

# PalestineCharityFinalReport1.d OCX

*by* Mohammad Shair

---

**Submission date:** 23-Jun-2024 02:20AM (UTC+0300)

**Submission ID:** 2405739289

**File name:** PalestineCharityFinalReport1.docx (6.3M)

**Word count:** 9586

**Character count:** 53647



**Faculty of Engineering and Technology  
Department of Computer Science  
COMP4300- Graduation Project**

**Second Semester**

**2023/2024**



**Palestine Charity**

---

**Group Members with IDs:**

Mohammad Shair 1200319

Iman Salameh 1201786

Naghm Taweelel 1200715

**Supervisor:**

Dr. Mamoun Nawahdah

## **Section – B**

**Title of Project:** Palestine Charity

**Project No:** 19

**Supervisor:** Dr. Mamoun Nawahdah

**Key Areas:** Charity, Gamification, Philanthropy, Web development, RWD  
(Responsive Web Design)

## **Abstract**

We are building a philanthropy website named Palestine Charity. The website shall give users around the world the ability to donate to trusted Palestinian charitable organizations. Our system would use gamification methods to maintain user engagement. Some gamification strategies include badges, virtual currencies, daily and weekly goals, and leaderboards.

In the last decade, more people around the world started recognizing the atrocities that the Palestinians go through, and after observing different social media sites, I found a common issue facing those who want to help. The issue is not having a trusted organization to donate to. Palestine Charity is made to solve this issue by having only trusted organizations on our website.

After observing other philanthropy websites, we noticed a lack of engagement between the users and the websites, so we decided to add gamification strategies to build engagement between the users and the website. Also to add a soul to our website, we are implementing some community features such as a news dashboard for each campaign and sub-campaigns where influencers can mirror existing campaigns, add some specialized content, and have a separate leaderboard.

Palestine Charity is developed so people around the world can show their support to the Palestinian needs.

## Table of Contents

## Table of Contents

<b>Abstract .....</b>	<b>3</b>
<b>Table of Contents .....</b>	<b>4</b>
<b>Acknowledgements .....</b>	<b>Error! Bookmark not defined.</b>
<b>Chapter 1: Introduction .....</b>	<b>9</b>
<b>1.1 Aim And Objectives .....</b>	<b>9</b>
<b>1.2 Introduction And Motivation .....</b>	<b>9</b>
<b>1.3 Methodology.....</b>	<b>10</b>
<b>1.4 Overview Of The Technical Area .....</b>	<b>10</b>
<b>1.5 Overview Of The Report.....</b>	<b>16</b>
<b>Chapter 2: Background.....</b>	<b>17</b>
<b>2.1 Related Papers: .....</b>	<b>17</b>
<b>Chapter 3: System Analysis .....</b>	<b>25</b>
<b>3.1 Product Description.....</b>	<b>25</b>
<b>3.1.1 System Objectives .....</b>	<b>25</b>
<b>3.1.2 System Main Features .....</b>	<b>25</b>
<b>3.1.3 Operating Environments .....</b>	<b>25</b>
<b>3.1.4 Constraints .....</b>	<b>26</b>
<b>3.1.5 Functional Requirements .....</b>	<b>26</b>
<b>3.1.6 Non-Functional Requirements .....</b>	<b>28</b>
<b>3.2 Functional Decomposition (Use Case Diagram).....</b>	<b>29</b>
<b>External Actors (Services): .....</b>	<b>31</b>
<b>3.2.2 Use Cases .....</b>	<b>31</b>
<b>3.2.3 Use Cases Diagram .....</b>	<b>32</b>
<b>3.3 System Models .....</b>	<b>33</b>
<b>3.3.2 Sequence Diagram .....</b>	<b>34</b>
<b>3.3.3 Activity Diagram .....</b>	<b>39</b>
<b>3.3.4 State Chart Diagram .....</b>	<b>41</b>
<b>3.4 System Architecture .....</b>	<b>43</b>
<b>3.4.2 Deployment Diagram .....</b>	<b>44</b>
<b>3.5 Data Management and Models.....</b>	<b>45</b>
<b>Chapter 4: Implementation and Testing: .....</b>	<b>46</b>
<b>4.1 Implementation:.....</b>	<b>46</b>
<b>4.2 Testing:.....</b>	<b>53</b>
<b>Chapter 5: Conclusion and Future Works.....</b>	<b>55</b>
<b>5.1 Review Of The Project .....</b>	<b>55</b>
<b>5.2 Future Works.....</b>	<b>55</b>
<b>Bibliography .....</b>	<b>56</b>
<b>Chapter6: Appendices .....</b>	<b>57</b>

## ***List of Tables/Figures***

Figure 1 Angular vs other frameworks.....	11
Figure 2 ExpressJs vs other backend Framework .....	15
Figure 3 Philanthropy Connections .....	18
Figure 4 Tiltify .....	18
Figure 5 Donorsee campaigns page .....	19
Figure 6 kiva .....	19
Figure 7 ShareTheMeal .....	20
Figure 8 BeanBeanBean .....	21
Figure 9 Freerice .....	21
Figure 10 benevity .....	22
Figure 11 comparasion between charity websites .....	23
Figure 12 Steam profile customization .....	23
Figure 13 Steam profile shop .....	23
Figure 14 steam badges .....	24
Figure 15 twitch leaderboard.....	24
Figure 16 Website Actors .....	31
Figure 17 UseCase Diagram.....	32
Figure 18 Class Diagram .....	33
Figure 19 Regestration Sequanse Diagram .....	34
Figure 20 Request Campaign Sequanse Diagram .....	35
Figure 21Donating Process Sequence Diagram .....	36
Figure 22 Sub-Campaign Creation Sequanse Diagram .....	37
Figure 23 Updating News Dashboard Sequence Diagram .....	38
Figure 24 Donation Process Activity Diagram .....	39
Figure 25 Creating Sub-Campaign Activity Diagram .....	40
Figure 26 Request To make A Campaign State Chart.....	41
Figure 27Buying Item State chart.....	42
Figure 28 Software Architecture .....	43
Figure 29 Deployment Diagram.....	44
Figure 30 Er-Diagram .....	45
Figure 31 landing page .....	46
Figure 32 log in form.....	47
Figure 33 side nav .....	47
Figure 34 user profile page .....	48
Figure 35 Customize profile .....	48
Figure 36 campaign .....	49
Figure 37 donation form .....	49
Figure 38 donation chart.....	50
Figure 39 leaderboard.....	50
Figure 40 Campaign NewsBoard .....	50
Figure 41 bookmarks page .....	51
Figure 42 shop .....	51
Figure 43 Inventory .....	52
Figure 44 API testing .....	53
Figure 45 cypress test .....	54
Figure 46 Donating To Campaign usecase .....	58
Figure 47 Campaign Creation UseCase .....	59
Figure 48 Profile Customization UseCase .....	60
Figure 49 Using Shop UseCase .....	61
Figure 50 Posting Campaign News UseCase .....	61

Figure 51 Sub-Campaign Creation UseCase .....62

## Acknowledgments

1

At first, we would like to thank our supervisor Dr. Mamoun Nawahda for his patient guidance and dedicated involvement throughout the process.

Our appreciation also extends to the instructors and the doctors, this couldn't been accomplished without their efforts during the previous years.

Most importantly, we most express our deepest gratitude to our parents and families for their continued encouragement and endless support.

# Chapter 1: Introduction

In this chapter, we will introduce the motivation of our “Palestine charity project”, Gamification features, as well as our aims and objectives.

27

## 1.1 Aim And Objectives

### 1.1.1 The Palestine Charity Project Aims To:

Create an interactive, captivating, philanthropic platform to transform the world to be more empathetic and connected while helping users make some meaningful contributions.

### 1.1.2 Objectives:

- To create a platform that draws users in to easily navigate and create their fundraising campaigns. Sharing their passions and interests in the form of philanthropy.
- To add gamification elements from successful platforms to increase the level of user engagement.
- Create a more community feel by building in leaderboards, badges, and social sharing that seem to have become ubiquitous now.
- To offer users a diverse catalog of causes to support. So the full spectrum of perhaps less popular but equally philanthropic activities.
- Ensuring transparency and showing the true reach of campaigns – regularly updating progress and results beyond what is typically stated to donors and contributors.
- Creating a digital currency or points system where actions or contributions are rewarded virtually, incentivizing supporters to keep engaging with the cause.

## 1.2 Introduction And Motivation

The 'Palestine Charity' project is like a new adventure in helping others. Think about helping people by giving money, but sometimes it's not easy to stay interested for a long time. The usual ways of doing this may not always keep people excited. People might stop giving money because they don't feel a strong connection to the cause, and they don't get updates on how their help is making a difference. That's where our special project comes in!

Our main goal is to prevent people from giving less to charities because they lose interest. With the rise of digital platforms and the use of engaging elements like leaderboards and badges, we see a great opportunity to change how people experience being charitable. The usual ways charities ask for help often don't show donors how their contributions make a real impact in real time, and that makes them feel disconnected. So, by adding in some fun features inspired by what works well in other apps, the "Palestine Charity" project wants to make giving more exciting and involving. This way, users not only help out financially but also feel a continuous and deep connection to the causes they care about.

### 1.3 Methodology

In developing and implementing the "Palestine Charity", our team will use the Waterfall methodology. The Waterfall methodology is a structured and linear approach that requires completing each phase of a project before the next phase can begin.

We chose the waterfall methodology based on our project requirements, which are simple and well-defined. The first interaction with the website is very serious, so we want the user to have the full experience when using the website.

## 1.4 Overview Of The Technical Area

### 1.4.1 Frontend

#### 1.4.1.1 Angular

Angular [1] is a development platform and application design framework for building smart and effective single-page applications.

**Angular as a platform consists of:**

- A framework based on components for creating scalable web applications.
- a group of well-integrated libraries covering many different functions, such as forms, management, routing, client-server communication, and much more.
- a group of tools for developers to aid in the creation, testing, and updating of your code.

**What does SPA (single-page application) mean?**

A single web page is loaded by an SPA (single-page application) [2], which then uses JavaScript APIs like Fetch to update the body content of that single document when new content needs to be displayed.

As a result, users can access websites without having to load entire new pages from the server. This can lead to improvements in performance and a more dynamic user experience, but there are some drawbacks as well, like increased work for SEO, maintaining state, implementing navigation, and conducting accurate performance monitoring.

Angular has a solution for all the drawbacks above.

### **Angular vs other Frameworks/libraries:**

Criteria	Angular	React	Vue
Type	Framework	Library	Framework
Language	TypeScript	JavaScript/TypeScript	JavaScript/TypeScript
Learning curve	Steep	Moderate	Easy
Architecture	Component-Based, MVVM	Component-Based, functional programming	Component-Based, MVVM
Ecosystem/Tooling	Rich, Integrated tools	Large, not Integrated	Large, Integrated
Scalability	Excellent	Good	Good
Community Support	Good	Very Good	Good
Built-in Features	Comprehensive	Minimal	Moderate
Development Approach	Opinionated	Flexible	Opinionated but flexible
Data Binding	Two way binding	One way	One way binding
Use Case	Large projects	Flexible	Small to medium

*Figure 1 Angular vs other frameworks*

#### **18 1.4.1.2 Tailwind CSS**

Tailwind CSS [3] is a utility-first CSS framework for rapidly building custom user interfaces. The framework [12]ers from traditional CSS frameworks like Bootstrap or Foundation, which provide pre-styled components, in that it offers low-level utility classes that let you build completely custom designs without ever leaving your HTML. This approach encourages a more component-driven design, which allows you to update and customize styles more easily by keeping your prototyping directly in your markup.

## Key Features of Tailwind CSS

- Utility-First: Tailwind uses utility classes to style elements, providing these classes you can use to build your designs directly into your HTML. This approach minimizes the amount of custom CSS you need to write and helps to speed up the development process.
- Responsiveness: Every utility in the framework is also supported by a responsive variant, making it especially easy to build designs that respond to different screen sizes just by swapping out classes with a different prefix, they are done with responsive modifiers.
- Customization: Tailwind is quite customizable. Through its configuration file, tailwind.config.js, you can set things such as your design system's colors, fonts, and breakpoints, ensuring that your utility classes will suit your project's design constraints.
- Performance: Tailwind can lead to much smaller, more efficient stylesheets because you only use what you need for a given project (Tree shaking). When used in conjunction with tools such as PurgeCSS it can be used to completely remove the aforementioned unused styles resulting in a final CSS bundle that's much, much smaller than it would have been without it.

### 1.4.1.3 SCSS

6

SCSS [4] is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets (CSS). It provides the functionality of CSS in addition to some great features that make your stylesheets more readable, maintainable and allows you to write DRY code. SCSS provides a CSS-compatible syntax, which means that every valid CSS stylesheet is a valid SCSS file with the same meaning.

## Key Features of SCSS

- Variables: Variables allow you to store things like colors, fonts, or any CSS value for reuse throughout your stylesheet. This ensures consistent use, more maintainability and makes your code easier to manage.
- Nesting: You can nest your CSS selectors in a way that follows the same visual hierarchy of your HTML, making your code cleaner and more readable.
- Partials and Imports: SCSS allows you to write your CSS in separate "partial" files that you can then import as a single file. This is a good way to organise your CSS code into smaller, more manageable chunks.
- Mixins: Mixins are a group of CSS declarations that you want to reuse throughout your site. A mixin can be included in other selectors and it can also take arguments; making your code DRY so that you do not repeat yourself.

- Functions: There are a lot of functions in SCSS for things like color manipulation, numerical operations and more, and you are also able to write your own functions.
- Operators: SCSS comes with a lot of useful operators you can perform mathematical operations directly within your SCSS files, this allows values to calculate dynamically in your styles.
- Control Directives: SCSS also includes a variety of handy control directives such as conditionals and loops for creating more powerful, complex styles in a simple, programmatic way.

#### 1.4.1.4 Ant design:

Ant design is a components framework that follows Google material items, this framework was used to help develop the project faster with premade components.

#### 1.4.1.5 ChartJs:

ChartJs is a well-known JavaScript library that helps build different charts with rich features.

### 1.4.2 Backend

#### 1.4.2.1 Expressjs:

Express.js [5], or simply Express, is a minimal Node.js web application framework ideal for building web applications and APIs. It is known for its simplicity, flexibility and scalability in building web applications, making it one of the most popular server-side development frameworks in the JavaScript ecosystem. Express provides a small layer of the core capabilities of web applications without covering the capabilities of Node.js itself.

Key features of Express.js:

- Simplicity - Express.js adheres to the phrase "minimalist"; an approach that provides the necessary tools to build web applications without strict conventions. Therefore, developers can structure their applications however they like.
- Middleware - Middlewares are essentially middleware function calls. Middleware functions have access to request and response objects and the next middleware function. They can execute code, make changes to the request and response objects, complete the request-response cycle, and call the next middleware function. This allows for a very flexible and modular design that can be easily extended with additional features.
- Routing - Express comes with a robust routing API that allows developers to specify routes based on HTTP methods and URIs. This feature simplifies the creation of RESTful web services and applications so that developers can easily manage incoming requests from clients such as browsers or other web services.

- Efficiency - Because Express is based on Node.js, it inherits its efficient, non-blocking I/O model, so it can serve multiple simultaneous connections with minimal overhead, making it very efficient and suitable for real-time web applications, such as chat applications or web APIs.
- Community Support and Ecosystem - As part of the Node.js community, Express has several modules and tools available that can be added to an application via npm, allowing many features to be added to the application. A large community provides a variety of resources, tutorials, and support. Compatibility
- Express can be integrated with many other frameworks and libraries in the JavaScript ecosystem, such as Angular (a front-end framework). This can be especially useful for building full-stack JavaScript applications, where developers can use JavaScript on both the server and client side.

### **Express.js vs other frameworks:**

Feature	Express.js	NET (ASP.NET Core)	Django
Language	JavaScript/Node.js	C#, F#, or Visual Basic	Python
Performance	High performance	High performance	Good
Scalability	Highly scalable with Node.js event-driven model	Scalable, benefits from strong typing and runtime	Scalable, with built-in features to support growth
Ease of Development	Minimalist, unopinionated, flexible	Opinionated	Opinionated
Learning Curve	Relatively low for JavaScript developers	Steeper for those unfamiliar with C# or .NET ecosystem	Moderate, Python is easy to learn, Django has more built-in features
Community and Ecosystem	Large, vibrant with numerous packages on npm	Large	Large

Use Case Flexibility	Great for building RESTful APIs, single-page applications, real-time applications	Suited for enterprise-level applications, microservices, web applications	Ideal for web applications
Development Speed	Fast development cycles due to JavaScript and npm ecosystem	Rapid development with .NET CLI and Visual Studio	Rapid development with batteries-included approach
Cross-Platform Development	Fully cross-platform with Node.js	Cross-platform with .NET Core	Cross-platform

Figure 2 ExpressJs vs other backend Framework

#### 1.4.2.2 Nodemailer:

Nodemailer is a well-known npm package that provide a mailing service so we can send emails to our users.

