Acknowledgment

First of all, we would like to thank our almighty God who gives us love, patience, healthy, wisdom, and the ability to finish this semester's project.

Our heartfelt appreciation goes to our adviser **Felix Edesa** for his guidance, valuable suggestions, moral support, and constant encouragement in the documentation part and also for constructive opinion and willingness to participate in each part of our documentation and his effective direction.

We would like to thank many people for their kindness and assistance during our project work. We truly appreciate the support of the people at the Mekedonia and Muday for their willingness to interview, patience in answering our numerous questions.

Table of Contents

[**List of tables** IV](#_Toc75866773)

[**List of figures** V](#_Toc75866774)

[**Abbreviations** VI](#_Toc75866775)

[**Abstract** VII](#_Toc75866776)

[**CHAPTER ONE: INTRODUCTION** 1](#_Toc75866777)

[1.1 Background of the organization 1](#_Toc75866778)

[1.2 Statement of the problem 1](#_Toc75866779)

[1.2.1 Existing System 1](#_Toc75866780)

[1.2.2 Major problem of the existing system 2](#_Toc75866781)

[1.2.3 Proposed system 2](#_Toc75866782)

[1.2.4 Advantage of the proposed system 3](#_Toc75866783)

[1.3 Motivation 4](#_Toc75866784)

[1.4 Scope and limitation of the project 4](#_Toc75866785)

[1.4.1 Scope of the project 4](#_Toc75866786)

[1.4.2 Limitation of the project 4](#_Toc75866787)

[1.5 Objectives 4](#_Toc75866788)

[1.5.1 General objective 4](#_Toc75866789)

[1.5.2 Specific objectives 4](#_Toc75866790)

[1.6 Methodology 5](#_Toc75866791)

[1.6.1 Data collection method 5](#_Toc75866792)

[1.6.2 System design and analysis tools 6](#_Toc75866793)

[**CHAPTER TWO: SYSTEM REQUIREMENT SPECIFICATION** 7](#_Toc75866794)

[2.1 Background Overview 7](#_Toc75866795)

[2.2 Functional Requirement 7](#_Toc75866796)

[2.3 Non-functional Requirement 8](#_Toc75866797)

[2.4 Feasibility Study 10](#_Toc75866798)

[**CHAPTER THREE: SYSTEM ANALYSIS AND MODELING** 13](#_Toc75866799)

[3.1 Overview 13](#_Toc75866800)

[3.2 Scenario-Based Modeling 13](#_Toc75866801)

[3.2.1 Use-case identification 13](#_Toc75866802)

[3.2.2 Actor identification and description 14](#_Toc75866803)

[3.2.3 Use-case Diagram 15](#_Toc75866804)

[3.2.4 Activity diagram 26](#_Toc75866805)

[3.3 Behavioral Modeling 30](#_Toc75866806)

[3.3.1 Sequence diagram 30](#_Toc75866807)

[3.3.2 State diagram 34](#_Toc75866808)

[3.4 Class Modeling 37](#_Toc75866809)

[3.4.1 Class diagram 37](#_Toc75866810)

[**Chapter Four: System Design** 38](#_Toc75866811)

[4.1 Overview 38](#_Toc75866812)

[4.2.2 Component Identification and Description 40](#_Toc75866813)

[4.2.3 Component Diagram 41](#_Toc75866814)

[4.3 Architecture of the System 42](#_Toc75866815)

[4.4 Database Design 44](#_Toc75866816)

[4.5 User Interface Design 46](#_Toc75866817)

[References 50](#_Toc75866818)

[Appendix 51](#_Toc75866819)

# 

# **List of tables**

[Table 1: System design and analysis tools 6](#_Toc75866820)

[Table 2: System development tools. 6](#_Toc75866821)

[Table 3: Use-case description for login. 16](#_Toc75866822)

[Table 4: Use-case description for Send profile for approval. 17](#_Toc75866823)

[Table 5: Use-case description for post needs. 17](#_Toc75866824)

[Table 6: Use-case description for register. 18](#_Toc75866825)

[Table 7: Use-case description for Make promise donation. 19](#_Toc75866826)

[Table 8: Use-case description for Make appointment. 20](#_Toc75866827)

[Table 9: Use-case description for Update profile. 21](#_Toc75866828)

[Table 10: Use-case description for Reject profile. 21](#_Toc75866829)

[Table 11: Use-case description for delete. 22](#_Toc75866830)

[Table 12: Use-case description for Approve profile. 22](#_Toc75866831)

[Table 13: Use-case description for Send profile for approval. 23](#_Toc75866832)

[Table 14: Use-case description for Approve profile. 23](#_Toc75866833)

[Table 15: Use-case description for Add agent. 24](#_Toc75866834)

[Table 16: Use-case description for Search. 24](#_Toc75866835)

[Table 17: Use-case description for Make online donation. 25](#_Toc75866836)

# **List of figures**

[Figure 1: Use-case diagram for the proposed system 15](#_Toc75866861)

[Figure 2:The Activity diagram for System admin. 27](#_Toc75866862)

[Figure 3:Activity diagram for organization officer 28](#_Toc75866863)

[Figure 4:Activity diagram for the donor. 29](#_Toc75866864)

[Figure 5:The sequence diagram for the organization officer. 31](#_Toc75866865)

[Figure 6:The sequence diagram for a donor. 33](#_Toc75866866)

[Figure 7:The State diagram for organization officer. 35](#_Toc75866867)

[Figure 8:The state diagram for a donor. 36](#_Toc75866868)

[Figure 9:class diagram. 37](#_Toc75866869)

[Figure 10:Component diagram 41](#_Toc75866870)

[Figure 11:Deployment diagram for the proposed system 41](#_Toc75866871)

[Figure 12:user interaction 42](#_Toc75866872)

[Figure 13: Architecture of the proposed system 43](#_Toc75866873)

[Figure 14: Database diagram 45](#_Toc75866874)

[Figure 15: Home Page 47](#_Toc75866875)

[Figure 16:Register Page 48](#_Toc75866876)

[Figure 17: Login page 49](#_Toc75866877)

# 

# **Abbreviations**

**GB Gigabyte**

**GIT Global Information Tracker**

**JWT JSON Web Token**

**HTTP HyperText Markup Language**

**PAYPAL Electronic Payment Website**

**EBAY Electronic Bay**

**ORG Organization**

**DOS Denial-of-service**

**SSL Secure Socket Layer**

**HMAC Hash-based Message Authentication Code**

**BLL** **Business Logic Layer**

**DAL Data Access Layer**

**Adobe XD Adobe Experience Design**

# **Abstract**

Many people want to donate and help charity organizations but they don't know, how to go about it, where to find such organizations on their choice of action, what the available ways are, and so on. Charity organizations also face many hardships in their advertising and fundraising activities. Hence, this web-based charity system addresses this issue by developing a web-based charity system as a hub of different charity organizations categorized based on their course of charity. It provides donors full information about each organization and ways to donate at any time from anywhere.

# **CHAPTER ONE: INTRODUCTION**

## 

## 1.1 Background of the organization

Charity is a fundamental norm of Ethiopians which is deeply waved into the culture. This act of philanthropy is advocated in every culture, religion, and value that exists in Ethiopia. Simply, we can state that it is part of what is being an Ethiopian. This adorable social value is also witnessed by foreigners as a manner of Ethiopians.

For centuries, charitable giving has been practiced in a traditional way where philanthropists directly provide money to those in need have support who usually seat on streets or public places. This mode of charity still constitutes the major share of charitable giving. However, nowadays, several local charity organizations are emerging in Ethiopia. This organization created a better platform where supporters can give their charity and those in need can get support. The existing number of charity organizations is also increasing since the belief in the concept of an organized form of charity is now being accepted widely.

Despite getting popularity, these charity organizations have several limitations especially in the way of getting donations. They have an inefficient mechanism of advertisement and collecting money and materials from donors. Consequently, the majority of them are financially unstable. More specifically, these organizations are not easily accessible for supporters and fundraisers in terms of physical distance and time. Moreover, most of these organizations lack a well-established department that promotes their services and announces the materials needed.

## 1.2 Statement of the problem

## 1.2.1 Existing System

We found some well-designed and established websites which are operating abroad took to match good practices. Just to name a few charhub.org, nonprofithub.org, charitynavigator.org, etc.

Since 2001**,** charity navigator has been empowering millions of donors by providing them with free access to data, tools, and resources to guide philanthropic decision-making. Through our ratings, nonprofits are equipped with the nonprofit sector's premier trust indicator and a powerful platform to raise awareness and funds [1].

To our knowledge, no hub serves as a place for all charitable organizations that can be found in Ethiopia. Some charity organizations have a basic promotional website which is not nearly enough for getting any charities directly through their website. So by combining the experience of those aboard charity hub with our countries demand a centralized hub we think it would be a perfect project and help those charity organizations in the way.

### 1.2.2 Major problem of the existing system

* No way enables donors to choose between charity organizations for their donation.
* The donation process is space and time-bounded.
* It focuses only on monetary donations.
* It does not encourage materials and labor donation.
* It does not provide enough and trustworthy information about a charity organization.
* It is a tedious and fatiguing job to perform.
* The way of collecting money is not participatory for all persons.
* The fraud person can collect money for their interest.

### 1.2.3 Proposed system

* The system will be able to supply solutions to the fundraising problem of charity organization’s main issues by first enabling individuals to donate online instead of calling, which is seen to be beneficial for both donors and charity organizations.
* Donors can also read about the charity organization’s history, and the variety of services offered and the Helps they gave out all on the web.
* The system will provide enough information about charity organizations' financial and kind or material needs for their service.
* The system will advertise any project which is started by those charity organizations to upgrade or expand their service.
* The system enables the donors can make a promise donation.
* The system will give full information about contacts, branches, email, location of those organizations.
* The system enables supporters to deliver their materials support from where they are through agents of these charity organizations.
* The system lets to book their appointment for those who want to support personally their support could be labor work like washing clothes or professional support like psychological consultancy.
* Moreover, the methodologies that are followed to gather support and address sufficient information are bashful and retarded.

### Advantage of the proposed system

* It encourages the charity donation to rely mainly on technology to publicize their firm and communicate with donors/individuals.
* It will provide the latest information about the status of these charity organizations.
* It is multi optional donation type like money, material, and kind.
* It increases trustworthiness in support gatherings activities.
* It saved the time of the donors on the donation process.
* It is more accessible for all people who have a mobile phone or a computer with an internet connection.
* It is participatory for all types of donors such as money donators, material donators.
* It facilitates material or kind support.
* It is a legal and recognized system.
* The donors can easily get all local charity organizations in one system and donate easily.

## 1.3 Motivation

The motivation why we started this project is because of the lack of a web-based charity system in Ethiopia, and it is known that charity is blessed work that deserves everyone's support. The measures that are taken to do so are under what is needed. Even there is no such software product to help this great activity.

1.4 Scope and limitation of the project

1.4.1 Scope of the project

This project is intended to create a web-based charity system for local charity organizations in Ethiopia and the donors can have multiple options to help these organizations by money, material, and kind. In addition to this donors can make a promise donation and make an appointment to these organizations before they go to these charity organizations and include all local charity organizations in a hub website.

