

# **Crowd Funded Donation Management System**

**R.A.P.B.M Jayasundara  
2023**



# **Crowd Funded Donation Management System**

**A thesis submitted for the Degree of Master of  
Information Technology**

**R.A.P.B.M Jayasundara  
University of Colombo School of Computing  
2023**



# Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: R.A.P.B.M Jayasundara

Registration Number: 2019/MIT/038

Index Number: 19550383




Signature:

Date: 17-11-2022

This is to certify that this thesis is based on the work of Mr. R.A.P.B.M Jayasundara under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by:

Supervisor Name: Prof. K. P. Hewagamage



Signature:

Date:

# Abstract

In simple terms, Charity is an act of kindness where a strong person helps a weak person. It is one of the key fundamentals of a highly developed society. There are lots of Non-Governmental Organizations (NGOs) that are conducting charity events on large scale but today even they also experienced difficulties in getting funds or other required things. Inflation and economic downturn made people care less about charity.

Building trust with the public is another biggest challenge in fundraising. Donors want to be sure that their money will make a difference. Even if there are only a few scammers out there, it discourages people from donating to charities in general. When one institution collapses, many individuals feel that all organizations are untrustworthy. Today people want to engage in charity/fundraising events to feel like they're a part of the story, not a passing observer.

“Charity” is a crowd-funded donation management system that is built to manage fundraising events. The main objective of this project is to provide a transparent, socially engaging platform for donors. This concept is heavily inspired by google’s local guide program. From creating a campaign to making a donation, the whole process is handled by the system. Authorized persons can verify the campaign's authenticity. This will increase the donor's trust in the fundraising campaign. The system was developed in a way that people can share donation campaigns on social media. It is very useful for the campaign organizers to raise awareness of their campaigns ultimately it’ll help to build a social network among donors. A clear dashboard with many useful reports are presented to the user very clearly. In addition, Performance, Security, Reliability, and Maintainability are some of the features which are considered during the implementation of the project.

The proposed system was implemented as a web-based system using Vue.js, a progressive javascript framework, and Google firebase was used as the backend database. The backend API server was developed using Node.Js. After implementing the project testing was carried out at different testing levels. Several manual testing techniques were used to test the front-end web application. The “Mocha” testing framework was used to carry out automated testing for back-end server testing. After the project testing, the system was deployed into the live environment for user evaluation. The user evaluation process was carried out to evaluate the system to get user feedback and improvements. Most of the participants were satisfied with the system and a few improvements were suggested to enhance the system.

# Acknowledgment

I would like to express my gratitude to my supervisor, Prof. K.P Hewagamage sir, for the guidance, encouragement, and useful advice given to me throughout the project.

Also, I would like to thank the staff of the postgraduate unit of the University of Colombo School of Computing (UCSC) for their support given during the project.

Finally, I would be negligent if I did not mention my family, particularly my parents. Their confidence in me has maintained my spirits and motivation strong throughout the process. This accomplishment would not have been possible without them. Thank you.

## Table of Contents

<b>Declaration</b> .....	i
<b>Abstract</b> .....	ii
<b>Acknowledgment</b> .....	iii
<b>Chapter 1 - Introduction</b> .....	1
1.1. Motivation.....	1
1.2. Background of the Study .....	1
1.3. Objectives .....	2
1.4. Scope of the Study.....	2
1.5. Dissertation Outline.....	4
<b>Chapter 2 - Background of the Study</b> .....	5
2.1. Introduction .....	5
2.2. Requirement Analysis .....	5
2.3. Review of Similar Systems .....	6
2.4. Related Technologies .....	10
<b>Chapter 3 - Design</b> .....	12
3.1. Introduction .....	12
3.2. Related Design Strategies.....	12
3.3. System Architecture .....	15
3.4. UML Diagrams .....	16
3.4.1. System Use case Diagram.....	16
3.4.2. Use case Scenarios.....	18
3.4.3. Activity Diagrams.....	20
3.4.4. Sequence Diagram .....	24
<b>Chapter 4 - Implementation</b> .....	27
4.1. Introduction .....	27
4.2. Implementation Environment .....	27
4.3. System Development tools and techniques .....	27
4.4. Reusable Components.....	28
4.5. Major Code Segments .....	28

<b>Chapter 5 - Testing and Evaluation</b> .....	35
5.1. Introduction .....	35
5.2. Related Testing Types.....	35
5.3. Test Cases.....	38
5.3.1. Test Cases for User Authentication .....	38
5.3.2. Test Cases for User Registration .....	39
5.3.3. Test Cases for Campaign Creation .....	39
5.3.4. Test Cases for Donation .....	40
5.3.5. Test Cases for Share Campaigns on Social Media .....	41
5.4. User Evaluation.....	42
5.5. Results of the Testing.....	43
<b>Chapter 6 – Conclusion</b> .....	47
6.1. Introduction .....	47
6.2. Critical Evaluation .....	47
6.3. Problem Encountered.....	47
6.4. Lessons Learned .....	48
6.5. Potential Future Work.....	48
<b>References</b> .....	49
<b>Appendix A - System Documentation</b> .....	51
<b>Appendix B - Design Documentation</b> .....	53
<b>Appendix C - User Manual</b> .....	62
<b>Appendix D - Management Reports</b> .....	76

## Table of Figures

Figure 1.1: Point System of Google's Local Guide.....	3
Figure 1.2: Badges of Google's Local Guide Platform .....	3
Figure 2.1: Tickets.lk Logo .....	7
Figure 2.2: Events.lk Logo .....	7
Figure 2.3: Dialog Star points donation for little hearts fund.....	7
Figure 2.4: Global Giving Website Home page.....	8
Figure 2.5: Mightycause Website Home page.....	8
Figure 2.6: Reasons to choose Vue.Js (Patel, J 2021).....	10
Figure 2.7: Sample log created using Winston .....	11
Figure 3.1: Paper prototype for sign in screen .....	12
Figure 3.2: Paper prototype for Sign Up screen .....	13
Figure 3.3: Paper prototype for Campaign page .....	13
Figure 3.4: Single Campaign Document .....	14
Figure 3.5: Sample Document Structure .....	15
Figure 3.6: High-Level Architecture diagram .....	16
Figure 3.7: System Use Case Diagram.....	17
Figure 5.1: Test execution hierarchy .....	36
Figure 5.2: Bottom-up incremental integration testing approach .....	37
Figure 5.3: Sample Mocha automated test case.....	38
Figure 5.4: User Evaluation Questionnaire.....	42
Figure 5.5: Overall result of the user evaluation.....	43
Figure 5.6: Overall result of user evaluation (Process evaluation).....	43
Figure 5.7: Result of similar system usage .....	44
Figure 5.8: User recommendation.....	44
Figure 5.9: Overall Automated Test Results .....	45
Figure 5.10: Test automation results .....	46
Figure A.1: Create a deployable build of the application.....	52
Figure C.1: Sign-in page.....	63
Figure C.2: Signup Page .....	64
Figure C.3: Campaign Creation Interface.....	65
Figure C.4: User Profile Page .....	66
Figure C.5: User's information panel .....	66
Figure C.6: Edit user information panel .....	67
Figure C.7: Global Chatbox.....	67
Figure C.8: Campaign Page .....	68
Figure C.9: FAQs section .....	69



Figure C.10: Campaign update .....	69
Figure C.11: Comment on campaigns .....	70
Figure C.12: Campaign approval page .....	71
Figure C.13: Campaign approval .....	71
Figure C.14: Approve/Reject reason box .....	72
Figure C.15: Share campaign on Social media .....	72
Figure C.16: Donation page .....	73
Figure C.17: Donation message .....	73
Figure C.18: Payment method selection .....	74
Figure C.19: Paystack payment gateway .....	74
Figure C.20: Users watchlist .....	75
Figure D.1: Administration dashboard .....	77
Figure D.2: Campaign Report .....	78
Figure D.3: Donation transaction Report .....	79
Figure D.4: User Report .....	80
Figure D.5: System Log report .....	80

## List of Tables

Table 2-1: Comparison of currently available systems .....	9
Table 3-1: Use case scenarios .....	19
Table 4-1: Implement Environment .....	27
Table 5-1: User Authentication Test Cases .....	39
Table 5-2: User Registration Test cases .....	39
Table 5-3: Create new campaign test cases .....	40
Table 5-4: Make a donation test case .....	41
Table 5-5: Share campaign on social media test cases.....	41
Table A-1: Hardware & Software requirements of on-premises deployment.....	51
Table A-2: Minimum requirement for cloud deployment .....	51
Table C-1: User roles of the system.....	62

## List of Acronyms

- SDLC: System Development Life Cycle
- CRUD: Create, Read, Update and Delete
- UUID: Universally Unique Identifier
- URL: Uniform Resource Locators. (Unique address to identify web pages)
- FAQ: Frequently Asked Question
- UI: User Interface
- SMTP: Simple Mail Transfer Protocol (Standard communication protocol for electronic mail transmission)
- API: Application Programming Interface (A way for two or more computer programs to communicate with each other)
- UML: Unified Modeling Language (A general-purpose, developmental modeling language)

# Chapter 1 - Introduction

Sri Lanka, the pearl of the Indian Ocean is a country well-known for its hospitality and kindness. As a country, we always like to share things with people and help those in need, whenever possible. When it comes to donations, people are very sympathetic and they would lend a hand, which means they can.

## 1.1. Motivation

Unfortunately, in Sri Lanka, almost every year a disaster happens. However, whatever the disaster could be, many people get together and organize donation campaigns, in their own capacities. They may be doing it using their social media account or through other groups, they have, such as school/ university batches or company-wise. Yet, at present, the pressing issue that concerns most of the empathetic Sri Lankans is that sometimes we may wonder whether our donations go to the people who need them. This occurs mainly due to a lack of transparency in some of the projects that were conducted on massive scales.

## 1.2. Background of the Study

Most of the time people take advantage of other people's sympathy. Sri Lanka police have arrested a man accused of siphoning off 300,000 rupees from a bank account that had collected donations for a child with a brain tumor, police said. The fund-raising program the man had stolen from is run by past pupils and teachers of the 14-year-old child's school in Horana, to support surgery, and donations are saved in a bank account belonging to her father. (Jayasinghe, C 2021).

Donors charge that Weerasinghe who is also a famed Instagram influencer has failed to provide any tangible proof of the dispersion of the amassed funds raised on behalf of Sri Lanka's Easter attack victims following the lapse of one and a half years. (WION Web Team, 2020)

The latest example is the "FOR SRI LANKA" fundraising campaign started by famous Sri Lankan Singer Yohani De Silva. Concerns were raised over this campaign after the US-based Sri Lanka Foundation denied any involvement with the project.

The problem of these above cases is the lack of transparency in the projects that were conducted on massive scales. There are a lot of incidents happening lately and these kinds of incidents will lose people the trust in donation campaigns.

In addition, at times, there could be issues with the unequal distributions of the donations that were made. For instance, sometimes when a disaster happens, in some areas people get many donations from people and in some areas people get nothing. Thus, a proper and equal distribution cannot be rather witnessed at times like that. Hence, we need a mechanism to manage this issue. As a method of addressing the aforementioned issue, I propose to streamline the donation campaign and monitor it systemically.

### 1.3. Objectives

Project objectives can be mentioned in the following categories.

- Provide donors to register and donate to selected donation campaigns. Once the user is registered they can donate anonymously or publicly.
- Allow volunteers can register and participate in the donation campaigns.
- Once the donation is complete users will be informed via email and they will be updated throughout the campaign regularly.
- Campaign creators can promote their donations through various social media to get public attention. Donors also can share these campaigns.
- Allow organizers to create timelines and add important events to the campaigns.
- Track users' donations and encourage them to donate by introducing a donor recognition program.
- Every campaign creation request will be manually validated. This process helps to eliminate fake donation campaigns circulating in public. To make this process faster system will provide functionalities like uploading necessary documents directly to the system, email approvals, etc.

### 1.4. Scope of the Study

Propose public donation management system will consist of five main modules.

- 1) Admin module (This module is used to monitor and manage administration functionalities)
  - Approve/Reject donation campaign request
  - Manage user registration
  - Disputes & complaints handling
  - Send reminders
- 2) Campaign module (This module contains all functionalities regarding campaign management & Campaign marketing)
  - Create Campaigns
  - Share campaigns through social media
  - Upload the progress of the campaigns
- 3) Donor management module (This module contains donor functionalities)
  - User Registration
  - Rate & comment on campaigns
  - Donate money for donation campaigns.
  - Donor recognition program.

Once a donor makes a donation he/she will get points accordingly. Donors can earn badges by earning points. These badges will be displayed in the user's public profile and they can share them through social media. This is called the donor recognition program which is similar to google's local guide program.

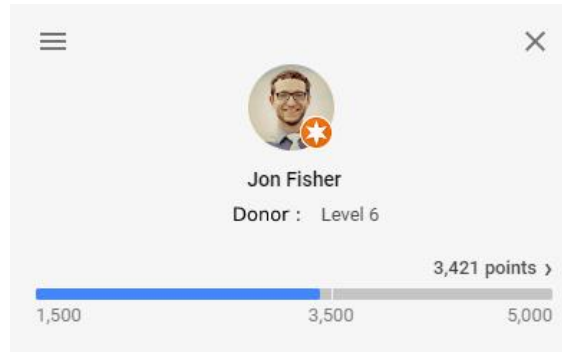
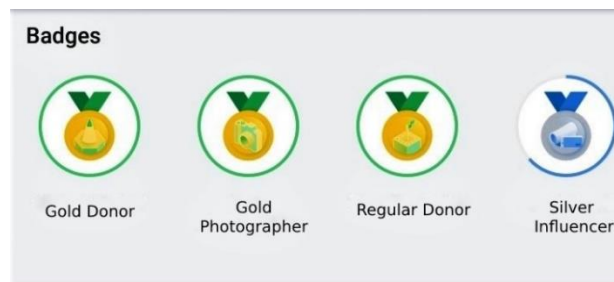


Figure 1.1: Point System of Google's Local Guide



**Number of Contributions**

Figure 1.2: Badges of Google's Local Guide Platform

- 4) Payment module (Since every campaign has one payment gateway it will be developed and integrate as a separate module.)
- 5) MIS Reporting for both campaign organizers and administration staff

For administration and staff System will provide a fully functional “Management dashboard” where they can monitor every donation campaign.

Campaign organizers have access to the “campaign manager dashboard”. It will provide reports for their running campaigns like

- Donor engagement reports.
- Campaign-related reports.
- Payment reports.
- Status reports etc.

All these reports can be exported in excel and PDF format and all these reports will be printer-friendly reports.

## 1.5. Dissertation Outline

This dissertation consists of six main chapters followed by Reference and Appendix.

- Chapter 1

This is the Introduction chapter which describes the main problem, motivation, objectives of the project, and scope of the project.

- Chapter 2

The background study of the project is described in chapter 2. An in-depth analysis of currently available similar systems and their technologies are described here. Further, it includes the requirement analysis of the project.

- Chapter 3

Chapter 3 is the Design chapter. It includes the design strategies and design architecture of the proposed system.

- Chapter 4

Chapter 4 is the implementation chapter. It describes the implementation environment, and development tools used to develop the system. It also includes the major code segments of the project.

- Chapter 5

This is the Testing and evaluation chapter. This chapter discusses whether the project objectives are satisfied. The testing methods, test cases, and their results are included in this chapter.

- Chapter 6

Chapter 6 is the conclusion chapter. It identifies any deficiencies in the final system and future improvements of the project.

# Chapter 2 - Background of the Study

## 2.1. Introduction

This chapter describes the background study of the system. It includes analysis of similar systems, requirement analysis, and related technologies.

## 2.2. Requirement Analysis

### Functional Requirements

The system consists of 5 modules. Each module is responsible for its respective tasks.

#### 1. Admin Module (AM)

- Campaign Organizer Verification – Admin staff can verify campaign organizers.
- Campaign Verification – Admin staff can verify campaign authenticity.
- Block/Flag Charity Scams – Admin staff can block or flag Charity Scams.
- Create Staff members – Super admin can create staff members.
- Change admin password – Admin staff can change their login password.

#### 2. User Management Module (UMM)

- Users Registration – Users can register to the system as a donor or campaign Organizers.
- Change user password – Users can change their passwords.
- Update user profile – Users can change their user profiles.
- Add campaigns to the watch list – Users can add donation campaigns to their future reference.
- Share campaigns via social media – Users can share donation campaigns on social media.
- Share user's profile via social media – Users can share their user profile on social media.
- Award donation points and badges to donors – Users can earn donation points through various activities.

#### 3. Campaign Management Module (CMM)

- Campaign creation – Campaign organizers can create donation campaigns.
- Donate to donation campaigns.
- Approve/reject campaign creation requests.

#### 4. Payment Module (PM)

- Facilitate online payments – Users can pay for donation campaigns through an online payment gateway.

#### 5. Report Module (RM)

- Generate user-related reports.
- Report campaign-related reports.
- Generate payment-related reports.



## Non Functional Requirements

### **Performance**

- The application will be developed using front end framework so that hosted server doesn't get much traffic. Also, response time should be minimal.
- Since the system is built on Vue.JS memory allocation (Borrelli P, 2021) and runtime performance (Comparison with Other Frameworks — Vue.js, n.d.) is better than other available technologies.

### **Integration**

- The system will be developed in modules so that each module can be integrated together.

### **Security**

- Since this website handles financial transactions security is a high-priority requirement. All user passwords are encrypted and stored securely. The system will provide social media login (Open ID authentication) to users as well.
- When the user is making a donation system will send the information to the bank website and redirect the user to the bank's payment gateway. The system won't store the payment method details (Credit card details) of the user.

### **Maintainability**

- To ensure maintainability each module of the system is developed in a way to achieve high cohesion / loose coupling. Ex-: All user account operations are handled under the user controller in the API.

### **User-friendliness**

- The entire system must be easy to use and user-friendly to all relevant users. This can be achieved by having responsive interfaces, appropriate color tones, easy navigations, search options, etc

## 2.3.Review of Similar Systems

In Sri Lanka, there are a lot of systems which had been specifically designed to carry out blood donations. (NBTS Online, 2021) Yet, the problem with these already existing systems is that their scope is too small or they only focus on one particular area. Thus, through the proposed system, such issues will be addressed and the system will be built to overcome these issues.

Currently, there are online websites such as tickets.lk, event.lk which are available for registrations for various charity events/ concerts. Hence, users are given the chance on these sites to purchase tickets through the websites and the collected funds will be transferred to the charity projects.



Figure 2.1: Tickets.lk Logo



Figure 2.2: Events.lk Logo

In addition, one of the leading network providers in Sri Lanka: Dialog has launched a campaign where you can donate your star points to donation projects like “Little hearts fund” (Kind Hearted Lankans 2020), and SOS Children’s Villages Help Age through their mobile phones. Yet, they have some conditions like people have to maintain a minimum star points balance to make a donation, so people don’t have much flexibility there.

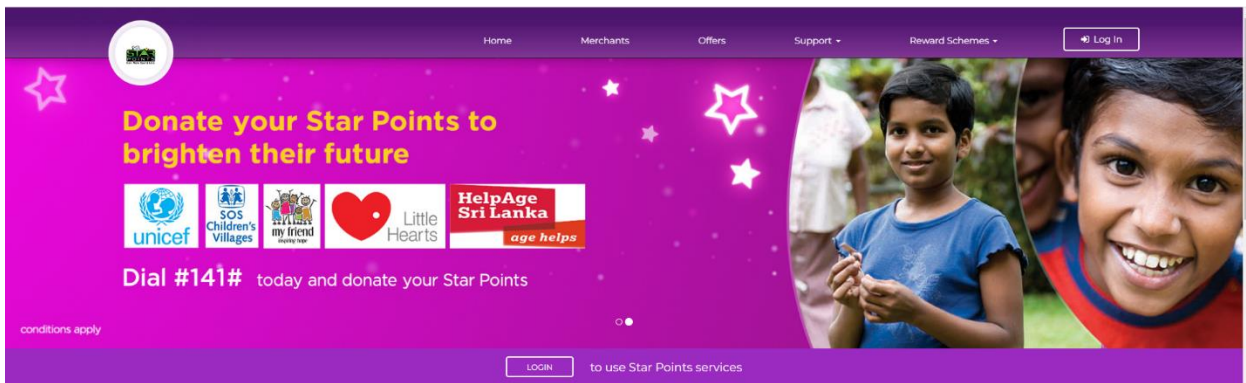


Figure 2.3: Dialog Star points donation for little hearts fund

If we consider outside Sri Lanka this concept is not new to European countries. There are lot of systems available to manage donations. Most of the non-profitable charity organizations and NGOs have their own systems where people can donate money to charities.