#### 1.4.2.3 Cloudinary:

Cloudinary is a web service that provides free cloud storage. We used clouddinary to save the users profile pictures

### 1.4.3 Database

#### 1.4.3.1 MongoDB:

5 MongoDB is an open-source, non-relational database management system that uses flexible documents instead of tables and rows to process and store various forms of data, it stores its data in JSON format rather than a relational format that you'd find in SQL database. This makes MongoDB flexible because data can be stored in a hierarchical format within the JSON document, making retrieving records easy and fast.

2 MongoDB cloud services consist of a comprehensive suite of data products that accelerate and simplify how you build with data for any application. With Atlas Database (the Database-as-a-Service for MongoDB), Search, and Data Federation, you can serve any class of workload through a common API.

#### **1.4.4 Testing**

##### **1.4.4.1 Jest:**

Jest is a JavaScript testing framework that provides an easy way to write unit tests, we used this framework to test backend files.

##### **1.4.4.2 Jasmine:**

Jest is a JavaScript testing framework that is provided and recommended by angular to write unit tests, we used this framework on testing frontend components and services.

##### **1.4.4.3 Karma:**

Karma is a test runner that uses the browser to display the test result elegantly, we used karma to display the frontend tests.

##### **1.4.4.4 Cypress:**

Cypress is a test tool that is recommended by angular, it provides end-to-end testing. We used Cypress to test our project functionalities.

#### **1.4.5 IDE**

We will use Webstorm for developing this project. It is widely supported in web development.

Webstorm is an IDE developed by JetBrains, the same publisher of IntelliJ.

### **1.5 Overview Of The Report**

The report will be organized into several chapters, each covering different aspects of the project.

#### **The chapters will include**

##### ***Chapter 1***

Introduction (current chapter).

##### ***Chapter 2***

Background.

##### ***Chapter 3***

System Analysis.

##### ***Chapter 4***

Conclusion and Future works.

## Chapter 2: Background

In this chapter what was read and reviewed from different papers on charity is discussed, similar apps. The following was found.

### 2.1 Related Papers:

#### 14 2.1.1 Relationship between perceived website design, trust and donation decisions online [6]:

Online fundraising is really important for charities, but we're not sure what makes people donate online. We wanted to know how perceptions of website design (like content, usability, and looks) influence the willingness to donate. We also asked questions about demographics (like age and gender) and trust in the organization. We checked 33 health-related websites and used statistics to analyze the data. What we found is that different things matter depending on the situation. When it's about donating your own money, what's on the website and trust in the organization matter the most. But when it's 14 about donating someone else's money, how the website looks is the most important. So, the design of the website and trust in the organization really matter. These differences could be because donating your own money feels more important, and it involves different ways of thinking. Still, more research is needed to understand other factors and how to make online donations in healthcare successful.

#### 20 2.1.2 Applying Gamification Design to a Donation-based Crowdfunding platform for improving user engagement [7]:

The study investigates how gamification might be included in donation-based crowdfunding platforms to improve user engagement. It draws attention to the quick expansion of crowdsourcing and how it helps finance a range of projects. The study presents gamification as a tactic to raise awareness and involvement in donation-based crowdfunding, acknowledging the difficulties associated with low user participation and engagement. The platform is designed using the MDA framework (Mechanic, Dynamic, Aesthetic), which emphasizes the aesthetic characteristics of user motivation, challenge, sensation, and fellowship and the dynamic features of time pressure, progression type, and feedback system. The study makes use of fundamental gamification elements such as avatars, leaderboards, points, levels, and badges. The evaluation shows that the incorporation of gamification components greatly boosts sustained user involvement in donation-based crowdfunding, based on a questionnaire completed by gamification specialists.

## 2.2 Similar Apps:

### 2.2.1 Philanthropy Connections

Much like 'Palestine Charity,' Philanthropy Connections [8] shares a commitment to philanthropic values and social impact. Both platforms emphasize community engagement and user customization as means to connect with their respective audiences. While 'Palestine Charity' employs gamification for enhanced user engagement, Philanthropy Connections brings direct, impactful support to local communities. Both platforms empower individuals to create positive change, showcasing different but parallel paths to a more compassionate world. Figure 3 shows the campaigns categorization.



Figure 3 Philanthropy Connections

### 2.2.2 Tiltify

Tiltify [9] Impact and 'Palestine Charity' share a dedication to philanthropy and social impact. While 'Palestine Charity' engages users through gamification, Tiltify Impact focuses on streamlining fundraising efforts. Both platforms empower individuals to contribute actively to positive change, emphasizing the transformative potential of collective giving." Figure 4 shows donating screen in tiltify.

The screenshot shows a dark-themed donation form for 'Tiltify Impact Fund'. It includes fields for 'Your Details' (Email, Anonymous), 'Donation Amount (USD)' (\$20.00), and 'Comments?'. On the right, there's a 'Donation Summary' section showing an amount of \$20.00, a fundraiser named 'HassanAbi', and a campaign titled 'PALESTINIAN MEDICAL ASSISTANCE'. There are also buttons for 'Dark mode' and 'Create a matching donation'.

Figure 4 Tiltify

### 2.2.3 DonorSee

DonorSee [10] is an online platform that links individual contributors with targeted nonprofit initiatives. Users can browse and support a variety of initiatives related to clean water, healthcare, and education. Donors get access to project details, beneficiary interactions, and updates on the effects of their contributions. DonorSee differs from "Palestine Charity" in that it places an emphasis on the face-to-face, direct interactions between donors and beneficiaries. Figure 5 shows campaigns screen.

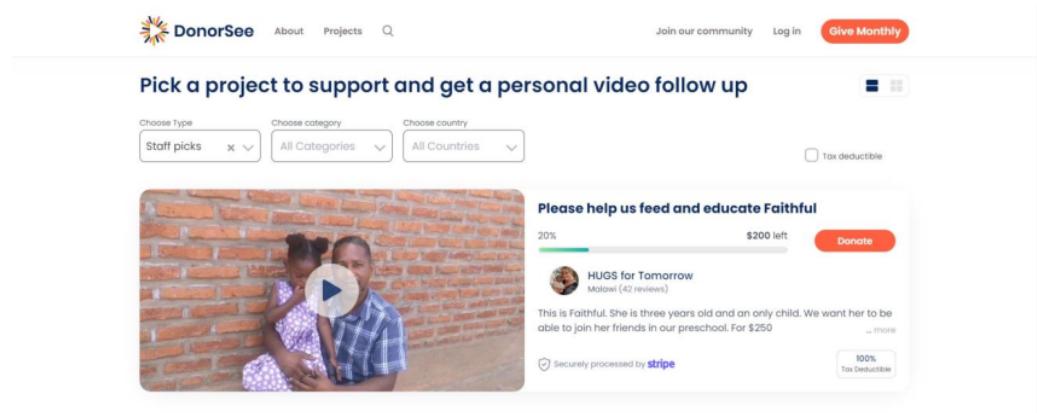


Figure 5 Donorsee campaigns page

### 2.2.4 Kiva

Kiva [11] is a platform that specializes in microfinance and connects lenders with borrowers in underserved communities worldwide. Users can browse profiles of entrepreneurs and small businesses seeking financial support. Lenders can make loans to these borrowers, helping them grow their businesses and improve their livelihoods. Kiva provides a unique approach to philanthropy by enabling individuals to directly impact the economic well-being of others. This differs from 'Palestine Charity,' which emphasizes gamification to engage users in a broader range of charitable activities. Figure 6 shows kiva main page.

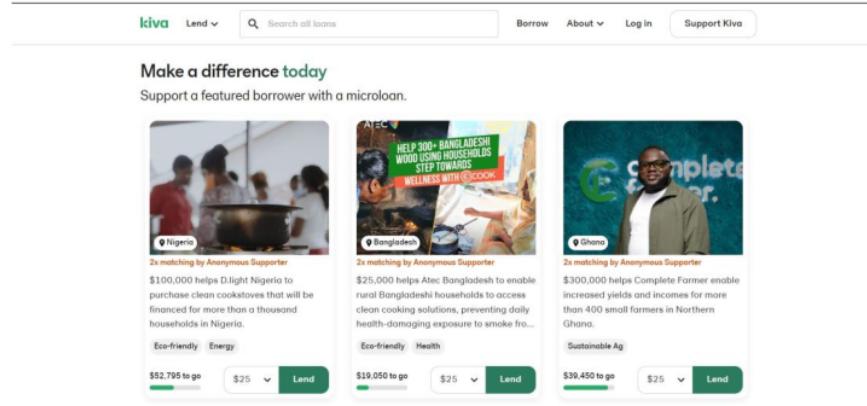


Figure 6 kiva

## 2.2.5 ShareTheMeal

Similarities: ShareTheMeal [12], developed by the United Nations World Food Programme, allows users to share meals with children in need around the world. Users can make small, easy donations directly through the app, contributing to global hunger relief efforts.

Differences: While ShareTheMeal encourages charitable giving, it lacks the gamification elements, personalized profiles, and interactive challenges found in "Palestine Charity." ShareTheMeal is more streamlined, focusing solely on direct donations, without the extensive social engagement and gaming-inspired features seen in the concept you've described. Figure 7 shows the main page of ShareTheMeal website.

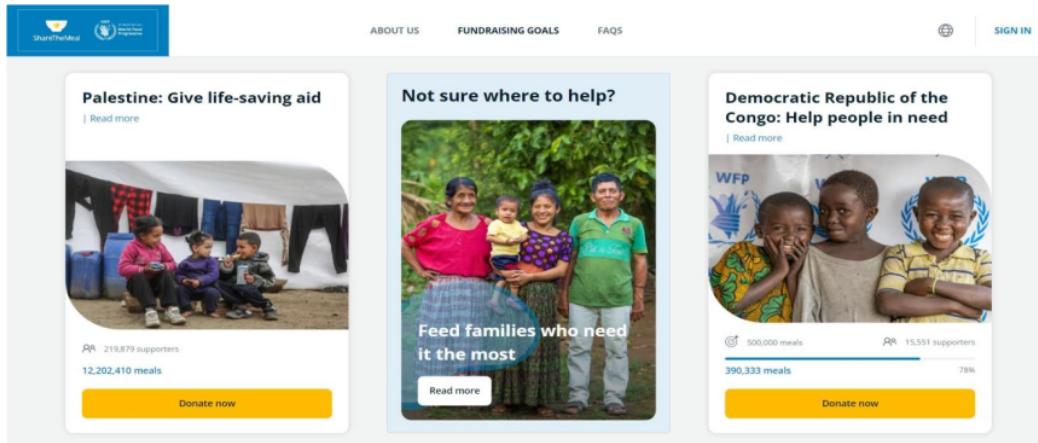


Figure 7 ShareTheMeal

## 2.2.6 BeanBeanBean:

BeanBeanBean [13] is a platform with a distinctive focus on environmental sustainability and tree planting initiatives. Users can explore profiles of various environmental projects and contribute to tree planting efforts worldwide. BeanBeanBean connects environmentally conscious individuals with projects seeking financial support to promote eco-friendly practices. This platform stands out from 'Palestine Charity' as it directs attention toward environmental causes, utilizing gamification to engage users in supporting tree-related activities and fostering a greener planet. While 'Palestine Charity' has a broader range of philanthropic activities, BeanBeanBean offers a specialized approach, allowing users to directly impact environmental well-being through tree planting initiatives. Figure 8 shows BeanBeanBean advertisement.



Figure 8 BeanBeanBean

#### 2.2.7 Freerice:

Palestine Charity and FreeRice [14] share a commitment to philanthropy but differ significantly in their focus areas, incentives, and features. "Palestine Charity" emphasizes a broader spectrum of causes with gamification elements and community engagement, while FreeRice focuses on global hunger relief through educational quizzes. The uniqueness of each app contributes to diverse user experiences, catering to different aspects of charitable giving and social impact. Figure 9 shows the game in freeRise.

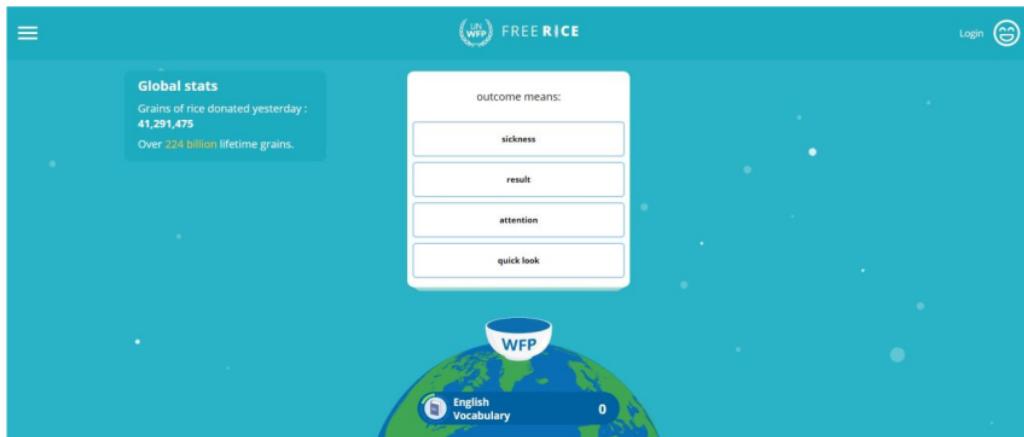


Figure 9 Freerice

#### 2.2.8 Benevity:

**Similarities:** Benevity [15] is a corporate social responsibility and workplace giving platform. Like "Palestine Charity," it connects employees with charitable causes, allowing them to donate, volunteer, and engage in corporate philanthropy.

**Differences:** While both platforms facilitate corporate giving, Benevity is tailored for workplace environments and lacks the direct user-driven gamification features found in "Palestine Charity." Benevity focuses on integrating corporate giving seamlessly into the workplace, offering a range of giving options and employee engagement tools. Figure 10 shows Benevity main page.

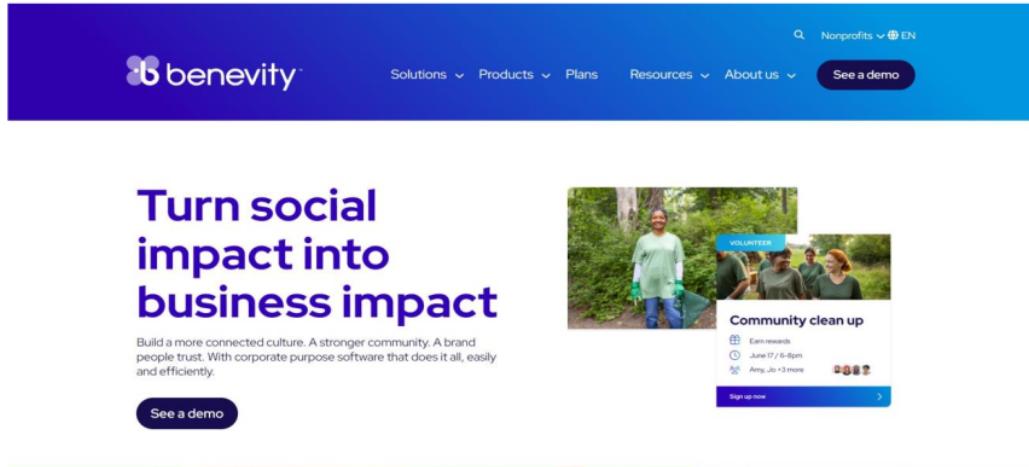


Figure 10 benevity

### Palestine charity vs other charity websites:

Feature / Platform	Gamification	Rich community features	Only Trusted Campaigns	Campaigns Diversity
Palestine Charity	✓	✓	✓	✓
Philanthropy Connections	✗	✓	✓	✓
Tiltify	✓	✓	✗	✓
DonorSee	✗	✓	✗	✓
Kiva	✗	✗	✗	✓

ShareTheMeal	✗	✗	✓	✗
BeanBeanBean	✓	✗	✓	✗
FreeRice	✓	✗	✓	✗
Benefity	✗	✓	✓	✓

Figure 11 comparasion between charity websites

## 2.3 Projects That Got Us Inspired:

### 2.3.1 Steam (Customization and Tokens):

- User Profiles and Customization: Allow users to create profiles with personal avatars like figure12 connected to the “gamer flair” theme and customization options. Users can select profile backgrounds and themes and express themselves in the app.



Figure 12 Steam profile customization

- Tokens: Implementing a virtual currency system within our charity app. Users can earn this virtual currency, which they can redeem for in-app rewards or recognition. Take a look at how Steam utilizes this concept in figure 13.