### 1.4.2 Limitation of the project

* The proposed system will not work without an internet connection.
* Security may be an issue.

## Objectives

## 1.5.1 General objective

The general objective of this project is to develop a web-based charity system.

### 1.5.2 Specific objectives

* + To develop a system that acts as a hub for all local charity organizations.
  + To develop a system that enables a donor can donate money online through the internet.
  + To develop a system that enables a donor can donate materials.
  + To develop a system that enables a donor can support charity organizations by their profession.
  + To develop a system that provides full information of charity organizations.
  + To design a system that provides a set of charity organizations in different categories.
  + To develop a system that enables donors to donate to more than one charity organization without changing the website.
  + To develop a system that enables automated report generation.
  + To develop a system that enables a promise donation for charity organizations.
  + To develop a system that enables online appointment setting for events.

## Methodology

1.6.1 Data collection method

The requirement elicitation process was conducted by interviews and observations. We interviewed major charity organizations. We observe a lot of disabled and needy peoples in charity organizations and those peoples are in a critical situation .those people need food, shelter, clothes, and medicine, and mentor. The main methodologies used in our data collection progress were conducting interviews and searching on the internet to get some additional information.

we have interviewed two local charity organizations. namely: Mekedonia and muday charity association. Mekedonia helps more than 5000 elders and people with a mental disorder whereas muday supports 450 mothers and 600 children. answering two core questions was the main objective of the interview. Those questions are:-

1. How do you raise money?
2. How do you promote charity activities?

Asking these questions, and we get the following answers:-

* On Fundraising:- mekedonia uses SMS and a direct donation from visitors and muday uses some income-generating activities like selling enjera and cultural handcrafts work as well as a direct donation from visitors to elevate its finance.
* On Promotion:- mekedonia uses flyers, banners, brochures, tv and radio advertisements. they tried to use social media that is Facebook and youtube, but as they told us, it has been hacked and also around 15 fake accounts were launched in their name so they stalled it. Muday also uses the same ways as mekedonia does.

Therefore our main findings on the above discussion are:

Those organizations have no reliable and secured electronic (web-based) system for fundraising and advertising their charity activities.

There is no local platform for a donor that provides them with a set of charity organizations with different categories and options.

### System design and analysis tools

Table 1: System design and analysis tools

|  |  |
| --- | --- |
| Documentation | Ms word |
| System Design and Modeling | Enterprise Architect |
| Database Design | MongoDB |
| Interface Design | Adobe XD |

* + 1. System development tools

Table 2: System development tools.

|  |  |
| --- | --- |
| Graphical Interface Framework | Reactjs |
| Database | MongoDB |
| Application Framework | NodeJs |
| Version Control Environment | Git |
| Web Server | Node.js |
| API Testing | Postman |

**CHAPTER TWO: SYSTEM REQUIREMENT SPECIFICATION**

## 2.1 Background Overview

Requirement analysis is a process of determining users' expectations from a system, it identifies and analyses the requirements for the new system using different models. To develop an achievable system that meets its users' needs, we have gathered requirements from various charity organizations and we have looked at different existing systems’ documentation which is relating to our project. So based on our gatherings we have identified the following functional and non-functional requirements.

## 2.2 Functional Requirement

Functional requirements define what is expected of a system or its components; they describe the behavior of a system by describing what it is required to do. The proposed system deals with online fundraising for the charity organization by collecting money, material, and professional help from volunteer companies and individuals.

* The system shall record all changes made as history for each charity organization.
* The system shall enable the registration of new donors.
* The system shall display the contact information of the charity organization.
* The system shall display a simple note on the current status of each charity organization.
* The system shall enable donors to donate money, materials.
* The system shall create a new profile for the new charity organization.
* The system shall enable to deletion of an existing charity organization profile.
* The system shall enable to update of charity organization information.
* The system shall give information on how to support charity using this platform.
* The system lets search for charity organizations.
* The system shall enable you to make an appointment with the desired charity organization
* The system shall send confirmation notifications for donors.
* The system shall promote ongoing projects of charity organizations.

## 2.3 Non-functional Requirement

* Scalability

As our service gets some traction and many people are drawn to it. It is expected to serve thousands of customers at once. We will build these things in mind. Node.js comes to be the best choice out in the market

Node.js popularity grew in the past years since 2017 this is because of its easy ways to scale. Developing a server-side application can be tricky, particularly when it has to be scalable. Node.js uses a single thread to handle non-blocking I/O calls, which means that it’ll take fewer resources for the application to accept concurrent connections compared to traditional approaches[2].

* Reliability

Considering our project's sensitive issue reliability is the number one thing we cannot compromise on. To deliver a high degree of confidentiality to our customers we considered aspects of the backend technology we used and the host.

Regarding our customer's sensitive data and privacy, we have every reason to be concerned about data breaches and other forms of cyber-crimes as the internet is not a safe place. This is why a secure and reliable web host is important. We make sure the host is as reliable and secure as possible by taking some measures.

* The web host runs on high-quality SSL protocols and certificates
* The host provides round the clock technical support
* The host offers a manual reboot service
* The web host associates with high-quality domain names
* A security expert vouches for the web host service
* Usability

Considering our user age( 20-60) and cultural demographic, we have to make the system as easy and simple as possible. That is why we choose the best front-end development technology.

In modern JavaScript creation, React.js is one of the most developmentally important frontend libraries you can use. It helps you to create highly reactive user interfaces that have a user experience that is fast and mobile-app-like.

* Maintainability

There is no perfect world. Things can go wrong and will go wrong, errors will happen. The best thing we can do is to minimize the risk and find a way to react fast it happen. To do that we will write a simple and maintainable code by:-

Recognize accidental complexity: Techniques that help me reduce complexity are declarative and [functional programming](https://codeburst.io/functional-programming-in-javascript-e57e7e28c0e5), and choosing composition over inheritance.

Start simple, build an evolvable architecture: Build the simplest thing that will work KISS(‘*keep it simple, stupid,’*), and keep the YAGNI(*"You aren't gonna need it"*) principle in mind. Only start creating abstractions once the code starts hurting to maintain or extend further. As soon as you see yourself repeating a piece of code more than two times, it’s time to see if you can create an abstraction to reuse the code.

* Security

Our system uses the best authentication and authorization ways to develop a system. Although there are many authentication methods there the one we are going to use is JSON Web Token(JWT). It is a state-less solution for authentication so there is no need to store any session data on the server which is perfect for restful APIs like the one we are building. because stateful API should always be stateless[3].

Node.js provides a built-in module called crypto that you can use to encrypt and decrypt strings, numbers, buffers, streams, and more. This module offers cryptographic functionality that includes a set of wrappers for OpenSSL's hash, HMAC, cipher, decipher, sign, and verify functions[4].

To prevent a compromised database(an attacker gained access to our database)

* We Strongly encrypt passwords with salt and hash (bcypt)
* We Strongly encrypt password reset token (SHA 256).

Even if the attacker gained access to our database he/she couldn't steal the user's password or reset them. To prevent Brute Force attacks (the attacker try to guess a password by trying random passwords until he finds the right one).

* We use bcrypt(to make login requests slow).
* We Implement rate-limiting(express-rate-limit).
* We Implement maximum login attempts.

To prevent cross-site scripting(xss) attacks(when the attacker try to inject scripts to run his/her malicious code)

* We Store JWT in HTTPonly cookies(never store jwt in local storage).
* The browser can only receive and send the cookie can not access or modify it.
* We sanitize user input data.
* We set special HTTP headers(helmet package).

To Prevent Denial-of-service (Dos) attacks (the attacker sends so many requests that the server breaks down and the application becomes available).

* We Implement rate-limiting(express-rate-limit).
* We Limit body payload(in body-parser).
  1. Feasibility Study

A feasibility study considers a project's relevant factors to determine the likelihood of completing a project successfully. It tries to find out whether a project is viable or not, under certain circumstances. We have conducted the feasibility studies that are most relevant to our project.

* Technical Feasibility

This assessment focuses on the availability of technical issue which is capable of converting the proposed idea into a working system. Since for front end and back end, development, node.js and react.js frameworks are proposed and MongoDB is selected for database design. These things make development easy and possible. The system is applicable and easy to use for any ordinary person who has basic computer skills. and it needs only an OS and 2GB ram computer with 250GB hard disk which is the least specifications of available computers.

* Operational feasibility

The proposed system solves the bottlenecks stated in the statement of the problem section of this document. We have a willingness of charity organizations to support the system solution. The current mode of operation is suitable for the application of the proposed system.

* It increases the income of charity organizations by increasing promotion.
* The social acceptability of the system is also high.
* There is no direct or indirect resistance from charity organizations we have interviewed so far.
* It saves the time, and effort of donors because they can easily get information about charity organizations.

From the above parameters, we can deduce that it satisfies the requirements identified and have pros listed under the advantages section of this document. So it is likely that the system will be used if it is developed.

* Economic Feasibility

This could also be referred to as cost/benefit analysis. In the context of our proposed system:-

* The system does not require extra finance or material facilities beyond the simple computer, smartphone, and data connection that are available throughout every organization.
* The development phase of the system also does not require any finance since the skill and equipment which are needed for the development are on hand already.
* It reduces the existing manual effort and expense for promotion and fundraising activities.
* The only money the system need is for maintenance cost, connection cost, operational cost, and server rent cost.
* Schedule feasibility

Schedule feasibility is a study of whether the project will be completed within a given time box. Everything needed including team dedication is one hand. So it is confirmed as time-bound feasible.

# **CHAPTER THREE: SYSTEM ANALYSIS AND MODELING**

## 3.1 Overview

In this section, we represented the requirement analysis using different modeling approaches; Scenario-based modeling (Use cases, Use case Diagrams, Activity Diagrams), Behavioral modeling (State Diagram, Sequence Diagram), and Class-based modeling (Class Diagrams). We will analyze the requirement analysis and specifications of the system to identify the necessary elements and components for modeling a system that results in a system having high efficiency and effectiveness.

## Scenario-Based Modeling

3.2.1 Use-case identification

* Approve
* Reject charity org registration
* Login
* Create
* Update
* Delete
* post needs
* send profile for approval
* Register
* make promise donation
* View Charities
* make online donation
* search
* make appointment
* add agent

### 3.2.2 Actor identification and description

System Admin: System Admin is an individual who controls the system .he checks the validity of the organization that wants to be registered. He or she is expected to have some sort of legal knowledge about charity-related things like legal aspects.

Charity Organization officer: Charity Organization officer is an individual who represents the organization which has legal documents and approved by the concerned body of government

Donor: A donor is a person who is interested in making any kind of donation or service that a charity organization may need. The donor can be representative of different organizations like NGOs.

