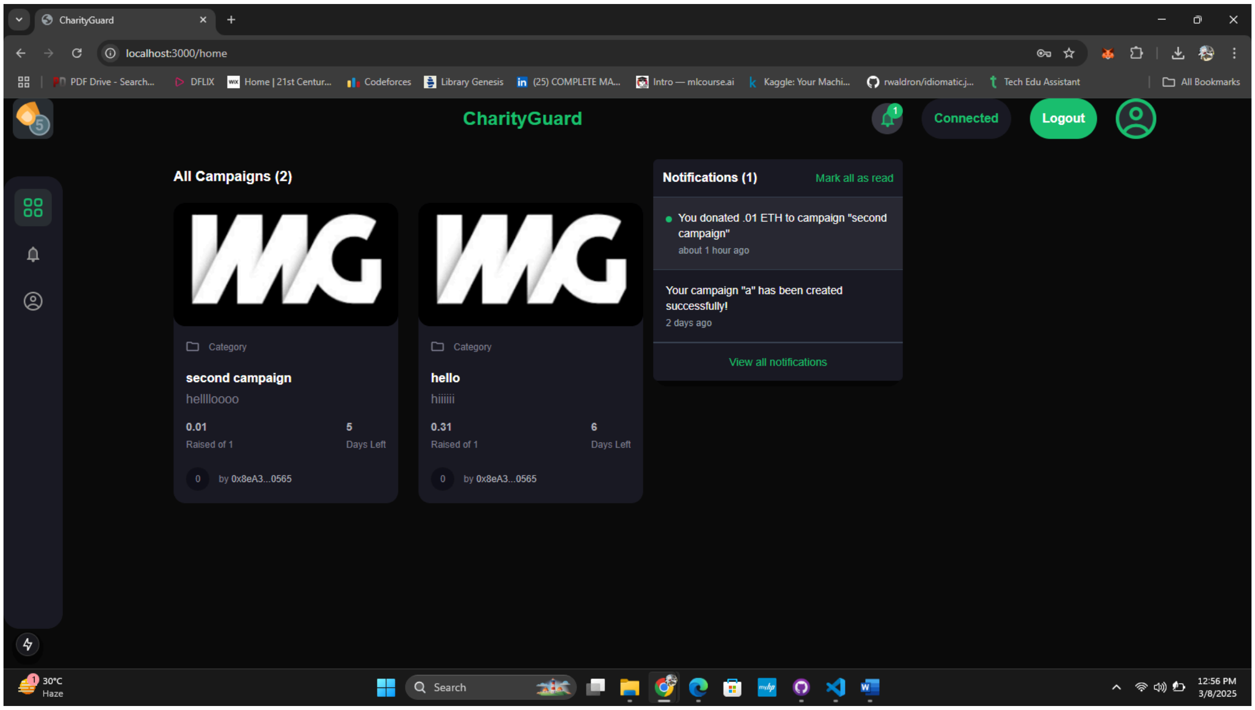
For Donors:

- Confirmation of successful donations
- Campaign completion notifications

For Campaign Creators:

- New donation alerts
- Campaign milestone achievements
- Approaching deadlines
- Campaign completion notifications

Managing Notifications



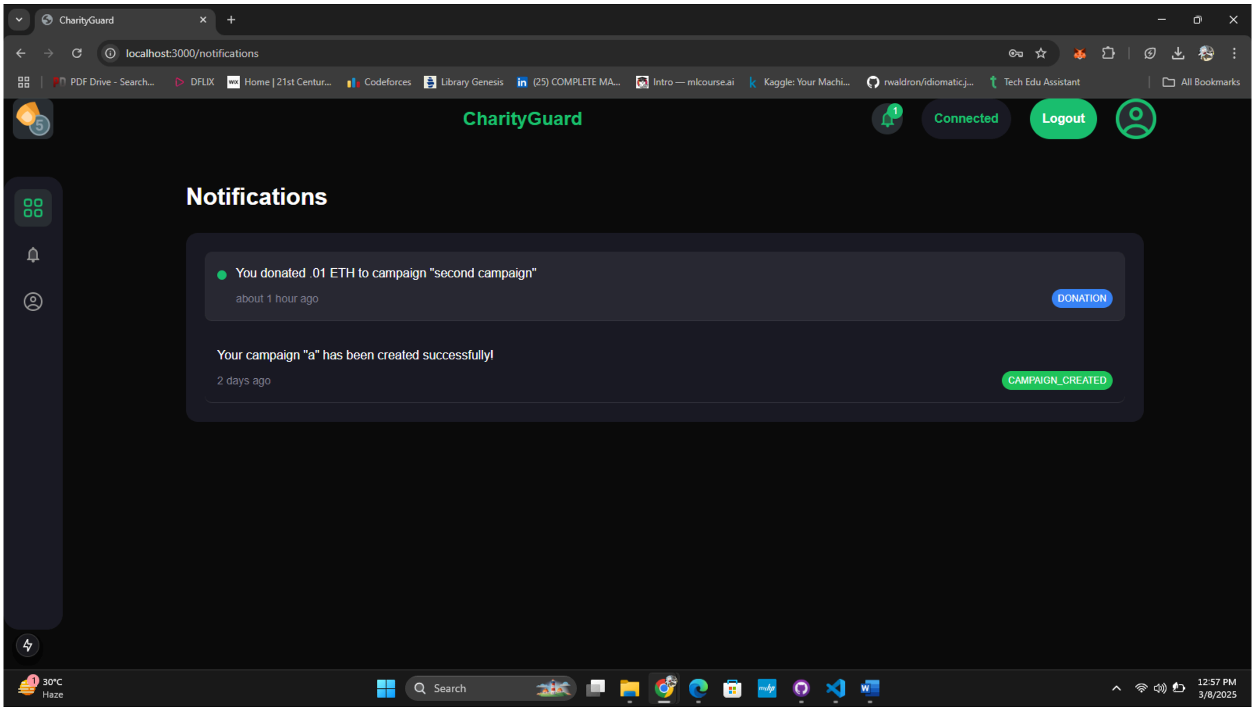
All Campaigns (2)