Global Giving is a nonprofit that supports other nonprofits by connecting them to donors and companies. Since 2002, they have helped trusted, community-led organizations from Afghanistan to Zimbabwe (and hundreds of places in between) access the tools, training, and support they need. (GlobalGiving.org, 2021)

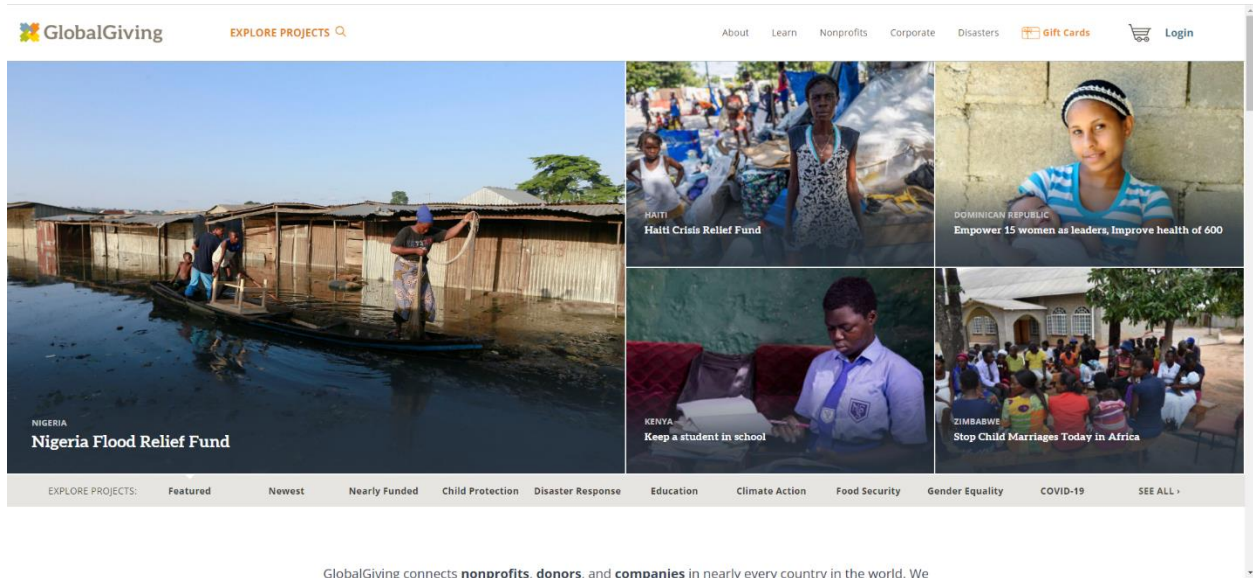


Figure 2.4: Global Giving Website Home page

Mightycause is another, a popular crowdfunding platform for individuals, teams, or organizations, is a great website for indexing your nonprofit organization to get exposure for your cause. Using mightycause’s nonprofit portal, any registered nonprofit can set up a professionally branded, completely custom charity fundraising page to collect donations at any time. (Mightycause: Nonprofit fundraising Made Easy, 2021)

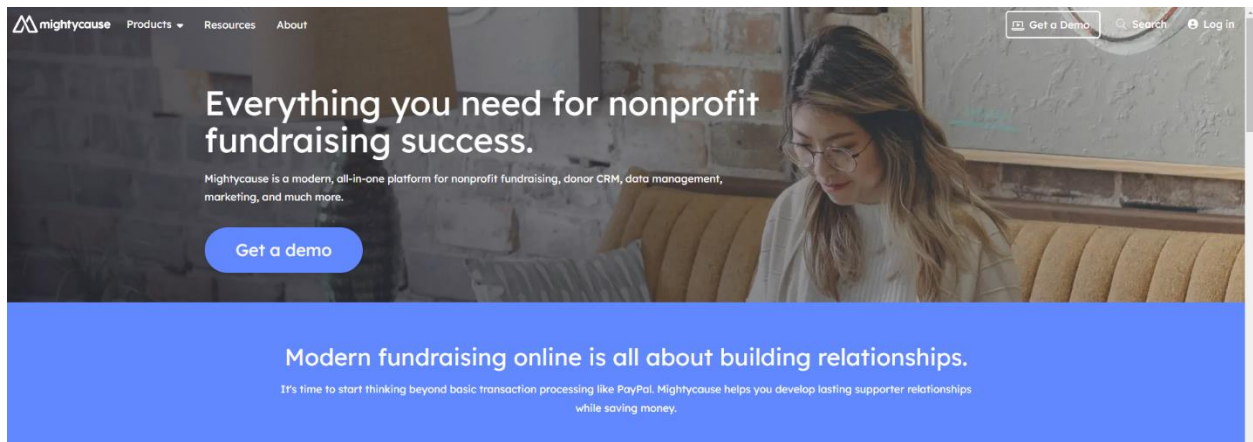


Figure 2.5: Mightycause Website Home page

Features	Tickets.lk & Events.lk	Global Giving	Mightycause	Dialog Donation	Proposed System
Donors can donate money for various charity projects	Anyone can donate money by registering.	Anyone can donate money by registering.	Anyone can donate money by registering.	Only Dialog customers can donate money to donation campaigns	Anyone can donate money by registering.
Any individual can create donation campaigns	Organizations can register and create campaigns.	Organizations can register and create campaigns.	After purchasing, individuals can create donation campaigns	Individuals can't create donation campaigns	Anyone can create donation campaigns.
Share donation campaigns through social media.	Yes	Yes	Yes	Yes	Yes
Explore projects?	Yes	Yes	No	Only a few projects are available to explore.	Yes
Campaign creation Fees	Free		\$59 month	Free	Free
Use your own branding.	No	No	Yes	No	Common template available for branding
Payment gateway integrated?	Yes	Yes	Yes	No. Star points or mobile credit used to make a donation	Yes.
Donor recognition program	No	Available	No	No	Available

Table 2-1: Comparison of currently available systems

## 2.4.Related Technologies

### Technologies used in Similar Systems

Most of the reviewed systems are web-based systems. Dialog donation is the only platform that has a strong mobile platform. They do have a website but the most popular platform is the mobile platform. Most websites are built using popular web technologies like PHP, HTML, Javascript, CSS, etc. Websites like event.lk are developed using word press.

Apart from dialog donation website and tickets.lk web sites all other websites have SSL installed for secure payment.

### Technologies used for the proposed System

Propose system was built using Vue.js, a powerful progressive front-end technology. Vue is a javascript framework. In 2019 Stack Overflow Developer Survey Vue.js become the most loved and most wanted web framework by developers (Stack overflow developer survey 2019).



Figure 2.6: Reasons to choose Vue.js (Patel, J 2021)

The main reason to choose Vue.js as the front-end framework is runtime performance (Comparison with Other Frameworks — Vue.js, n.d.). It was easy to learn the language because of the proper documentation and great community support.

For the backend API of the proposed system, Node.js was used. Node.js is open source cross-platform backend Javascript runtime server environment. According to Stack Overflow Developer Survey 2021, Node.js is the 6<sup>th</sup> most popular technology chosen by developers. (Stack overflow developer survey 2021)

Node.js is composed of Google's V8 JavaScript engine which allows its applications to perform at high speed, handling multiple requests with ease. (Tomislav C, 2013) Sometimes the size of the application is essential when choosing the correct technology. If the application is large it directly affects its performance of the application. Node.js is lightweight and has a lot of open-source community-developed libraries available. It also has good documentation and great community support. These are the main reasons to choose Node.js to develop the proposed system.

Google firebase is used as the backend database of the proposed system. Firebase offers the Firebase Real-time Database, a NoSQL database hosted in the cloud. It enables developers to store and sync JSON data in real-time. It is also equipped with cloud storage, which allows developers to store images and videos. All the images of the proposed system are uploaded to firebase cloud storage. Using firebase cloud functions these uploading can be automated.

Ex -: Automatically resize images when uploading.

Logging is also an important aspect in any software product. For software maintenance phase it is necessary to have complete logs for analysis. For the Charity application Winston logs were used to log system activities. winston is designed to be a simple and universal logging library with support for multiple transports. (Winston).

```
sysLog.log
{"level":"info","message":"User Authenticated Sucessfully","request":"/api/user/authenticate","timestamp":"2022-11-19T08:04:28.868Z"}
{"level":"info","message":"User Authenticated Sucessfully","request":"/api/user/authenticate","timestamp":"2022-11-19T08:05:04.526Z"}
{"level":"error","message":"User lqblyRwIey1JjL6V8Chj Found Sucessfully","request":"/api/user/getUser","timestamp":"2022-11-19T08:12:32.570Z"}
|
```

Figure 2.7: Sample log created using Winston

# Chapter 3 - Design

## 3.1. Introduction

This chapter describes the design of the system. It includes a related design strategy, system architecture, and UML diagrams of the system.

## 3.2. Related Design Strategies

- System Design

For system design according to three-tier architecture. More about the system design is discussed under system architecture.

- UI Design

User Interface design is very important because it plays a major role in customer acceptance of the system. For UI design paper prototyping technique is heavily used here. It's simple and easy to change compared to other methods. It is also the cheapest technique among others. For each user interface, two designs are drawn and the best one is selected for implementation.

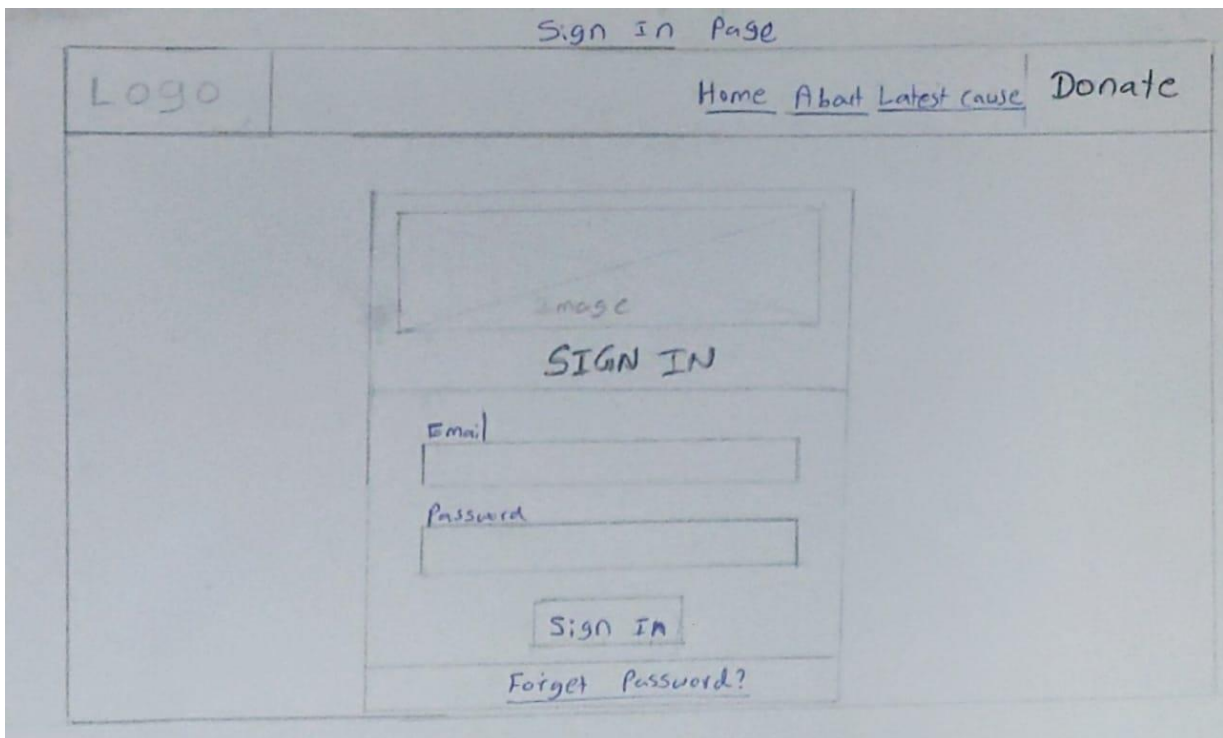


Figure 3.1: Paper prototype for sign in screen

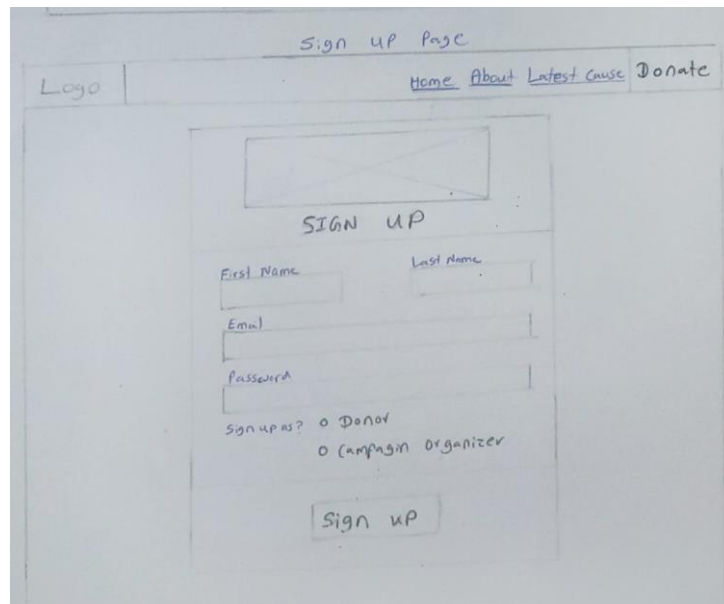


Figure 3.2: Paper prototype for Sign Up screen

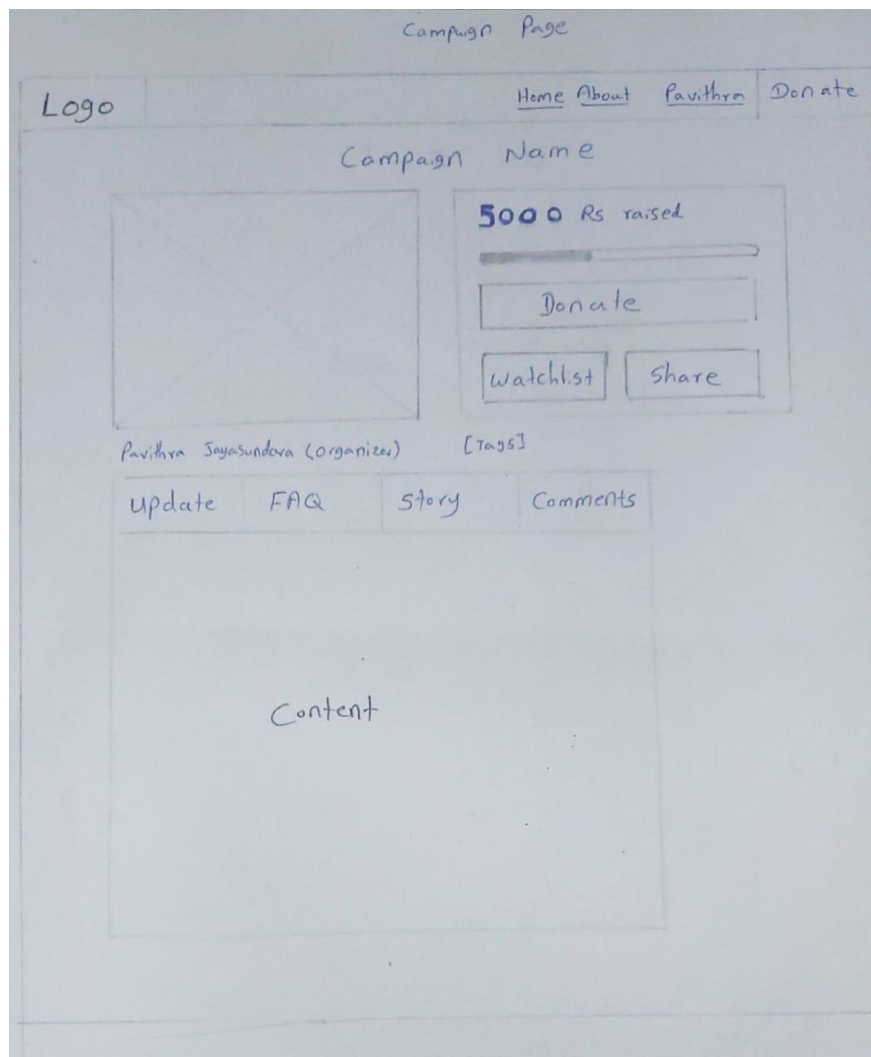


Figure 3.3: Paper prototype for Campaign page

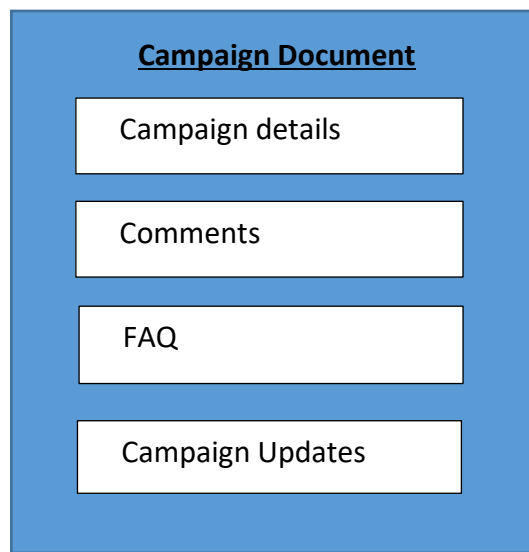


- Database Design

The system database is built according to the fundamental properties of NoSQL databases. To achieve faster access to the data everything is stored in a separate collection according to the access frequency.

Ex -: Most of the time users are frequently searching for donation campaigns. They are not searching or updating user details or viewing other users' details. So everything related to the campaign is stored in separate collections. But user details are saved under one collection.

When quarrying the data in Google firebase it retrieves the full document. For example, assume the user wants to read the comments of a specific donation campaign. If we store comments inside the Campaign collection like below, when retrieving the data all the details including FAQ, and campaign updates campaign details will be sent to the user.



*Figure 3.4: Single Campaign Document*

But if we store everything in separate collections like below only the comments document will be sent to the user. Using this approach data duplication can happen but the performance of the database is heavily increased.

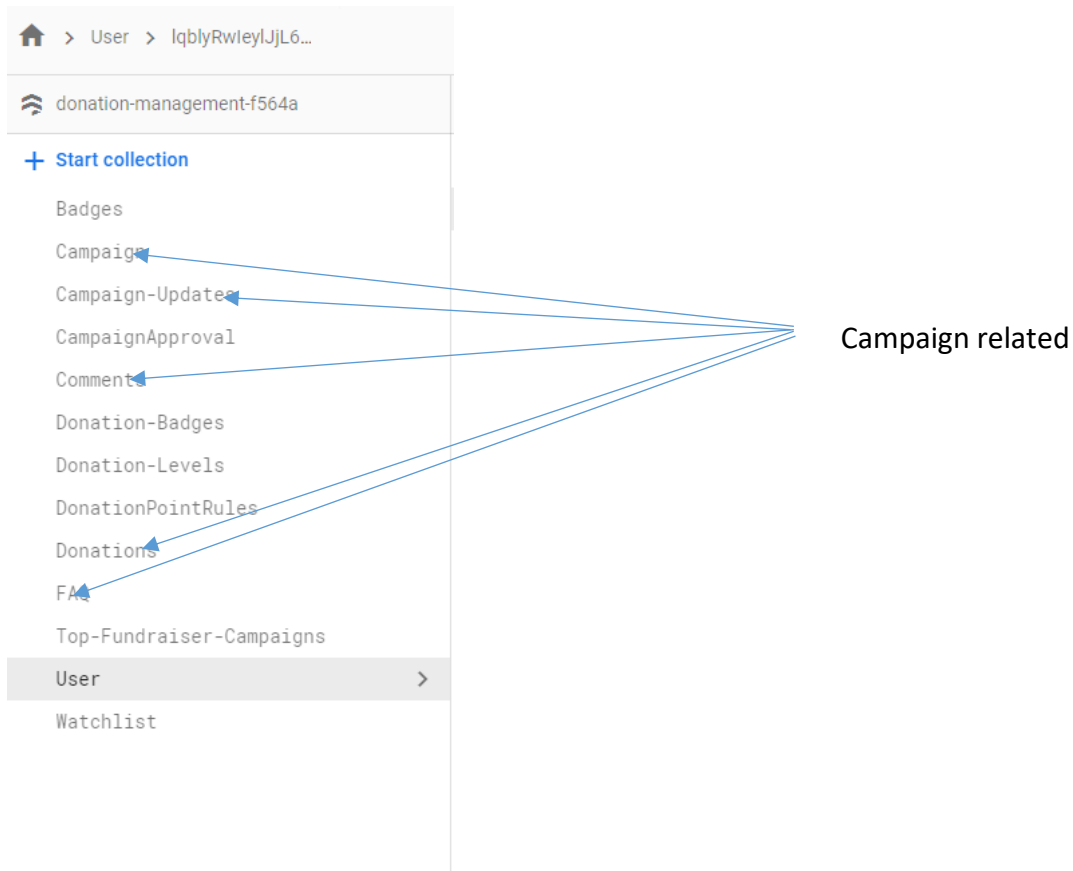


Figure 3.5: Sample Document Structure

### 3.3. System Architecture

The donation management system is built according to the pillars of the 3 tier architecture. Users can interact with the front-end system via any device and it is called a separate API server connect to the database. Google firebase is used as the main database. Google cloud storage is used to store images. Front end server can upload images directly to googol cloud storage but cannot request images from it. To request images, front end server has to invoke the API function and the API will deliver the requested resources.