### Use-case Diagram

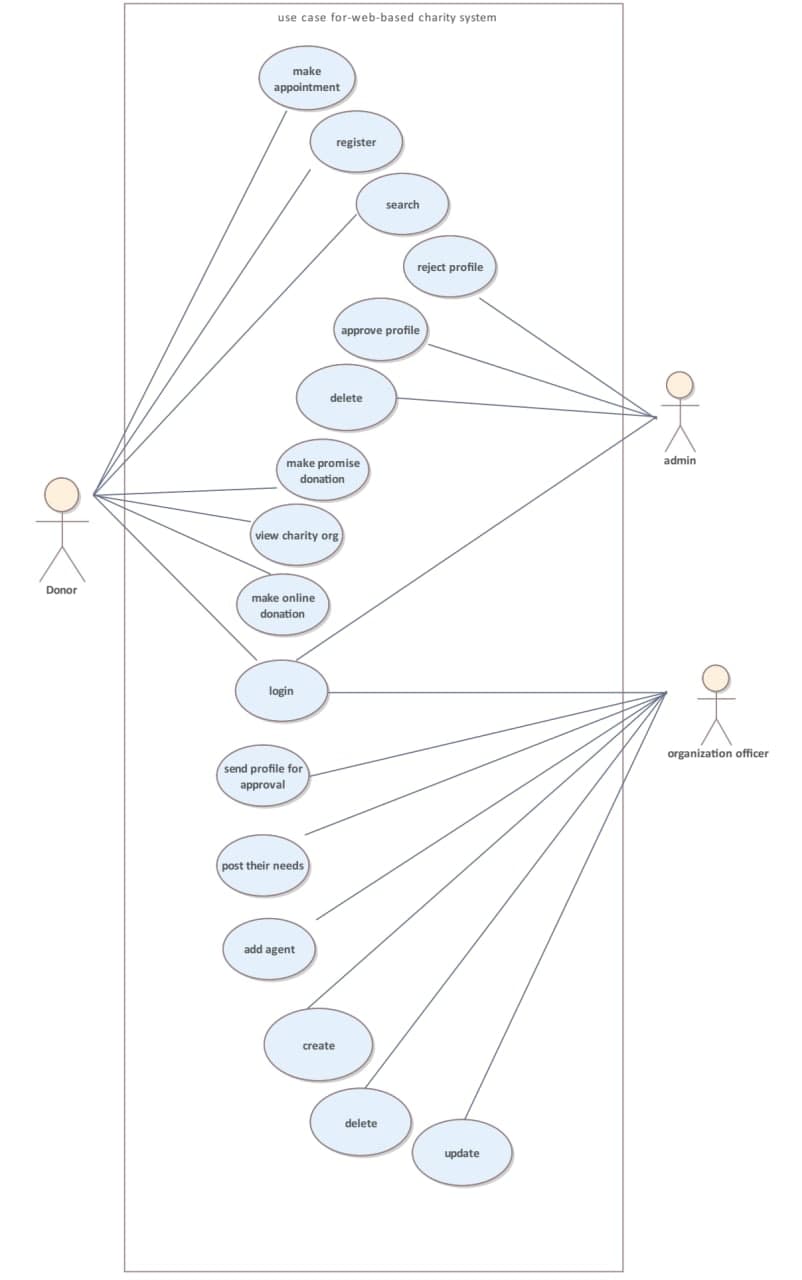


Figure 1: Use-case diagram for the proposed system

|  |  |
| --- | --- |
| Use Case | Log in |
| Primary Actor | System Admin/charity organization officer/donor |
| Goal In Context | Enable the actors to access the system |
| Preconditions | The agent has a valid username and password and is not already logged in |
| Trigger | The agent requires access to the system to perform their job |
| Scenario | 1. Enter the email address and password 2. Click login 3. Check for credential validity  * If it is invalid  1. Show error and enable to edit it 2. Enable the user to try again a couple of times  * If the user forgot their password  1. Click reset password 2. Enter email address 3. Press reset |

Table 3: Use-case description for login.

Table 4: Use-case description for Send profile for approval.

|  |  |
| --- | --- |
| Use Case | Send profile for approval |
| Primary Actor | Organization officer |
| Goal In Context | Get the organization profile approved |
| Preconditions | If some organization creates an account on the system and send for approval  The organization is not registered before |
| Trigger | Get approved and rejected |
| Scenario | 1. fill all the forms and upload all legal papers 2. Submit 3. Check for validity  * If forms are invalid  1. User edit error files and enter again 2. Resubmit |

Table 5: Use-case description for post needs.

|  |  |
| --- | --- |
| Use Case | Post needs |
| Primary Actor | Organization officer |
| Goal In Context | To get donations for what the organization is lacking at the moment |
| Preconditions | 1. The organization’s profile must be approved 2. The organization is logged in |
| Trigger | New projects come or there is some work to do |
| Scenario | 1) user list all its needs in items  2) post all needs  3) Check for validity   * If forms are invalid  1. User edit error files and enter again 2. Resubmit |

Table 6: Use-case description for register.

|  |  |
| --- | --- |
| Use Case | Register |
| Primary Actor | Donor |
| Goal In Context | Create a new account |
| Preconditions | The donor is not registered before |
| Trigger | Donor get access to the service |
| Scenario | 1. select sign up 2. Fill the form 3. Check for credential validity  * If it is invalid  1. Show error and enable to edit it 2. Enable the user to try again a couple of times |

Table 7: Use-case description for Make promise donation.

|  |  |
| --- | --- |
| Use Case | Make promise donation |
| Primary Actor | Donor |
| Goal In Context | Donate |
| Preconditions | The donor is registered  The donor must be logged in the system |
| Trigger | Donor gives donation |
| Scenario | 1. Select a charity organization to help 2. Select his/her way of helping  * If he/she select item donation  1. Get agent information 2. Send an email to the regional agent  * If it is a financial donation   If the donator domestic   1. Give charity bank information and redirect the page to its bank page   If abroad donator   1. Fill online payment form 2. Let the payment API deal with entered credential   If the donor is out of Ethiopia  Use API to donate |

Table 8: Use-case description for Make appointment.

|  |  |
| --- | --- |
| Use Case | Make appointment |
| Primary Actor | Donor |
| Goal In Context | To set a schedule to help or attend in the event |
| Preconditions | The donor is registered  The donor is logged in |
| Trigger | The donor will be remembered at the time when the appointment is set. |
| Scenario | 1. select the organization to help and attend the event 2. Enter the appropriate time 3. The organization will receive a message about an appointment |

Table 9: Use-case description for Update profile.

|  |  |
| --- | --- |
| Use Case | Update profile |
| Primary Actor | Donor/charity organization official |
| Goal In Context | Edit the pre-entered posts and status |
| Preconditions | The donor is registered  The donor is logged in |
| Trigger | The Agents profile is changed or updated |
| Scenario | 1. select posts and profile contents to update 2. Edit the profile in its content manner 3. Check for the validity of the edited file  * If the entered file is invalid  1. Show error and enable to edit it 2. Enable the user to try again a couple of times |

Table 10: Use-case description for Reject profile.

|  |  |
| --- | --- |
| Use Case | Reject profile |
| Primary Actor | System Admin |
| Goal In Context | To reject the invalid profile of the organization. |
| Preconditions | Admin is already logged in. |
| Trigger | The organization officer is denied to access the system. |
| Scenario | 1.system admin must be logged in to the system.  2.review profile sent from the organization officer.  3.select reject to reject the profile. |

Table 11: Use-case description for delete.

|  |  |
| --- | --- |
| Use Case | Delete |
| Primary Actor | System Admin |
| Goal In Context | To delete organizations and donor accounts and posts. |
| Preconditions | The admin must be logged in to the system |
| Trigger | Organizations or doners can not access the system. |
| Scenario | 1.system admin reviews the organization's status.   * If it is accumulated complaints on organization, * If the organization is closed or suspended its service. * And if the donor is invalid like terrorism etc   2.system admin select delete option |

Table 12: Use-case description for Approve profile.

|  |  |
| --- | --- |
| Use Case | Approve profile |
| Primary Actor | System admin |
| Goal In Context | Check the legality of a profile of the organization. |
| Preconditions | The system admin must be in the system /logged in. |
| Trigger | The organization can post, add agent, etc |
| Scenario | 1) the admin first logged in  2)the system admin review the profile sent from the organization officer  3)approved the profile |

Table 13: Use-case description for Send profile for approval.

|  |  |
| --- | --- |
| Use Case | Send profile for approval |
| Primary Actor | Charity organization |
| Goal In Context | Make a legit profile and get it approved |
| Preconditions | All forms and required documents must be summited |
| Trigger | New projects come or there is some work to do |
| Scenario | 1) add on the necessary files |

Table 14: Use-case description for Approve profile.

|  |  |
| --- | --- |
| Use Case | Approve profile |
| Primary Actor | System admin |
| Goal In Context | Check the legality of a profile of the organization. |
| Preconditions | The system admin must be in the system /logged in. |
| Trigger | The organization can post, add agent, etc |
| Scenario | 1) the admin first logged in  2)the system admin review the profile sent from the organization officer  3)approved the profile |

Table 15: Use-case description for Add agent.

|  |  |
| --- | --- |
| Use Case | Add agent |
| Primary Actor | Organization officer |
| Goal In Context | The organization officer add their agent information |
| Preconditions | The organization is registered  The organization must be logged in |
| Trigger | The organization agent information will be added to the system |
| Scenario | 1. the organization officer select add agent on its dashboard 2. the organization officer fill in all necessary information about the agent   3)Check for the validity   * If the entered data is invalid show an error and try again   4)If it is valid the information will add and agent are added to the system |

Table 16: Use-case description for Search.

|  |  |
| --- | --- |
| Use Case | Search |
| Primary Actor | Donor |
| Goal In Context | The donor can search organization in different parameter |
| Preconditions | The donor is registered.  The donor is logged in. |
| Trigger | The donor will get data that is related to its search result |
| Scenario | 1)type right over the search  2)click the search button  3) if the search match the result will display on the display area  Else the search is not matching msg will pop up |

Table 17: Use-case description for Make online donation.

|  |  |
| --- | --- |
| Use Case | Make online donation |
| Primary Actor | Donor |
| Goal In Context | A donor can donate online to the organization |
| Preconditions | The donor is registered.  The donor is logged in. |
| Trigger | A donor can donate whatever the donor want |
| Scenario | 1. select organization. 2. select contribute to the selected organization. 3. fill the donation form.   3)Check for the validity   * If the entered data is invalid show an error and try again. * If the entered data are valid.  1. And if the donor help in material and kind. 2. The donor will get agent information. 3. An email will send to the agent  * If the donor wants to help in financial support   + - If the doner is a domestic donor the system will redirect to an online banking page.     - If the donor out of country residents online payment form must be filed.   8)the confirmation will send to the donor. |

* + 1. Activity diagram

An activity diagram is a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. it deals with all types of flow control by using different elements such as fork, join. here in our system use an activity diagram to show the activities of doners, organization officers. activity diagram to show the activities of doners, organization officers.