Category	Category
second campaign	hello
hellllooo	hiiiiii
0.01 Raised of 1	0.31 Raised of 1
5 Days Left	6 Days Left
0 by 0x8eA3...0565	0 by 0x8eA3...0565

Notifications (1) Mark all as read

- You donated .01 ETH to campaign "second campaign"
  about 1 hour ago
- Your campaign "a" has been created successfully!
  2 days ago

View all notifications



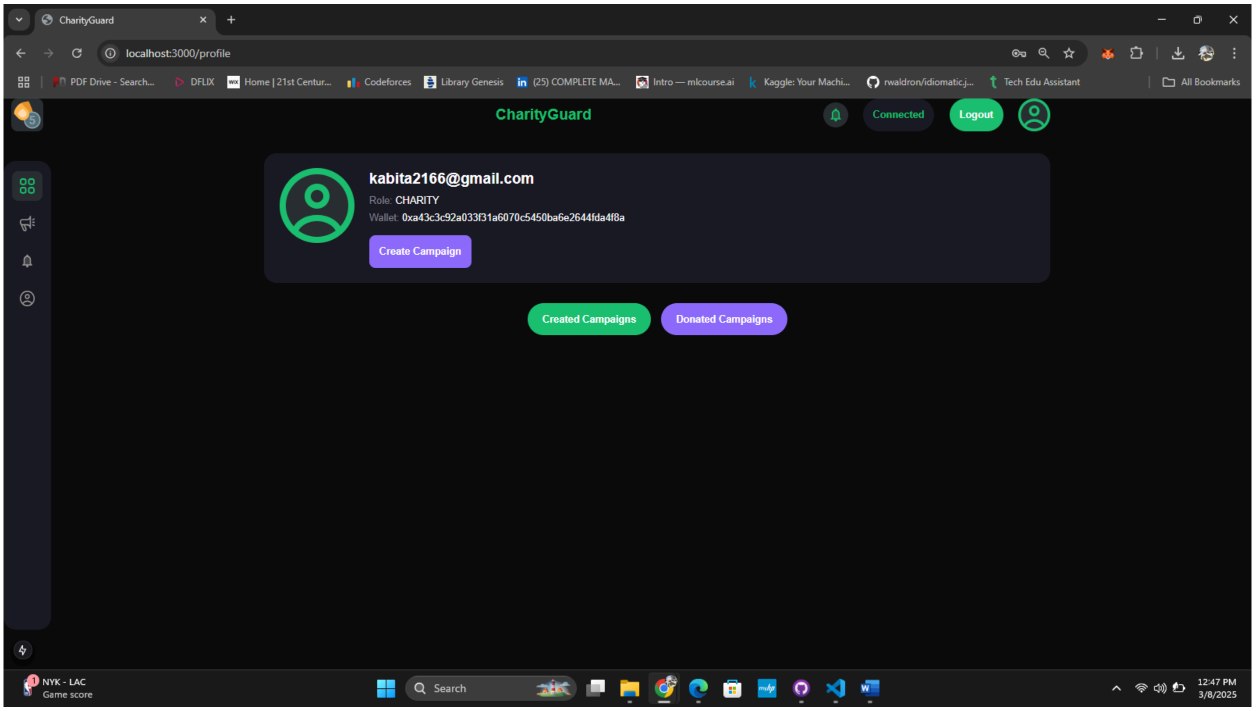
Notifications

- You donated .01 ETH to campaign "second campaign"
  about 1 hour ago DONATION
- Your campaign "a" has been created successfully!
  2 days ago CAMPAIGN_CREATED