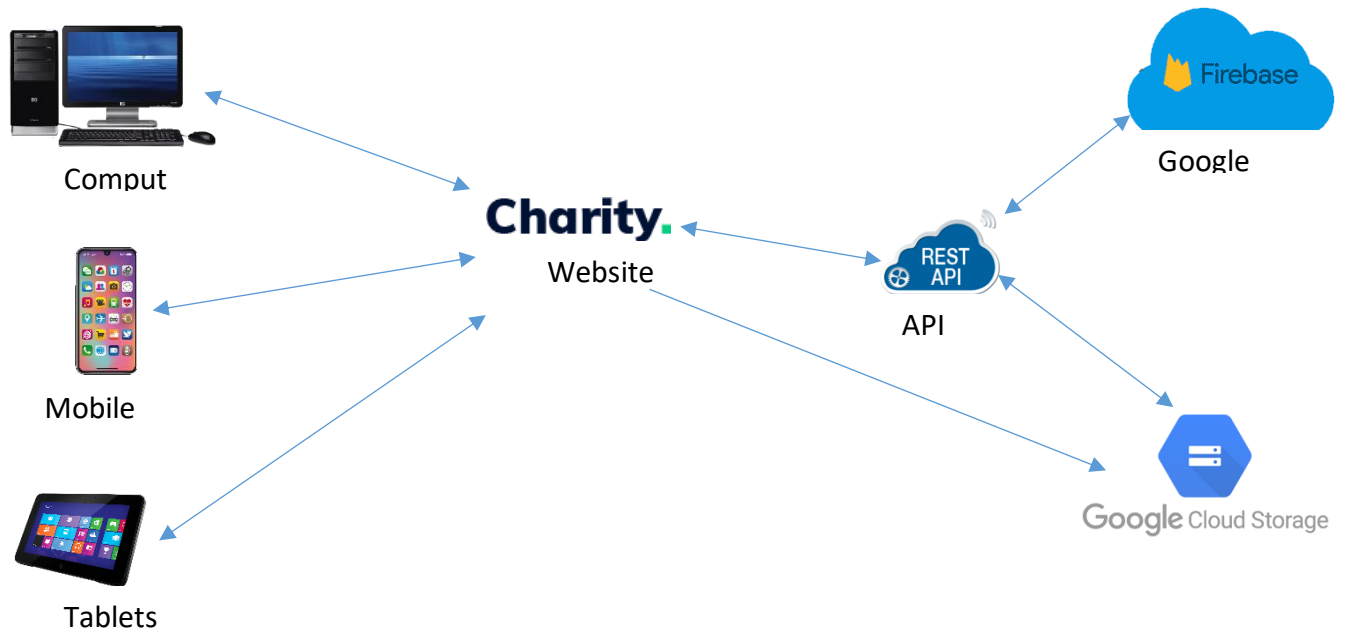


Figure 3.6: High-Level Architecture diagram

### 3.4.UML Diagrams

#### 3.4.1. System Use case Diagram

System Actors -: Donor

Campaign Organizer

Administration Staff

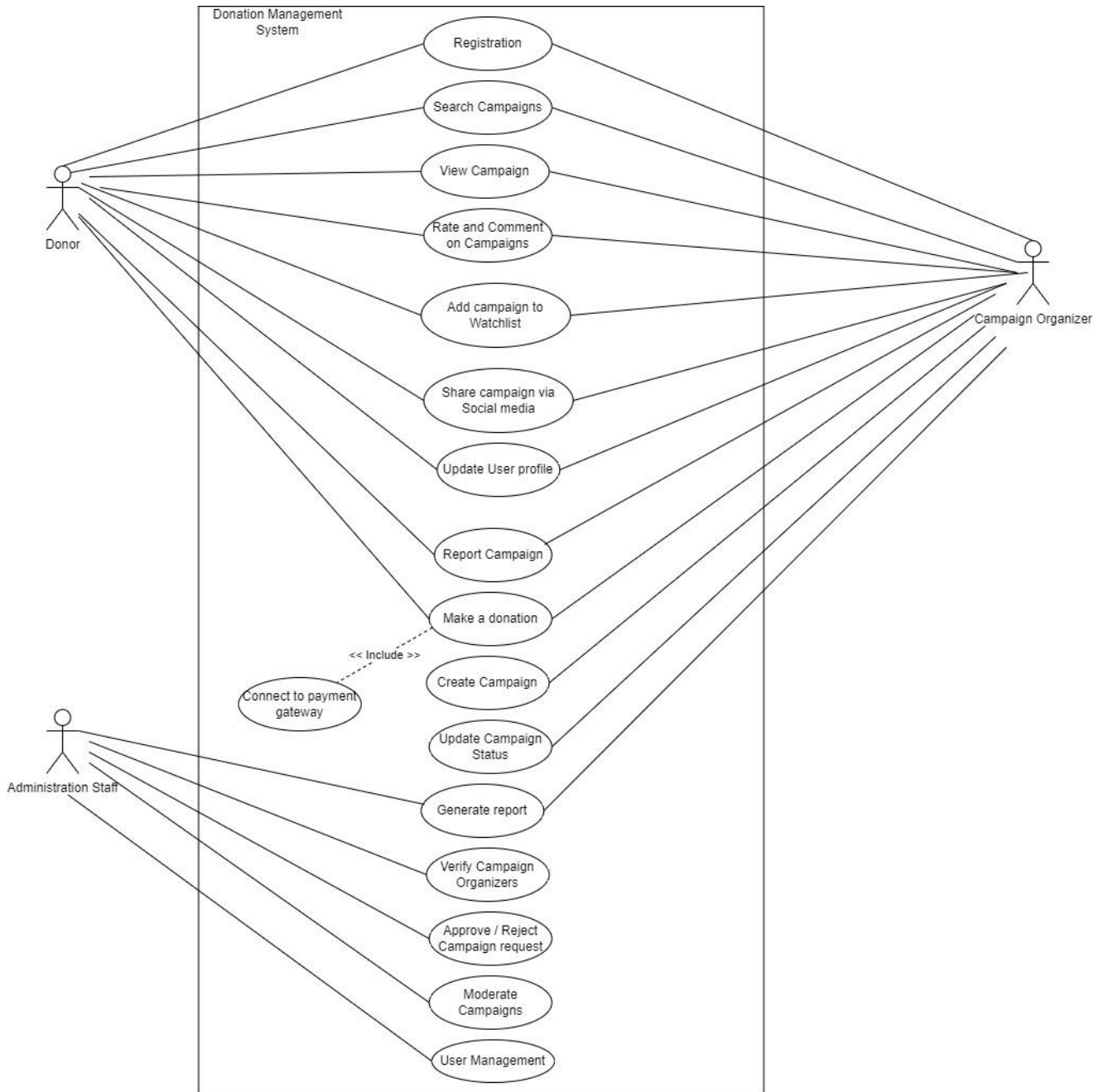


Figure 3.7: System Use Case Diagram

### 3.4.2. Use case Scenarios

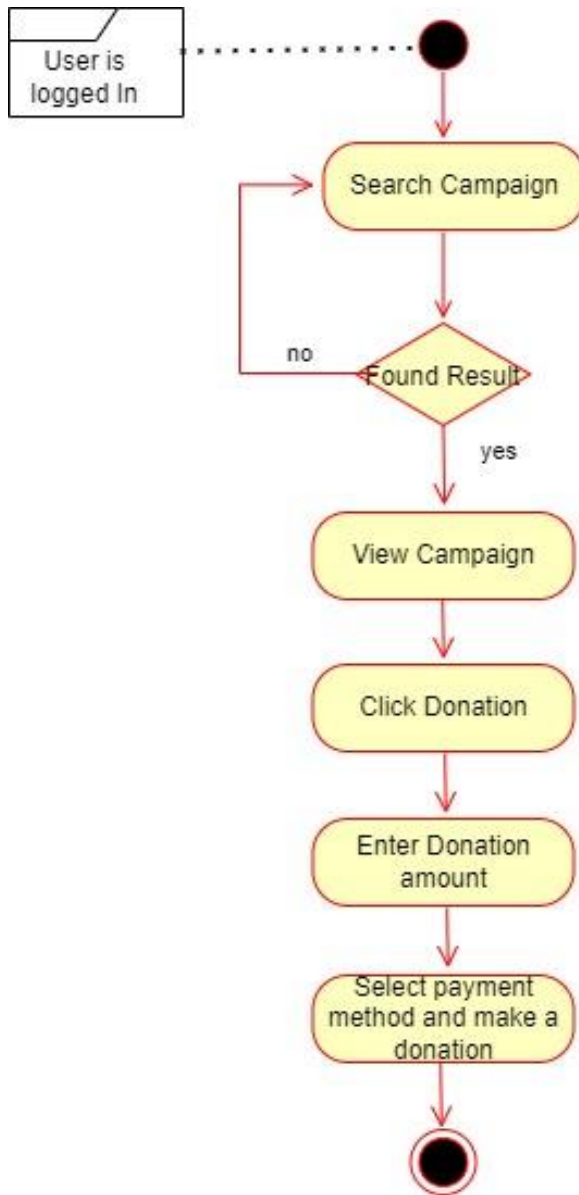
Class of Use Case	Use Case	Description of the use case
Use case related to login to the system.	<ol style="list-style-type: none"> <li>1) Login as an Admin.</li> <li>2) Log in as a donor.</li> <li>3) Log in as a campaign creator.</li> <li>4) Change Password for Users.</li> </ol>	<ol style="list-style-type: none"> <li>1) Admins can log in to the system through the admin portal.</li> <li>2) Login donors to the system.</li> <li>3) Login campaign creators to the system.</li> <li>4) Change the password for donors/campaign creators.</li> </ol>
User case related to registration of the users.	<ol style="list-style-type: none"> <li>1) Register donor.</li> <li>2) Register campaign organizer.</li> </ol>	<ol style="list-style-type: none"> <li>1) Create a user account and store the user's personal details, contact details, etc.</li> <li>2) Create a campaign organizer account and store his/her personal details, contact details, etc.</li> </ol>
Use case related to user profile management.	<ol style="list-style-type: none"> <li>1) Change personal details by donor or campaign organizer.</li> </ol>	<ol style="list-style-type: none"> <li>1) Updating existing user account details by the user.</li> </ol>
Use case related to creating a donation campaign.	<ol style="list-style-type: none"> <li>1) Create a donation campaign by the campaign organizer.</li> <li>2) Share the campaign through social media.</li> </ol>	<ol style="list-style-type: none"> <li>1) Create a donation campaign and send it for verification.</li> </ol>
Use case related to campaign verification.	<ol style="list-style-type: none"> <li>1) Approve campaign creation request by admin.</li> </ol>	<ol style="list-style-type: none"> <li>1) Admin staff go through the campaign details validate them with necessary parties for authenticity and verify them.</li> </ol>
Use case related to donation campaign.	<ol style="list-style-type: none"> <li>1) Donor search donation campaigns.</li> <li>2) Add the campaign to the watch list.</li> <li>3) View the user's watch list.</li> <li>4) Remove items from the watch list.</li> </ol>	<ol style="list-style-type: none"> <li>1) Search for various donation campaigns using global search and filters.</li> <li>2) Users can add donation campaigns to their watch list for future reference.</li> </ol>

	5) Share the campaign through social media.	3) Users can view current items in the watch list. 4) Users can remove the item from the current watch list. 5) Users can share donation campaigns through social media.
Use a case related to donating. (Payment)	1) The donor will select the donation campaign and donate. 2) Complete the donation through the payment gateway.	1) Make a payment for the selected donation campaign. 2) For credit card payments donors will redirect to the payment gateway.
Use case related generate reports.	1) Generate the following report for users <ul style="list-style-type: none"> <li>• Successful donations made so far.</li> </ul> 2) Generate reports for campaign organizers. <ul style="list-style-type: none"> <li>• Created/running/finished campaigns details report.</li> <li>• Collected payments.</li> </ul> 3) Generate reports for administration staff. <ul style="list-style-type: none"> <li>• Created/running/finished campaign detail report.</li> <li>• Payment report.</li> <li>• Registered Users and campaign manager-wise report.</li> </ul>	1) Users of the system can select and generate necessary reports according to their authorization.