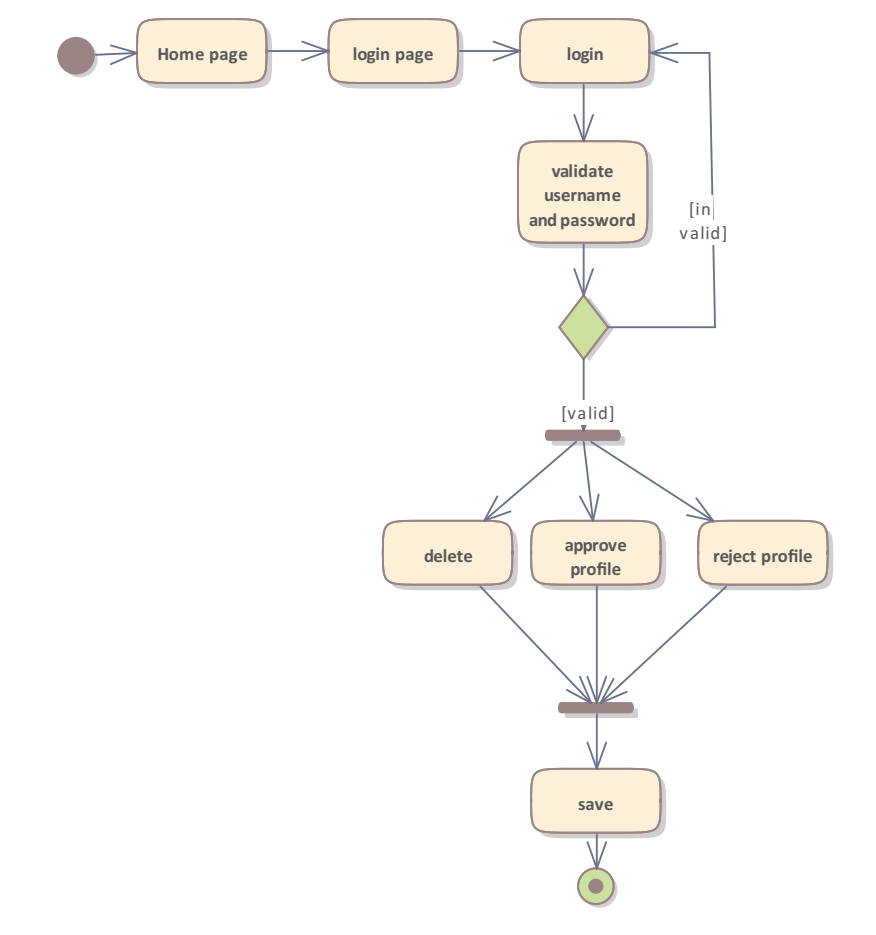


Figure 2:The Activity diagram for System admin.

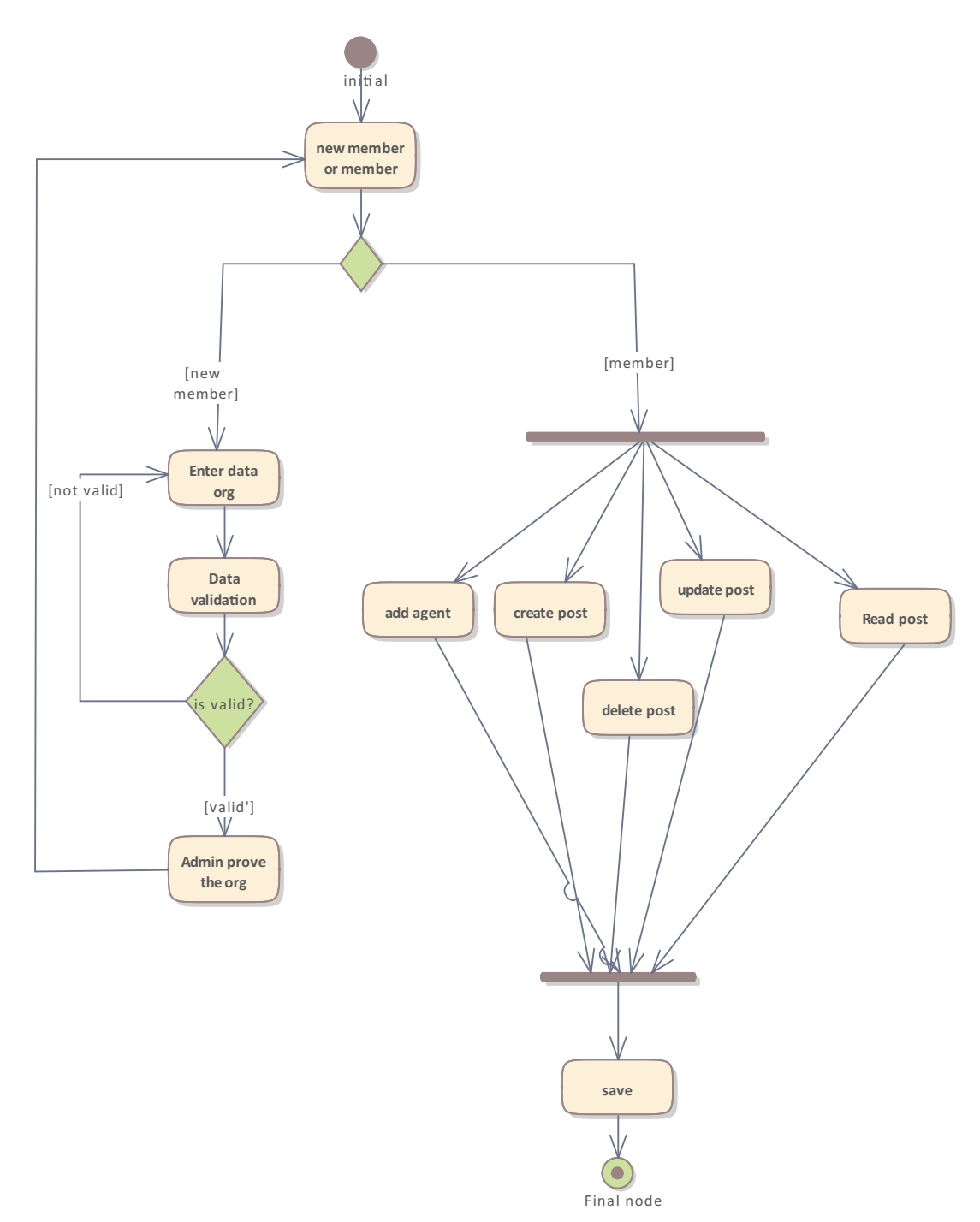


Figure 3:Activity diagram for organization officer

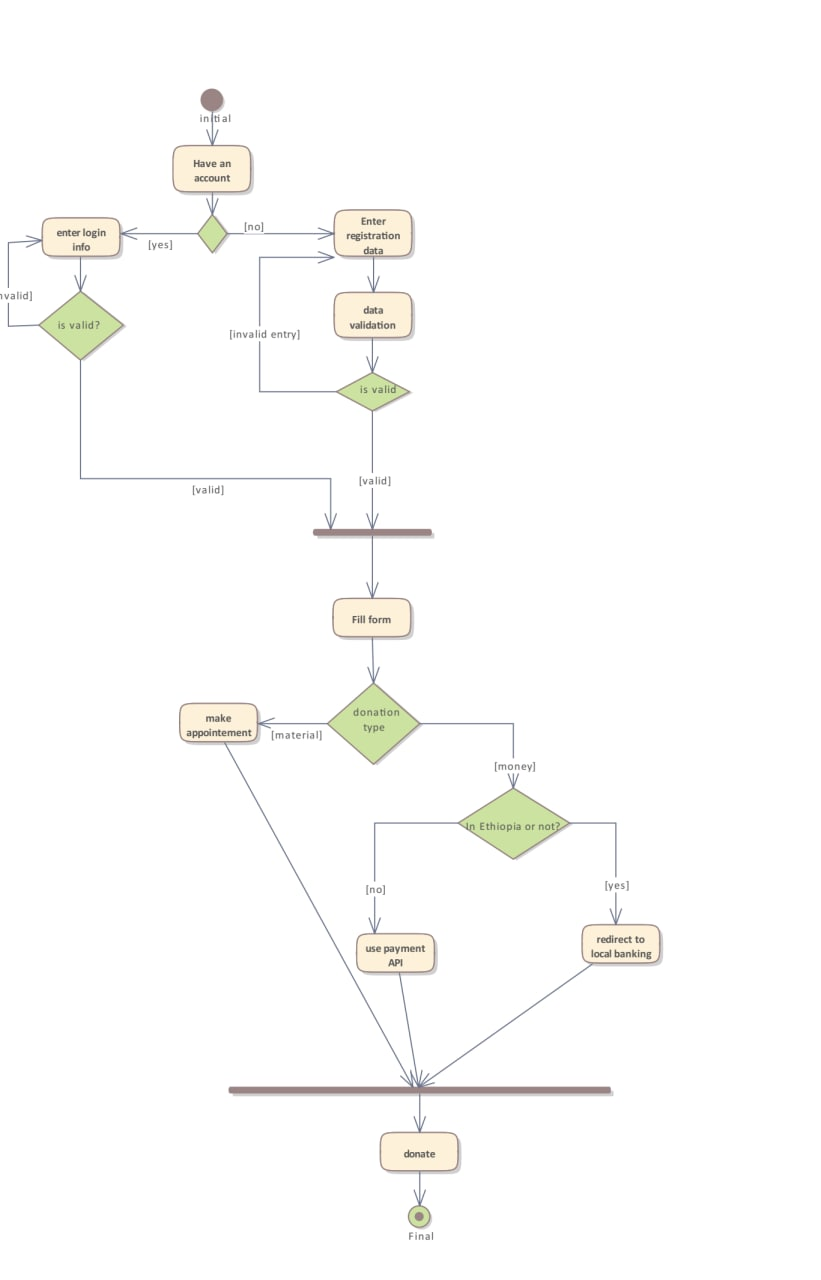
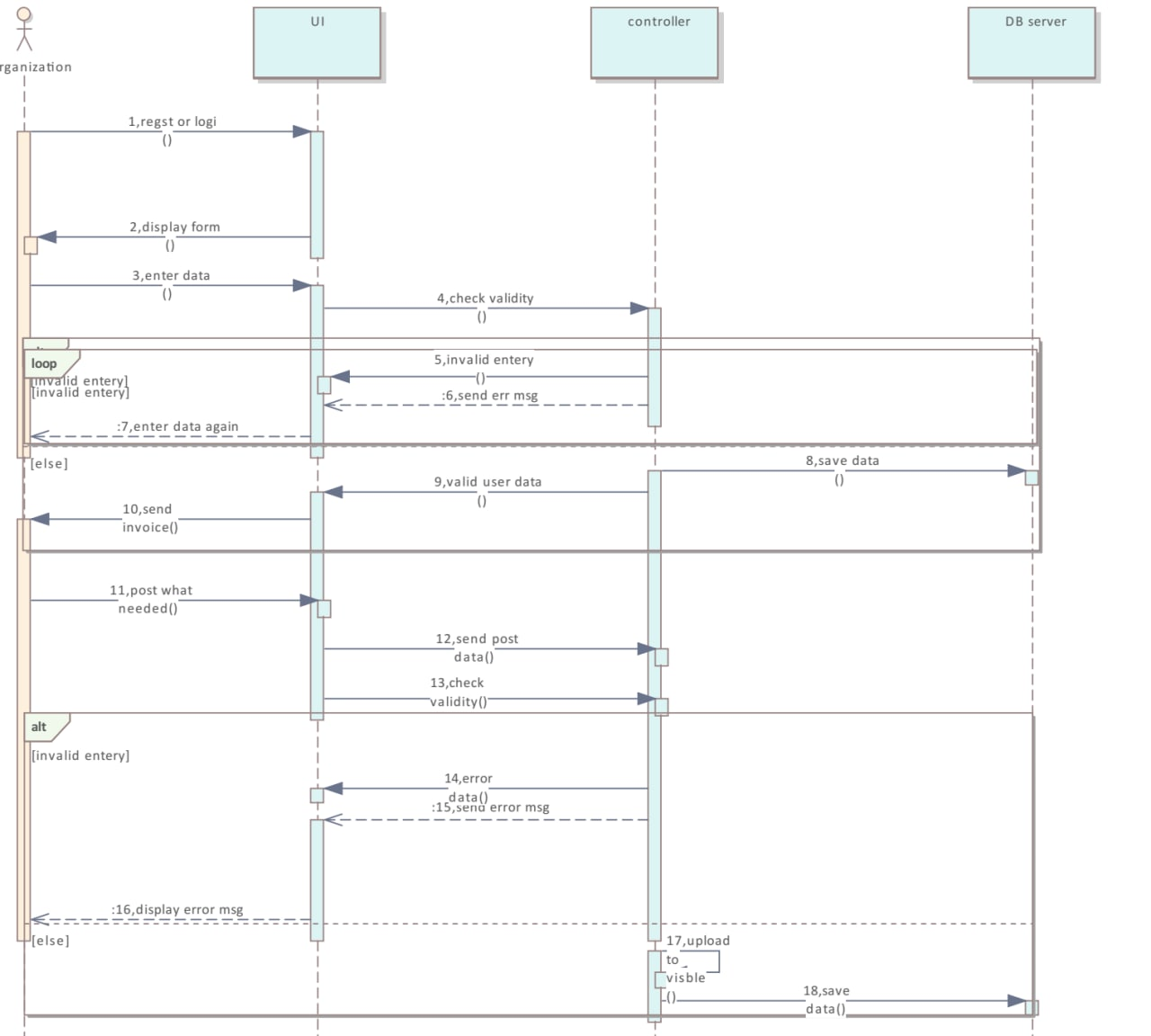