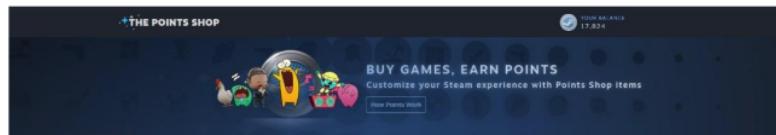


Figure 13 Steam profile shop

- Badges and Emblems: Create a system of badges or emblems (like figure 14) that users can earn for hitting different charitable milestones. For example, earning a badge for the first \$100 donated, a poster for volunteering five times in one month, or a profile picture border for giving to a variety of causes.

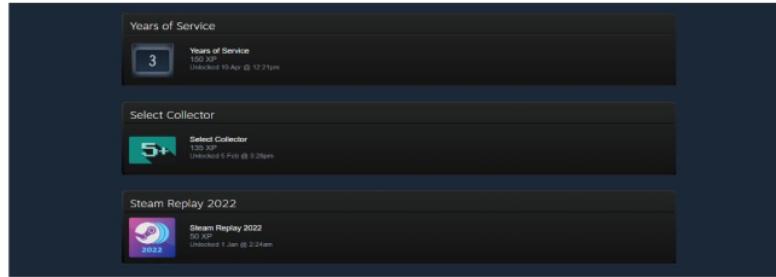


Figure 14 steam badges

### 2.3.2 Twitch (Leaderboards and Social Interaction):

- Leaderboards: Develop leaderboards (like figure 15) to showcase top contributors and fundraisers within your charity app. Public recognition can incentivize healthy competition and keep users motivated to continue their support.

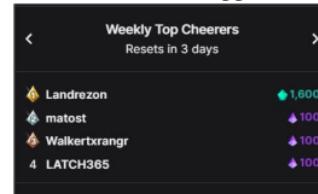


Figure 15 twitch leaderboard

- Daily and Weekly Challenges: It takes time to get gamers to consistently use your app. They are after all looking for that next level, weapon, or unlock. Offering daily and weekly charitable challenges can let users know a new challenge awaits each day and their regular usage will earn them special recognition. Partially hidden leaderboards can also encourage them to use the app more frequently, to find out how they're stacking up. The end result of all of this activity is a user that will overcome the biggest obstacle in this space, which is keeping your app on a user's home screen and regularly getting their attention.
- Social Sharing and Community: Gamers are not introverts, the stereotype couldn't be further from the truth. Allow them to share their in-app activities and achievements on social media and inspire their family or friends to join the cause. Further build community through enabling users to form teams or groups within your app. Teams of like-minded gamers can form and tackle causes they are most passionate about together.

# **Chapter 3: System Analysis**

This chapter introduces the main software engineering requirements, diagrams, and system components. Each of these components has a vital role in the functionality of the project.

## **3.1 Product Description**

In a world of combining gaming platforms and charity websites, Palestine Charity provides a path of interactivity, engagement, and empowerment that enables people to actively make a greater, more compassionate, and connected world

### **3.1.1 System Objectives**

- Help people to donate simply and easily.
- Connect people who wanna Palestine, so they can support each other and work together.
- Work with influencers to encourage people and decrease the spirit of competition to donate
- Make users interested and excited to donate by making it fun and rewarding using gamification.
- Track the donation process to know the stage it has reached.

### **3.1.2 System Main Features**

- Donating process in an easy and friendly way.
- Request to make a campaign.
- Customize the profile and shopping
- Allow different organizations to request to make a campaign
- Get badges and earn points for being generous
- Dashboard to explain the campaigns for the donors.
- Letting people create sub-campaigns, so they can support, and encourage each other and work together.

### **3.1.3 Operating Environments**

The system is designed to operate in a web-based environment that can be accessed through web browsers. You can work on the system through web browsers on mobile phones.

### **3.1.4 Constraints**

#### **◆ Internet Limitations**

Users need to be connected to the Internet to access and use the website effectively.

#### **◆ Device Compatibility**

The system must be available and work across various devices (smartphones, laptops and tablets).

#### **◆ Data Entry Limitations**

All required data (such as Donors, Organizations, and Influencers details, etc.) must be entered into the system correctly.

#### **◆ Privacy restrictions**

The system must comply with privacy laws and maintain the confidentiality of user data (especially the financial ones like visa).

#### **◆ User Credibility and Trust**

Building trust and credibility between donors and organizations.

### **3.1.5 Functional Requirements**

#### **3.1.5.1 Signing up**

**UR01:** User signs up for the first time

**SR1.** The system shall provide a user registration form. It shall include the user's name, first name, last name, email address, and password.

**SR2.** It shall validate that the inputs match the required format and enforce the field necessary.

**SR3.** The system shall display appropriate error messages if there are any issues with the registration form.

**SR4.** Upon successful registration, the system shall send a confirmation email to the user's provided email address.

### 3.1.5.2 Log In

**UR02:** User logs into the web page:

**SR1.** The system shall provide a user login form. It shall include the user's name and password.

**SR2.** The system shall provide the user with the option "Remember me" to keep him logged in upon reopening the web page.

### 3.1.5.3 Donating To Campaign

**UR03:** User want to donate a campaign:

**SR1.** The system shall allow users to browse and select campaigns based on their interests.

**SR2.** The system shall display campaign details, including name, description, and goal.

**SR3.** The system shall provide users and campaign creators with updates on donating progress.

**SR4.** The system shall provide reward donors with virtual rewards for their contributions, to encourage user participation and provide positive reinforcement.

**SR5.** The system shall allow users to have the option to donate anonymously or provide their information. This allows users to choose the level of privacy for their donations based on campaign requirements or their personal preferences.

### 3.1.5.4 Campaign Creation

**UR04:** The user requests to make a campaign

**SR1.** The system shall allow authorized users the ability to create campaigns.

**SR2.** The system shall provide a campaign creation form containing details such as campaign name, description, goal, and duration, simplifying the process for users to communicate their campaign ideas effectively.

**SR4.** The system shall.

### 3.1.5.5 Profile Creation

**UR05:** The user creates her/his profile

**SR1.** The system shall allow users to create accounts and customize their profiles, profile customization options can include avatar bios, etc.

**SR2.** The system shall provide users the ability to customize the visibility of their profiles, and choose between public or private visibility settings, some users may prefer a public profile to showcase their charitable efforts while others may choose a more private setting.

### 3.1.5.6 Shop

**UR05:** The user uses the store to customize the profile:

**SR1.** The system shall notify users of the profile customization store, highlighting new features such as animated backgrounds and avatars, the notifications must include information about virtual currency available for customization.

**SR2.** The system shall allow users to navigate to the store via a clear and intuitive profile customization path, with virtual currency as an accepted payment method.

**SR3.** The system shall display to the user the virtual currency cost of each avatar and live wallpaper, it should be linked to the virtual currency costs.

**SR4.** The system shall offer users the option to preview the live wallpaper and selected avatar together, as well as display the cumulative virtual currency cost, before making a

purchase.

### 3.1.5.6 Sub-campaign creation

**UR06:** The user(influencer) creates a sub-campaign

**SR1.** The system shall provide influencers with the option to create a sub-campaign, prompting them to set specific goals.

**SR2.** The system shall provide Influencers the ability to share their sub-campaigns on social media platforms, using features like integrated share buttons.

**SR3.** The system shall provide updates on progress towards sub-campaign objectives, including funds accumulated and achievements unlocked.

**SR4.** The system shall provide donors participating in the sub-campaign with virtual badges, points, and recognition within the app, which increase user engagement.

**SR5.** The system must manage the successful conclusion of the sub-campaign. When the sub-campaign goal is reached, ensure funds are appropriately allocated to the broader campaign.

### 3.1.5.7 Sharing Updates

**UR07:** The campaign admin posts updates on the campaign dashboard

**SR1.** The system shall present to campaign admins a text input or editor to compose and post updates.

**SR2.** The system shall provide a user interface for admins to select a specific campaign from the list they are administering.

## 3.1.6 Non-Functional Requirements

### Usability

- The app shall have a simple interface that is intuitive to navigate and understand for many ages of users.
- The dashboard shall be visually appealing and easy to comprehend.
- Clear instructions and guidance should be provided throughout the system.
- Users should receive error messages that guide them
- User Onboarding Time: Users should be able to understand and perform the main functionalities within 3 minutes maximum of using the app for the first time.

### Security

10

- Secure Communication: Ensure that all communication between the client-side (Angular) and server-side (ExpressJS) is encrypted using HTTPS.

### Performance

- response time: The application should respond to user interactions within 1 second for main functionalities and 3 seconds for secondary functionalities.
- Loading Time: The app should load within 5 seconds, ensuring a smooth and responsive user experience.

## **Scalability**

- The app shall be designed to handle a growing number of users and potential increases in data volume without significant performance degradation.
- The app shall be scalable to accommodate future updates, enhancements, and increased user traffic.

## **Reliability:**

- It shall be reliable and available whenever the user wants to use it (24h/7).
- It shall handle errors gracefully and provide appropriate error messages when there are connectivity issues or data retrieval problems.

## **3.2 Functional Decomposition (Use Case Diagram)**

### **3.2.1 Actors (actor list and description of their roles)**

In this section we are going to describe the actors of our app.  
Let's start with the main actors:

#### **3.2.1.1 App Admins**

The app admins play a crucial role in the app, they can approve the organizations and campaigns, and approve influencers. So they can prevent any scammy campaign or suspicious users from becoming influencers.

The app admins or (support team in the future) are expected to deal with users problems and complaints and use this feedback to improve the website.

#### **3.2.1.2 Organization**

Organizations should have the ability to start campaigns, which is the main part of the app. Organization admins should have the ability to share updates on the campaign news board. The organization admin should have the ability to control sub-campaigns by approving them or disapproving them if they don't want to connect with influencers. Organization admins should have a tab where they can see more detailed analytics about the campaigns.

#### **3.2.1.3 Influencer**

Influencers should have the ability to make sub-campaigns of existing ones.

### **3.2.1.4 Donor**

Donor is the main actor in our app, donors should have the ability to donate to campaigns and enjoy the gamification features.

	Make Donation	Create campaign	Create Sub-Campaign	profile customization
App Admin	✓	✗	✗	✓
Organization	✓	✓	✓	✓
Influencer	✓	✗	✓	✓
Donor	✓	✗	✗	✓

outsiders	✓	✗	✗	✗
-----------	---	---	---	---

Figure 16 Website Actors

### External Actors (Services):

#### 3.2.1.5 Social Media

Users should have the ability to share campaigns and achievements on social media.

#### 3.2.1.6 Payment Service

The user should have multiple payment methods such as PayPal, PalPay, and credit cards.

#### 3.2.1.7 Authentication Service

The system should use authentication services to implement more secure authentication methods.

### 3.2.2 Use Cases

Here we will explain the main and designed use cases.

#### 3.2.2.1 Donating To a Campaign:

After navigating the dashboard, donors should have the ability to donate to a desired campaign.

#### 3.2.2.2 Campaign Creation:

After logging in as an organization, the organization should have the ability to create a campaign.

#### 3.2.2.3 Profile Customization:

The users should have the ability to customize their profile by changing their profile photo, profile background, and much more.

#### 3.2.2.4 Using The Shop:

All users can access the shop. Users can buy profile-customizable items from the shop. None of the items can be bought with real money. This decision was taken to keep donors interested in donating money to the campaigns, not spending it on in-app items.

#### 3.2.2.5 Posting Campaign News:

For each campaign, there should be a news board, where campaign admins can post new updates about the campaign, this feature would keep the donors in touch with beneficiaries.

#### 3.2.2.6 Organization admin approves sub-campaign:

Organization admin should have the ability to approve or disapprove sub-campaigns. We made this feature so the organization can prevent connecting with unwanted influencers.

#### 3.2.2.7 Launching Sub-campaigns:

After the app admins approve an influencer the influencer should have the ability to launch sub-campaigns of already existing ones, he needs to be approved by the campaign admin too.

We made this feature so smaller communities can feel their impact, the sub-campaign includes all the elements of normal campaigns.

### 3.2.3 Use Cases Diagram

In this use case diagram, there are four actors: admin, organization, influencer, and donor. The user can sign in or sign up using service authentication. Donors can donate, access campaign details, shop, customize their profile, they can view dashboards, Sharing campaigns using social media, donating uses payment services. Influencers can create sub-campaigns, get campaign approval from organizations that can create campaigns, and view analysis.

Campaign creation requires admin approval. All the actors decide as users, who can sign in or sign up via service authentication, As shown in Figure 17.

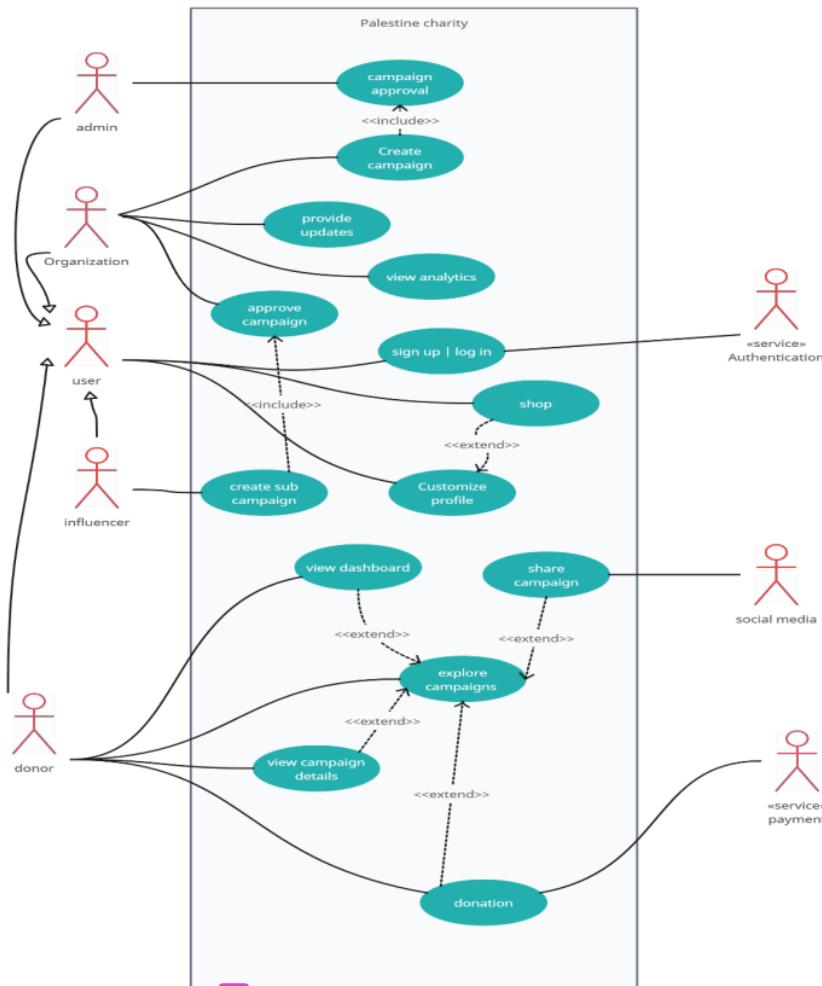


Figure 17 UseCase Diagram

26

### 3.3 System Models

#### 3.3.1 Class Diagram

Class diagram provides a static view of the Palestine charity website that contains 9 entries that user has a profile that can open the shop, The user might be a donor that can donate to any campaign, Campaign that the organization creates, LeaderBoard that the campaign has, Admin that has to approve the influencers, Influencer, and organization that must approve the sub-campaign that influencer has created it. as shown in Figure18.