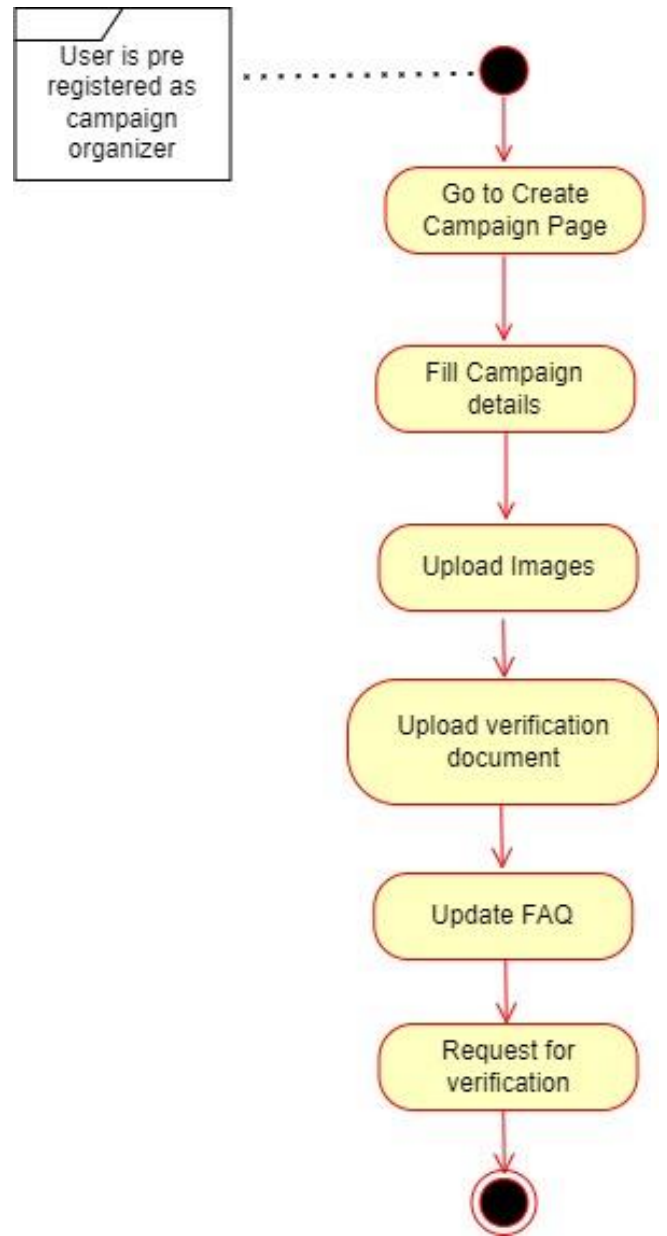
Table 3-1: Use case scenarios

### 3.4.3. Activity Diagrams

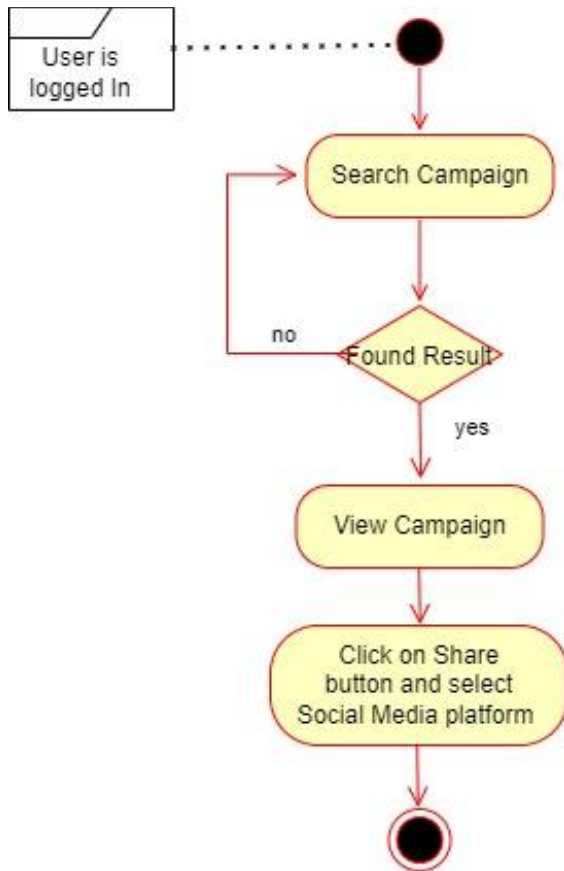
#### 1. Donate to a Campaign



#### 2. Create Campaign



### 3. Share Campaign

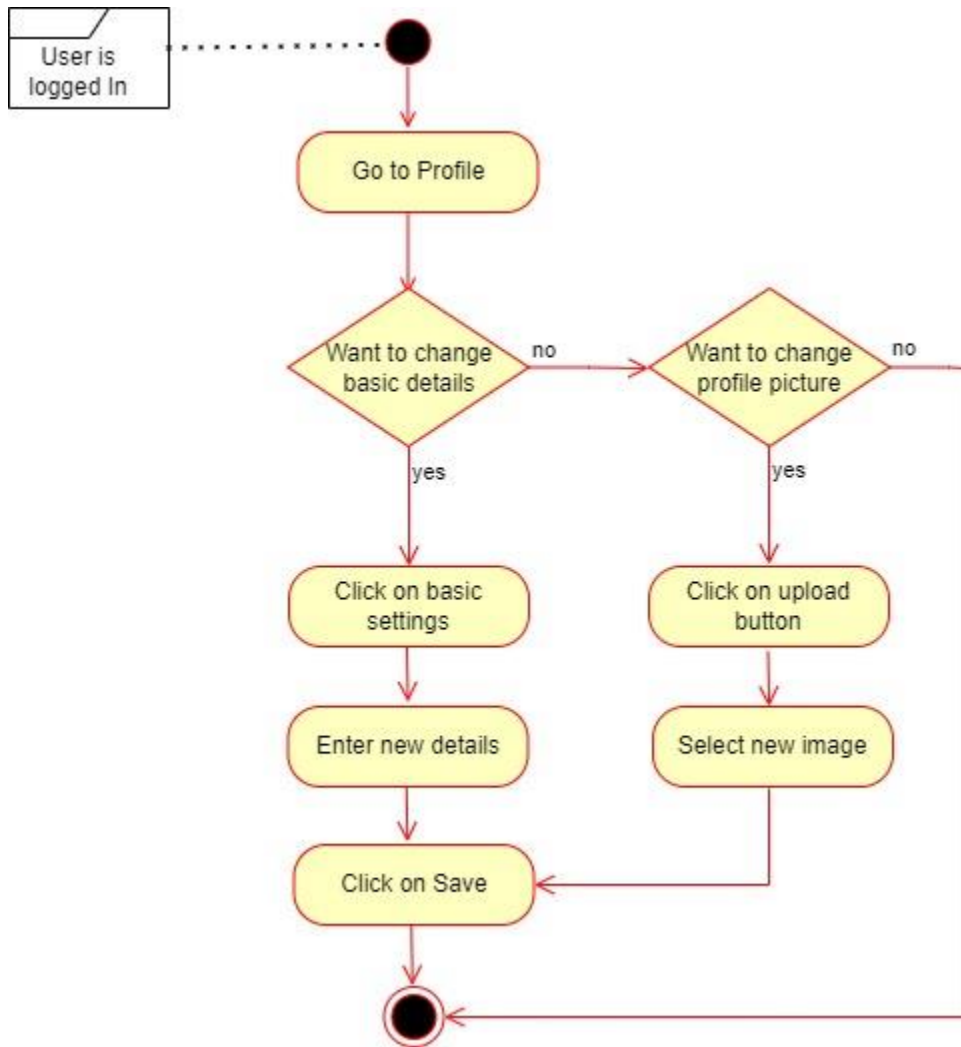


### 4) Add a campaign to the watchlist

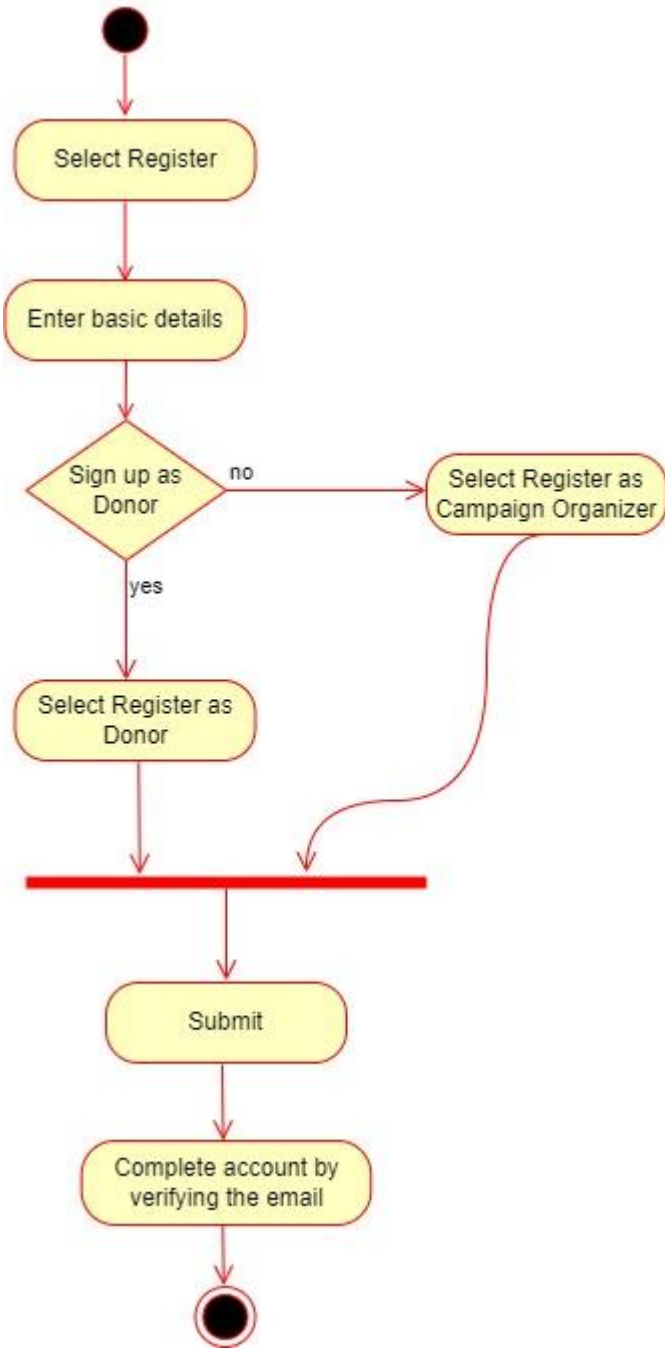




### 5) Update user profile details

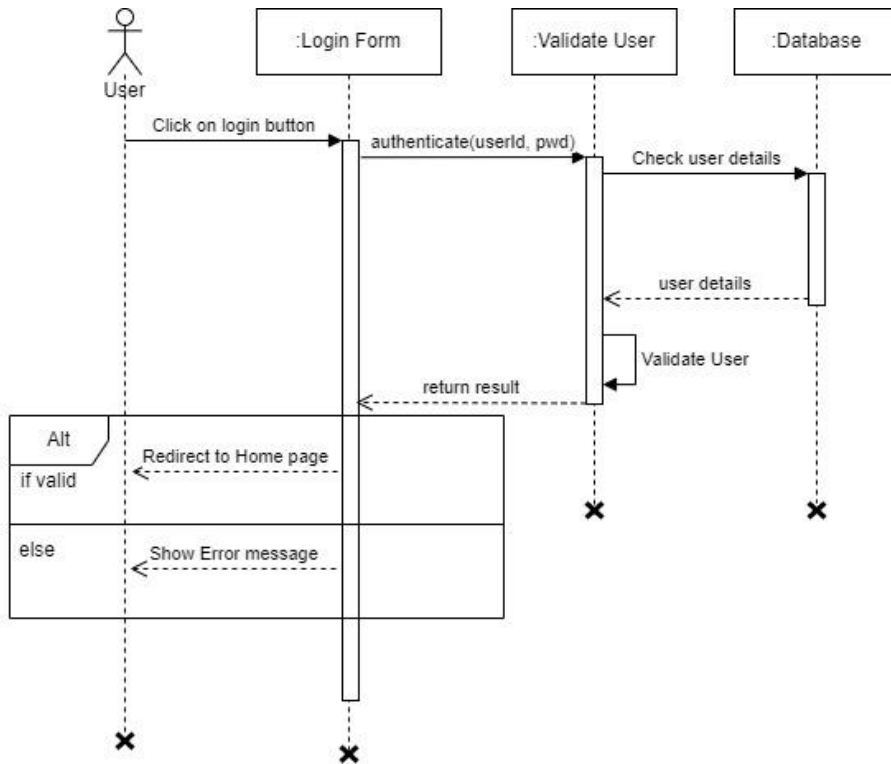


6) Register user

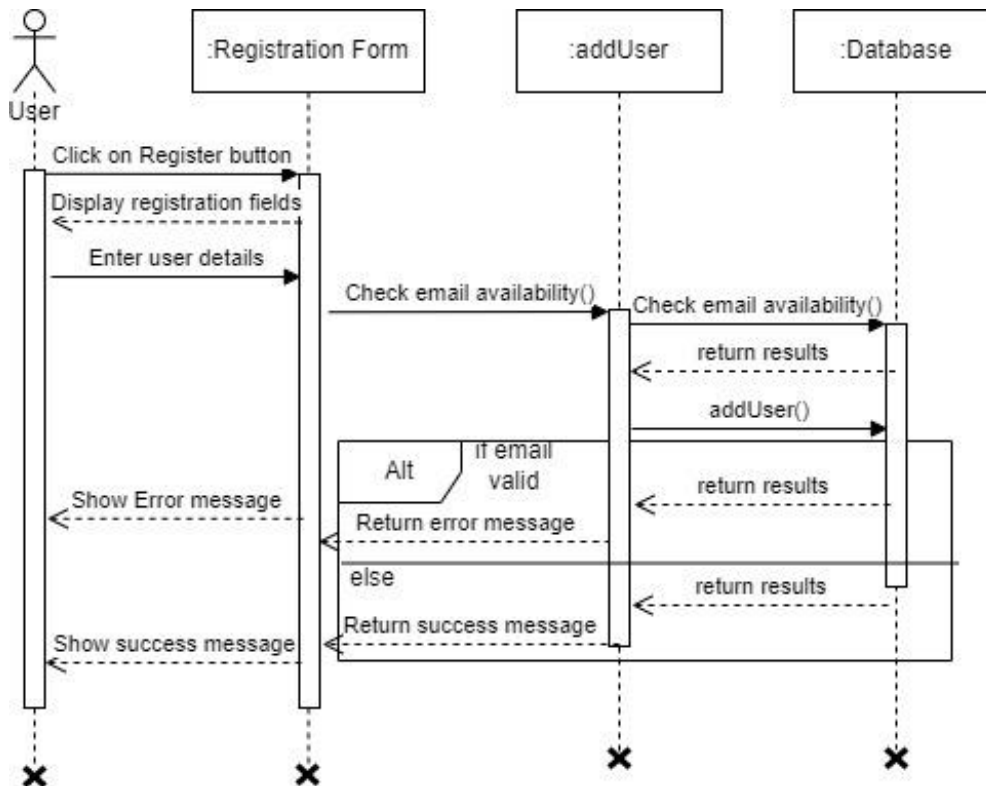


### 3.4.4. Sequence Diagram

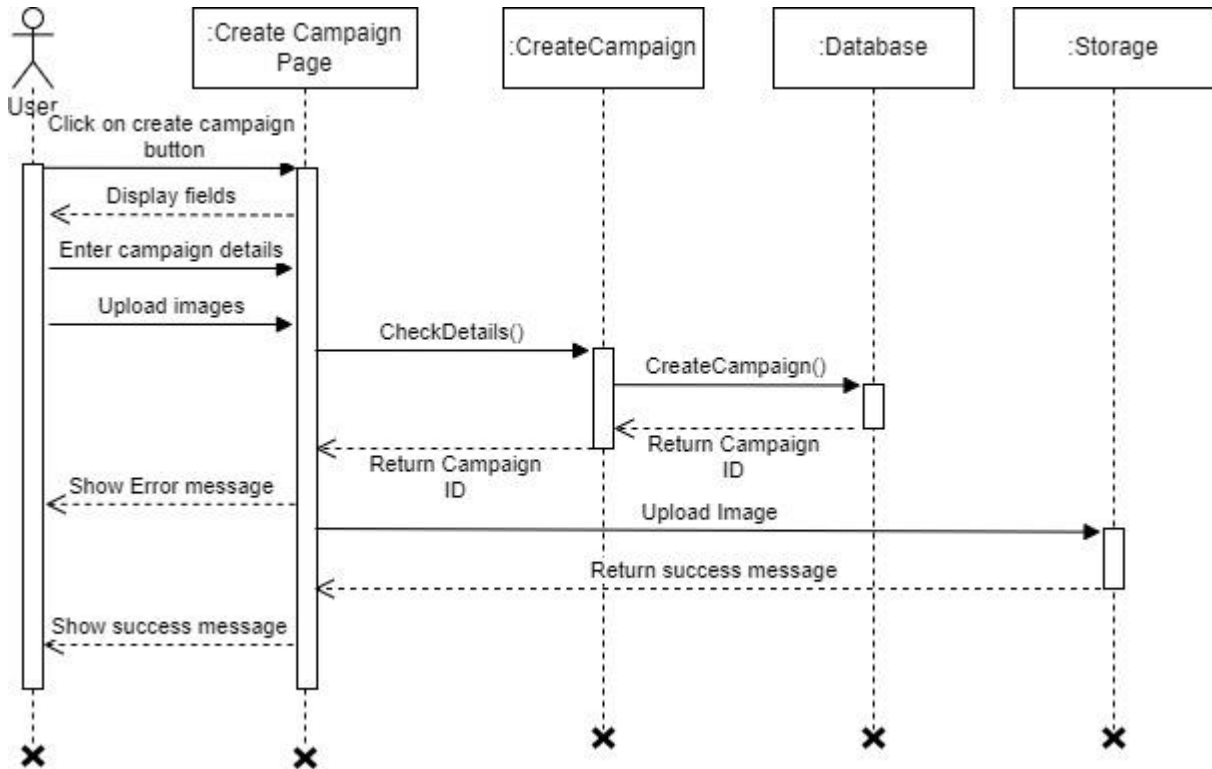
#### 1) Login



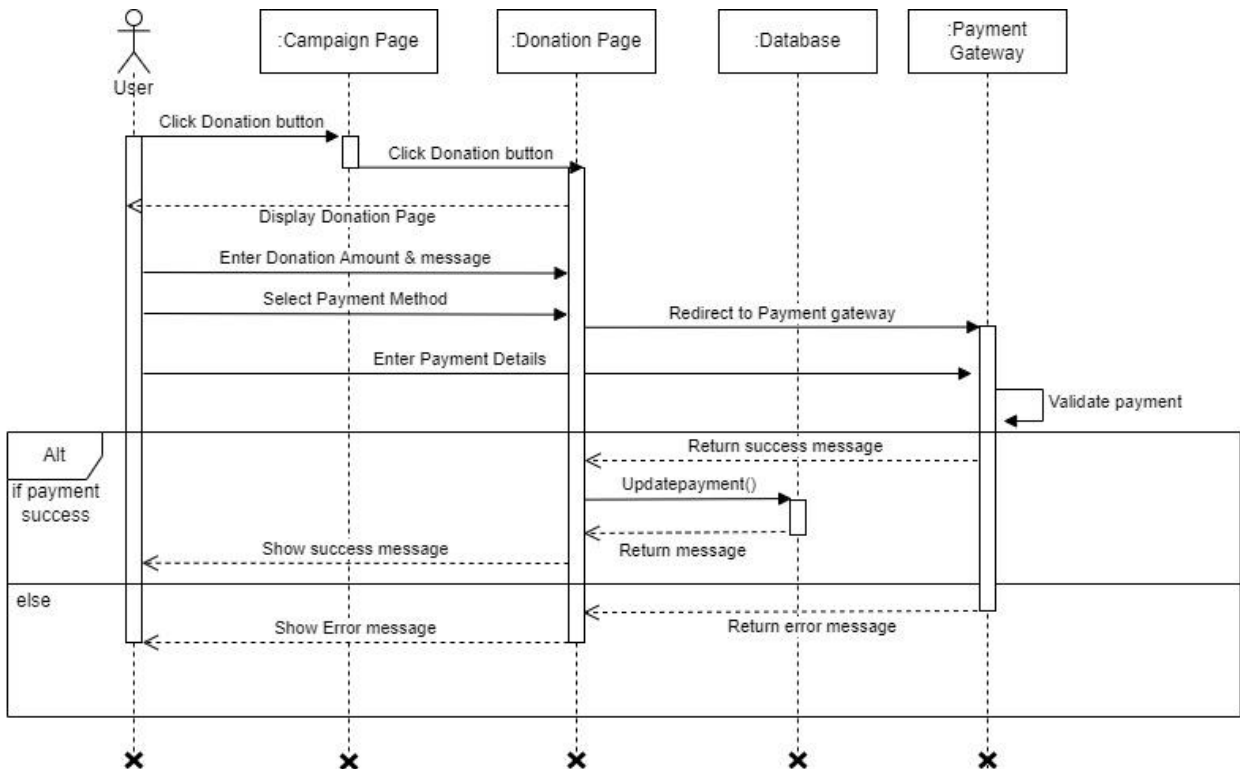
#### 2) Registration



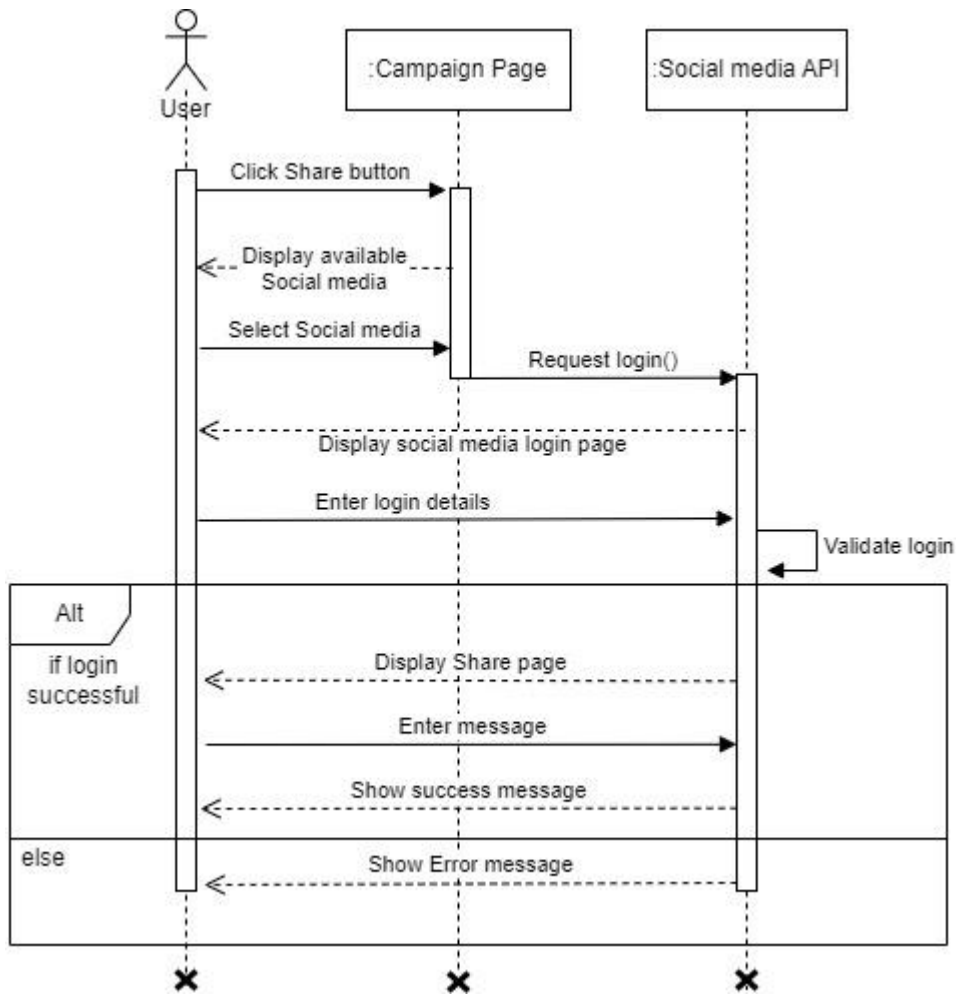
### 3) Create Campaign



### 4) Make a donation



5) Share the campaign on social media



# Chapter 4 - Implementation

## 4.1. Introduction

After completing the design phase implementation phase is carried out according to the output of the previous phase using the selected design tools and techniques. This is a very important phase in SDLC because the output of this phase is the system itself. As per the design architecture, the system is divided into modules and developed individually. After that one by one, each module is integrated accordingly. In this chapter, major code segments of the system, tools, and techniques used for development and the implementation environment are discussed.

## 4.2. Implementation Environment

The following table 4-1 describe the minimum hardware and software requirement which need for setup the implementation environment.

Hardware Requirement	Software Requirement
Intel(R) Pentium(TM) 4 @ CPU 3.0GHz or higher	Windows 10 or Ubuntu Desktop 16.04
4GB RAM	NodeJs LTS Version 18.12.1
160 GB HDD	Visual Studio Code 1.7

Table 4-1: Implement Environment

## 4.3. System Development tools and techniques

The following tools and techniques are used while developing the system.

- Draw.io -: Open source cross-platform graph drawing tool. This is been used to draw the diagrams of the system. (Diagrams.net)
- Photopea -: Open source online photo editing tool (Advanced photo editor). This is used to create/resize images for the system.
- Visual Studio Code -: This system is used to code the system. It's a free built-on open-source platform with many useful features.
- Vue.js -: Progressive javascript framework which is used to develop the front-end application
- NodeJs -: Powerful javascript runtime environment. This is used to develop the server side of the system.
- CSS -: This is used for styling the system.
- JSON -: JSON is used to pass data between the front-end application and the back-end server.

## 4.4. Reusable Components

Vuetify -: Bootstrap equivalent (A material design framework for vue.js)

Google Firebase -: Realtime cloud-hosted NoSQL database (Google)

Font Awesome icons -: Scalable vector icons (Gandy, Font awesome)

Vue2-editor -: Reusable rich text editor (Vue2-editor)

Vue-social-sharing -: Reusable component for social media sharing (Vue-Social-sharing)

Email.js -: Email sending package for any SMTP server (Emailjs)

Vue-json-excel -: Used to export JSON data to excel (Vue-Json-Excel)

Vue-charts -: Reusable chart component (Vue-chartjs)

Vue-paystack -: Payment gateway component (Vue-Paystack)

Uuid -: Reusable component for creating UUID (UUID)

Winston Log -: Simple and universal logging library (Winston)

## 4.5. Major Code Segments

The entire system is coded as modules and reusable components. The main important modules are been mentioned below with a brief description of their functionality.

### Database connection

A database connection is a must for most of today's applications. To perform any CRUD operations

```
config.js > ...
1 | // Import firebase package
2 | const admin = require('firebase-admin');
3 |
4 | // Import firebase Credentials
5 | const serviceAccount = require('./donation-management-f564a-firebase-adminsdk-hg53s-47bb8926d5.json');
6 |
7 | // Initialize Firebase app
8 | admin.initializeApp({
9 |   credential: admin.credential.cert(serviceAccount)
10 | });
11 |
12 | // Get a reference to the database
13 | const db = admin.firestore();
14 |
15 | // Export database reference
16 | module.exports = db;
```

## Front end configuration

```
src > main.js > ...
1 | // Import required packages
2 | import Vue from 'vue'
3 | import App from './App.vue'
4 | import router from './router'
5 | import vuetify from './plugins/vuetify'
6 |
7 | import { library } from '@fortawesome/fontawesome-svg-core'
8 | import { FontAwesomeIcon } from '@fortawesome/vue-fontawesome'
9 | import { fas } from '@fortawesome/free-solid-svg-icons'
10 | import { fab } from '@fortawesome/free-brands-svg-icons'
11 | import VueSocialSharing from 'vue-social-sharing'
12 | import VueSession from "vue-session"
13 | import JsonExcel from "vue-json-excel";
14 | import EmojiPicker from '@zaichaopan/emoji-picker';
15 |
16 | // Register component globally
17 | Vue.component("downloadExcel", JsonExcel);
18 | Vue.component('font-awesome-icon', FontAwesomeIcon)
19 | library.add(fas)
20 | library.add(fab)
21 |
22 | Vue.use(VueSocialSharing);
23 | Vue.use(VueSession)
24 | Vue.use(EmojiPicker);
25 |
26 | Vue.config.productionTip = false
27 |
28 | new Vue({
29 |   router,
30 |   vuetify,
31 |   render: h => h(App)
32 | }).$mount('#app')
33 |
```

## SignIn component

Below the code segment is the front-end application code for login functionality. It validates the user input such as the user has entered the correct email format and both the email and password fields are not empty. After the basic user input is validated it'll invoke a server function to authenticate the user.

```
<script>
// Import forget password component
import ForgetPassword from "@components/layouts/ForgetPassword.vue";
export default {
  name: "SignInComp",
  components: {
    ForgetPassword
  },
  data() {
    return {
      showPassword: false,
      valid: true,
      dialog: false,
      email: "",
      // Field validation
      emailRules: [
        (v) => !!v || "E-mail is required",

```



```

    (v) => /.+@.+\.+/.test(v) || "Please enter valid E-mail address",
  ],
  password: "",
  passwordRules: (v) => !!v || "Password is required",
}
},
methods: {
  validateUser() {
    var myHeaders = new Headers();
    myHeaders.append("Content-Type", "application/json");

    var raw = JSON.stringify({
      email: this.email,
      password: this.password,
    });

    var requestOptions = {
      method: "POST",
      mode: "cors",
      headers: myHeaders,
      body: raw,
      redirect: "follow",
    };

    // Fetch request
    fetch("http://localhost:3000/api/user/authenticate", requestOptions)
      .then(async (response) => {
        const resdata = await response.json();

        // check for error response
        if (!response.ok) {
          // Pass error message to parent component
          this.$emit('message', "Wrong Username or Password");
        }

        // Create session
        this.$session.start();
        this.$session.set("user_token", resdata.data.token);
        this.$session.set("user_name", resdata.data.userName);
        this.$session.set("role", resdata.data.role);
        this.$session.set("fullname", resdata.data.fullname);

        if(resdata.data.profileImg !=null){
          this.$session.set("profileImg", resdata.data.profileImg);
        }
      })
  }
}

```

```

// Redirect to Home Page
this.$router.push('/Home');
})
.catch((error) => {
  this.errorMessage = error;
  console.error("There was an error!", error);
});
},
closeDialog: function(){
  this.dialog = false;
}
}
}
</script>

```

Once the authenticate function invokes it retrieves the request parameters and checks if they are available in the Users collection in firebase. If it found a document that means the given email address and password are correct. Then it passes the response back to the front-end application.

```

// Authenticate User
const authenticate = async (req, res, next) => {
  try {
    // Retrieve parameters
    const email = req.body.email;
    const password = req.body.password;

    // Create reference to User collection
    const userRef = db.collection('User');
    // Check if email and password are exists
    const snapshot = await userRef.where('email', '==', email).where('password', '==', password).get();

    // Check if response is empty
    if (snapshot.empty) {
      console.log('No matching documents. ');
      return res.status(400).json({
        status: 'error',
        msg: 'User Authenticated Failed',
      });
    }

    snapshot.forEach(doc => {
      console.log(doc.id, '=>', doc.data());

      let fullname = doc.data().firstName + ' ' + doc.data().lastName;

      // Send data to front end
      res.status(200).json({
        status: 'success',
        data: { 'token': doc.id, 'userName': doc.data().firstName, 'role': doc.data().role, 'fullname': fullname, 'profileImg': doc.data().profileImg },
        msg: 'User Authenticated Successfully',
      });
    });
  } catch (er) {
    res.status(500).json({
      status: 'error',
      error: er,
    });
  }
}
}

```

## Make a donation

The payment gateway is invoked by the following code segment. Customer email, payment amount, payment reference, and paystack public key are passed as parameters.

```
<paystack
  :amount="getamount"
  :email="email"
  :paystackkey="paystackkey"
  :reference="reference"
  :callback="callback"
  :close="close"
  :embed="false">
  <div style="margin-top: 6px">
    Credit or Debit Card
  </div>
</paystack>
```

Once the user is made the payment callback function will call and pass the transaction status back to the system. The system will read the transaction response and pass the information to the backend server to update the transaction in the database. Please refer to the below code segment,

```
var myHeaders = new Headers();
  myHeaders.append("Content-Type", "application/json");

  var raw = JSON.stringify({
    campaignId: this.id,
    name: this.displayName,
    amount: Number(this.toggle_exclusive),
    message: this.message,
    userId: this.userId,
    trxref: response.trxref,
  });

  var requestOptions = {
    method: "POST",
    mode: "cors",
    headers: myHeaders,
    body: raw,
    redirect: "follow",
  };

  fetch("http://localhost:3000/api/payment/donate", requestOptions)
    .then(async (response) => {
      const resdata = await response.json();
```

```

    // check for error response
    if (!response.ok) {
        // get error message from body or default to response.statusText
    }
    console.log(resdata.data.status);
})
.catch((error) => {
    this.errorMessage = error;
    console.error("There was an error!", error);
});
}

```

## Create Campaign

```

const create = async (req, res, next) => {
  try {
    // Check if the request body is empty
    if (!req.body) {
      return res.status(400).json({
        status: 'error',
        msg: 'Body is Required',
      });
    }
    const data = req.body;

    var createdAt = new Date();
    console.log(createdAt.toISOString().slice(0, 10));

    // Create a dummy ID for the campaign
    var id = db.collection('Campaign').doc().id;

    // Map request parameters to Campaign object
    Campaign.campaignName = req.body.campaignName;
    Campaign.campaignStartDate = req.body.campaignStartDate;
    Campaign.campaignEndDate = req.body.campaignEndDate;
    Campaign.campaignDescription = req.body.campaignDescription;
    Campaign.goalAmount = parseFloat(req.body.goal);
    Campaign.raiedAmount = 0.0;
    Campaign.city = req.body.city;
    Campaign.province = req.body.province;
    Campaign.campaignStatus = "Request";
    Campaign.dateCreated = createdAt;
    Campaign.createdBy = req.body.createdBy;
    Campaign.id = id;
  }
}

```

```

Campaign.headerImg = "";
Campaign.mainImg = "";
Campaign.noOfDonations = 0;
Campaign.shortDescription = req.body.shortDescription;

// Add Campaign object as a document to firebase
const snapshot = await db.collection('Campaign').doc(id).set(Campaign).then(() => {
  console.log("Campaign Created Successfully");
});

// Create relevant campaign request document
const campaignStatusData = {
  campaignstatus: 'Request ',
  approvedBy: "",
  comment: "",
  approvedDate: ""
};

// Send the response to user
const response = await db.collection('CampaignApproval').doc(id).set(campaignStatusData).then(() => {
  return res.status(200).json({
    status: 'Success',
    data: id,
    msg: 'Campaign Created Successfully',
  });
});
} catch (error) {
  return res.status(500).send(error.message);
}
}

```

# Chapter 5 - Testing and Evaluation

## 5.1.Introduction

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that the software product is Defect free. (Hamilton, 2022) It is a process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include

- Preventing bugs
- Reducing development costs
- Improve customer satisfaction.
- Improve product quality.
- Improve security.
- Improving performance etc.

Testing is an important phase in SDLC that need to carry out during the implementation phase. The testing phase can be divided into two sub-segments. Testing phase and evaluation phase. Testing is carried out during the implementation phase and evaluation can be conducted after implementing the system or a prototype of the system.