Figure 4:Activity diagram for the donor.

3.3 Behavioral Modeling

3.3.1 Sequence diagram

A sequence diagram is used for a particular scenario of a [use case](https://en.wikipedia.org/wiki/Use_case), the events that external actors generate, their order, and possible inter-system events.



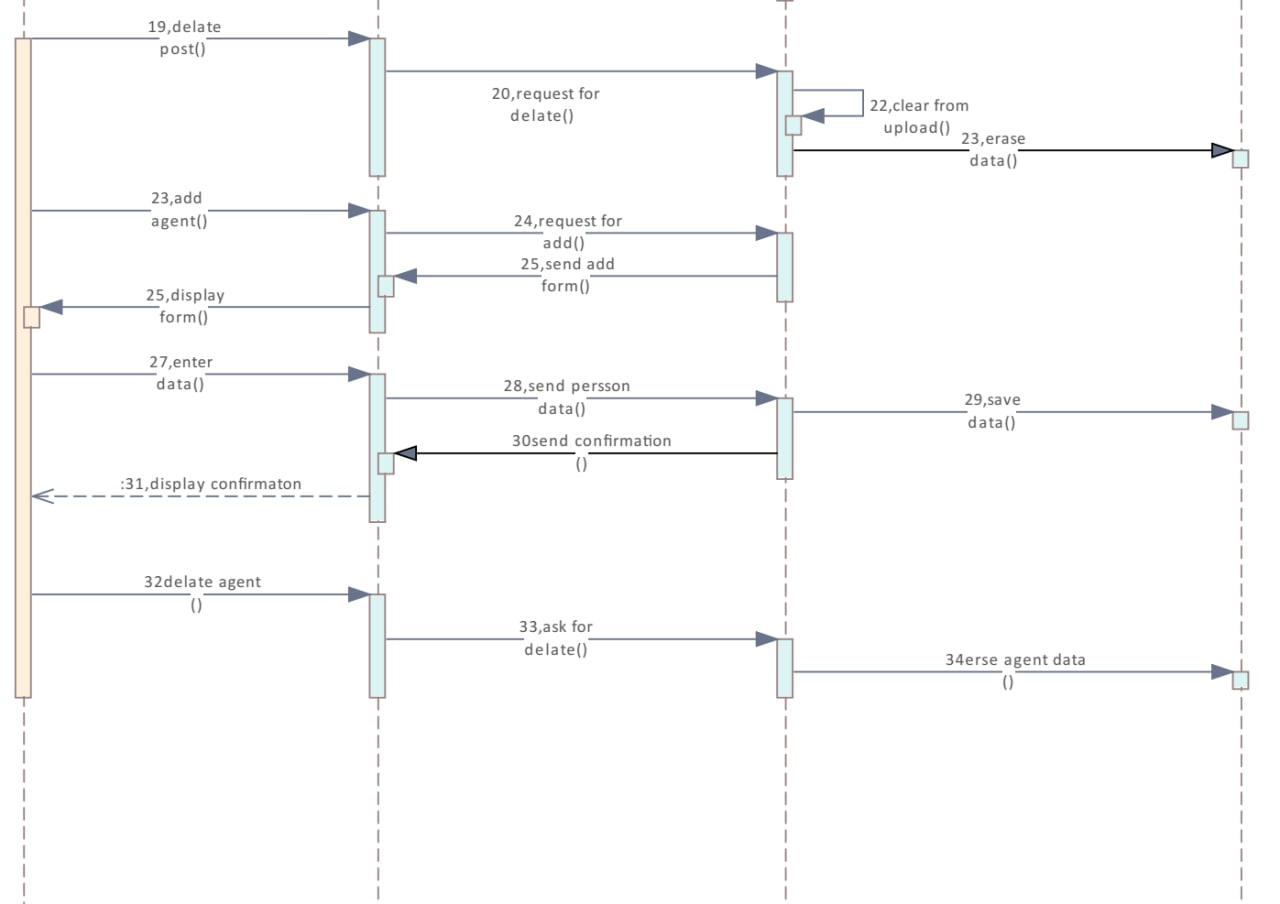


Figure 5:The sequence diagram for the organization officer.

Process: for an organization

1. register or login

2.display form

3.enter data

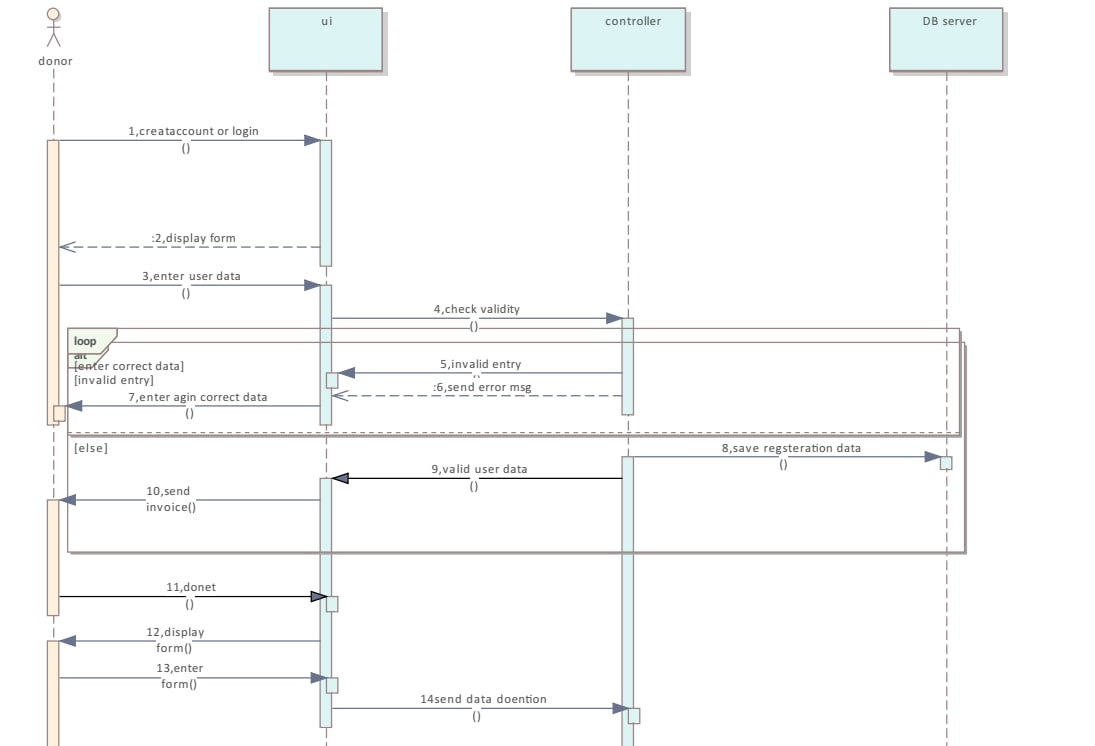
4.check validity

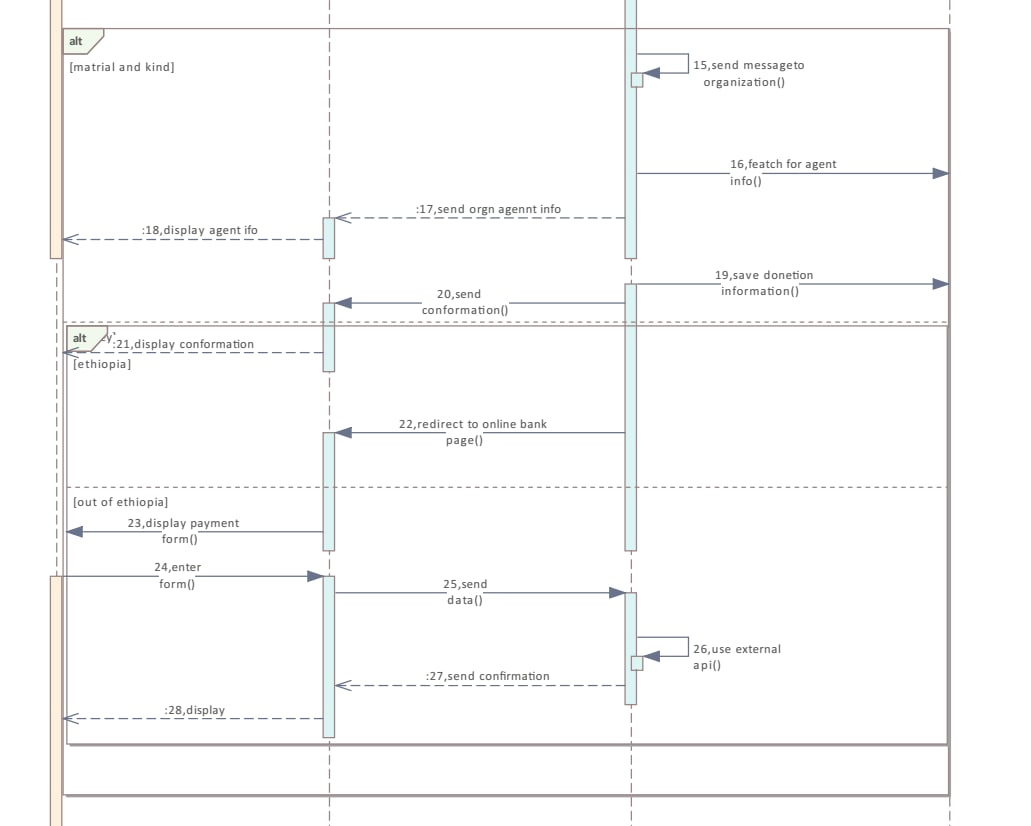
5.invalid entry

6.send err msg

7enter data again

1. save data
2. Valid user data
3. Send invoice
4. Post what needed
5. Send post data
6. Check validity
7. Error data
8. Send error msg
9. Display error msg
10. Upload to visible
11. Save data
12. Delete post
13. Request for deleting
14. Clear from uploaded
15. Erase data
16. Add agent
17. Request for add
18. Send add form
19. Display form
20. Enter data
21. Send person data
22. Save data
23. Send confirmation
24. Display confirmation
25. Delete agent
26. Ask for deleting
27. Eraseagentdata