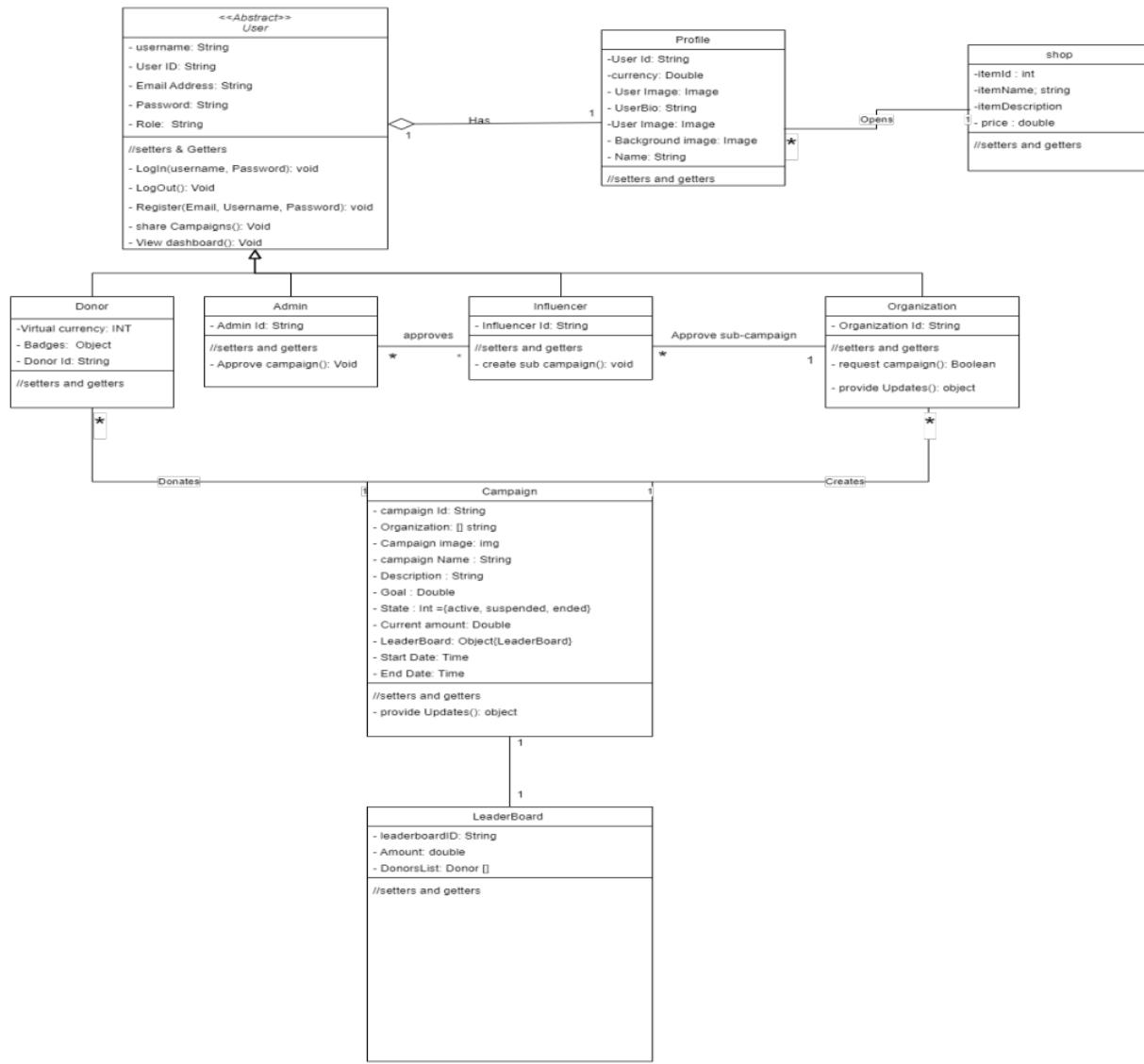


Figure 18 Class Diagram

24

### 3.3.2 Sequence Diagram

#### 3.3.2.1 Registration

Figure 19 shows the sequence diagram for the user in the registration process that depicts the interactions between three objects: UI, Reregistration Controller, and Database, showing the flow of messages exchanged between them over time.

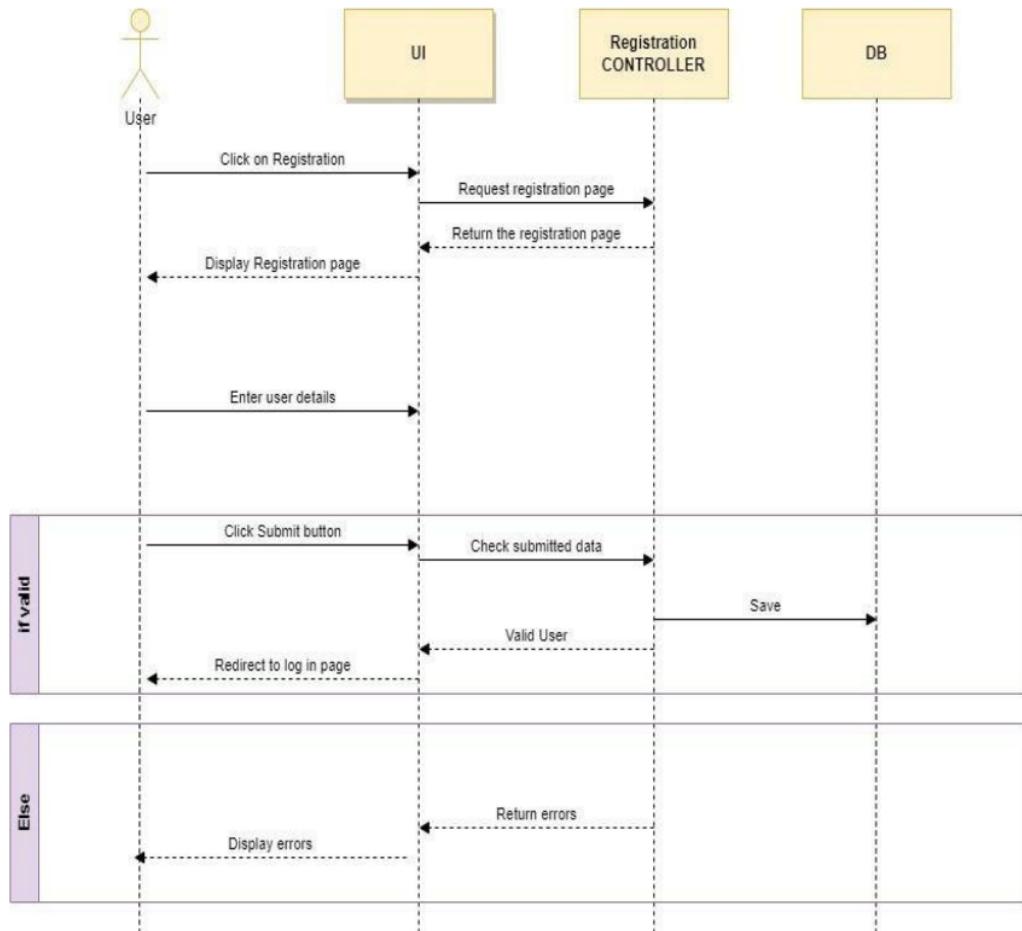


Figure 19 Registration Sequence Diagram

### 3.3.2.2 Request To Make a Campaign

Figure 20 shows the sequence diagram for the Organization Requesting to make a campaign that depicts the interactions between five objects: UI, Campaign controller, Admin Controller Database, and Notification Service, showing the flow of messages exchanged between them over time.

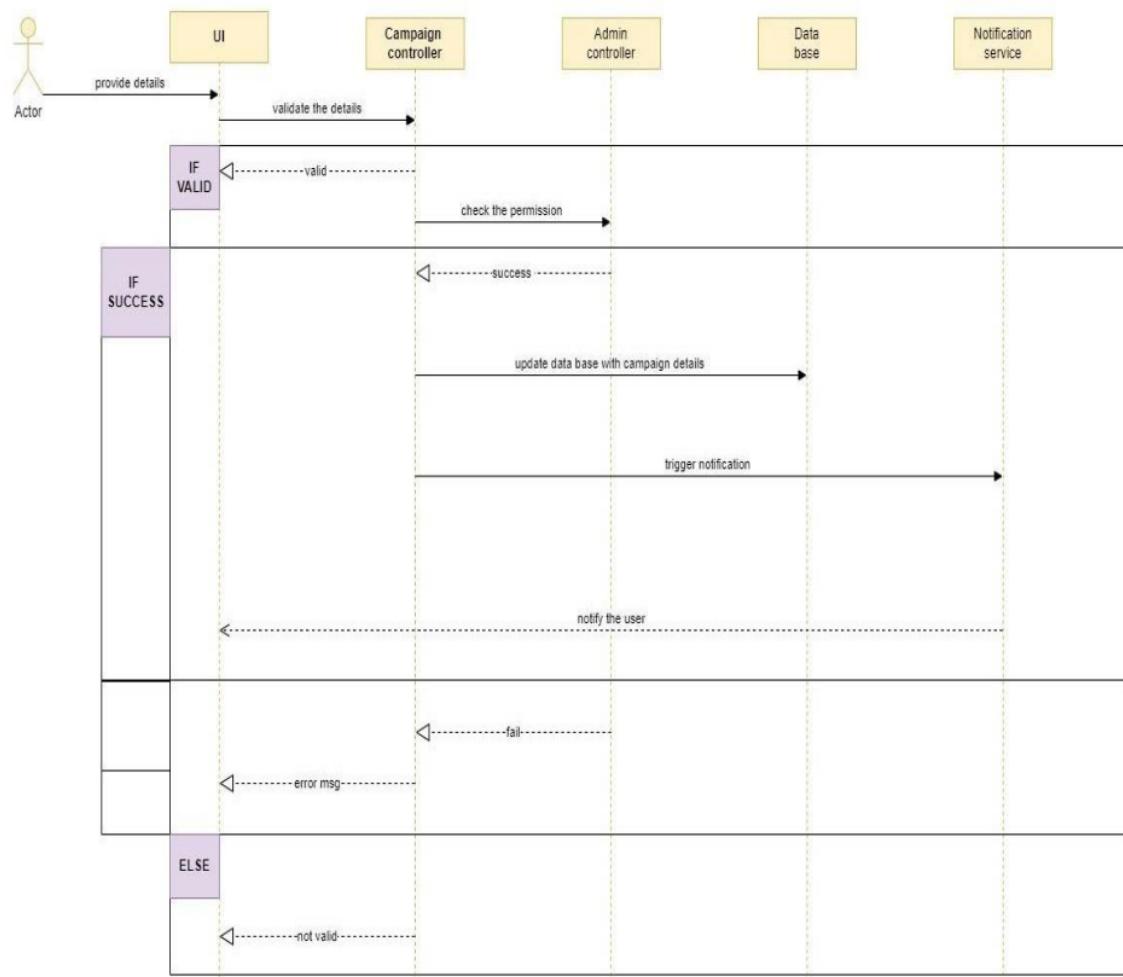


Figure 20 Request Campaign Sequence Diagram

### **3.3.2.3 Donating Process**

Figure 21 shows the sequence diagram for an Organization Requesting to make a campaign, that depicts the interactions between five objects: UI, Campaign, Admin Controller, Database, and Notification Service, showing the flow of messages exchanged between them over time.

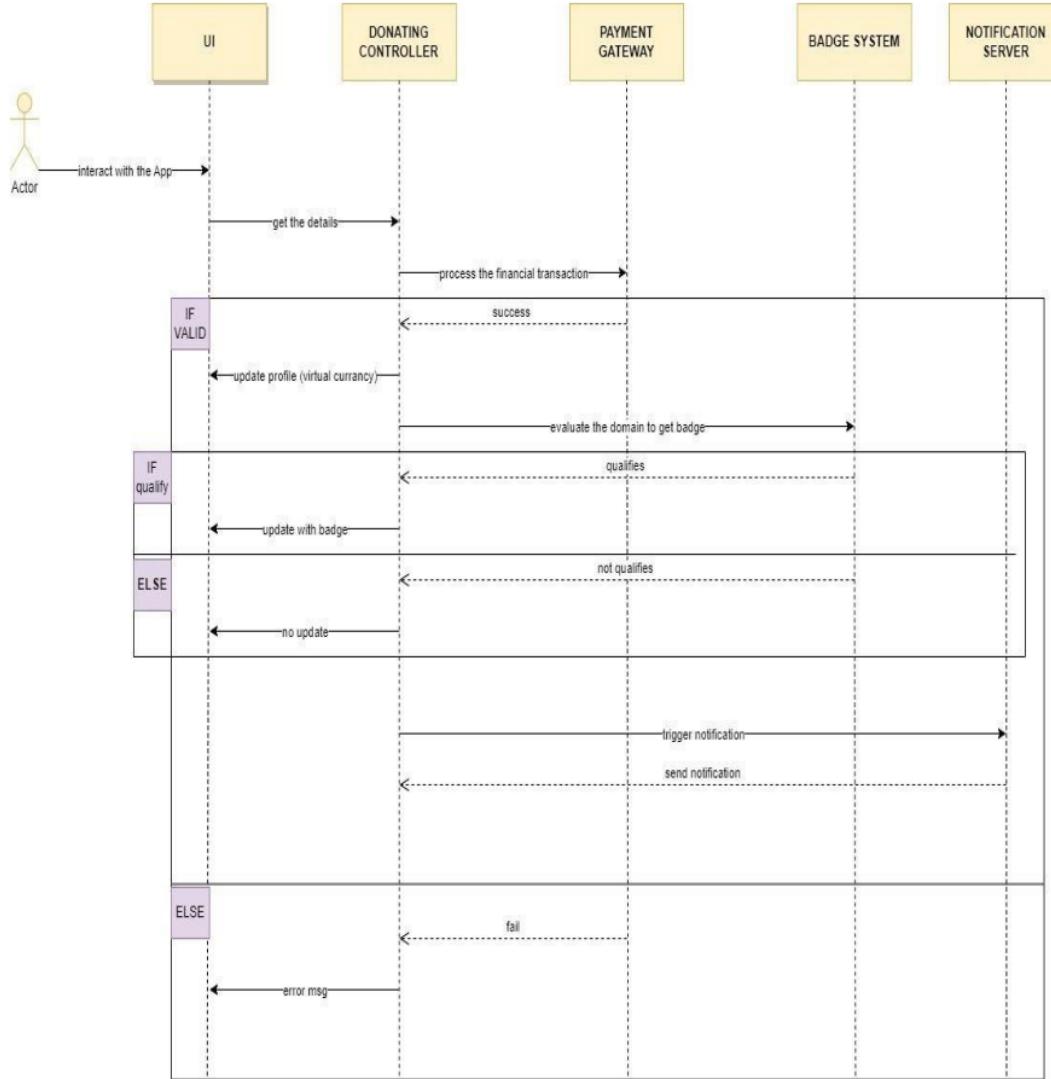


Figure 21 Donating Process Sequence Diagram

### **3.3.2.4 Create a sub campaign:**

Figure22 shows the sequence diagram for an Influencer Creating a sub-campaign, that depicts the interactions between four objects: UI, Campaign Controller, Organization, and Database, showing the flow of messages exchanged between them over time.

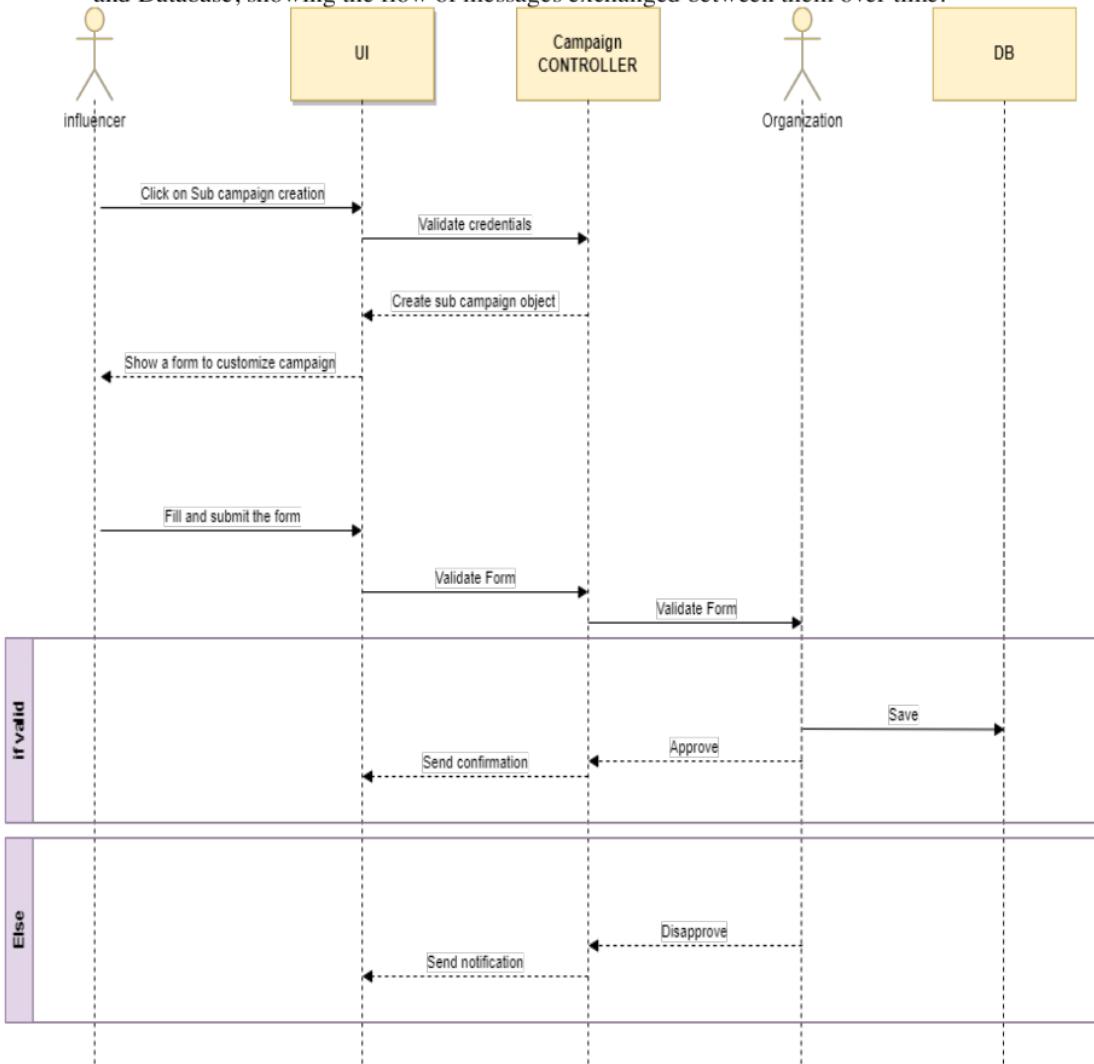


Figure 22 Sub-Campaign Creation Sequanse Diagram

### 3.3.2.5 News Dashboard:

Figure23 shows the sequence diagram for News Dashboard to update the information for each campaign. This diagram depicts the interactions between five objects: Campaign Admin, UI, system, Database, and Donor showing the flow of messages exchanged between them over time.