## 5.2.Related Testing Types

There are many testing types available today. To test, the “Charity” platform thoroughly, a combination of few test types are used.

### Manual Testing

Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools (Hamilton, 2022). Although manual testing is considered the most primitive testing type we have to conduct it because 100% automated testing can not be done in a practical scenario. There are many manual testing methods with specific objectives and strategies available today. When developing the Donation management system testing types are used in the following order.

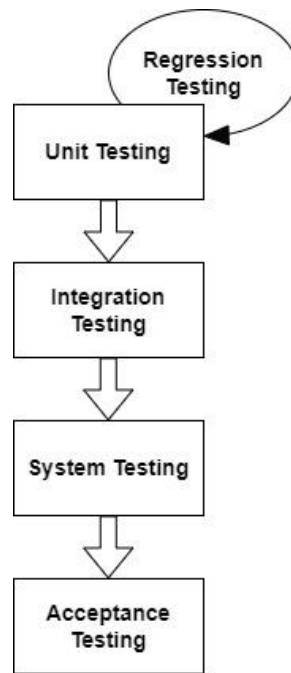


Figure 5.1: Test execution hierarchy

### 1. Unit testing

Units are the smallest testable component in any application. In unit testing, individual modules are built and tested by the developer. When developing the Charity system unit tests are conducted from the beginning of the coding stage to minimize errors. Since I have followed the bottom-up approach in developing the system it is easy to conduct unit tests.

When developing the system I divided the entire system into modules. For each module, sub-modules are created according to their functionality. These sub-modules are tested using

### 2. Regression testing

Once the code base is changed after modification we have to ensure that previously run tests are still passed after modification. For that, we are conducting regression testing. When developing the donation management system after each modification regression testing is performed to verify that recent code changes aren't affecting the existing features.

### 3. Integrated testing

When individually built software components are combined we have to make sure that it won't affect the working system. The main difference between regression testing and integration testing is that regression testing verifies only whether previously executed tests are still passed after making changes to the system and interaction testing verifies whether newly added features work properly with existing modules.

For the integrated testing, I have used the bottom-up incremental testing approach where two modules are integrated together and tested for the proper functioning of the application. After that other related modules are integrated incrementally and tested accordingly.

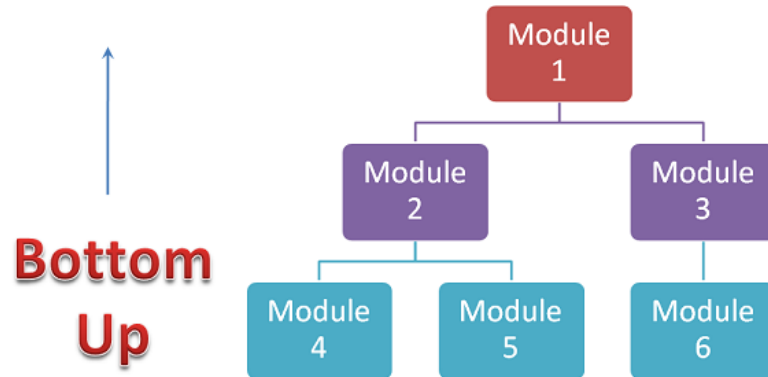


Figure 5.2: Bottom-up incremental integration testing approach

#### 4. System Testing

Once the system is developed and integrated completely entire system should be tested. This is called system testing.

#### 5. Acceptance Testing

Acceptance testing is a method of testing that is used to determine whether a software system satisfies the requirements specification.

- Automated Testing

Automated testing is a testing technique that used special tools to execute test cases. Once the automated testing is configured there is no manual intervention is needed.

For the Charity system, I have used an automated testing strategy to test the back-end application. For this, I've used the mocha testing library. Mocha is a javascript framework for node.js programs.



```

/**
 * Test Get User by ID
 */
describe("POST /api/user/getUser", ()=>{
  it("Get specific User by ID", (done)=>{
    const id = {
      id: "lqblyRwIeyl7jL6V8Chj"
    };
    chai.request(server)
      .post("/api/user/getUser")
      .send(id)
      .end((err, response)=>{
        response.should.have.status(200);
        response.body.should.be.a('object');
        response.body.should.have.property('status').eq("success");
        response.body.should.have.property('status').should.be.a('object');
        done();
      });
  });
});

```

Figure 5.3: Sample Mocha automated test case

### 5.3. Test Cases

This section describes the system test cases carried out when doing the manual testing of each module and its functionality. The system is developed and tested module by module to reduce the complexity of both development and testing. Following are some of the test cases and their corresponding results which are carried out for the major system modules.

#### 5.3.1. Test Cases for User Authentication

Test Case ID	Description	Testing Steps	Expected Output	Actual Results	Pass/Fail
TU01	Check user login with valid data	1) Go to the website login page 2) Enter the Email address 3) Enter the Password 4) Click submit button	<ul style="list-style-type: none"> <li>The user should be able to log into the system.</li> <li>If the user is a donor home page should be displayed.</li> <li>If the user is an administrative staff dashboard page should be displayed</li> </ul>	As expected	Pass
TU02	Check user login with an invalid email address	1) Go to the website login page 2) Enter the Email address 3) Enter the Password 4) Click submit button	<ul style="list-style-type: none"> <li>The user should not be able to log into the system.</li> <li>An error message should be displayed to the user</li> </ul>	As expected	Pass
TU03	Check user login with	1) Go to the website login page	<ul style="list-style-type: none"> <li>The user should not be able to log into the system.</li> </ul>	As expected	Pass

	an invalid password	2) Enter the Email address 3) Enter the Password 4) Click submit button	<ul style="list-style-type: none"> <li>The error message should be displayed</li> </ul>		
TU04	Check user login with null data	1) Go to the website login page 2) Click submit button	<ul style="list-style-type: none"> <li>The user should not be able to log into the system.</li> <li>An error message should be displayed to the user</li> </ul>	As expected	Pass

Table 5-1: User Authentication Test Cases

### 5.3.2. Test Cases for User Registration

Test Case ID	Description	Testing Steps	Expected Output	Actual Results	Pass/Fail
TU01	Register the user with a valid email address	1) Go to the website signup page 2) Enter the Email address 3) Enter the Password 4) Select the account type 5) Click on the Register button	<ul style="list-style-type: none"> <li>The user will redirect to the email confirmation page.</li> <li>An email confirmation should be sent to the user's given email address.</li> </ul>	As expected	Pass
TU02	Register the user with an already existing email address	1) Go to the website signup page 2) Enter the Email address 3) Enter the Password 4) Select the account type 5) Click on the Register button		As expected	Pass
TU03	Register user without filling all fields	1) Go to the website signup page 2) Click the Register button	<ul style="list-style-type: none"> <li>The user should be able to log into the system.</li> <li>Empty fields should be highlighted in red and an error message should be displayed to the user.</li> </ul>	As expected	Pass

Table 5-2: User Registration Test cases

### 5.3.3. Test Cases for Campaign Creation

Test Case ID	Description	Testing Steps	Expected Output	Actual Results	Pass/Fail
TU01	Create a campaign by filling in	1) Go to the Campaign create page. 2) Fill in all details.	<ul style="list-style-type: none"> <li>A successful message should be displayed.</li> <li>The user will be redirected to the campaign dashboard.</li> </ul>	As expected	Pass

	all required fields	3) Upload campaign images. 4) Upload support documents. 5) Click on Submit button.			
TU02	Create a campaign without filling in all required fields	1) Go to the Campaign create page. 2) Fill few details. 3) Upload campaign image. 4) Upload support documents. 5) Click on Submit button.	<ul style="list-style-type: none"> <li>Required fields should be highlighted in red and an error message should display to the user.</li> </ul>	As expected	Pass
TU03	Register user without uploading campaign images	1) Go to the Campaign create page. 2) Fill in all details. 3) Upload support documents. 4) Click on Submit button.	<ul style="list-style-type: none"> <li>Required fields should be highlighted in red and an error message should display to the user.</li> </ul>	As expected	Pass
TU04	Register user without uploading support document	1) Go to the Campaign create page. 2) Fill in all details. 3) Upload campaign image. 4) Click on Submit button.	<ul style="list-style-type: none"> <li>Users should be able to create a campaign even without uploading the support documents.</li> <li>A successful message should be displayed and the user will be redirected to the campaign dashboard.</li> </ul>	As expected	Pass

Table 5-3: Create new campaign test cases

#### 5.3.4. Test Cases for Donation

Test Case ID	Description	Testing Steps	Expected Output	Actual Results	Pass/Fail
TU01	Donate with valid credit card details	1) Go to any campaign page. 2) Click on donate button 3) Enter the donation amount. 4) Enter donor name and message (Optional) 5) Click on the pay online button. 6) Enter valid credit card details. 7) Click on the Pay button	<ul style="list-style-type: none"> <li>The amount should be deducted from the user's credit card.</li> <li>A success message should be displayed to the user.</li> <li>Donation points should be calculated and displayed on the user's profile page.</li> </ul>	As expected	Pass
TU02	Donate with invalid	1) Go to any campaign page. 2) Click on donate button	<ul style="list-style-type: none"> <li>The amount should not be deducted from the user's credit card.</li> </ul>	As expected	Pass

	credit card details	3) Enter the donation amount. 4) Enter donor name and message (Optional) 5) Click on the pay online button. 6) Enter invalid credit card details. 7) Click on the Pay button	<ul style="list-style-type: none"> <li>• An error message should be displayed to the user.</li> <li>• Donation points won't be calculated,</li> </ul>		
--	---------------------	--	---	--	--

Table 5-4: Make a donation test case

### 5.3.5. Test Cases for Share Campaigns on Social Media

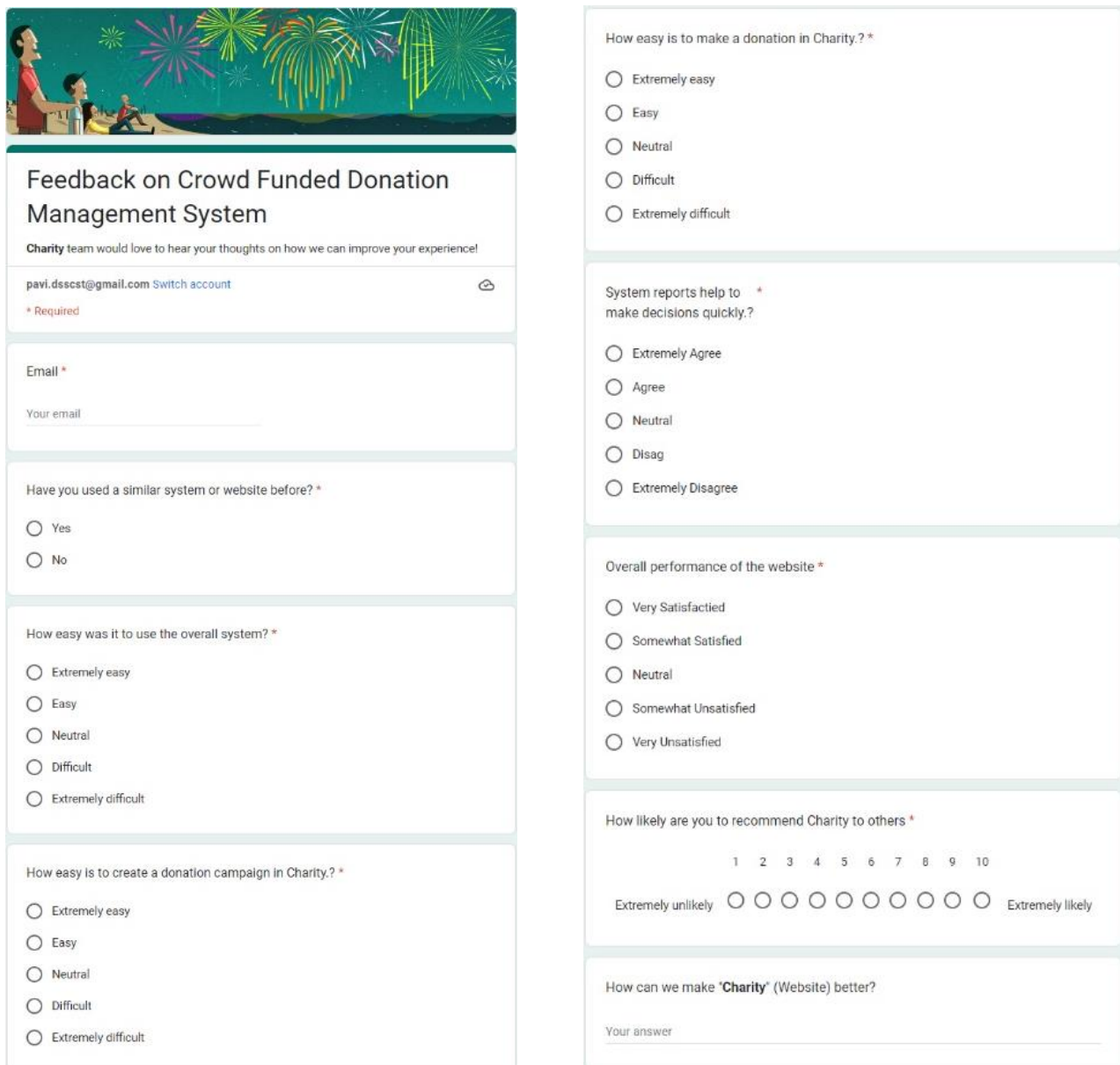
Test Case ID	Description	Testing Steps	Expected Output	Actual Results	Pass/Fail
TU01	Share the campaign with valid social media credentials.	1) Go to any campaign page. 2) Click on the share button. 3) Select social media by clicking the social media icon. 4) Click on the share button 5) Enter a valid selected social media account credential on the popup menu. 6) Enter the message 7) Click on the share button	<ul style="list-style-type: none"> <li>• The selected campaign should be shared on social media.</li> </ul>	As expected	Pass
TU02	Share campaign with valid social media credentials	1) Go to any campaign page. 2) Click on the share button. 3) Select social media by clicking the social media icon. 4) Click on the share button 5) Enter the invalid social media account credential on the popup menu. 6) Enter the message 7) Click on the share button.	<ul style="list-style-type: none"> <li>• An error message should pop up.</li> <li>• Users should not be able to share the campaign on social media.</li> </ul>	As expected	Pass

Table 5-5: Share campaign on social media test cases

## 5.4. User Evaluation

User evaluation is considered the most important process to do when developing a system. This is done after the implantation process in the system development life cycle. This is the stage in the software development process in which the system is tested within the user's environment.

Normally user evaluation is conducted by selecting different types of users. Since this is an open system for everyone there is no specific client. So when doing the user evaluation I developed an open questionnaire and distribute it among various people to get their feedback. When distributing the questionnaire I didn't filter out the users because the system is developed as an open system.



**Feedback on Crowd Funded Donation Management System**  
Charity team would love to hear your thoughts on how we can improve your experience!

pavi.dsscst@gmail.com [Switch account](#)

\* Required

Email \*

Your email

Have you used a similar system or website before? \*

Yes  
 No

How easy was it to use the overall system? \*

Extremely easy  
 Easy  
 Neutral  
 Difficult  
 Extremely difficult

How easy is to create a donation campaign in Charity.? \*

Extremely easy  
 Easy  
 Neutral  
 Difficult  
 Extremely difficult

How easy is to make a donation in Charity.? \*

Extremely easy  
 Easy  
 Neutral  
 Difficult  
 Extremely difficult

System reports help to make decisions quickly.? \*

Extremely Agree  
 Agree  
 Neutral  
 Disag  
 Extremely Disagree

Overall performance of the website \*

Very Satisfied  
 Somewhat Satisfied  
 Neutral  
 Somewhat Unsatisfied  
 Very Unsatisfied

How likely are you to recommend Charity to others? \*

1 2 3 4 5 6 7 8 9 10

Extremely unlikely            Extremely likely

How can we make 'Charity' (Website) better?

Your answer

Figure 5.4: User Evaluation Questionnaire

## 5.5. Results of the Testing

The overall result of the user evaluation is displayed below.

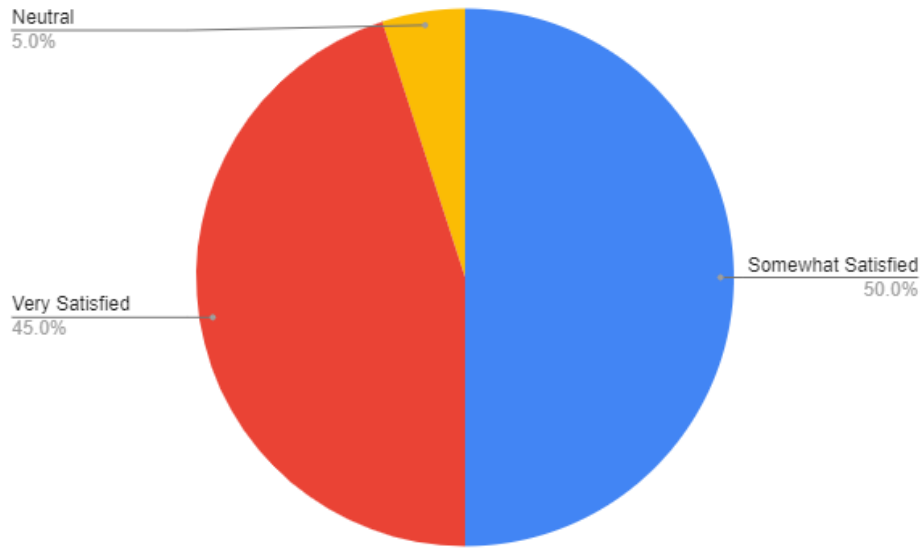


Figure 5.5: Overall result of the user evaluation

45% of the users are very satisfied with the overall system functionality and performance of the system and 50% of the users are somewhat satisfied with the system. Only 5% of the users responded as neutral. (see figure 5.5)

More than 55% of users say that it is easy to create a campaign and make a donation in the system. 32.2% of users say that process is extremely easy. Only 2.9% of the users say that the whole process is difficult. (see figure 5.6)

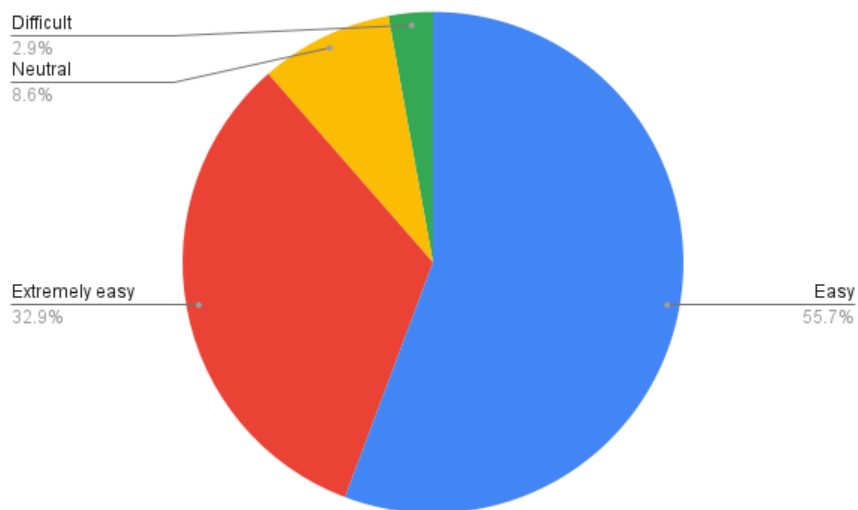


Figure 5.6: Overall result of user evaluation (Process evaluation)

A few interesting things popped up during the user evaluation questionnaire. One of them is that none of the participants used a platform like this before. (see figure 5.7) Although it is a new initiative participants like to get involved.

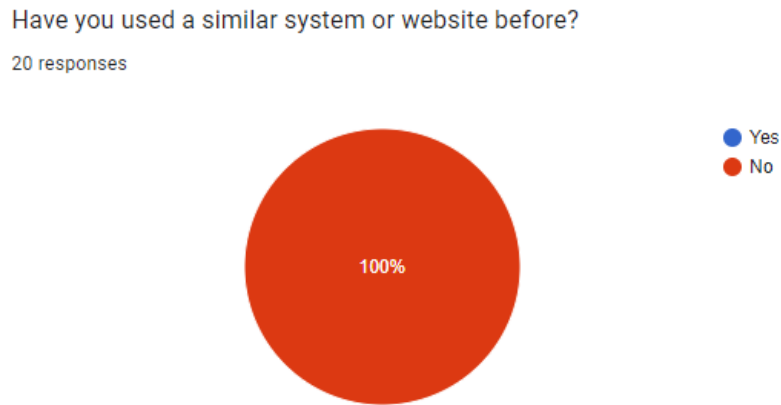


Figure 5.7: Result of similar system usage

Most of them are likely to recommend the “Charity” platform to others. (See figure 5.8)

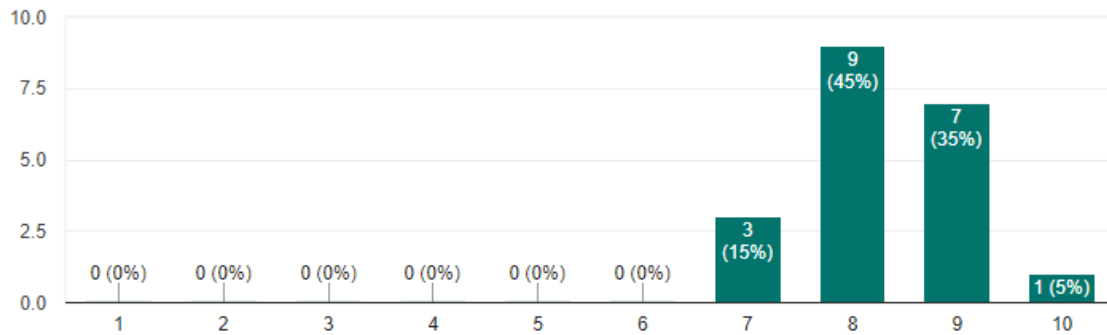


Figure 5.8: User recommendation

The test result of the automated testing is displayed below. Each API endpoint is tested thoroughly with the mocha test automation library to increase the quality of the software.