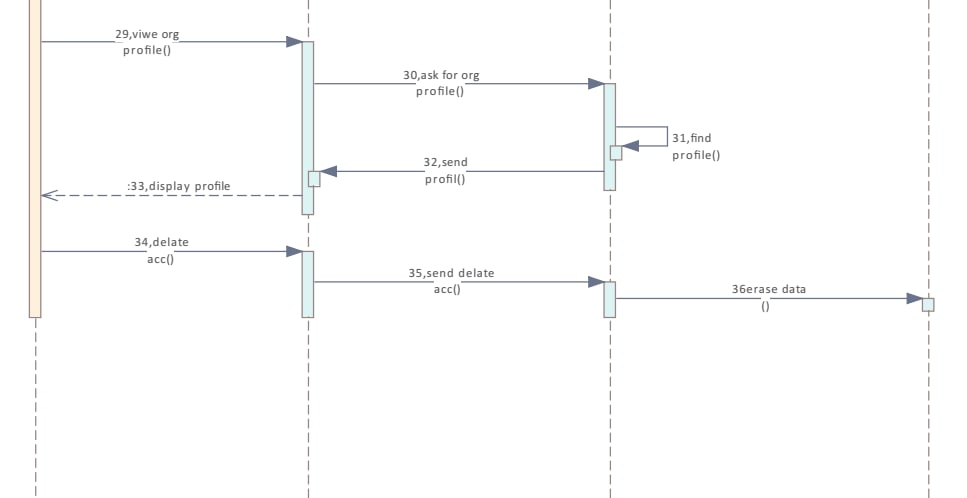


Figure 6:The sequence diagram for a donor.

Process for the donor

1. creat account

1. display form
2. Enter user data
3. Check validity
4. Invalid entry
5. Send error message
6. Enter agent correct data
7. Save registration data
8. Valid user data
9. Send invoice
10. Donate
11. Display form
12. Enter form
13. Send data donation
14. Send a message to the organization
15. Featchfor agent information
16. Sendorgnizationagent information
17. Display agent info
18. Save donation information
19. Send confirmation
20. Display confirmation
21. Redirect online banking page
22. Display payment form
23. Enter form
24. Send data
25. Use external payment API
26. Send confirmation
27. Display
28. View organization profile
29. Ask for an organization profile
30. Find profile
31. Send profile
32. Display profile
33. Delete account
34. Send delete account
35. Erase data

3.3.2 State diagram

A state diagram is used to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of [states](https://en.wikipedia.org/wiki/State_(computer_science)). Here in our system online charity system for local charity organizations we use a state diagram to show how donors can donate and how the organization officer can register and do different activities.

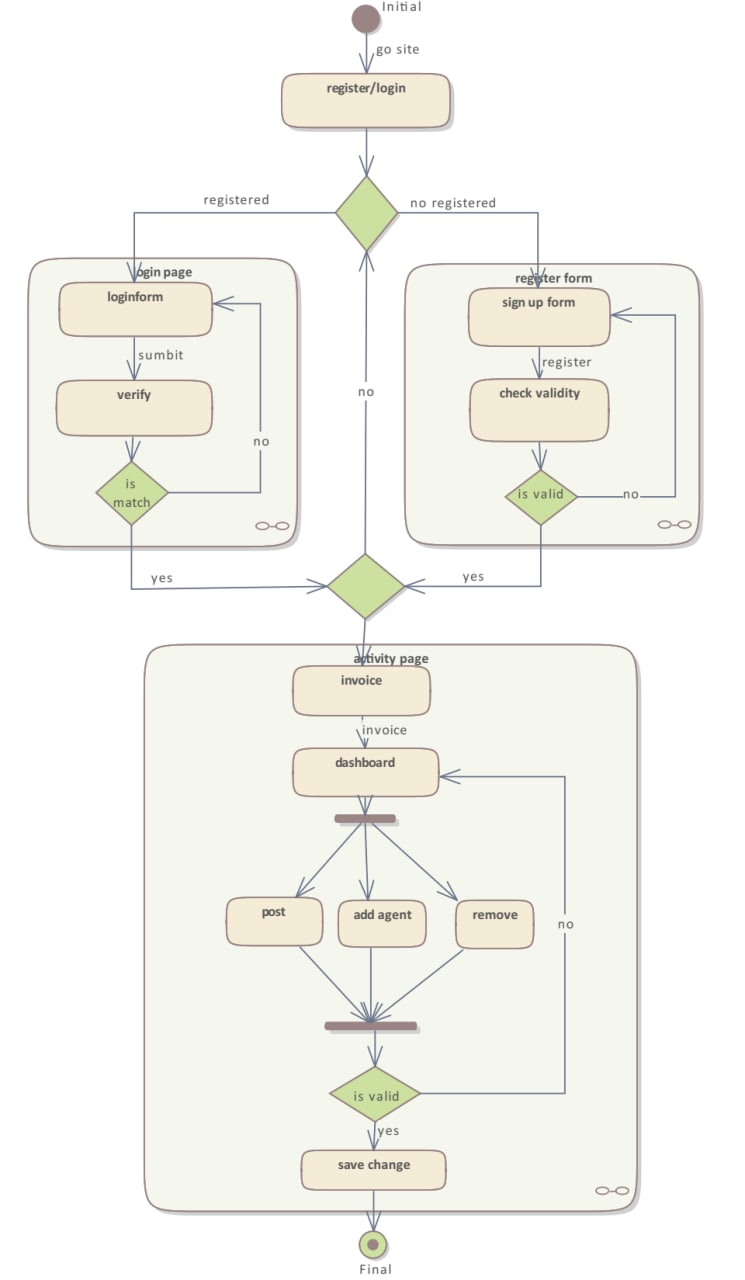


Figure 7:The State diagram for organization officer.

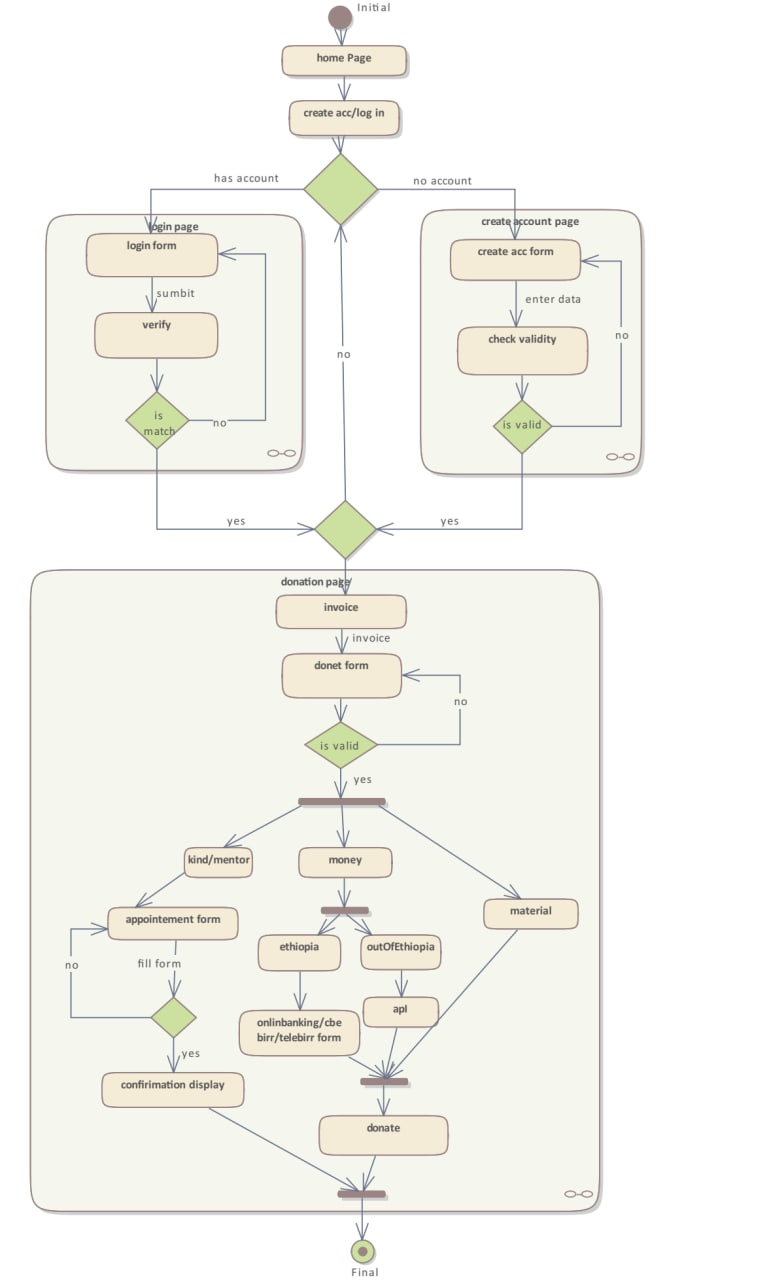


Figure 8:The state diagram for a donor.

3.4 Class Modeling

3.4.1 Class diagram

Here, a class diagram is required to describe the static view of the proposed online charity for local charity organizations. It is also helpful in visualizing objects within a system and the kinds of static relationships that exist among them.

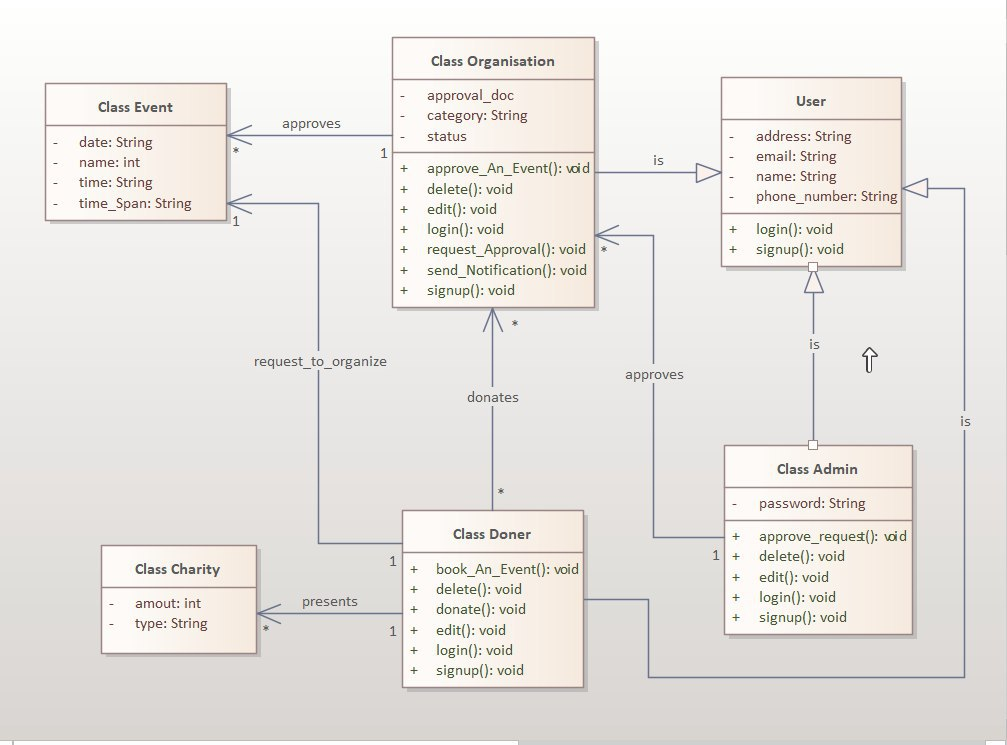


Figure 9:class diagram.