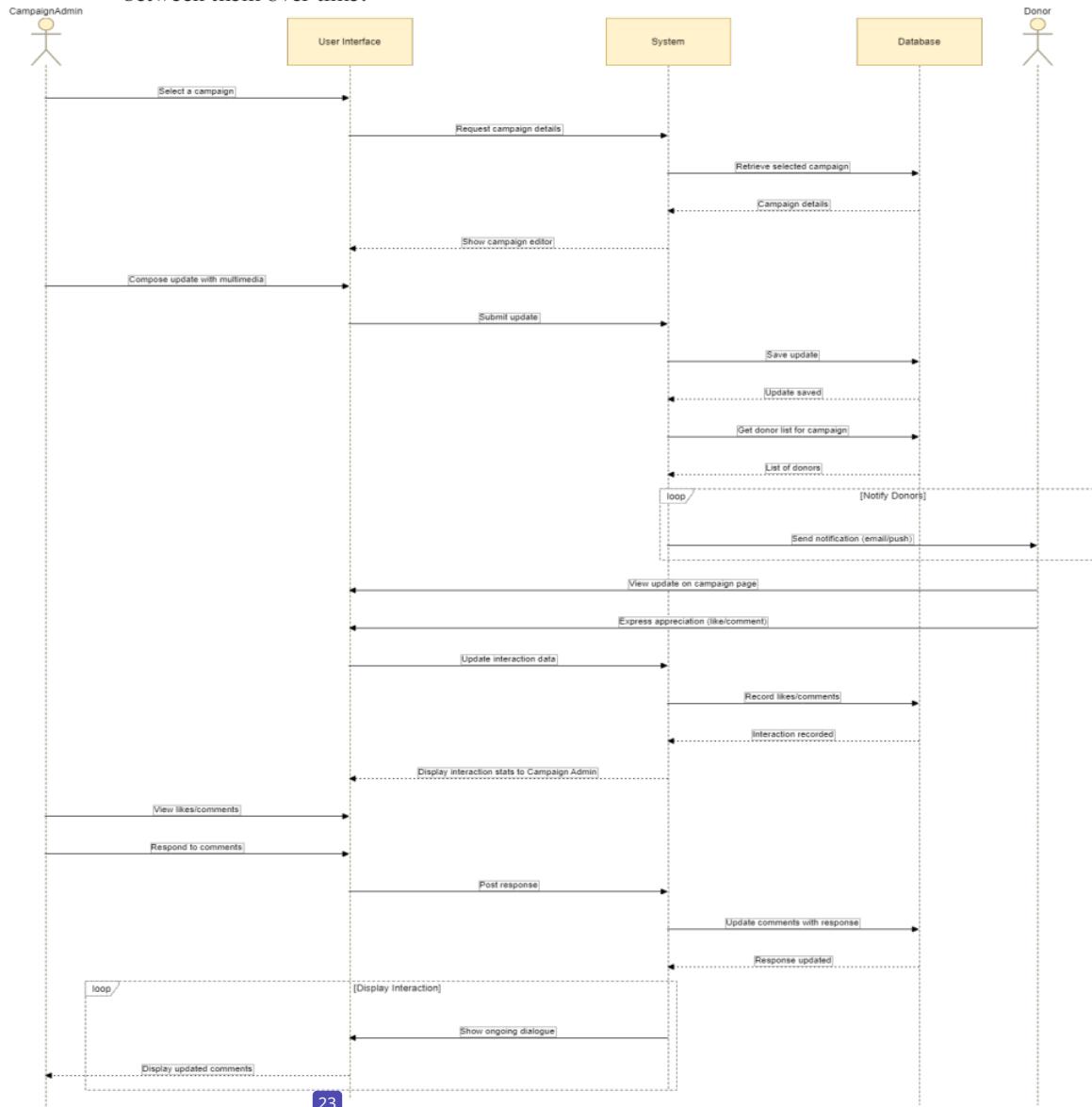


Figure 23 Updating News Dashboard Sequence Diagram

23

### 3.3.3 Activity Diagram

#### 3.3.3.1 Donation Process:

This activity diagram shown in Figure24 shows the steps that the user Donates to a Campaign and checks if the user's action meets the criteria for earning badges or virtual currency.

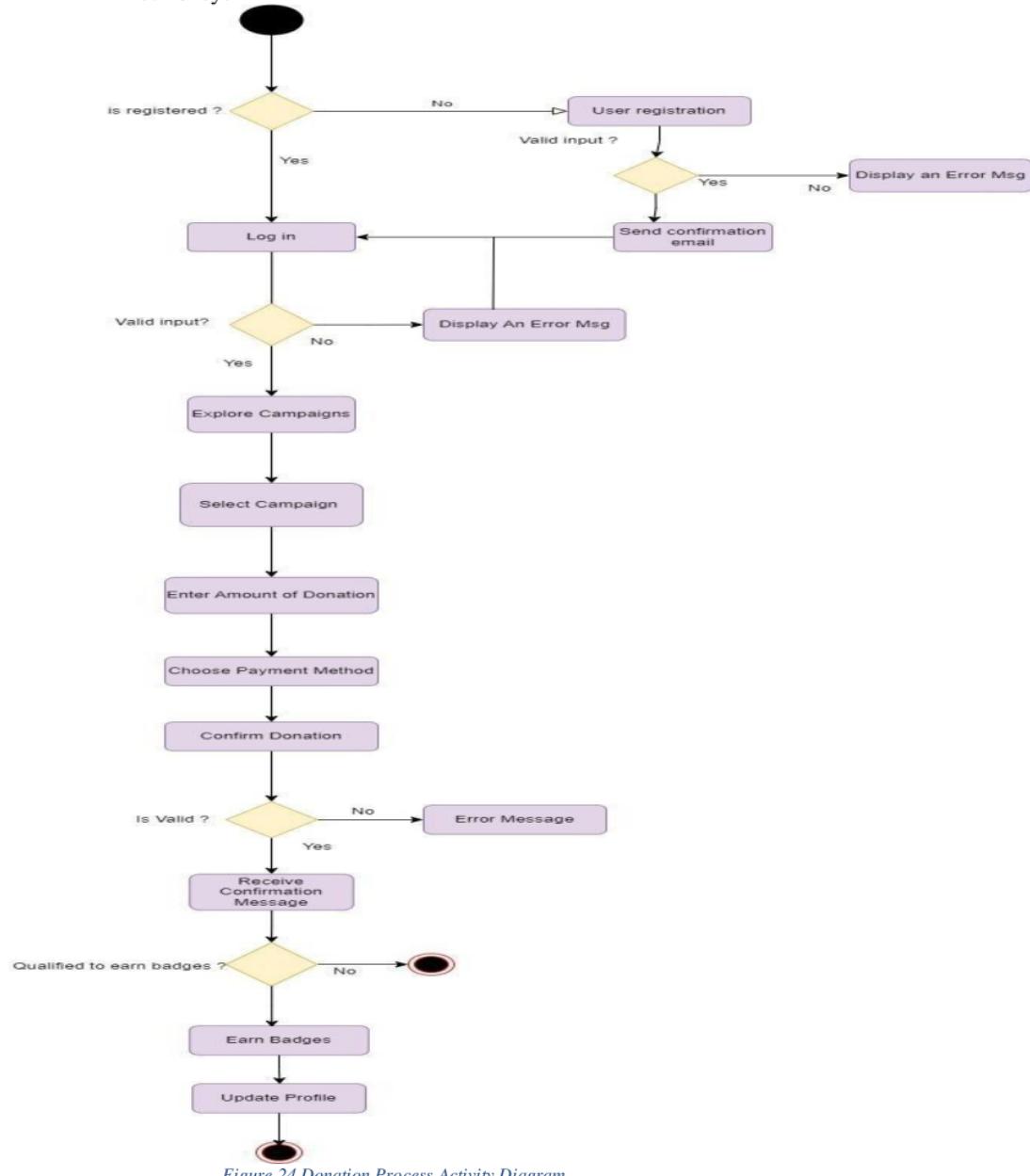


Figure 24 Donation Process Activity Diagram

### **3.3.3.2 creating sub-campaign:**

This activity diagram shown in Figure 25 shows the steps that the user makes to create a sub-campaign.

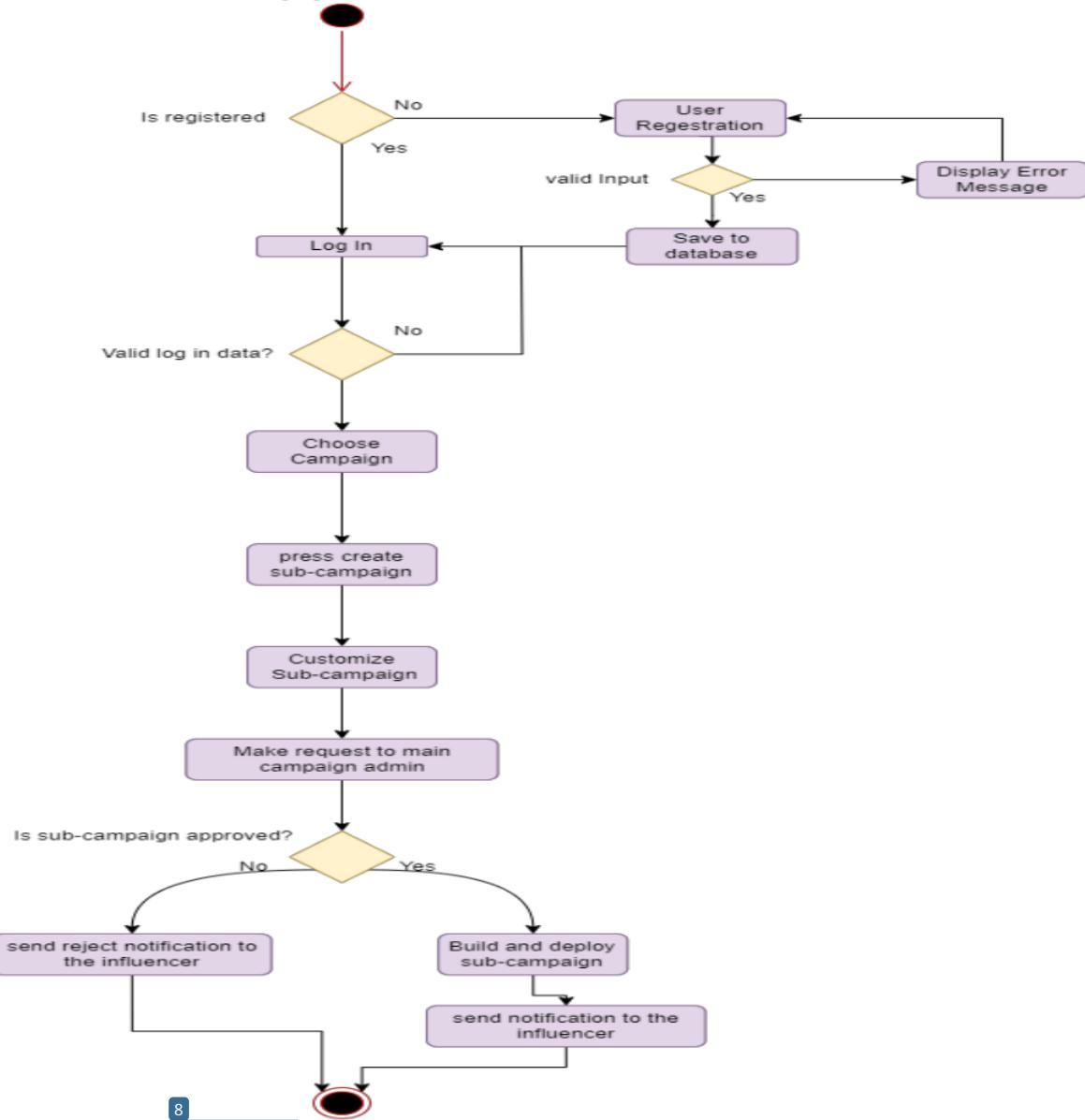


Figure 25 Creating Sub-Campaign Activity Diagram

### 3.3.4 State Chart Diagram

#### 3.3.4.1 Requesting To Make A Campaign :

Figure 26 shows the user (Organizations) Requesting to make a Campaign.



Figure 26 Request To make A Campaign State Chart

### **3.3.4.2 buying Items:**

Figure 27 shows the user buying an item from the shop.

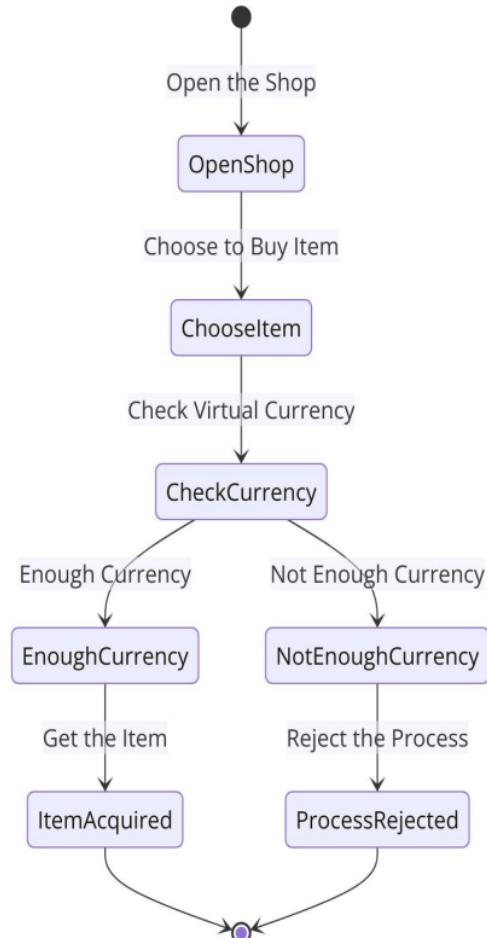


Figure 27Buying Item State chart

## 3.4 System Architecture

### 3.4.1 Software Architecture

Figure 28 shows the software architecture based on 3 layer architecture and shows various components of the system.