Figure 5.9: Overall Automated Test Results

## User API

\test\user.js

<b>POST /api/user/getUser</b>		^
\test\user.js		
⌚ 260ms	📄 2	✅ 2
✅	Get specific User by ID	174ms ⌚
✅	Get specific User by providing wrong User ID	86ms ⌚
<b>POST /api/user/authenticate</b>		^
\test\user.js		
⌚ 285ms	📄 3	✅ 3
✅	Authenticate User by correct username and password	99ms ⌚
✅	Authenticate user by wrong email	93ms ⌚
✅	Authenticate user by wrong password	93ms ⌚
<b>POST /api/user/getAllUsers</b>		^
\test\user.js		
⌚ 1s	📄 5	✅ 5
✅	Get All Users by all parameter	473ms ⌚
✅	Get All Users by passing one role and all statuses	231ms ⌚
✅	Get All Users by role and status	165ms ⌚
✅	Get All Users by passing null values	172ms ⌚
✅	Get All Users without passing parameters	12ms ⌚
<b>POST /api/user/addUser</b>		^
\test\user.js		
⌚ 13ms	📄 3	✅ 2 ⏸ 1
⏸	Add User	0ms ⌚
✅	Add User without passing parameters	7ms ⌚
✅	Add User by already exists Email	6ms ⌚



## Campaign API

\test\campaign.js



### POST /api/campaign/getCampaigns

\test\campaign.js



🕒 2.2s 📄 3 ✓ 3

✓ IT should return all Campaign details of Specific User	2s 🕒
✓ IT should not return Campaign details of the User	208ms 🕒
✓ IT should not return Campaign details of the User	4ms 🕒

### POST /api/campaign/getWatchlist

\test\campaign.js



🕒 555ms 📄 3 ✓ 3

✓ IT should return the requested Watchlist details	361ms 🕒
✓ IT should not return the requested Campaign details	190ms 🕒
✓ IT should not return the requested Campaign details	4ms 🕒

Figure 5.10: Test automation results

# Chapter 6 – Conclusion

## 6.1.Introduction

Today people are moving away from donation campaigns because there are a lot of scammers out there who perform unethical work by taking charity as their cover. Lack of trust is the main concern of any charity event. Not like in the early days, now people think twice before donating because everyone faces the impact of the increase in the cost of living. Today most charity organizations had to face the issue of finding donors.

## 6.2.Critical Evaluation

The “Charity” platform is a crowd-funded donation management system that was built to add transparency to fundraising events. With the power of technology, fundraising should be easy and convenient for everyone. The main purpose of the system is to provide a safe transparent socially engaging platform for everyone to conduct hassle-free donation campaigns.

From the requirement analysis phase, all the functional and non-functional requirements are focused to achieve the objectives of the system as well as the satisfaction of the users. Then the system was designed with five major modules: Campaign Management, Administration, User management, Payment management, and Reporting module. Afterward, the system was implemented using Vue.js and Node.js. When developing the system special attention was given to the payment module and report module. Using the report module both operation staff and fundraising organizers can closely monitor their progress and help them to make better decisions. Throughout the implementation phase testing was carried out and after implementation also separate user evaluation was carried out for quality assurance purposes. The satisfaction level of the users can be understood with the user’s feedback.

## 6.3.Problem Encountered

There was some problem encountered throughout the different phases of the project. The main problem is to find ways to attract donors to the platform. Just a donation website will not be able to keep donors for a long time. Designing a socially engaging donor network platform is a challenging task.

One of the main problems encountered in the implementation phase is that the system there is a new version of Vue.js (Vue 3) was released but the support modules only support the previous version. Ex-: When the implementation started Vuetify (A variant of Bootstrap) was not optimized to Vue 3. So the project has to be developed using an older version of Vue.js.

Integrating a payment gateway became a challenging task because registering for a payment gateway required a business Registration in Sri Lanka. Alternatives need to be found to complete the payment module. For that matter, the paystack payment gateway was chosen to integrate with the system.

## 6.4. Lessons Learned

As a student who studying a recognized degree program, I have learned practical ways to apply theories learned in the classroom. I trained to complete the project within a limited time period but from beginning to end I managed to follow the full System Development Life Cycle (SDLC) during the project period.

At that time I gained a lot of knowledge and experience working with a front-end framework (Vue.js), and NoSQL databases (google firebase). Also, I learned to set up an automated testing framework and conduct different types of testing throughout the project cycle.

## 6.5. Potential Future Work

As mentioned in the above introduction Charity is a crowd-funded donation management platform. These are a few suggestions to improve the platform in the future.

- **Adding more payment options**

Currently, the system only supports credit cards and debit cards when donating. But if more payment platforms are integrated then it is easy for donors to donate. Especially if we can include mobile platforms like dialog star points, dialog gene, Mobitel M cash, and dialog EZ cash with charity system more people can be attracted to use the Charity platform.

- **Add a newsletter**

Although newsletters are somewhat old, it is still considered as a good marketing tool. Although the Charity website is not a product sale website we can use newsletters to inform people about the charity events happening nearby.

- **Provide a live broadcasting facility**

Most of the time people like to be involved/participate in charity events physically. But due to factors like long distance and time, they couldn't do it. By providing a live broadcasting facility campaign organizers can live broadcast their charity events globally. It will also increase the transparency of the event as well.

- **Add monthly donation plans**

Sometimes just people don't like to visit donation websites each month and make a donation. For people like that, we can introduce monthly donation plans. By adding a monthly donation option they don't have to visit the website every time. All they have to do is select an ongoing donation campaign and select a monthly donation plan. Every week or every month progress of the donation campaign can be updated by email newsletter.

# References

Advanced photo editor (no date) Photopea. Available at: <https://www.photopea.com/> (Accessed: January 22, 2022).

A material design framework for vue.js (no date) Vuetify. Available at: <https://vuetifyjs.com/en/> (Accessed: January 30, 2022).

Borrelli, P., 2021. Angular vs. React vs. Vue.js: Comparing performance. [Blog] logrocket.com, Available at: <https://blog.logrocket.com/angular-vs-react-vs-vue-js-comparing-performance> (Accessed 13 January 2022).

Diagrams.net - free flowchart maker and diagrams online. Flowchart Maker & Online Diagram Software. [Online] Available at: <https://app.diagrams.net/> (Accessed: December 12, 2021).

Capan, T. (2013) Why the hell would I use node.js? A case-by-case tutorial, Toptal Engineering Blog. Toptal. Available at: <https://www.toptal.com/javascript/why-the-hell-would-i-use-node-js> (Accessed: January 13, 2022).

Emailjs (no date) npm. Available at: <https://www.npmjs.com/package/emailjs> (Accessed: April 28, 2022).

Gandy, D. (no date) Font awesome, Font Awesome, the iconic font and CSS toolkit. Available at: <https://fontawesome.com/v4/> (Accessed: March 15, 2022).

GlobalGiving, 2021, GlobalGiving.org. [Online] Available at: <https://www.globalgiving.org/> (Accessed 28 November 2021).

Google (no date) Google. Available at: <https://firebase.google.com/> (Accessed: January 23, 2022).

Hamilton, T. (2022) What is software testing? definition, Guru99. [Online] Available at: <https://www.guru99.com/software-testing-introduction-importance.html#:~:text=Software%20Testing%20is%20a%20method,or%20more%20properties%20of%20interest>. (Accessed: November 2, 2022).

Jayasinghe, C. 2021 'Sri Lanka police nab man for siphoning off money from 14-y-o child's brain surgery fund'. [Online] Available at: <https://economynext.com/sri-lanka-police-nab-man-for-siphoning-off-money-from-14-y-o-childs-brain-surgery-fund-84867/> (accessed 28 November 2021).

Kind Hearted Lankans 2020, Kind Hearted Lankans. [Online] Available at: <https://www.kindheartedlankans.com> (Accessed 05 July 2021).

Mightycause.com, 2021, Mightycause: Nonprofit fundraising Made Easy. Available at: <https://www.mightycause.com> (Accessed 28 November 2021).

Nbts.health.gov.lk. 2021. National Blood Transfusion Service Online. Available at: <http://www.nbts.health.gov.lk/index.php/component/k2/item/321-appointment-booking-system> (Accessed 28 November 2021).

Stack overflow developer survey 2019 (no date) Stack Overflow. Available at: <https://insights.stackoverflow.com/survey/2019#technology> (Accessed: January 12, 2022).

Stack overflow developer survey 2021 (no date) Stack Overflow. Available at: <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language> (Accessed: January 12, 2022).

UUID (no date) npm. Available at: <https://www.npmjs.com/package/uuid> (Accessed: October 2, 2022).

Vue-chartjs (no date) npm. Available at: <https://www.npmjs.com/package/vue-chartjs> (Accessed: October 20, 2022).

Vue2-editor (no date) npm. Available at: <https://www.npmjs.com/package/vue2-editor> (Accessed: March 29, 2022).

Vue-Json-Excel (no date) npm. Available at: <https://www.npmjs.com/package/vue-json-excel> (Accessed: July 3, 2022).

Vue-Paystack (no date) npm. Available at: <https://www.npmjs.com/package/vue-paystack> (Accessed: October 2, 2022).

Vue-Social-sharing (no date) npm. Available at: <https://www.npmjs.com/package/vue-social-sharing> (Accessed: April 2, 2022).

V2.vuejs.org. n.d. Comparison with Other Frameworks — Vue.js. [online] Available at: <https://v2.vuejs.org/v2/guide/comparison.html> (Accessed 13 January 2021).

WION Web Team, 2020, '#IshiniGate: Sri Lankan influencer accused of siphoning \$82,000 raised for Easter attack victims'. Available at: <https://www.wionews.com/south-asia/ishinigate-sri-lankan-influencer-accused-of-siphoning-82000-raised-for-easter-attack-victims-331773> (Accessed 28 November 2021).

Winston (no date) npm. Available at: <https://www.npmjs.com/package/winston> (Accessed: July 19, 2022).

# Appendix A - System Documentation

Hardware and software requirement for deployment of the system is explained here in this system documentation.

The following table shows the minimum requirements required for on-premises deployment of the full system (front-end application and backend server)

Hardware Requirement	Software Requirement
Intel(R) Pentium(TM) 4 @ CPU 3.0GHz	Windows 10 or Ubuntu Ubuntu 14.04
2GB RAM	
160 GB HDD	

*Table A-1: Hardware & Software requirements of on-premises deployment*

Table A-2 shows the minimum required virtual machine

Cloud providers	Minimum required VM
Amazon Web service	t1.micro or t2.micro instance
Microsoft Azure	f1-micro instance
Google Cloud	Basic_A0 Virtual machine

*Table A-2: Minimum requirement for cloud deployment*

Required instance for Cloud deployment

Deployment instruction for the front-end application.

- Use npm run build command to create a deployable build. It'll create a deployable folder called dist.

```
File                               Size                               Gzipped
dist\js\chunk-vendors.9ea2780c.js  322.74 KiB                        104.60 KiB
dist\js\about.a088c52b.js         112.11 KiB                        28.97 KiB
dist\js\app.51e3fbf2.js           16.85 KiB                          8.70 KiB
dist\css\chunk-vendors.2515bea0.css 365.62 KiB                        43.67 KiB
dist\css\about.3500d70f.css       31.84 KiB                          5.32 KiB
dist\css\app.2db3d4e8.css          0.63 KiB                           0.26 KiB

Images and other types of assets omitted.

DONE Build complete. The dist directory is ready to be deployed.
INFO  Check out deployment instructions at https://cli.vuejs.org/guide/deployment.html

PS D:\MIT Project\donation-management>
```

Figure A.1: Create a deployable build of the application

- Files inside the dist folder can be directly copied to the client-server.

Deployment instructions for the server application.

There are a lot of ways to deploy a node js server on different operating systems. Please refer (Deploying Node/Express app on a windows server 2019) for more details on how to deploy the nodejs server on a windows server.

# Appendix B - Design Documentation

## Use case scenarios in detail

Following is the detailed elaboration of use case scenarios mentioned in Chapter 3.

### **Use Case 1: Log in to the system**

Primary Actor -: Donor/Campaign Organizer, Administration Staff

Pre-Condition -: Active Internet connection should be available.

Main Scenario -:

- 1) Enter URL or Google search “Charity” website.
- 2) Click Sign in button
- 3) Enter Username (Email address) and Password and click login
- 4) System checks for the authenticity of the username and password.
- 5) According to user roles system will load the relevant landing page. For normal donors system will load the normal dashboard. For campaign organizers system will enable more features. For admin staff system will load the admin dashboard.

Alternative Scenario -:

4(a) Authorization failed.

4(a) 1. An error message will pop up to the user informing that his/her username or Password is incorrect.

4(a) 2. The system asks the user to reenter the username or password.

### **Use Case 2: Change Password**

Primary Actor -: Donor/Campaign Organizer, Administration Staff

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The actor who is performing this action should be a registered user.

Main Scenario -:

- 1) Click forgot password link on the login screen.



- 2) Enter the email address used to register for the website.
- 3) A password reset link will be sent to the given email address
- 4) The user has to log in to his/her email and click the password reset link.
- 5) A new window will open and the user can type his/her new password.
- 6) Enter a new password.
- 7) Click Save changes.

### **Use Case 3: Register Donor**

Primary Actor -: Donor

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should have a valid email address.

Main Scenario -:

- 1) Enter URL or Google search “Charity” website.
- 2) Click the Signup button
- 3) Select the donor radio button.
- 4) The donor will initiate the registration process.
- 5) A basic questionnaire is given to the donor which is relevant to his/her personal details and contact details.
- 6) The donor answers the basic questionnaires and clicks the register button.
- 7) An invitation email will be sent to the user's given email address to confirm the user's email address validity.

Alternative Scenario -:

5(a) Email already exists.

5(a) 1. An error message will pop up to the user informing that his/her email address is  
Already taken.

5(a) 2. The system asks the user to reenter the email address.

5(b) Donor doesn't answer the mandatory questions.

5(b) 1. An error message will pop up to the user informing that he/she didn't provide  
answers to all the mandatory questions.

5(b) 2. The system will highlight the question which isn't answered by the user.

## Use Case 4: Register Campaign Organizer

Primary Actor -: Campaign Organizer

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should have a valid email address.

Main Scenario -:

- 1) Enter URL or Google search “Charity” website.
- 2) Click the Sign-up button.
- 3) Select the Campaign Organizer radio button.
- 4) The campaign Organizer will initiate the registration process.
- 5) A basic questionnaire is given to the Campaign Organizer which is relevant to his/her personal details and contact details.
- 6) The Campaign Organizer answers the basic questionnaires and clicks the next button.
- 7) The system validates the user’s input and asks the user to submit the necessary documents. (NIC, address, verification documents, etc.)
- 8) An authorized person submits the reports and clicks next.
- 9) The system validates the entries.
- 10) Click the register user button to create an account.
- 11) An invitation email will be sent to the user's given email address to confirm the user’s email address validity.

Alternative Scenario -:

6(a) Email already exists.

6(a) 1. An error message will pop up to the user informing that his/her email address is Already taken.

6(a) 2. The system asks the user to reenter the email address.

6(b) User doesn’t answer the mandatory questions.

6(b) 1. An error message will pop up to the user informing that he/she didn’t provide answers to all the mandatory questions.

6(b) 2. The system will highlight the question which isn’t answered by the user.

8(a) User doesn’t submit the required documents.

8(a) 1. An error message will pop up to the user informing that he/she didn’t upload the necessary documents.

8(a) 2. The system will highlight the necessary field and ask the user to submit the required documents again.

8(b) User tries to submit unsupported files.

8(b) 1. An error message will pop up to the user informing that donor is trying to upload unsupported file to the system.

8(b) 2. The system will highlight the necessary field and ask the user to submit the donor's reports again.

### **Use Case 5: Change Personal details**

Primary Actor -: Donor/Campaign Organizer

Pre-Conditions -:

- 1) An Active Internet connection should be available.
- 2) The User should have an account created and already login to the system.

Main Scenario -:

- 1) Click Username and select the profile tab.
- 2) The system prompts the filled application data of the login user.
- 3) Donor changes the personal and contact details.
- 4) Click the save button to save changes to the database.
- 5) The system authenticates the updated data and displays a success message.

### **Use Case 6: Create a Campaign**

Primary Actor -: Campaign Organizer

Pre-Conditions -:

- 1) An Active Internet connection should be available.
- 2) The user should log in to the system.

Main Scenario -:

- 1) Click create campaign button in the navigation bar.
- 2) The system will prompt the basic questionnaires about the donation campaign.
- 3) The campaign Organizer will initiate the campaign registration process
- 4) The Campaign Organizer answers the basic questionnaires and clicks the next button.

- 5) The system will display another questionnaire that includes donation methods (Bank details or physical donation), contact details, proof of letters, etc. (User has to upload the proof of letters for evaluation).
- 6) The Campaign Organizer answers the questionnaire and clicks the register button.
- 7) The system will save data and create pending campaign requests for validation. After creating the campaign summarized email will be sent to the campaign organizer.

Alternative Scenario -:

4(a) User doesn't answer the mandatory questions.

4(a) 1. An error message will pop up to the user informing that he/she didn't provide answers to all the mandatory questions.

4(a) 2. The system will highlight the question which isn't answered by the user.

6(a) User doesn't submit the required documents.

6(a) 1. An error message will pop up to the user informing that he/she didn't upload the necessary documents.

6(a) 2. The system will highlight the necessary field and ask the user to submit the required documents again.

6(b) User tries to submit unsupported files.

6(b) 1. An error message will pop up to the user informing that donor is trying to upload an unsupported file to the system.

6(b) 2. The system will highlight the necessary field and ask the user to submit the donor's reports again.

## **Use Case 7: Verify donation campaign**

Primary Actor -: Administrator/Staff

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should log in to the system.

Main Scenario -:

- 1) Click the Campaign menu and select the dashboard link in the navigation bar.
- 2) The system will show all donation campaign details.
- 3) The user clicks on pending requests.
- 4) The system will filter only the pending donation campaigns and display them to the user.
- 5) The user clicks on the selected donation campaign request. Users can type the search query on the search bar and find any donation campaign.
- 6) The system will display all details about the donation campaign.
- 7) Users can go through the details download uploaded documents and verify the details. Users might have to contact people outside of the system manually to confirm the details.
- 8) Once the manual verification is done user clicks on approve button to verify the donation campaign.
- 9) An email will be sent to the campaign organizer informing them that his donation campaign is approved and live.

Alternative Scenario -:

2(a) No donation request available.

2(a) 1. The system will inform the user that there is no request received.

8(a) User clicks the reject button to reject the donation campaign request.

8(a) 1. An email will send to the campaign manager informing them that his/her campaign creation request is rejected by the administration staff.

## **Use Case 8: Search Donation campaigns**

Primary Actor -: Donor

Pre-Conditions -:

- 1) An active Internet connection should be available.

Main Scenario -:

- 1) Click on the make a donation link on the navigation panel.
- 2) The system will display the most popular donation campaigns.
- 3) Users can type his/her search query in the search bar.
- 4) The system will display results accordingly.

Alternative Scenario -:

4(a) No results available.

4(a) 1. The system will inform the user that there are no results for the search query.

### **Use Case 9: Share the donation campaign via social media.**

Primary Actor -: Donor/Campaign organizer

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should log in to the system.
- 3) Users should have a valid social media login to share the content.

Main Scenario -:

- 1) Select a donation campaign.
- 2) The system will redirect to the selected campaign page and display campaign details.
- 3) Click on desired social media icon. (Ex-: Facebook, Twitter, etc)
- 4) The system will redirect to the selected social media login page.
- 5) User authenticate the social media login and share it.

### **Use Case 10: Add a donation campaign to the watch list.**

Primary Actor -: Donor

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should already log in to the system.

Main Scenario -:

- 1) Select the donation campaign and go to the campaign page.
- 2) The system will display campaign details.
- 3) Click on add to watch list button.
- 4) The system will display a successful message and be added to the user's watch list.

### **Use Case 11: View the User's watch list.**

Primary Actor -: Donor

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should already log in to the system.

Main Scenario -:

- 1) Click the user name and select watch list.
- 2) The system will display the user's current watch list campaigns.

### **Use Case 10: Remove campaigns from the user's watch list.**

Primary Actor -: Donor

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should already log in to the system.

Main Scenario -:

- 1) Click the user name and select watch list.
- 2) The system will display the user's current watch list campaigns.
- 3) Click on the watch list item.
- 4) The system will display the campaign details.
- 5) Click on the remove watch list button.
- 6) The system will remove the campaign from the user's watch list.

### **Use Case 12: Donate to a donation campaign**

Primary Actor -: Donor

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should log in to the system.

Main Scenario -:

- 1) After login into the system select the donation campaign from the landing page or search page.
- 2) The system will display the campaign details.
- 3) The user clicks on the make a donation button.
- 4) The system will pop up a basic questionnaire about the payment. Here users have the option to select his/her donation to be anonymous.
- 5) The user fills out the payment questionnaire and clicks the pay button.
- 6) The system will redirect the user to the payment gateway.
- 7) The user enters his payment details.
- 8) The system gets the confirmation from the payment gateway and displays it to the donor.
- 9) The system will add donation points accordingly to the user's account.

5(a) User doesn't answer the mandatory questions.

5(a) 1. An error message will pop up to the user informing that he/she didn't provide answers to all the mandatory questions.

5(a) 2. The system will highlight the question which isn't answered by the user.

### **Use Case 13: Generate Reports.**

Primary Actor -: Campaign manager, Administration staff

Pre-Conditions -:

- 1) An active Internet connection should be available.
- 2) The user should already log in to the system.

Main Scenario -:

- 1) Go to the reports dashboard
- 2) Select the required report.
- 3) The system will display filtering criteria for each report.
- 4) Select necessary filtering criteria.
- 5) Click on generate report button.
- 6) System will display the report output.



# Appendix C - User Manual

## Introduction

This user document is created to educate users about how to interact with the system easily. The audience of the document includes every system user mentioned in table B-1. An assumption was made that the users are not technically mature.

## Getting Started

### User access, roles, and privileges

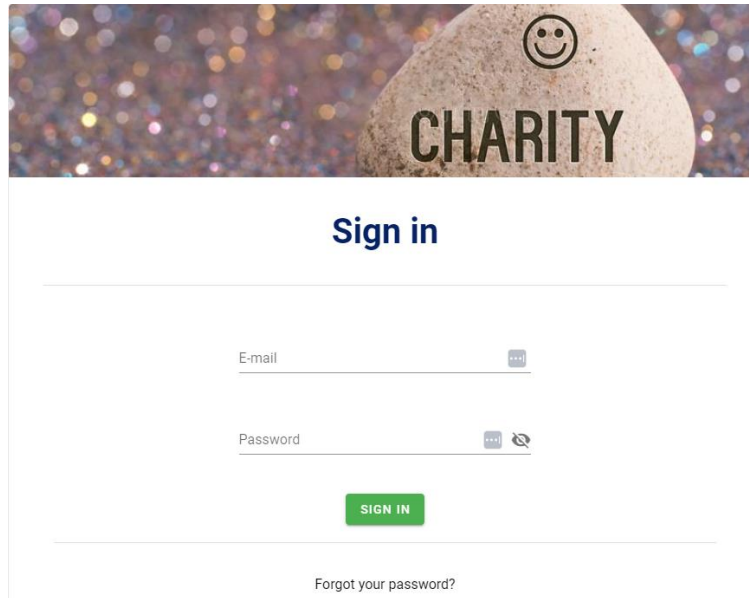
The users of the application are system administrators, operation staff, campaign organizers, and donors. Each user has a specific role assigned according to the functions they performed in the system.