# **Chapter Four: System Design**

## 4.1 Overview

This phase introduces an object modeling technique were objects including data and process. It uses the asset of symbols to represent various components and relationships within the system graphically.

Object modeling requires many types of diagrams to represent the proposed system. Hence, these diagrams will form the foundation for the system design and very important to make implementation easy. Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. However, during systems design, the focus is on the processes, data structures, and software and hardware components necessary to implement it.

4.2 System Design . 4.2.1 Design Goals

Design goals provide a consistent set of criteria that should be taken into consideration when making design decisions. They represent the expected qualities of the system and define the values against which design options are evaluated. Design goals should be used as a guide when choosing a system design and they should be realized in the design chosen. The non-functional requirements of a system provide the basis for inferring most design goals and we use the eight golden rules of interface design principles for better user interface design(by Shneiderman). The design goals we choose are performance, strive for consistency, seek universal usability, offer informative feedback, prevent errors, permit easy reversal of actions, reduce short-term memory load, reliability and maintainability.

* Performance

The performance of a system is measured by its latency and throughput. The latency should be minimized while the throughput should be maximized working with the available bandwidth. For our system to accomplish this we will use a lightweight graphic interface so it will load fast, and we will choose a fast and dependable server.

* Strive for consistency

Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent color, layout, capitalization, fonts, and so on, should be employed throughout. Exceptions, such as required confirmation of the delete command or no echoing of passwords, should be comprehensible and limited in number.

* Seek universal usability

Recognize the needs of diverse users and design for plasticity, facilitating the transformation of content. Novice to expert differences, age ranges, disabilities, international variations, and technological diversity each enrich the spectrum of requirements that guides design. Adding features for novices, such as explanations, and features for experts, such as shortcuts and faster pacing, enriches the interface design and improves perceived quality.

* Offer informative feedback

For every user action, there should be interface feedback. For frequent and minor actions, the response can be modest, whereas, for infrequent and major actions, the response should be more substantial. Visual presentation of the objects of interest provides a convenient environment for showing changes explicitly.

* Prevent errors

As much as possible, design the interface so that users cannot make serious errors; for example, gray out menu items that are not appropriate and do not allow alphabetic characters in numeric entry fields. If users make an error, the interface should offer simple, constructive, and specific instructions for recovery. For example, users should not have to retype an entire name-address form if they enter an invalid zip code but rather should be guided to repair only the faulty part. Erroneous actions should leave the interface state unchanged, or the interface should give instructions about restoring the state.

* Permit easy reversal of actions

As much as possible, actions should be reversible. This feature relieves anxiety since users know that errors can be undone, and encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data-entry task, or a complete group of actions, such as entry of a name-address block.

* Reduce short-term memory load

Humans’ limited capacity for information processing in short-term memory (the rule of thumb is that people can remember “seven plus or minus two chunks” of information) requires that designers avoid interfaces in which users must remember information from one display and then use that information on another display. It means that cellphones should not require reentry of phone numbers, website locations should remain visible, and lengthy forms should be compacted to fit a single display.

* Reliability

The reliability of a system is usually defined as the probability that the system will operate without failure for a specified number of uses or a specified period. We will achieve this goal by using redundant hardware that will take on the services of the failed ones.

* Maintainability

The system should be easily extensible to meet the constantly changing needs of its users and to add new functionalities to the system.

### 4.2.2 Component Identification and Description

**Registration** – this registration component handles the donor and charity organization registration process. It provides an interface used for approval and donation.

**Donation** - this donation component is responsible for all processes regarding donations.

**Approval**- this approval component is about approving the charity organization profile whether is legal or illegal. If it is legal then approved. If it is illegal then rejected and it uses an interface provided by the registration component.

**Databas*e*** *–* access to the databases of the system is provided by this component. All transactions to and from the database will be handled by it.

**Accounts** *–* this accounts component is responsible for all account management issues. It uses the database component.

### 4.2.3 Component Diagram

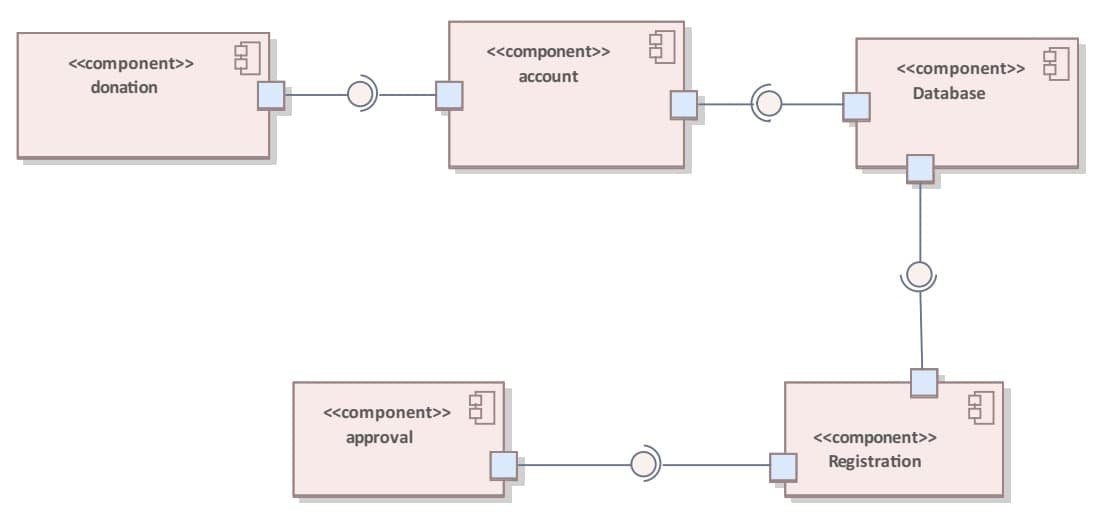


Figure 10:Component diagram

4.2.4 Deployment diagram

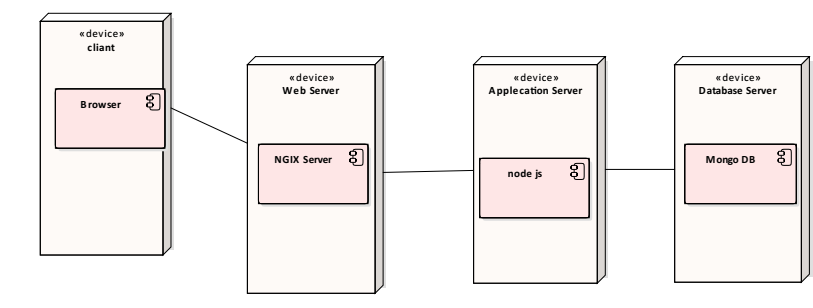


Figure 11:Deployment diagram for the proposed system

## 4.3 Architecture of the System

The first phase of the software design is defining the software architecture. Software architecture is defined as a high-level view of the system, the relationships among the subcomponents, and their properties [5]. We chose the 3-tier architecture design model which divides presentation, business logic, and data access into three separate tiers or layers. The system will comprise of user interfaces (or web pages) as Presentation Layer, a Business Logic Layer (BLL) that defines the functional part of the system and acts as a link between the presentation layer and the Data Access Layer (DAL) which is in charge of retrieving and storing the data. Figure 12 shows the 3-tier architecture while Figure 13 shows a high-level view of the system and the user-system interaction.

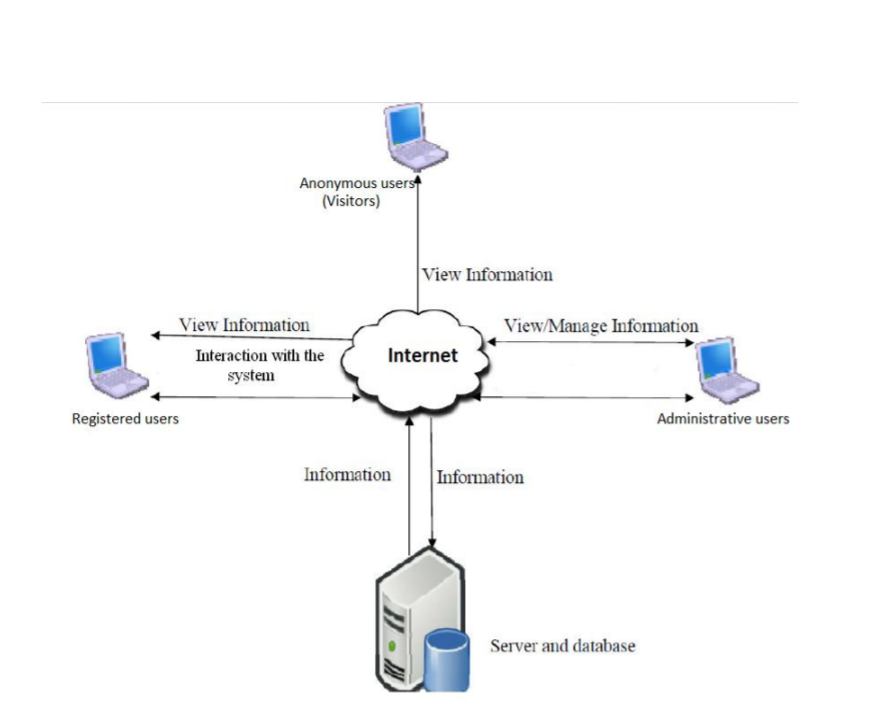


Figure 12:user interaction

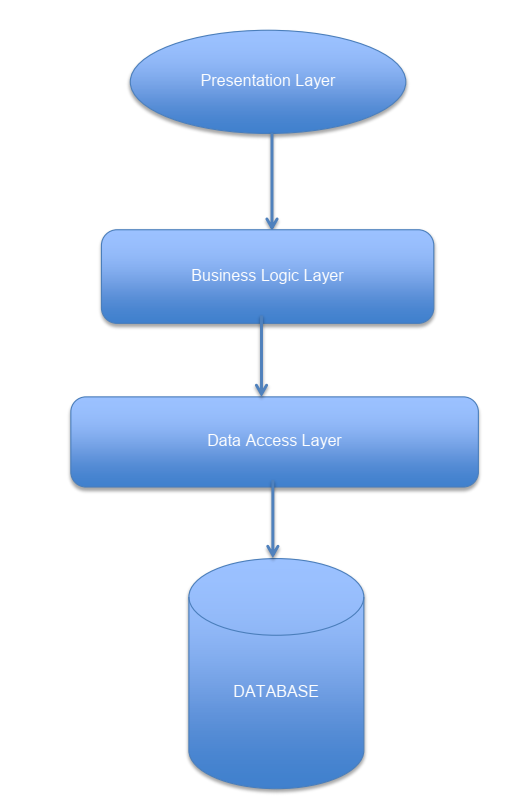


Figure 13: Architecture of the proposed system

4.4 Database Design

Database Entities:

* **Beneficiary:** is a person who gets support from a charity organization.
* **A charity organization:** is a non-profit organization whose aim is to help the charity person.
* **Donor**: is a person who gives charity.
* **Charity:** is any things that are given to the charity organization .it may be money, materials, or kind.
* **Charity agent:** isa person who is assigned by the organization officer to collect any materials donation for the organization.
* **Admin**: is a person who controls the system.
* **Location**: is a place where each charity organization, donors, and charity agents are found.