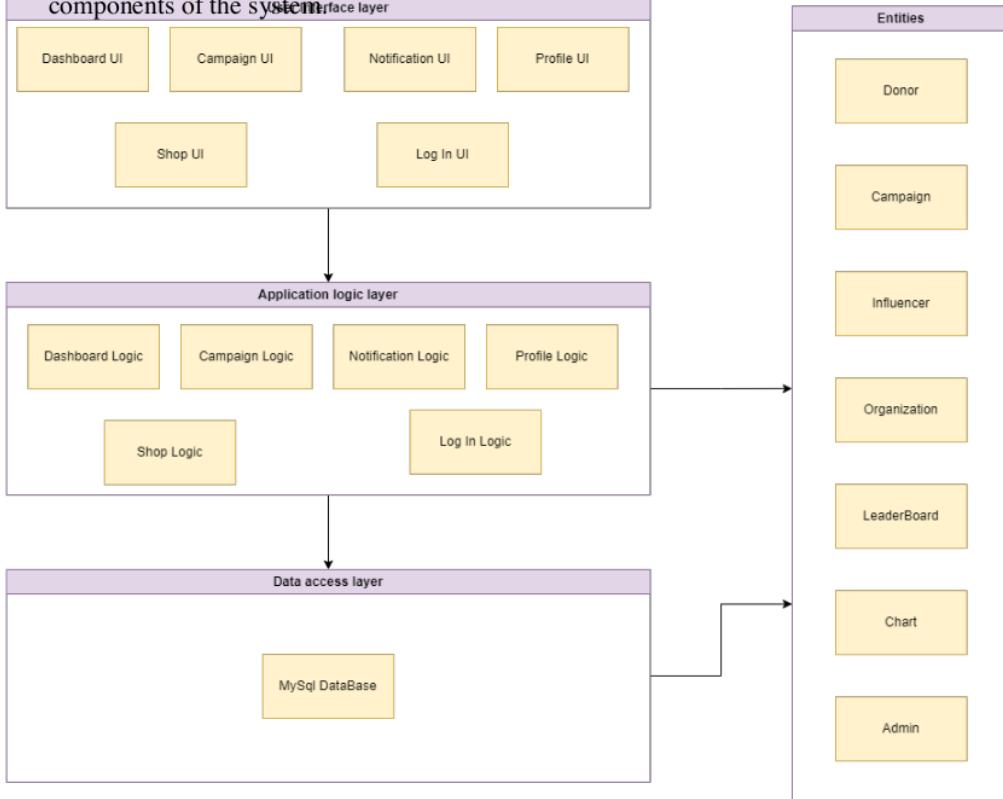


Figure 28 Software Architecture

### 3.4.2 Deployment Diagram

Figure 29 shows the Deployment diagram and highlights what components are on the server side and are on the client side.

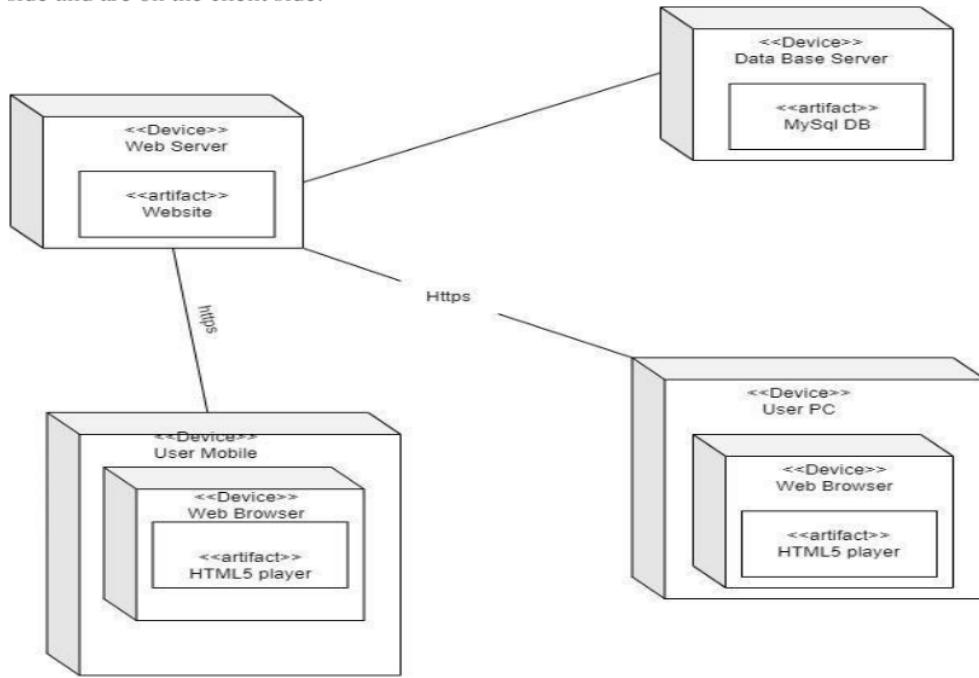


Figure 29 Deployment Diagram

### 3.5 Data Management and Models

Figure 30 shows our er-diagram, which represents the entities, their attributes, and the relationships between them within a database in Palestine Charity app.

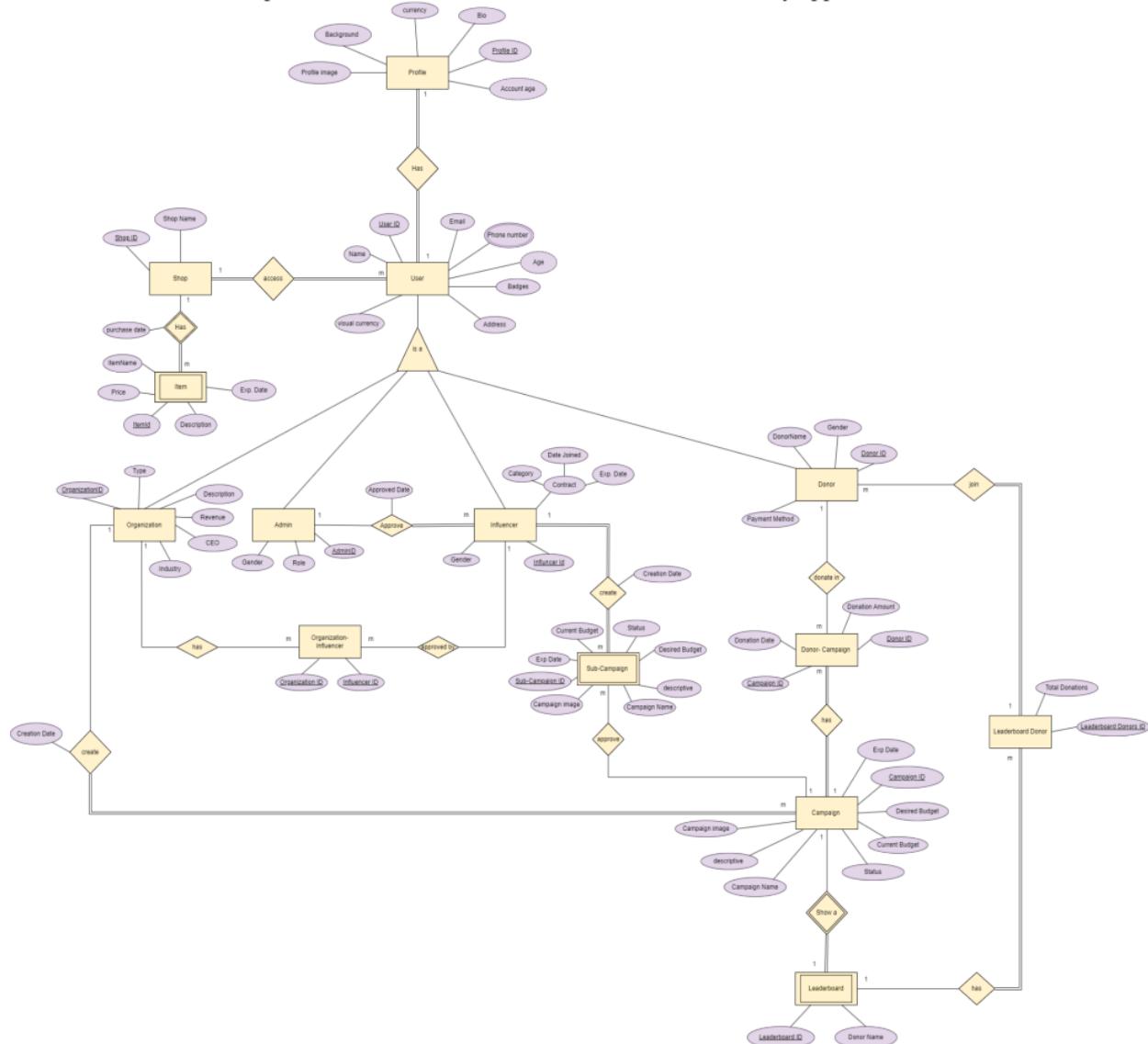


Figure 30 Er-Diagram

## 6 Chapter 4: Implementation and Testing:

In this chapter, we are going to cover the implementation process, the following table shows the development stages:

Table 4 work distribution

Task	Assigned To	Progress	Start	End
<b>Initiation</b>				
Deep Dive in Technologies	Iman, Nagham, Mohammad	50%	2/16/24	2/29/24
Design Interface	Mohammad	5%	2/29/24	3/31/24
Implement database	Iman, Nagham		2/29/24	3/10/24
Connect DB To express	Iman, Nagham		3/11/24	3/14/24
Prepare Rest API's	Mohammad, Iman, Nagham		3/15/24	5/4/24
Connect Frontend and backend	Mohammad		3/20/24	5/19/24
Prepare final report	Mohammad, Iman, Nagham		5/20/24	5/30/24

### 4.1 Implementation:

In this section we will display the features and interfaces of the project:

#### 4.1.1 landing page:

When the user opens the website the landing page would be the first page he would see:

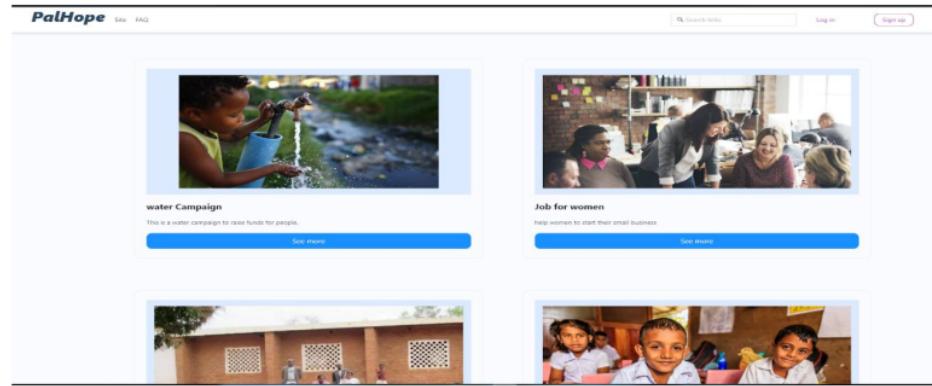


Figure 31 landing page

this page contains the campaigns on our website, and by being the landing page the user can get to the donation process smoothly.

#### 4.1.2 login/signup:

Users have to be signed up to have the full user experience and features:

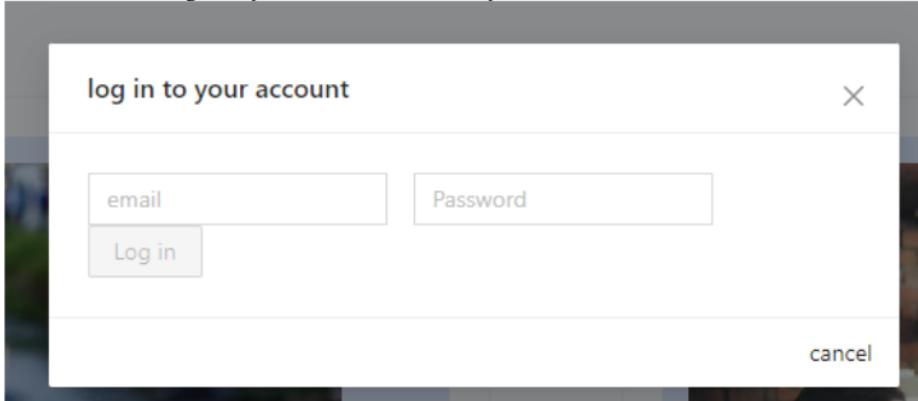


Figure 32 log in form

a simple login form with cookies integration so the user can get on his account quickly, the system also has an auto-login feature, so refreshing the tab or closing it won't end the user session.

This is a safe decision because no valuable data is shown in our system, the secession will end after 1 month to prevent users from forgetting the password.

#### 4.1.3 After logging in:

After logging in a side nav appears and the user gets access to the other pages and features:

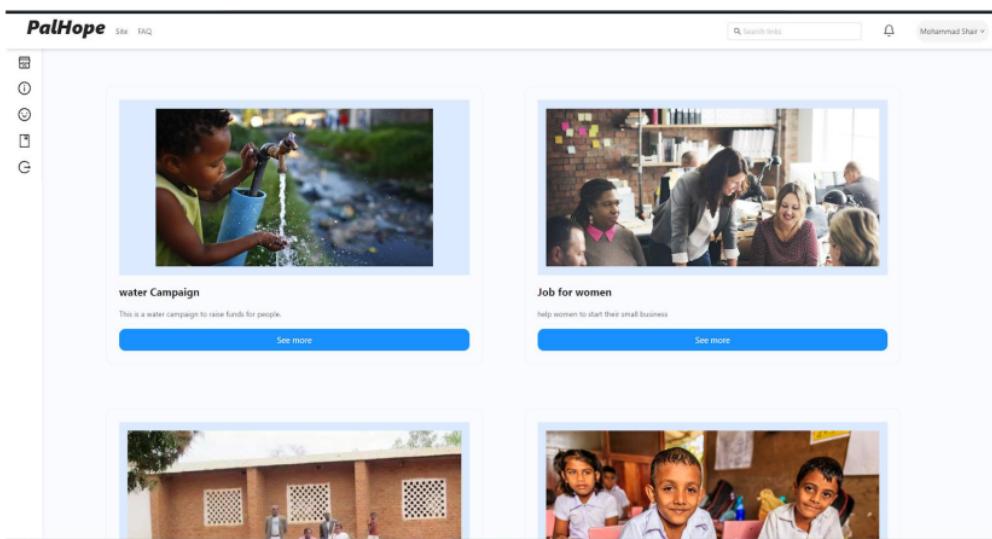


Figure 33 side nav

the user gets access to the user profile page, bookmarks, and the shop.

#### 4.1.4 user profile:

On this page, the user can modify his profile, add a picture, or background picture, or change the default biography:

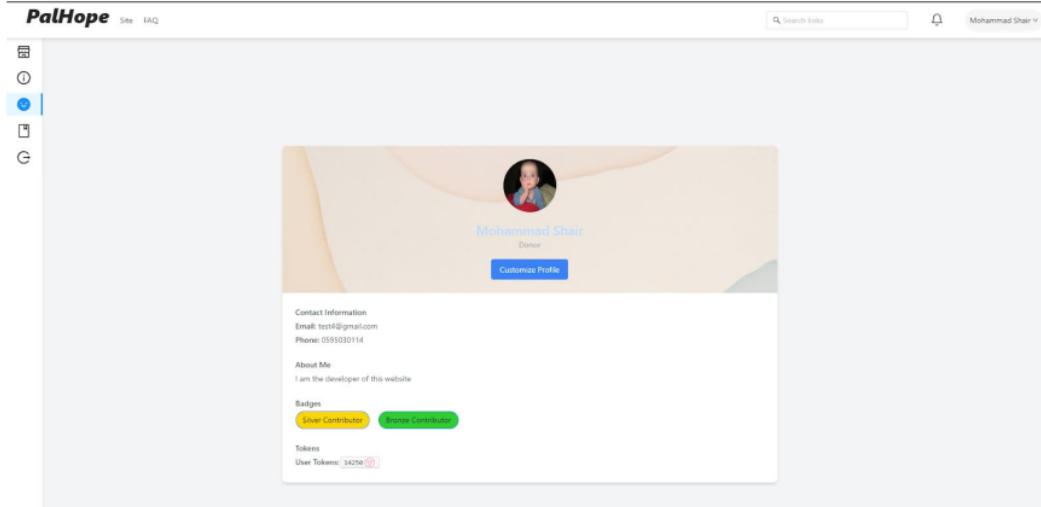


Figure 34 user profile page

as seen in Figure (34) the profile shows the user information, badges, and tokens.

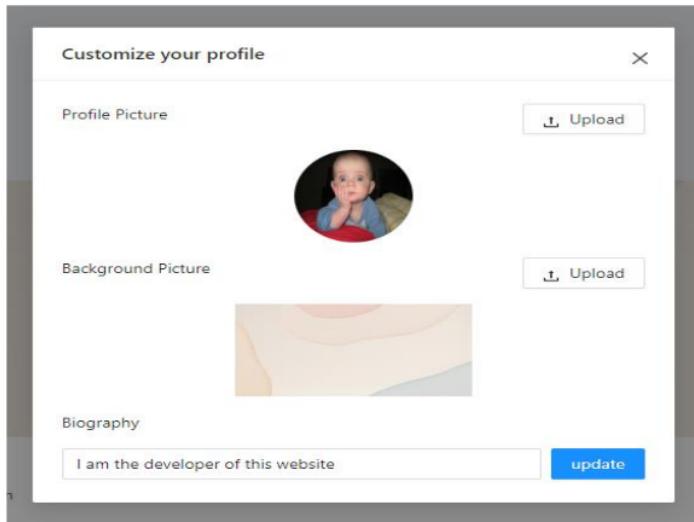


Figure 35 Customize profile

the customize profile button would show the form in figure(35)

This design is inspired by big companies like Meta, showing users the current profile and background pictures to make it more user-friendly.