User	Types of access rights
Administrator	<ul style="list-style-type: none"><li>• User management</li><li>• Role management</li><li>• System parameter configuration</li><li>• Campaign management</li><li>• Approvals</li><li>• Report generation</li></ul>
Operation staff	<ul style="list-style-type: none"><li>• User management</li><li>• Role management</li><li>• Campaign management</li><li>• Approvals</li><li>• Report generation (Restricted access)</li></ul>
Campaign Manager	<ul style="list-style-type: none"><li>• Campaign management</li><li>• Report generation (Restricted access)</li></ul>
Donor	<ul style="list-style-type: none"><li>• Make donation</li></ul>

*Table C-1: User roles of the system*

## Login Interface

Once the system is deployed properly user can access the system using any internet browser (Google Chrome, Firefox, etc.) by entering the URL <http://localhost/SignIn> and click enter. The following screen will appear for the user to sign in. Once the user enters the username(email) and password system will validate the user. If the user is valid will direct to the home page. Otherwise, the system will pop up an error message. Only valid account holders can log in to the system



CHARITY

### Sign in

E-mail

Password

SIGN IN

[Forgot your password?](#)

*Figure C.1: Sign-in page*

## Sign-up Interface

If the user doesn't have a valid user account he/she needs to signup for a new account. Go to the signup page and fill in basic details. Here user can choose to create the account as a normal donor or a campaign manager. If the user is registered as a campaign organizer he/she needs to upload verification documents in order to verify his/her account.

After successfully submitting the form user will receive an email confirmation to verify his/her account. Once the user clicks the email verification link their account will be activated can they can sign in to the "Charity" website.

CHARITY

## Sign Up

First Name

Last Name

Email Address

Password

Sign up as?

Donor

Campaign Organizer

REGISTER

*Figure C.2: Signup Page*

### **Campaign creation**

Campaign organizers can create new campaigns. Click on create campaign button on the navigation panel and figure C-3 will be displayed. Each field is validated by vuetify lazy validation. Users can upload campaign images and verification documents. Examples of verification documents are medical records, doctors' approval documents, police reports, consent letters, etc.

When creating a campaign short description has 200 character limit. It will appear in the campaign tile along with the goal amount and raised amount. But for the long description use can tell their story as they want. A rich text box is given for users to add multimedia content for long descriptions which appear in the campaign story panel.

## Create Campaign

The form is titled "Create Campaign" and contains the following fields and controls:

- Campaign Name:** A text input field.
- Campaign Period:** A date picker field showing "2022-11-15".
- Goal:** A text input field with "LKR" as a placeholder.
- City:** A text input field.
- Province:** A dropdown menu.
- Category:** A dropdown menu.
- Campaign Image:** A field with a camera icon for uploading an image.
- Short Description:** A large text area with a "0" character count at the bottom right.
- Rich Text Editor:** A toolbar with options for Normal, Bold (B), Italic (I), Underline (U), Strikethrough (ABC), Bulleted List, Numbered List, Indent, Outdent, Text Color, Background Color, Link, Unlink, Image, and Source code. Below the toolbar is a large text area for the description.
- Support Documents:** A section with a camera icon and a text input field labeled "Support Documents". To the right, it says "Please Upload all your support documents here".
- Agree to all Terms and Conditions:** A checkbox.
- Buttons:** A green "CREATE CAMPAIGN" button and a red "CANCEL" button.

Figure C.3: Campaign Creation Interface

## User Profile

Users can view their profile by clicking on the user name and clicking on the profile button in the navigation bar and the below figure will show up. It includes the user's contribution to donation campaigns, badges he/she earns by donating or participating in donation campaigns, etc.

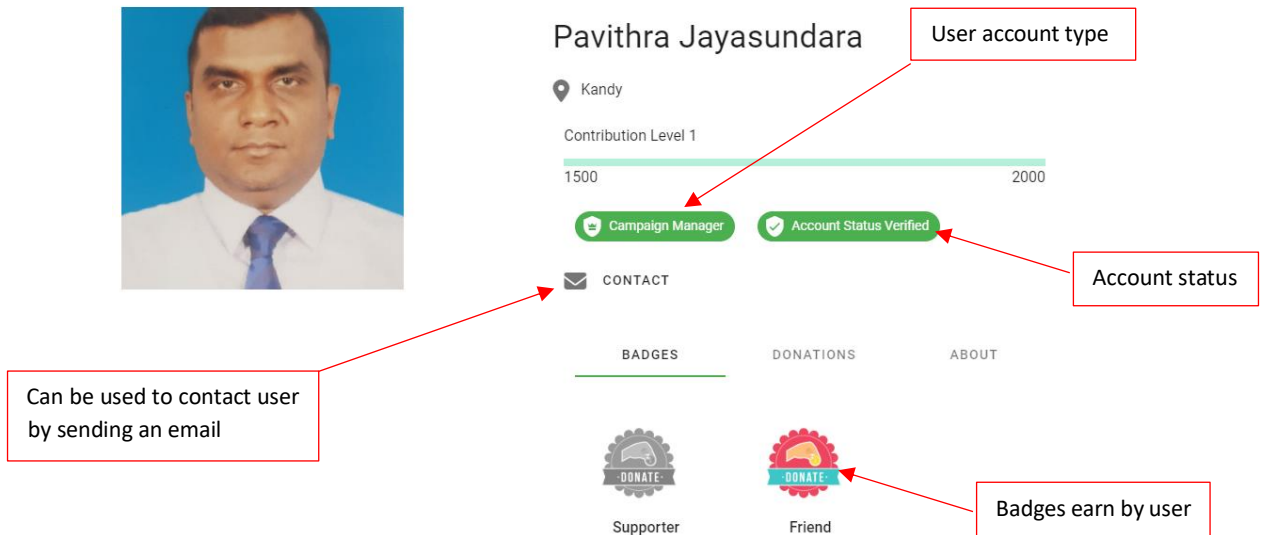


Figure C.4: User Profile Page

On the profile page, users can edit their personal information and change their profile pictures. To change the image click on the profile image and it will open a file upload dialog box. Users can select any .jpg or .png file and click ok to upload the image.

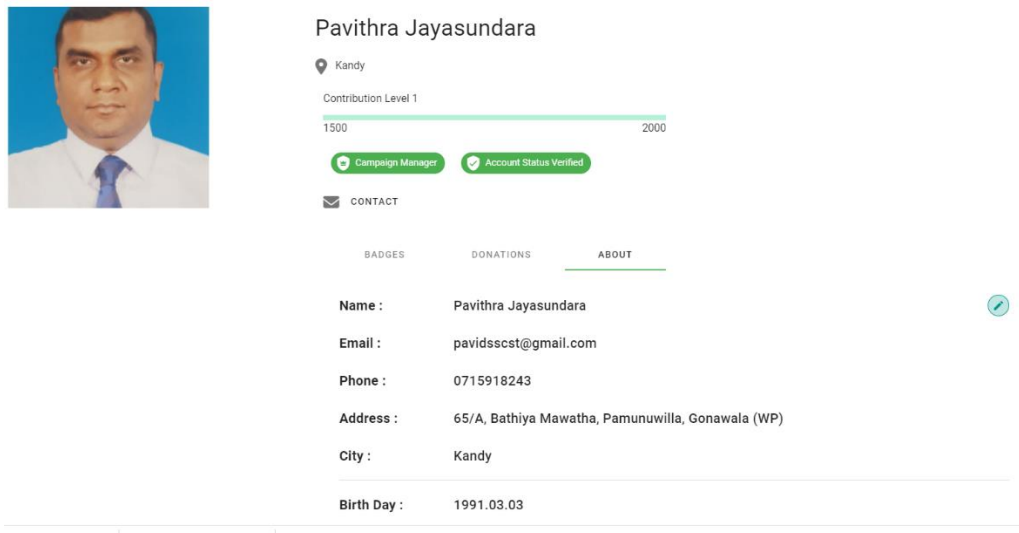
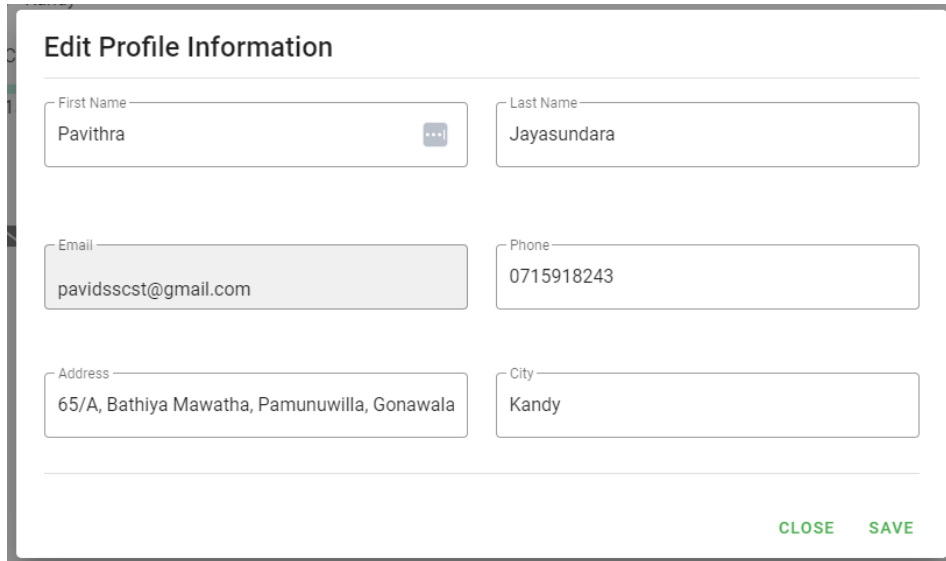


Figure C.5: User's information panel

To edit information click on the about tab and click on the pencil button and it will open a popup window like figure C-6. Here users can change their details but they can't change the email address which they used to sign up for Charity.



The image shows a web form titled "Edit Profile Information". It contains several input fields arranged in a grid. The first row has "First Name" (Pavithra) and "Last Name" (Jayasundara). The second row has "Email" (pavidsscst@gmail.com) and "Phone" (0715918243). The third row has "Address" (65/A, Bathiya Mawatha, Pamunuwilla, Gonawala) and "City" (Kandy). At the bottom right, there are two buttons: "CLOSE" and "SAVE".

First Name	Pavithra	Last Name	Jayasundara	
Email	pavidsscst@gmail.com	Phone	0715918243	
Address	65/A, Bathiya Mawatha, Pamunuwilla, Gonawala		City	Kandy

CLOSE SAVE

Figure C.6: Edit user information panel

## Global Chat

Whenever user faces an issue with the system they can easily send a chat message to operation staff. On every page, there is a chat message icon appears on the right side of the page. By clicking the icon it'll open the chat box like the below figure.

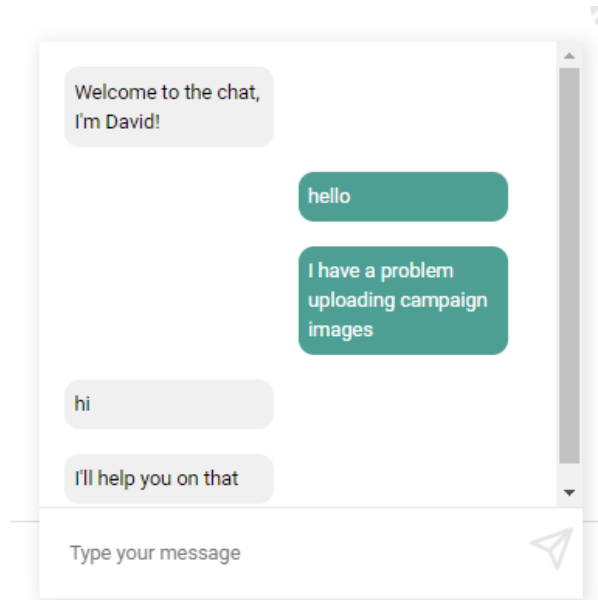


Figure C.7: Global Chatbox

## Campaign Page

Each campaign created in the Charity platform has a separate campaign page which a unique identifier. All the necessary details such as the campaign story, amount raised, and no of donations are displayed here. Once the campaign is verified, a verification tag will also be displayed here.

### Edit campaign details

Once the campaign is created and approved by the operation staff only the person who created the campaign can edit the campaign details.

The screenshot shows a charity campaign page for "Ben Kollock's Fight Against Leukemia". The page features a navigation bar with "HOME", "ABOUT", "LATEST CAUSES", "PAVITHRA", and "DONATE". The campaign details include a photo of Ben Kollock and his wife, a progress bar showing 3500 (LKR) RAISED and 1095 DONORS, and a goal of LKR 650,000. A "Verified Campaign" tag is present. The page also has a "DONATE" button, "WATCHLIST", and "SHARE" buttons. A "CONTACT" button is visible for the campaign organizer, Pavithra Jayasundara. The campaign description is provided below the photo. The footer includes the Charity logo, "Fundraise for" categories (Medical, Education, Memorial, Non Profit), "Learn more" options (How It Works), and "Resources" (About Us, Contact Us). The page is dated 2022.

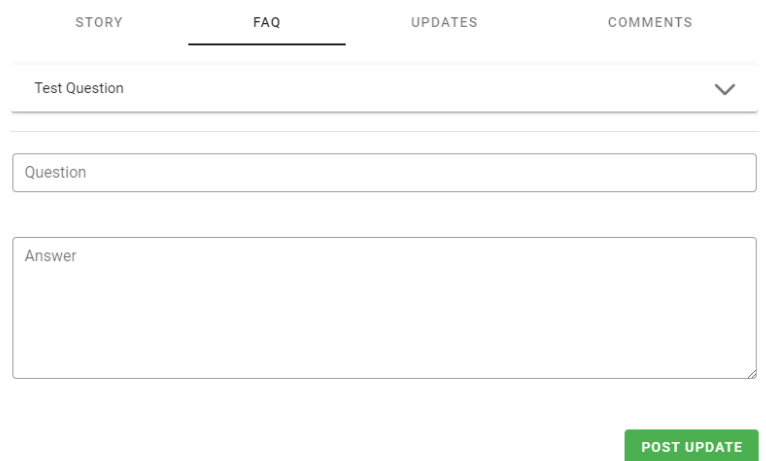
**Annotations:**

- Only the verified campaign has this campaign verification tag**: Points to the "Verified Campaign" tag.
- Any campaign can be shared via social media using share button**: Points to the "SHARE" button.
- Users can contact campaign organizers using contact**: Points to the "CONTACT" button.
- Campaign edit button.**: Points to the "CONTACT" button.
- Campaign description**: Points to the "STORY" section.

Figure C.8: Campaign Page

## FAQ section

To add, update or delete FAQs users can navigate to the FAQ post FAQ questions. Only the campaign organizer has the authorization to update this section.

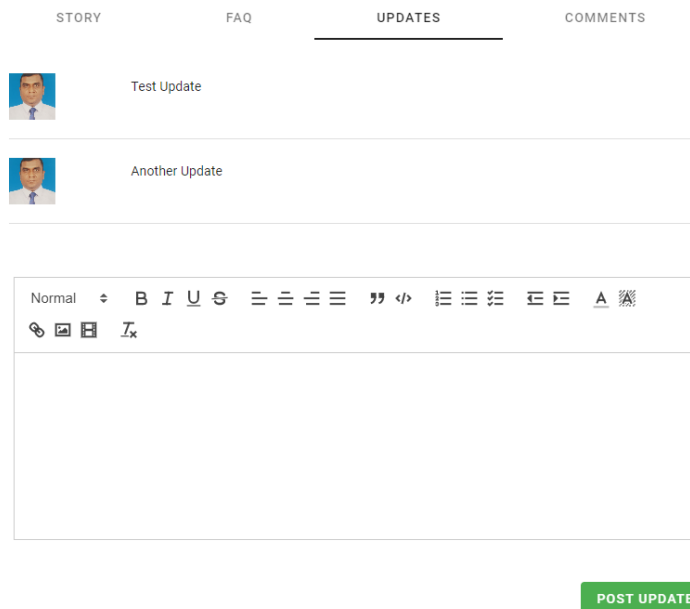


The screenshot shows a navigation bar with four tabs: STORY, FAQ (which is underlined), UPDATES, and COMMENTS. Below the navigation bar is a form with a header "Test Question" and a dropdown arrow on the right. The form contains two text input fields: "Question" and "Answer". At the bottom right of the form is a green button labeled "POST UPDATE".

Figure C.9: FAQs section

## Current campaign updates

To update the campaign status campaign organizers can use the updates section on the campaign page. Only the campaign organizer has the authorization to update this section.



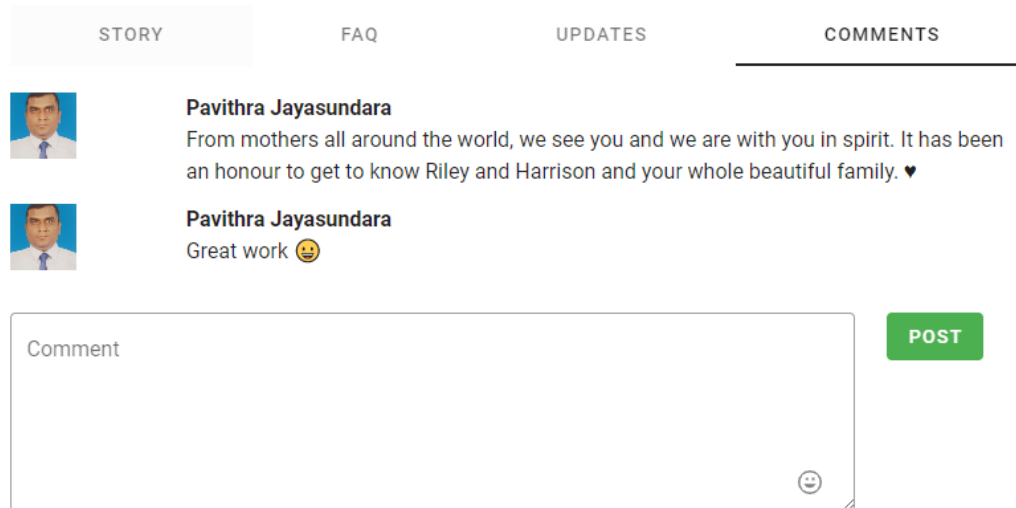
The screenshot shows a navigation bar with four tabs: STORY, FAQ, UPDATES (which is underlined), and COMMENTS. Below the navigation bar are two update entries, each with a profile picture of a man in a suit and a text label: "Test Update" and "Another Update". Below these entries is a rich text editor with a toolbar containing icons for text formatting (Normal, Bold, Italic, Underline, Strikethrough, Text Color, Background Color, Bulleted List, Numbered List, Indent, Outdent, Link, Unlink, Image, Video, Code) and a "POST UPDATE" button at the bottom right.

Figure C.10: Campaign update



## Comment on campaigns

Anyone who has a valid “Charity” account can post comments. To post a comment on a campaign navigate to the campaign page and select the comments tab. The following figure will appear for the user to post their comments.




*Figure C.11: Comment on campaigns*


## **Approve campaign**



Once the campaign is created by the campaign organizer it'll appear in the campaign request approval screen. Operation staff has to approve the campaign to appear in public. To approve the campaign go to reports -> approvals and select campaign requests. It'll display the approval screen as figure C-12

Campaign Status  
Request

**FILTER**



**Campaign Requests** Search 

Campaign Name	Campaign Start Date	Campaign End Date	City	Campaign Status	Actions
Donation Campaign for SOS Village	2022-07-01	2022-07-23		Request	
Ben Kollock's Fight Against Leukemia	2022-08-27	2022-09-30		Request	

Rows per page: 10 ▼ 1-2 of 2 < >

View Campaign icon

Figure C.12: Campaign approval page

Click the view campaign button to view the campaign details. Once the operation staff verifies the campaign details they can decide whether to approve the campaign or not by clicking either approve button or the reject button.

 Approve campaign for Ben Kollock's Fight Against Leukemia
**APPROVE** **REJECT**

## Ben Kollock's Fight Against Leukemia



3600

(LKR) RAISED

1095

DONORS

 Verified Campaign

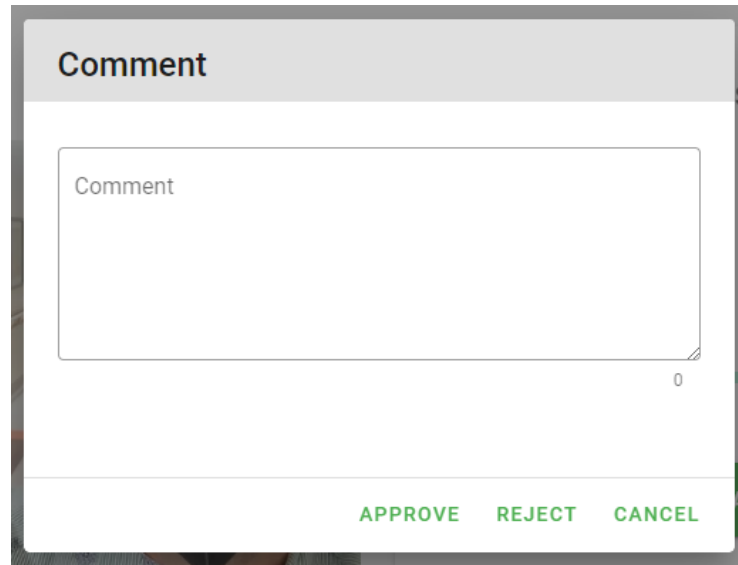
Goal: LKR 650,000

**DONATE**

 WATCHLIST
 SHARE

Figure C.13: Campaign approval

Once the user click approves button they have to enter the reason as a comment. After that user can approve the campaign and it will appear in public. Rejecting a campaign similar to the approval process and following the same steps.