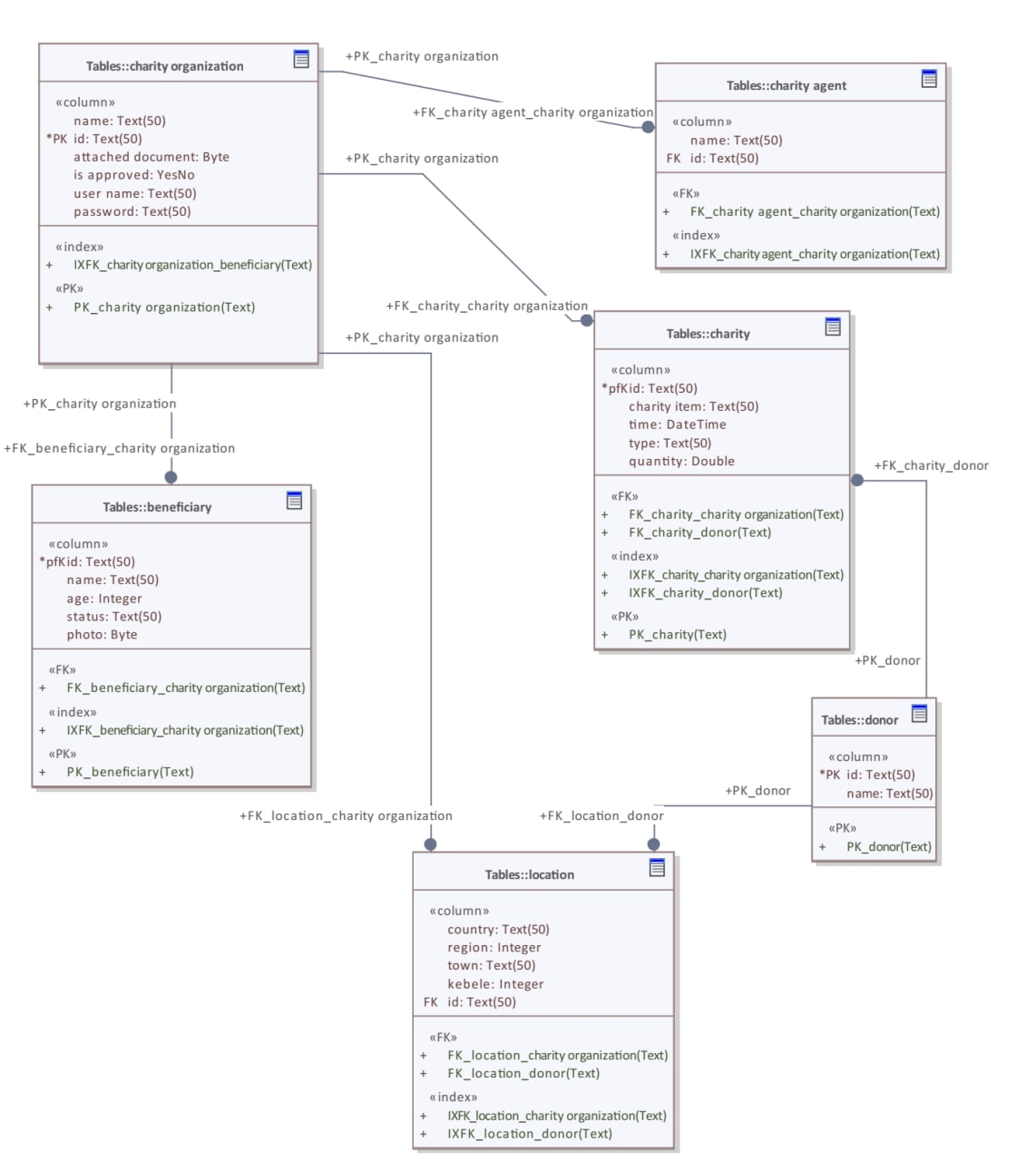
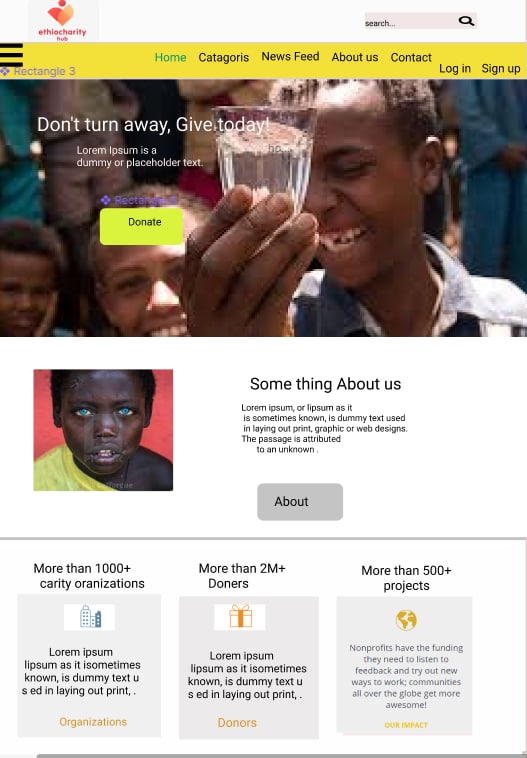


Figure 14: Database diagram

4.5 User Interface Design



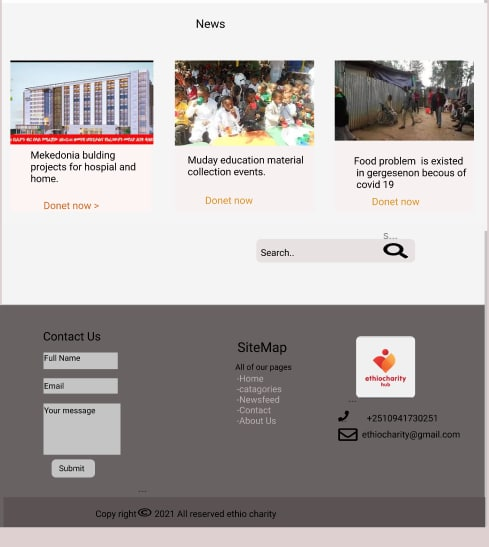


Figure 15: Home Page

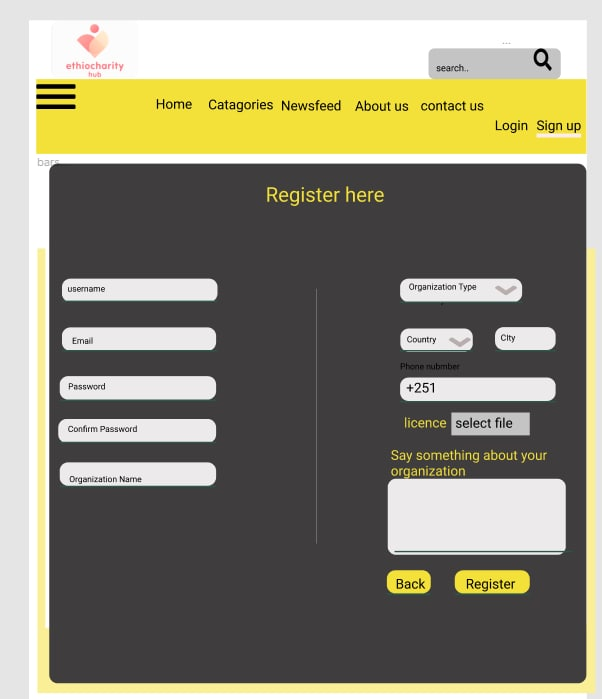


Figure 16:Register Page

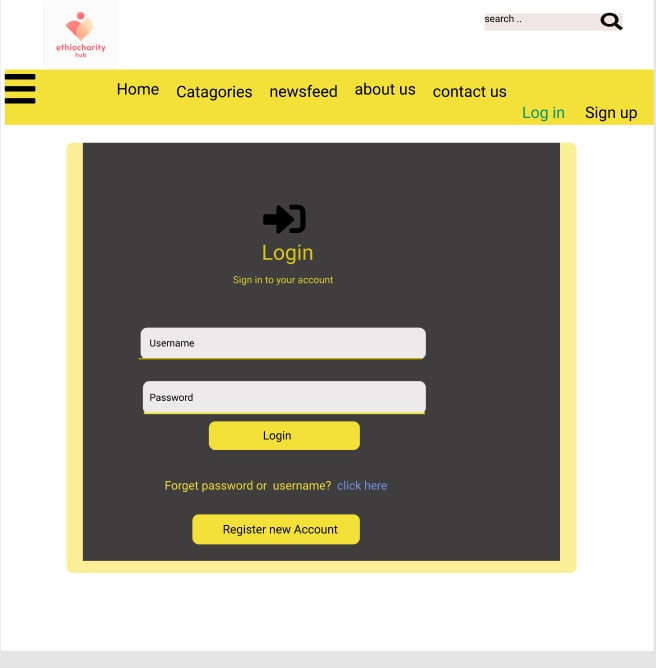


Figure 17: Login page

# References

[1]. ” Your Guide To Intelligent Giving,”(2020,Aug. 15). [online]. Accessed on: june. 08, 2021. Available at : <https://www.charitynavigator.org/index.cfm?bay=content.view&cpid=8658>.

[2].”Node.js v16.4.0 Documentation”(2020,sep 15).[online]. Accessed on: june. 08, 2021.

Availabl at: https://nodejs.org/api/documentation.html.

[3]. “Web Application Security Best Practices: A Developer’s Guide,”(2020,May. 15). [online]. Accessed on: june. 08, 2021. Available at: <https://securityintelligence.com/posts/web-application-security-best-practices-developers-guide/>

[4]. “Web Application Security Best Practices: A Developer’s Guide,”(2020,May. 15). [online]. Accessed on: june. 08, 2021. Available at: https://securityintelligence.com/posts/web-application-security-best-practices-developers-guide/

[5]. Bellinaso, “Marco, ASP.NET 2.0 website programming, Problem - Design - Solution”, 2006. p 48.

# Appendix

Donation - *something that is given to a charity*.

Charity organization - *is an organization whose primary objectives are philanthropy and social well-being*

Charity - *the voluntary of giving help, typically in the form of money, to those in need*.