Alerts are implemented to keep users updated about the current process.

#### 4.1.5 campaign page:

The main features of the website are implemented here, figure(36) shows the page:

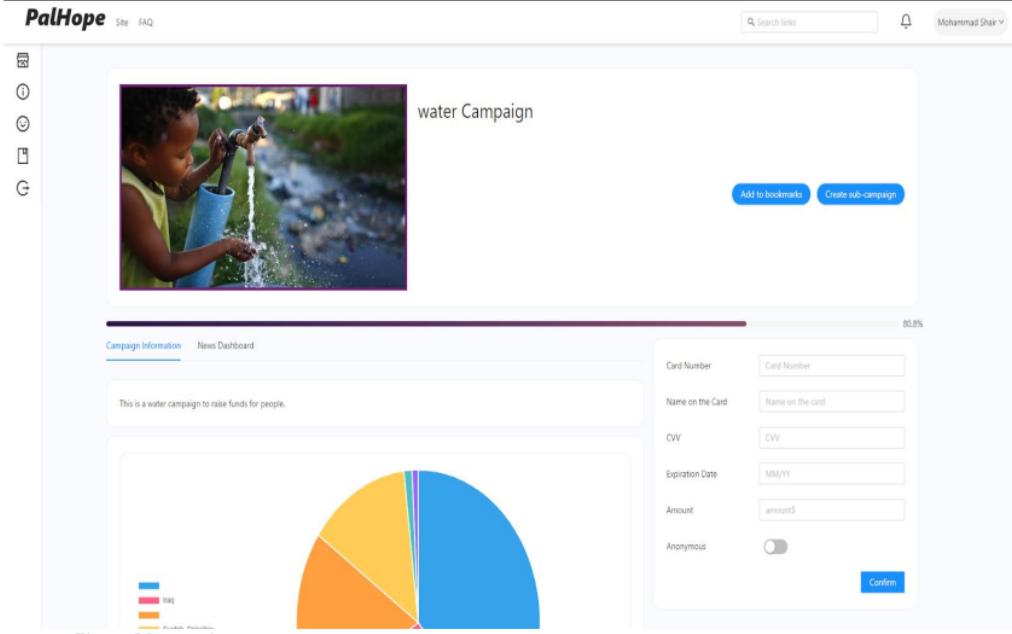


Figure 36 campaign

First, the page shows the campaign details such as campaign name, description, and organization, the header has two buttons “Add to favorites” which would allow the user to add the campaign to his bookmark page for easy access to the desired campaigns if the campaign is already bookmarked then the button would turn to “remove from bookmarks”, and a “Create sub-campaign” a button that appears to influencers only which opens a form so the influencer can start his sub-campaign.

Second, the donation segment of the page shown in Figure (37), this segment gives the user to donate to the desired campaign with a credit card. The user can donate to campaigns as an anonymous user so the username will appear on the leaderboard as anonymous, even in this case the user will get rewards for the donation. Also, the user can donate to any campaign while not logged in but no rewards would be gained.

The screenshot shows a donation form. It includes fields for "Card Number", "Name on the Card", "CVV", "Expiration Date", "Amount", and an "Anonymous" toggle switch. Each field has a corresponding input box. A "Confirm" button is located at the bottom right of the form.

Figure 37 donation form

Third an interactive pie graph as shown in Figure (38) shows the countries that contributed the most to this campaign

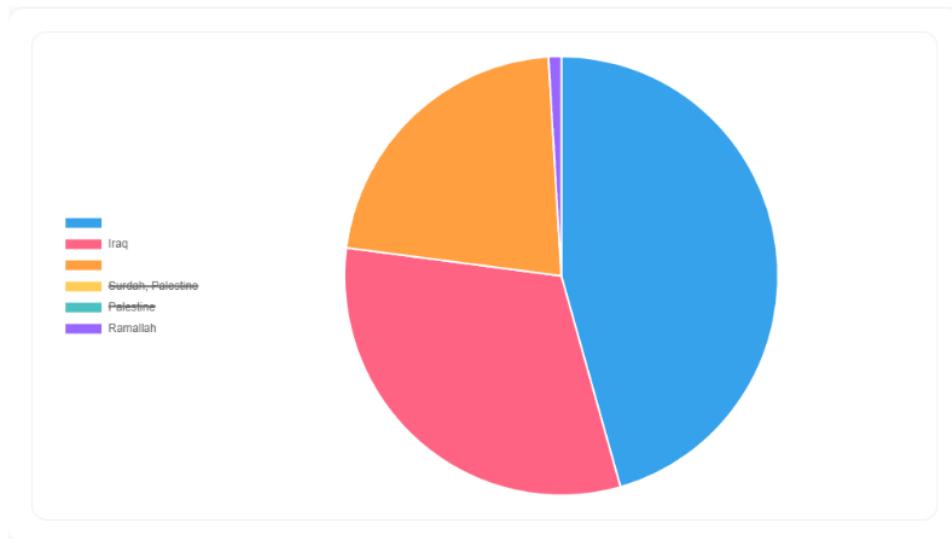


Figure 38 donation chart

the page also contains a leaderboard that shows the top contributors as shown in Figure(39)

Rank	Name	Amount Donated
0	Ali karakra	1003
1	Anonymous	694
2	Anonymous	515
3	Anonymous	500
4	Mohammad	500
5	John Doe	250
6	Anonymous	90
7	Anonymous	52
8	Anonymous	50

Figure 39 leaderboard

Also, the user can navigate to the campaign news dashboard page to check the latest updates from the campaign owners as shown in Figure(40)

The screenshot shows the 'Campaign Updates' section of a news dashboard. At the top, there is a navigation bar with 'Campaign Information' and 'News Dashboard' tabs, with 'News Dashboard' being the active tab. Below the navigation is a heading 'Campaign Updates'. A central area displays a message icon with a speech bubble containing the text 'No Data'. Below this message is a large input field for posting new updates, with a placeholder 'Post an update' and a small camera icon. In the bottom right corner of the update field, there is a small circular icon with a plus sign, likely for adding media.

Figure 40 Campaign NewsBoard

As shown in the figure the organization has access to add new updates to the news dashboard.

#### 4.1.6 Bookmark page:

The user-bookmarked campaigns appear on this page as shown in Figure(41)

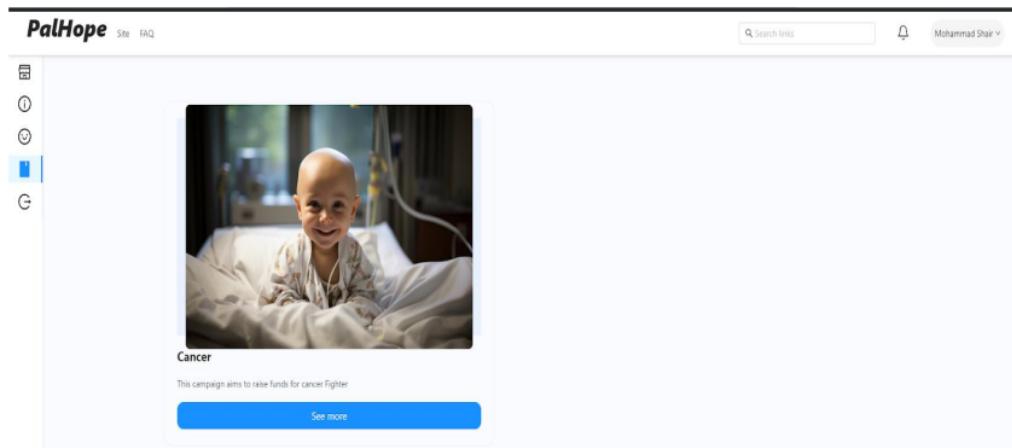


Figure 41 bookmarks page

#### 4.1.7 Shop:

The user can buy different types of items just as Fonts, profile pictures, or background pictures, from the shop to customize the profile figure(42) shows a sample of the shop Items.

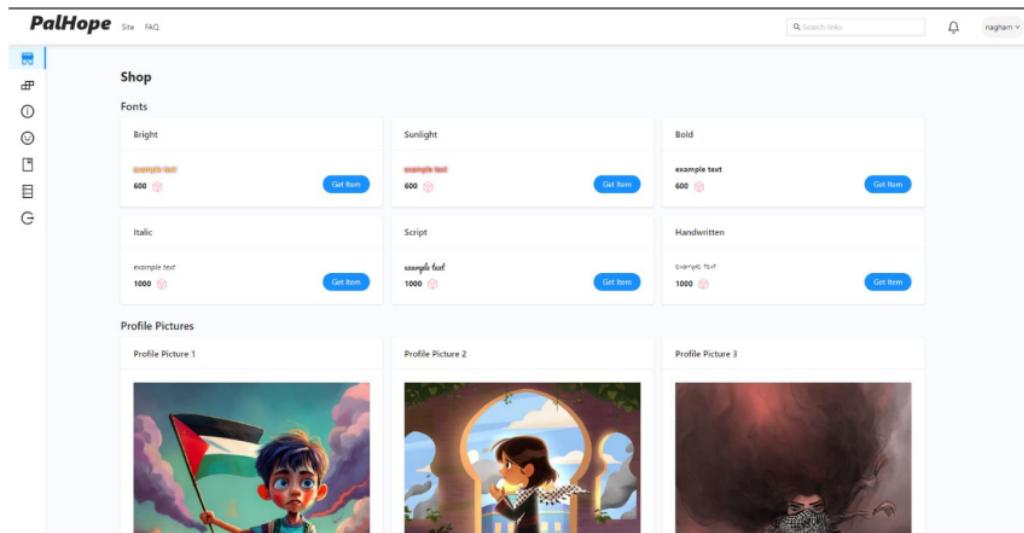


Figure 42 shop

#### 4.1.8 Inventory:

The inventory page is where the user can find the purchased Items to use, as shown in Figure() the user can pick the font or profile picture.

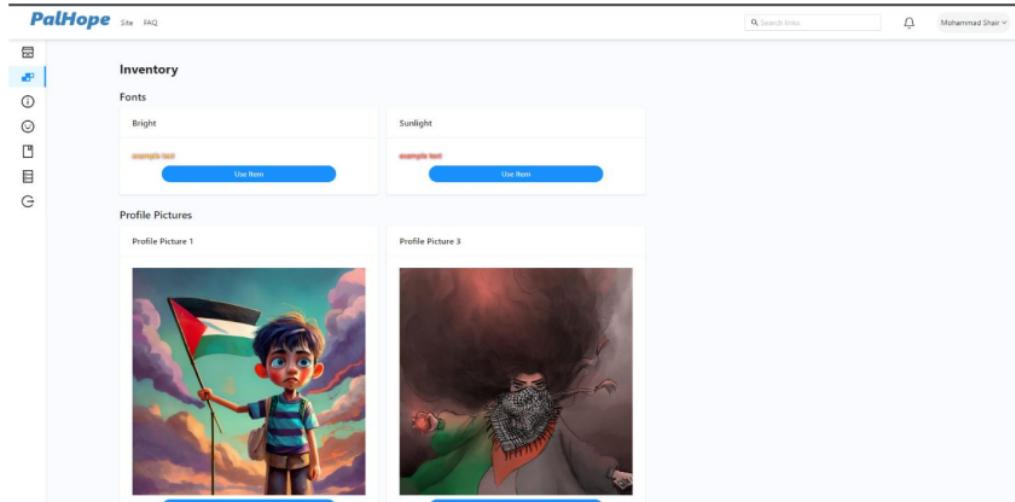


Figure 43 Inventory

#### 4.1.9 Creating campaign/ sub-campaign:

Authenticated organizations can create campaigns using the form shown in Figure (43)

A screenshot of a campaign creation form. It includes fields for 'Campaign Image' (with a 'Choose File' button and placeholder 'No file chosen'), 'Campaign Name' (a text input field), 'Goal Amount' (a text input field), 'End Date' (a date picker with placeholder 'Select date'), 'Description' (a text area), and a large blue 'Add Campaign' button at the bottom.

Figure 44 creating campaign form

#### 4.1.10: Approving new influencers and organizations:

Admins can approve new created accounts of influencers and organizations using the page shown in Figure ()

#### 4.1.11: Checking donation history:

Users can check their previous donation records on the page shown in Figure ()

## 4.2 Testing:

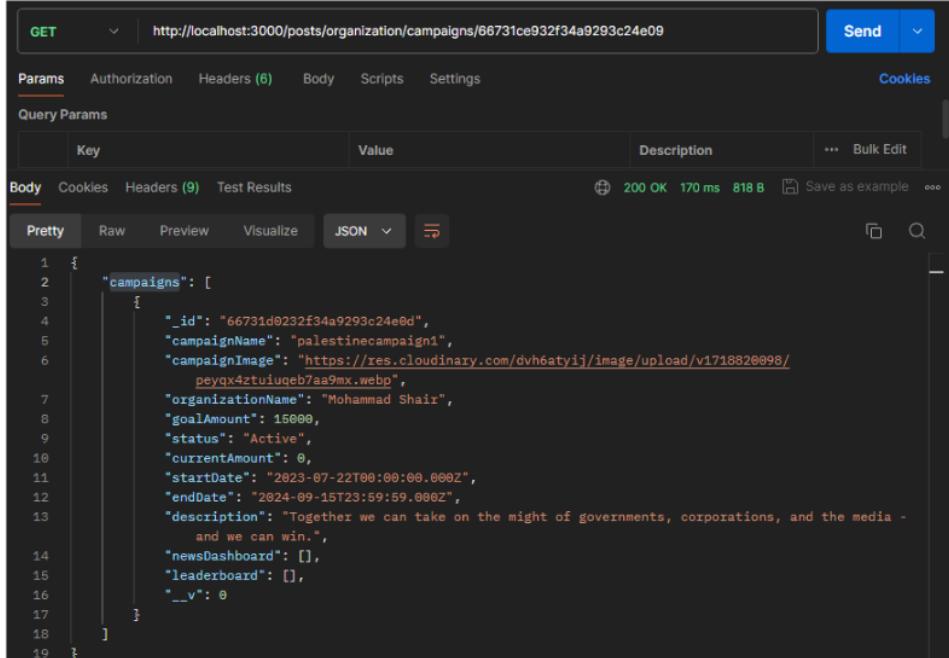
In this section, we are going to share the test results of our system.

### 4.2.1 Automated Testing:

These tests are automated which means that we used software for testing.

#### 4.2.1.1 API Testing:

We used Postman to test the APIs of our project individually, the Figure(44) shows an API test of getting the campaigns of an organization by id.



```
1 {
2   "campaigns": [
3     {
4       "_id": "66731ce932f34a9293c24e09",
5       "campaignName": "palestinecampaign1",
6       "campaignImage": "https://res.cloudinary.com/dvh6atyij/image/upload/v1718820098/
7         peyqx4ztuiqeb7aa9mx.webp",
8       "organizationName": "Mohammad Shair",
9       "goalAmount": 15000,
10      "status": "Active",
11      "currentAmount": 0,
12      "startDate": "2023-07-22T00:00:00Z",
13      "endDate": "2024-09-15T23:59:59Z",
14      "description": "Together we can take on the might of governments, corporations, and the media - and we can win.",
15      "newsDashboard": [],
16      "leaderboard": [],
17      "__v": 0
18    }
19  }
```

Figure 44 API testing

Other API tests are provided in the Appendices

#### 4.2.1.2 Unit Testing:

We used Jasmine and Jest for unit testing, the Figure(45) shows frontend unit tests using Jasmine.

```
DashboardService
  #getCampaigns
    • should fetch campaigns and update the BehaviorSubject
  #handleError
    • should handle various error messages correctly
  • should be created

ComponentsComponent
  • should create

ProfileService
  • SPEC HAS NO EXPECTATIONS should be created

  #getUserInfo
    • SPEC HAS NO EXPECTATIONS should return user info when the user is authenticated
    • SPEC HAS NO EXPECTATIONS should return null when the user is not authenticated

ProgressBarComponent
  • should create

PagesComponent
  • should create
```

The Figure() shows Jest test for the backend

Other Unit tests are provided in the Appendices

#### 4.2.1.3 End-To-End Testing:

We used Cypress to perform these tests, while Unit testing tests each component individually e2e testing tests the whole feature and project functionalities, the Figure() shows a test of the shop features.

The screenshot displays two side-by-side windows. On the left is the Cypress Test Runner interface, showing a spec file for a 'Shop Component' with several test cases. One test case, 'should buy an item', includes a detailed code snippet for visiting the shop, selecting an item, and performing a purchase. On the right is a screenshot of the 'PalHope' application's 'Shop' page. This page features sections for 'Fonts', 'Profile Pictures', and 'Background Pictures'. Each section contains two items, each with a preview image, a size (e.g., 100, 150, 200, 250), and a 'Get Icons' button.