Figure: 38 (Notifications)

- Click the notification bell icon in the navigation bar
- View all recent notifications
- Click a notification to navigate to the relevant campaign
- Mark notifications as read individually or all at once

11.10 User Profiles



kabita2166@gmail.com

Role: CHARITY

Wallet: 0xa43c3c92a033f31a6070c5450ba6e2644fda4f8a

Figure: 39 (User Profile)

Viewing Your Profile:

- Click "My Profile" in the navigation menu
- View your registered information: email address, connected wallet address,user role

User Profile Features

For Donors:

- View donation history
- Total amount donated
- List of supported campaigns

For Campaign Creators:

- View created campaigns
- Campaign performance metrics
- Donation notifications
- Viewing Other Users' Profiles

Click on a user's name or wallet address from a campaign page
See their public profile including: Campaigns they've created, Role in the platform, Wallet address

11.11 Security Features

Wallet Integration
CharityGuard integrates with MetaMask for secure transaction signing
Users maintain full control of their private keys
All blockchain transactions require explicit user confirmation

Email Verification
All user accounts require email verification during registration

Smart Contract Security
Campaign funds are managed through secure smart contracts
Fund disbursement follows predefined rules and allocations
Campaign creators cannot withdraw funds directly; funds are transferred automatically to recipients

11.12 Troubleshooting

Wallet Connection Problems:

- Ensure MetaMask is installed and logged in
- Check if you're on the correct network
- Try refreshing the page
- Disconnect and reconnect the wallet

Transaction Failures:

- Ensure you have sufficient ETH for the donation and gas fees
- Check network congestion and adjust gas prices if necessary
- Verify that the campaign is still active

Missing Notifications:

- Check your browser permissions for notifications
- Ensure you're logged in with the correct account
- Verify that the campaign creator's wallet address is correctly linked to their user account

Campaign Creation Issues:

- Ensure all required fields are completed
- Check that recipient addresses are valid Ethereum addresses
- Verify that allocation percentages total 100%

12. Challenges Faced

Building and launching CharityGuard came with several challenges:

1. **Smart Contract Security**: Ensuring smart contracts were secure was crucial to prevent hacking and fund loss (Buterin, 2014).
2. **User Adoption**: Many users are unfamiliar with blockchain, making wallet integration and transaction fees confusing (Nakamoto, 2008).
3. **Transaction Fees**: Gas fees can be high, making small donations or any kind of transaction very expensive. (Gudgeon et al., 2020).
4. **Scalability**: As the number of campaigns and transactions grows, the platform must remain fast and responsive (Wood, 201.

13. Future Scope

There are many ways to improve CharityGuard in the future:

1. **Support for More Blockchains**: Adding networks like Binance Smart Chain and Solana could lower fees and increase efficiency (Gudgeon et al., 2020).
2. **Community Governance (DAO)**: Users could vote on platform updates and campaign approvals through a decentralized decision-making system (Buterin, 2014).
3. **Global Donations**: Enabling multiple cryptocurrencies and stablecoins or, fiat coins that would allow more people to participate.
4. **Mobile App**: A dedicated mobile app would make the platform easier to use on the go.

5. **AI-Based Fraud Detection**: Machine learning could help detect suspicious activities and prevent fraud.

14. Conclusion

CharityGuard is a promising solution for making charitable donations more transparent and efficient. By using blockchain, it removes trust issues, reduces costs, and ensures that donations are used correctly.

Although there are challenges, such as security and transaction fees, the platform has great potential. Future updates like multi-chain support, fraud detection, and DAO governance could make it even more effective.

With continuous improvements, CharityGuard could set a new standard for decentralized philanthropy, ensuring that every donation makes a real difference.

References

1. Buterin, V. (2014). "A Next-Generation Smart Contract and Decentralized Application Platform." Ethereum Whitepaper. Retrieved from https://ethereum.org/en/whitepaper/
2. Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System." Retrieved from https://bitcoin.org/bitcoin.pdf
3. Wood, G. (2015). "Ethereum: A Secure Decentralised Generalised Transaction Ledger." Ethereum Yellow Paper. Retrieved from https://ethereum.github.io/yellowpaper/paper.pdf
4. Gudgeon, L., Moreno-Sanchez, P., Roos, S., & Bonneau, J. (2020). "SoK: Layer-Two Blockchain Protocols." Retrieved from https://arxiv.org/abs/2006.01760