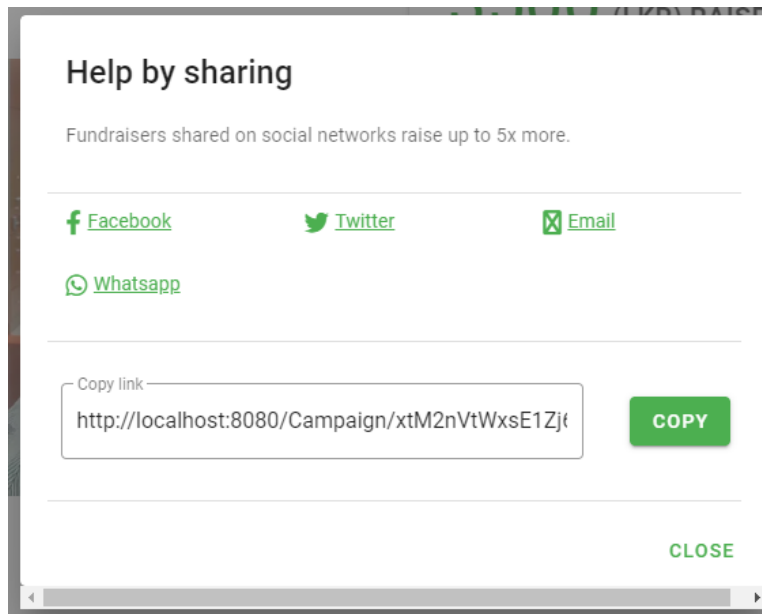


The image shows a modal dialog box titled "Comment". It features a large text input area with the placeholder text "Comment". Below the input area, there are three buttons: "APPROVE", "REJECT", and "CANCEL", all in green text. A small "0" is visible in the bottom right corner of the input area.

Figure C.14: Approve/Reject reason box

### Share donation campaign on social media

Go to the campaign page and click on the share button. The following screen will appear user to select the social media or the user can copy the shareable campaign page link to share it manually.

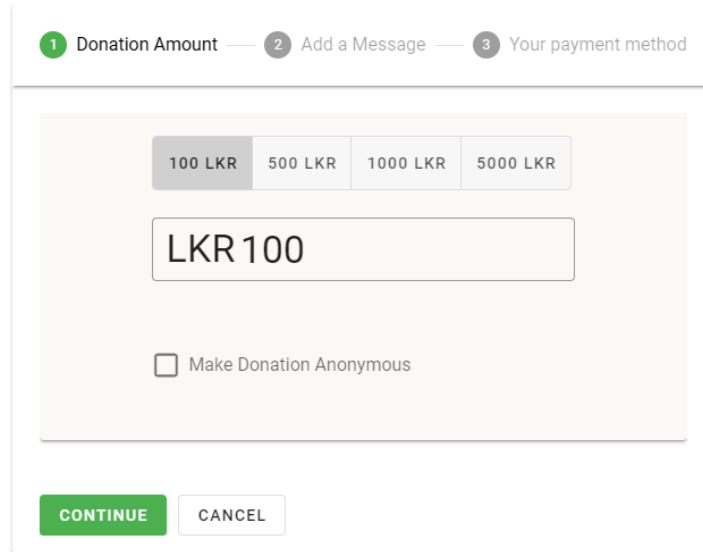


The image shows a modal dialog box titled "Help by sharing". It contains the text "Fundraisers shared on social networks raise up to 5x more." Below this, there are four social media sharing options: "Facebook", "Twitter", "Email", and "Whatsapp", each with its respective icon. At the bottom, there is a "Copy link" section with a text input field containing the URL "http://localhost:8080/Campaign/xtM2nVtWxsE1Zjt" and a green "COPY" button. A "CLOSE" button is located in the bottom right corner.

Figure C.15: Share campaign on Social media

## Make a donation

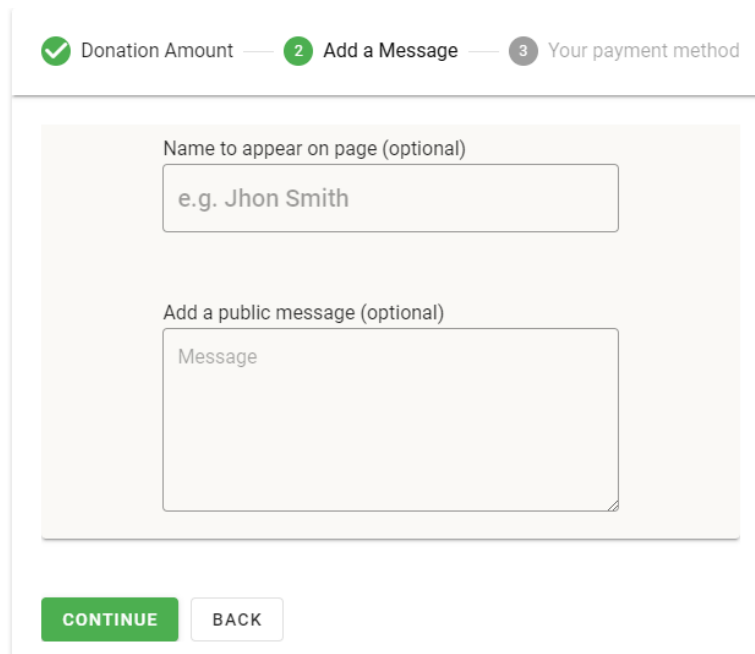
To make a donation click on donate button on any campaign page. It will direct the user to the donation page in figure C-16. Users can select the pre-defined amount or user can type any amount they want in the amount text box. Users also can select whether to make this donation anonymously by checking the “Make donation anonymous” button.



The screenshot shows a donation page with a progress indicator at the top: 1 Donation Amount (active), 2 Add a Message, and 3 Your payment method. Below the progress bar, there are four buttons for pre-defined amounts: 100 LKR, 500 LKR, 1000 LKR, and 5000 LKR. The 100 LKR button is selected. Below these buttons is a text input field containing "LKR 100". Underneath the input field is a checkbox labeled "Make Donation Anonymous" which is currently unchecked. At the bottom of the form are two buttons: "CONTINUE" (green) and "CANCEL" (white).

Figure C.16: Donation page

Once the user enters the donation amount user can enter a public message. See figure C-17.



The screenshot shows the next step in the donation process. The progress indicator at the top is: 1 Donation Amount (checked), 2 Add a Message (active), and 3 Your payment method. The main content area has two sections. The first is "Name to appear on page (optional)" with a text input field containing "e.g. Jhon Smith". The second is "Add a public message (optional)" with a larger text area containing the placeholder "Message". At the bottom of the form are two buttons: "CONTINUE" (green) and "BACK" (white).

Figure C.17: Donation message

Once the user adds a message they will direct to select the payment method. Currently, the system supports only credit/debit card transactions.

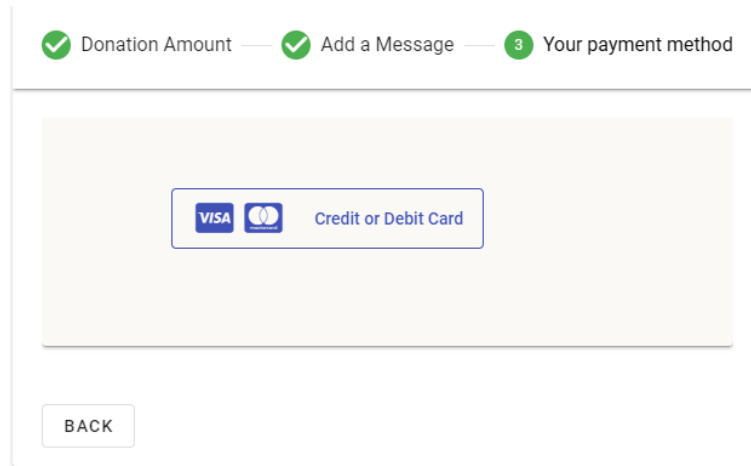


Figure C.18: Payment method selection

Once users select the payment method (Credit or Debit card) they will direct to the paystack payment gateway.

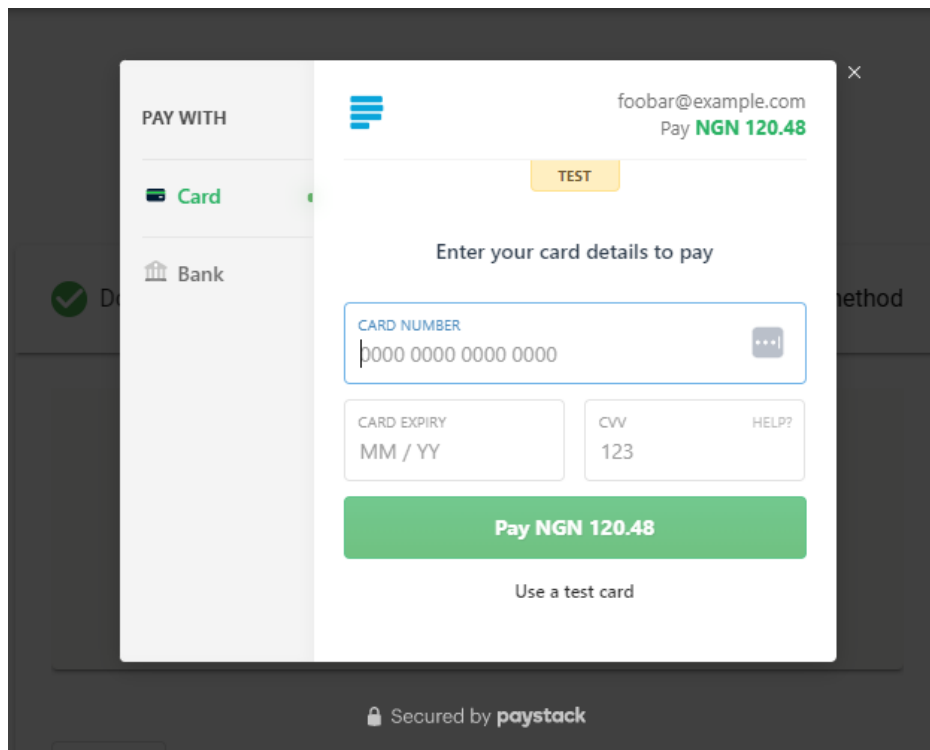


Figure C.19: Paystack payment gateway

## Add a campaign to the watchlist

A watchlist is a collection or a list of items that users need to keep track of. Any campaign can be added to the user's watchlist and accessed later. To view the watchlist select user's name in the navigation panel in select the Watchlist button. It will direct the user to their watchlist page (see figure C.20)

# Watchlist

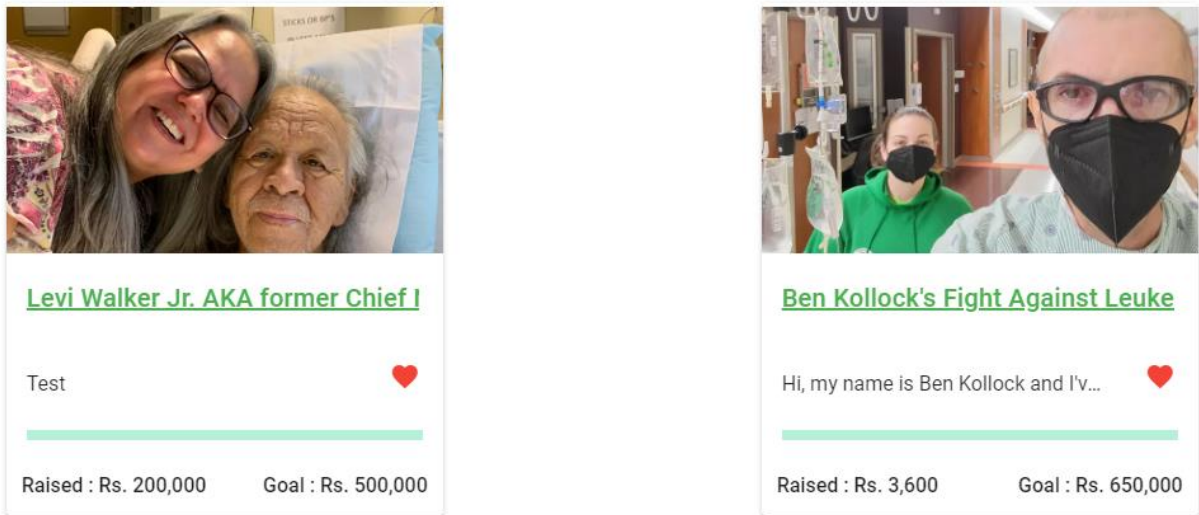


Figure C.20: Users watchlist

To add a campaign to the watchlist user has to click on the grey color heart-shaped icon in the campaign tile. If the campaign is added to the watchlist heart-shaped icon turns to red color.

# Appendix D - Management Reports

Management Information System Reports often simply referred to as MIS reports are the key resources used by the management of the company or an organization for faster decision-making.

## Types of MIS reports

There are many types of MIS reports are available for users. A few of the important report types are mentioned below.

- 1) Accounting & Financial Reports
- 2) Sales Reports
- 3) Budget Reports
- 4) Production Reports
- 5) Summary Reports
- 6) Trend Reports
- 7) Inventory Reports

## Components of the MIS Reports

MIS reports are not a standalone entity. It has many components which interact together to provide meaningful output.

- People – people who use the information system
- Data – the data that the information system records
- Business Procedures – procedures put in place on how to record, store and analyze data
- Hardware – these include servers, workstations, networking equipment, printers, etc.
- Software – these are programs used to handle the data. These include programs such as spreadsheet programs, database software, etc.

## Uses of MIS reports

The main purpose of MIS reports is to facilitate information for the decision-makers to help for effective and faster decision-making. Apart from that MIS reports help to

- Employees within the organization have easy access to information but for outsiders, we need to provide only the necessary information. Using MIS reports we can easily facilitate communication within and outside the organization.
- Record keeping.

The Donation management system provides a wide variety of reports to its users. Basically, reports are categorized under Administration reports and users' reports. Each report can be downloaded in both excel and PDF formats.

## Administration Reports

Administration reports are available in the administration dashboard. Once the administrative officer logs on to the system, they are redirected to the admin dashboard. Here admin can get a quick glance at the entire summary of the system.

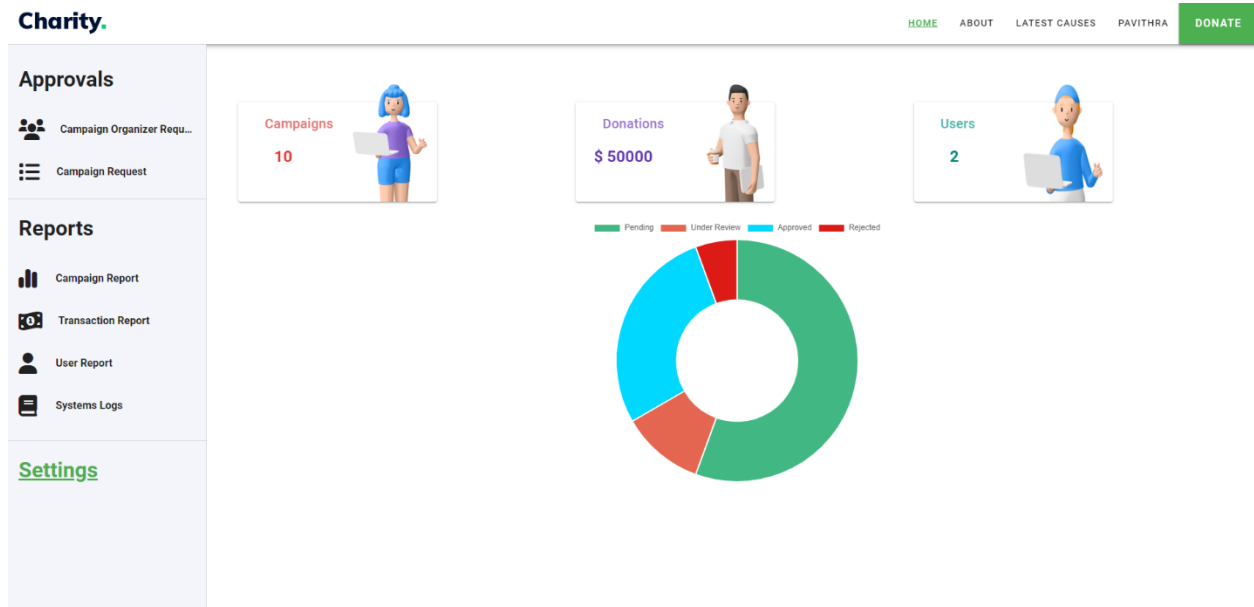


Figure D.1: Administration dashboard

Following are the reports provided by the system for administration officers.

### 1) Campaign Summary Report

The campaign summary report will display a summary of all the campaigns. From here admin can get an idea of

- What are the current running campaigns?
- Which campaigns are pending approval?
- No of donations each campaign received.
- Raised amount vs. goal amount for each campaign.
- Amount of donation each campaign received.
- Status of each campaign etc.

Note -: By clicking the action button admin can navigate to the respective campaign page to view more details of the campaign.

Admin can use the following filters to filter the output

- Campaign Status
- Campaign start date



- Campaign end date

The screenshot shows a web interface for a charity organization. At the top right, there are navigation links: HOME, ABOUT, LATEST CAUSES, PAVITHRA, and a prominent DONATE button. On the left, a sidebar contains 'Approvals' (with sub-items for Campaign Organizer Requests and Campaign Requests) and 'Reports' (with sub-items for Campaign Report, Transaction Report, User Report, and Systems Logs). Below the sidebar is a 'Settings' link. The main content area features a 'Campaign Report' table with filters for Campaign Status (set to 'All'), Campaign Start Date (2022-08-01), and Campaign End Date (2022-11-30). A 'FILTER' button is present. The table has columns for Campaign ID, Campaign Name, Start Date, End Date, Goal, Raised Amount, No of Donations, Campaign Status, Created By, and Created Date. Two rows of data are visible:

Campaign ID	Campaign Name	Start Date	End Date	Goal	Raised Amount	No of Donations	Campaign Status	Created By	Created Date	Actions
xtM2nVtWxsE1Zj6mx0uC	Ben Kollock's Fight Against Leukemia	2022-08-27	2022-09-30	650000	3500	0	Request	IqbyRwiefJjLEv8Chj	2022-08-27	ⓘ
2GDNh8rap1s76uHmFbh4	Levi Walker Jr. AKA former Chief Noco-a-Homa	2022-08-28	2022-08-31	500000	20000	0	Approved	IqbyRwiefJjLEv8Chj	2022-08-28	ⓘ

At the bottom of the table, it indicates 'Rows per page: 10' and '1-2 of 2' with navigation arrows.

Figure D.2: Campaign Report

## 2) Donation Report (Transaction Report)

In donation, report will display the summary of all the donations made by the donor using the payment gateway. Using this report admin can

- View all transactions made by donors for a given time period.
- Status of each transaction and their payment types.

Admin can use the following filters to filter the output

- Payment Status
- Payment from date
- Payment to date

The screenshot displays the 'Donation Transactions' report in a web application. The top navigation bar includes 'HOME', 'ABOUT', 'LATEST CAUSES', 'PAVITHRA', and a green 'DONATE' button. The left sidebar has 'Approvals' (with 'Campaign Organizer Requ...' and 'Campaign Request') and 'Reports' (with 'Campaign Report', 'Transaction Report', 'User Report', and 'Systems Logs'). Below the sidebar is a 'Settings' link. The main content area shows filters for 'Payment Status' (All), 'Payment From Date' (2022-08-01), and 'Payment To Date' (2022-11-30), with a 'FILTER' button. Below the filters are CSV and PDF download icons. The report title is 'Donation Transactions' with a search bar. The table below has the following data:

Donor Name	Donated Date	Amount	Payment Type	Payment Status	Actions
Saviru Rajapakse	2022-08-06	5000	Credit Card	Paid	👁️

At the bottom right of the table, there is a 'Rows per page' dropdown set to 10 and a pagination indicator '1-1 of 1' with navigation arrows.

Figure D.3: Donation transaction Report

### 3) User Report

This report will display a summary of all users who are using the system. Using this report admin can

- View user information.
- View user profile and their behavior.

Note -: By clicking on the action button admin can navigate to the respective user's profile to view more details about the user.

Admin can use the following filters to filter the output

- User role
- Account Status
- User-created from date
- User-created to date

**Charity.** [HOME](#) [ABOUT](#) [LATEST CAUSES](#) [PAVITHRA](#) [DONATE](#)

**Approvals**

- Campaign Organizer Requ...
- Campaign Request

**Reports**

- Campaign Report
- Transaction Report
- User Report
- Systems Logs

[Settings](#)

User Role: All Account Status: All Created From Date: 2022-11-01 Created To Date: 2022-11-30 **FILTER**

**User Report**

Name	Email	Address	City	Phone No	User Role	User Status	Account Status	Created Date	Actions
Pavithra	pavidsscst@gmail.com	65/A, Bathiya Mawatha, Pamunuwilla, Gonawala (WP)	Kandy	0715918243	Campaign Manager	Active	Verified	06-09-2021	👁️
Saviru	test@gmail.com				Administrator	Block	Verified	[object Object]	👁️

Rows per page: 10 1-2 of 2

Figure D.4: User Report

#### 4) System Log Report

For each user activity, a log entry will be recorded. Using the system log report admin can view those user activities.

Admin can use the following filters to filter the output

- Log type
- Log created from date
- Log created to date

**Charity.** [HOME](#) [ABOUT](#) [DASHBOARD](#)

**Settings**

- Donation Rules
- Contribution Levels
- Donation Badges

**Logs**

- System Logs

[Reports](#)

Log type: All Campaign Start Date: 2022-11-01 Campaign End Date: 2022-11-30 **FILTER**

**System Logs**

Level	Message	Method	Time Stamp
error	User lqblyRwleyLJL6V8Chj Found Successfully	/api/user/getUser	2022-11-19T08:12:32.570Z
info	User Authenticated Successfully	/api/user/authenticate	2022-11-19T08:05:04.526Z
info	User Authenticated Successfully	/api/user/authenticate	2022-11-19T08:04:28.868Z

Rows per page: 10 1-3 of 3

Figure D.5: System Log report

## **Campaign Organizer's Report**

### 1) Transaction Report

In the transaction report, the campaign organizer can view how well the campaign is performing. (See figure D.3)

### 2) Campaign Report

In the campaign report, the organizer can view how well the campaigns are performing. Here he/she can get the following information.

- Details of each campaign.
- No of donations each campaign received.
- Amount of donation each campaign received.
- Raised amount vs. goal amount for each campaign.

By clicking the action button campaign organizer can navigate to the campaign page.