Figure 45 cypress test

More e2e test is provided in the Appendices.

#### 4.2.2 Manual Testing:

Each feature was tested manually too, we tested each feature after developing and we provided the website to a small circle of people to test the functionality and the UI. Some errors were shown by these tests like buttons appearing for not registered users, but all caught errors was handled.

## Chapter 5: Conclusion and Future Works

### 5.1 Review Of The Project

As a conclusion for all the project, we will provide a web application that helps donors to donate to the desired campaigns and causes. The application should give organizations a straightforward way to build a campaign. Influencers should have the ability to make sub-campaigns of existing ones.

Gamification is the major part of our project, the users should feel rewarded for every activity they do. With leaderboards badges and tokens, the user would always be engaged with our app.

### 5.2 Future Works

- Continue working on the privacy and policy so the payment process can be done in the app.
- Getting the trust and approval of other campaigns.
- Deploying the project on fully trusted deployment servers.
- Provide support for different languages.
- Providing 2-step authentication to improve the security of the project.
- Employ an audit and accounting team.

## Bibliography

- [1] Angular, "What is angular," Google 15 8 2023. [Online]. Available: <https://angular.io/docs>. [Accessed 8 2 2024].
- [2] Mdn, "Mdn web docs," Mdn, 20 12 2023. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>. [Accessed 8 2 2024].
- [3] Tailwind, "Utility First FUNDimentals," TailIwnd, [Online]. Available: <https://tailwindcss.com/docs/utility-first>. [Accessed 8 2 2024].
- [4] S. documentation, "SASS documentation," SASS, [Online]. Available: <https://sass-lang.com/documentation/>. [Accessed 9 2 2024].
- [5] ExpressJs, "glossary," ExpressJs, [Online]. Available <https://expressjs.com/>. [Accessed 9 2 2024].
- [6] Louisa Küchler, Meinald T. Thielsch, Guido Hertel, "Are you willing to donate?: relationship between perceived website design, trust and donation decisions online," in *MuC '20: Proceedings of Mensch und Computer 2020*, 2020.
- [7] H. Golrang, Entertainment Computing, sciencedirect, 2019.
- [8] philanthropyconnections, "philanthropy connections," philanthropy connections Foundation, 2024. [Online]. Available: <https://philanthropyconnections.org/>. [Accessed 10 12 2023].
- [9] tiltify, "tiltify," tiltify, 2024. [Online]. Available: <https://tiltify.com>. [Accessed 10 12 2023].
- [10] DonorSee, "DonorSee," DonorSee , 2024. [Online]. Available: <https://donorsee.com>. [Accessed 10 12 2023].
- [11] Kiva, "Kiva," Kiva, 2024. [Online]. Available: <https://www.kiva.org>. [Accessed 26 11 2023].
- [12] ShareTheMeal, "ShareTheMeal," ShareTheMeal, 2024. [Online]. Available: <https://sharethemeal.org>. [Accessed 9 12 2023].
- [13] BeanBeanBean, "BeanBeanBean," BeanBeanBean, 2024. [Online]. Available: <https://beanbeanbean.com>. [Accessed 15 12 2023].
- [14] FreeRice, "FreeRice," FreeRice, 2024. [Online]. Available: <https://freerice.com/home>. [Accessed 9 2 2024].
- [15] Benevity, "Benevity," Benevity, 2024. [Online]. Available: <https://benevity.com>. [Accessed 18 1 2024].

# Chapter6: Appendices

## 5.1. Use Case Specifications

### 5.1.1 Donating To a Campaign:

participating actors	Donors
preconditions	<ul style="list-style-type: none"><li>• donor completed the user registration process</li></ul>
Flow of events	<ol style="list-style-type: none"><li>1. The donor creates a user profile and provides the necessary information about him.</li><li>2. The donor customizes their profile with a personal avatar and chooses a theme that reflects their interest.</li><li>3. The donor explores unique campaigns.</li><li>4. The system must display on the campaign page detailed information about the campaign, including its location, beneficiaries, and funding goal.</li><li>5. The donor decides to contribute to the campaign.</li><li>6. The donor chooses the donation amount and has the option to set up a recurring donation.</li><li>7. The system provides multiple payment methods.</li><li>8. The donor enters her payment information securely.</li></ol>
postconditions	<ul style="list-style-type: none"><li>• The donor receives virtual currency points for her donation.</li><li>• The donor receives a confirmation message that the donation was completed successfully</li><li>• The system will inform the donor with regular updates about the progress of the campaign she has donated to. These updates help her stay connected to the cause and see the tangible impact of her contribution.</li><li>• The donor shares her achievement on social media, encouraging friends and followers to support the cause.</li><li>• The system updates the set of default donor badges on her user profile, in recognition of her contribution to the campaign.</li></ul>
Alternatives	<ol style="list-style-type: none"><li>1. The donor entered invalid payment information. The system displays an error message and asks the donor to correct the input.<sup>22</sup></li><li>2. If the campaign does not reach its funding goal. The system provides the donor with the latest developments about the campaign status and suggests other campaigns for him.</li></ol>

	3. The donor decides not to set up a recurring donation. The system recognizes a one-time contribution without affecting regular updates.
--	---

Figure 46 Donating To Campaign usecase

### 5.1.2 Campaign Creation:

<b>Participating actors</b>	Organization owner, Administrators
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The web's backend system is operational.</li> <li>• Administrators are available to review and approve campaign requests.</li> <li>• The organization owner is a registered user and has logged into the "Palestine Charity".</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. The organization owner moves to create a campaign section</li> <li>2. The organization owner provides the necessary campaign details for example the title and description about this campaign and the fundraising goal etc .</li> <li>3. The organization owner sets a goal amount and a deadline for the campaign</li> <li>4. The owner of the organization explains the importance of the impact of this campaign.</li> <li>5. The organization owner submits the campaign request.</li> <li>6. The backend system generates a notification to administrators about this campaign.</li> <li>7. The Administrators review campaign details.</li> <li>8. The administrators approve this campaign.</li> <li>9. The organization owner engages with donors by a dashboard that shows campaign progress.</li> </ol>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The organization owner campaign is live and available to potential donors.</li> <li>• The organization owner receives a notification confirming that his campaign is continuing.</li> <li>• The organization owner receives real-time notifications for every donation made to his campaign.</li> <li>• The organization owner and the donors receive a detailed report on the impact after the campaign ends.</li> <li>• The organization owner interacts with donors through thank you letters and personal updates.</li> </ul>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1. The organization owner campaign request was</li> </ol>

	<p>rejected. The system provides a reason for rejection, and the organization has the option to edit and resubmit the campaign.</p> <ol style="list-style-type: none"> <li>2. The organization owner is experiencing technical problems during the campaign creation process. The system provides support options or instructions for troubleshooting.</li> <li>3. The campaign may receive a low response. The system will suggest promotional strategies or provide support to improve campaign visibility.</li> <li>4. Perhaps he wants to extend the duration of the campaign. The system allows him to edit campaign details, including the deadline, before the current deadline is reached.</li> </ol>
--	---

Figure 47 Campaign Creation UseCase

### 5.1.3 Profile Customization:

Participating actors	Users
Preconditions	<ul style="list-style-type: none"> <li>• The system's profile customization features are accessible</li> </ul>
Flow of events	<ol style="list-style-type: none"> <li>1. The user goes to the profile section to check his default settings.</li> <li>2. The user clicks on her default avatar to update it.</li> <li>3. The user selects a new avatar from a variety of options.</li> <li>4. The system successfully updates the user's profile picture.</li> <li>5. The user explores the customization options for her profile, choosing a background photo and theme.</li> <li>6. The system implements the user's choice without errors.</li> <li>7. The user goes to the Badges section to view and view the virtual badges she has earned.</li> <li>8. The user accesses the privacy settings and edits them according to her preferences.</li> <li>9. The system updates changes to privacy settings immediately.</li> <li>10. The user explores the user engagement history section to review her past charitable activities.</li> <li>11. The user decided to switch to a different theme to update the look of her profile.</li> <li>12. The user chooses a new theme, and the system applies the changes seamlessly without loading errors.</li> </ol>

<b>Postconditions</b>	<ul style="list-style-type: none"> <li>the user's profile has been customized with a new avatar, background and theme.</li> <li>The system uploads a comprehensive timeline detailing her donations, and her achievements.</li> <li>The system displays an attractive screen that displays the user's contributions and achievements through the badges she obtained.</li> </ul>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1. The user may encounter a technical problem while updating her avatar. The system provides troubleshooting options or prompts the user to try again.</li> <li>2. The user faces difficulty in adjusting privacy settings.</li> <li>3. User interaction history section failed to load due to a temporary issue. The system informs the user about the problem and suggests trying again later or provides an alternative way to access the information.</li> </ol>

Figure 48 Profile Customization UseCase

#### 5.1.4 Using The Shop:

Participating actors	User, System
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The user has a stable internet connection.</li> <li>• The user's device supports the visual elements and interactive elements of the animated backgrounds and avatars.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. The system notifies the user about the Profile Customization Shop, highlighting new features like animated backgrounds and avatars.</li> <li>2. Information about the available virtual currency is displayed in the notification.</li> <li>3. The user goes to the "Profile Customization Shop" section within the system.</li> <li>4. The user goes to the Avatars section and chooses a special avatar that matches her interests.</li> <li>5. Each item has a clearly defined virtual currency cost.</li> <li>6. The system displays the cumulative virtual currency cost before making a virtual currency purchase.</li> <li>7. The user confirms the purchase of virtual currency, and the system debits the virtual currency from the user's account.</li> </ol>

<b>Postconditions</b>	<ul style="list-style-type: none"> <li>Virtual currency is debited from the user's account.</li> <li>The user profile is customized with the animated background and selected avatar.</li> </ul>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>The user may encounter a technical problem while updating her avatar. The system provides troubleshooting options or prompts the user to try again.</li> </ol>

Figure 49 Using Shop UseCase

### 5.1.5 Posting Campaign News:

<b>Participating actors</b>	Donors, campaign admin
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>Donors who follow the campaign are registered users of the system with valid login .</li> <li>The "Palestine Charity" system is operational and allows admins to post updates.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>The admin moves to the list of campaigns</li> <li>The admin selects the specific campaign for which he wants to publish an update</li> <li>The admin will have access to the campaign dashboard where he can update the campaign information.</li> <li>The admin writes information about recent accomplishments or challenges the campaign has faced.</li> </ol>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>The donors engage with the update, expressing appreciation through shares and likes.</li> <li>The campaign admin successfully communicated recent developments and strengthened the sense of community among donors.</li> </ul>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>The campaign admin is experiencing technical problems while trying to publish the update. The application displays an error message and asks him to try again or contact support.</li> <li>Some donors may provide comments or ask questions in response to the update. The campaign administrator may need to respond to these interactions on the campaign dashboard.</li> </ol>

Figure 50 Posting Campaign News UseCase

### 5.1.6 :Sub-campaign creation:

<b>Participating actors</b>	Influencers, organization owner, admin
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>The "Palestine Charity" system supports the creation of sub-campaigns.</li> </ul>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>The Influencer logs into the "Palestine Charity" system.</li> <li>The influencer creates the sub-campaign and sets a specific target amount for it.</li> <li>The influencer submits the sub-campaign for approval to the organization owner and admin, providing a comprehensive overview of its goals and expected impact.</li> <li>The influencer shares her sub-campaign on her social media accounts, encouraging followers to contribute.</li> <li>The influencer's followers make donations to her campaign, and the system provides real-time updates on progress toward her campaign goal.</li> <li>Donors who donate to this sub-campaign receive badges, virtual currency, and recognition for their contributions.</li> <li>An influencer keeps donors informed of the impact they are making, sharing stories and photos the target receives.</li> <li>The sub-campaign reaches its goal, contributing to the success of the broader campaign.</li> </ol>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>Inflouencer's sub-campaign has been successfully created and approved.</li> <li>Followers (donors) make donations and earn virtual badges.</li> <li>Funds raised contribute to the wider campaign, making a significant impact.</li> <li>Influencers and donors get recognition for their charitable efforts.</li> </ul>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>The sub-campaign may face delays in approval.</li> <li>If the sub-campaign faces challenges in reaching its goal, the influencer may introduce new incentives or reach out to her followers for support.</li> </ol>

Figure 51 Sub-Campaign Creation UseCase



# PalestineCharityFinalReport1.docx

## ORIGINALITY REPORT



## PRIMARY SOURCES

- |   |   |      |
|---|---|------|
| 1 | Al-Ahmad, Bilal Ibrahim. "Domain ontology based detection approach to identify effect types of security requirements upon functional requirements.", Proquest, 2015.<br>Publication | <1 % |
| 2 | <a href="http://www.mongodb.com">www.mongodb.com</a><br>Internet Source   | <1 % |
| 3 | <a href="http://gist.github.com">gist.github.com</a><br>Internet Source   | <1 % |
| 4 | Hoda Golrang, Ehram Safari. "Applying gamification design to a donation-based crowdfunding platform for improving user engagement", Entertainment Computing, 2021<br>Publication    | <1 % |
| 5 | <a href="http://dabar.srce.hr">dabar.srce.hr</a><br>Internet Source   | <1 % |
| 6 | <a href="http://www.packtpub.com">www.packtpub.com</a><br>Internet Source   | <1 % |
| 7 | <a href="http://repository.lib.cuhk.edu.hk">repository.lib.cuhk.edu.hk</a>  |      |

Internet Source

<1 %

8

github.com

Internet Source

<1 %

9

export.arxiv.org

Internet Source

<1 %

10

doria.fi

Internet Source

<1 %

11

jvek.weebly.com

Internet Source

<1 %

12

dev.to

Internet Source

<1 %

13

repository.mercubuana.ac.id

Internet Source

<1 %

14

dl.gi.de

Internet Source

<1 %

15

www.cmarix.com

Internet Source

<1 %

16

Bassoumi, Youssef. "Course Management Support Application-iMaster.Schedule",  
Instituto Politecnico de Braganca (Portugal),  
2024

Publication

<1 %

17

msatechnosoft.in

Internet Source

<1 %

18	<a href="https://designsystem.gov.ae">designsystem.gov.ae</a>	<1 %
19	<a href="https://www.mun.ca">www.mun.ca</a>	<1 %
20	<a href="https://www.researchgate.net">www.researchgate.net</a>	<1 %
21	<a href="https://dspace.plymouth.ac.uk">dspace.plymouth.ac.uk</a>	<1 %
22	Belhaj, Bechir. "Course Management Support Application-iMaster. Report", Instituto Politecnico de Braganca (Portugal), 2024 Publication	<1 %
23	<a href="https://informatika.un&gt;tag-sby.ac.id">informatika.un&gt;tag-sby.ac.id</a>	<1 %
24	<a href="https://repository.sustech.edu">repository.sustech.edu</a>	<1 %
25	<a href="https://9pdf.net">9pdf.net</a>	<1 %
26	<a href="https://digilib.uinsgd.ac.id">digilib.uinsgd.ac.id</a>	<1 %
27	<a href="https://sahrис.sahra.org.za">sahrис.sahra.org.za</a>	<1 %
28	<a href="https://www.h2020fairshare.eu">www.h2020fairshare.eu</a>	<1 %

---

Exclude quotes

Off

Exclude bibliography

Off

Exclude matches

< 4 words

# PalestineCharityFinalReport1.docx

---

PAGE 1

---

PAGE 2

---

PAGE 3

---

PAGE 4

---

PAGE 5

---

PAGE 6

---

PAGE 7

---

PAGE 8

---

PAGE 9

---

PAGE 10

---

PAGE 11

---

PAGE 12

---

PAGE 13

---

PAGE 14

---

PAGE 15

---

PAGE 16

---

PAGE 17

---

PAGE 18

---

PAGE 19

---

PAGE 20

---

PAGE 21

---

PAGE 22

---

PAGE 23

---

PAGE 24

---

PAGE 25

---

---

PAGE 26

---

PAGE 27

---

PAGE 28

---

PAGE 29

---

PAGE 30

---

PAGE 31

---

PAGE 32

---

PAGE 33

---

PAGE 34

---

PAGE 35

---

PAGE 36

---

PAGE 37

---

PAGE 38

---

PAGE 39

---

PAGE 40

---

PAGE 41

---

PAGE 42

---

PAGE 43

---

PAGE 44

---

PAGE 45

---

PAGE 46

---

PAGE 47

---

PAGE 48

---

PAGE 49

---

PAGE 50

---

PAGE 51

---

PAGE 52

---

PAGE 53

---

PAGE 54

---

PAGE 55

---

PAGE 56

---

PAGE 57

---

PAGE 58

---

PAGE 59

---

PAGE 60

---

PAGE 61

---

PAGE 62

---

PAGE 63

---