

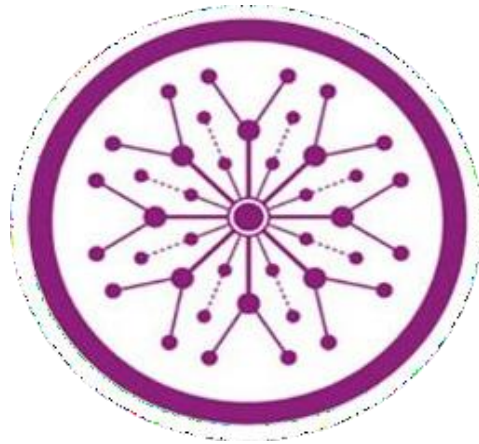
Right Charity

Final Year Project

Session 2019-2023

A project submitted in partial fulfillment of the degree of

BS in Information Technology



Department of Information Technology

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

Type (Nature of project)	<input checked="" type="checkbox"/> Development <input type="checkbox"/> Research <input type="checkbox"/> R&D			
Area of specialization				
FYP ID	FYP-BITM-F22-001			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BITM-F19-002	Muneeb Asif	bitm-f19-002@superior.edu.pk	
(ii)	BITM-F19-025	Anas Zaid	bitm-f19-025@superior.edu.pk	
(iii)	BITM-F19-029	Muhammad Jawad	bitm-f19-029@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I **Muneeb Asif** S/D of **Muhammad Asif**, group leader of FYP under registration no **FYP- BITM-F22-001** at Information Technology Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: _____ Name of Group Leader: Muneeb Asif Signature: _____

Name of Supervisor: Sir Ifraseab Afzal

Designation: Senior-Lecturer

Signature: _____

HoD: Asad Ali Naqvi

Signature: _____

Project Report

Right Charity

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____

Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____

Signature: _____

/55

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____

Signature: _____

Dedication

First of all, we dedicate our project to Allah Almighty

And to whom the world owes its existence

Muhammad (Peace Be upon Him)

This humble effort is dedicated to

Our beloved parents who brought us

To the level of excellence where we

Are studying today looking for most

Promising and gleaming future ahead

For which they scarified most of the

Time of their life

&

To our respected and genius teachers

Who guided us throughout academic career!

And all those people

Who have remembered us in their prayers!

A lot of thanks for all my teachers!

Acknowledgements

Above all, I owe much tribute to the Almighty God who gave us a life worth living and I thank Him for giving us the strength to accomplish this project. The success and accomplishment of this project stems from efforts and dedication offered by many individuals whose support was either direct or indirect. I thank all of them for their devotion.

The author would like to express the sincere gratitude to their project advisor **Sir Irfaseab Afzal** for his vigilant supervision, intellectual guidance, constructive advice and very kind attitude throughout the course of project.

Appreciate to the efforts of all our teachers whose teachings have brought us to this stage of academic zenith.

In the last but not the least, special gratitude with the deepest sense of respects to our parents whose love and affections kept us steep fast and enabled us to attain targets and goals of academic life. The authors are extremely indebted to their brothers and sisters whose constant encouragement provides us with the impetus that was necessary for attaining academic initiatives.

Executive Summary

The proposed system has functionality to facilitate the deserving peoples. Different organizations and volunteers donate funds to the NGO's, Trust organization in the society. We are developing this online donation system using blockchain to keep record of all transaction and also record of all donors and consumers. This system is also known as Right Charity. According to the survey this system is already using in different countries all over the world but we are using blockchain. This system has both donor and consumers, they can register and get services via online using this system. The deserving people who need funds according to their demands can send request to organization using this system. The requests from the deserving peoples and from the NGO or Trust organizations goes to the system and then forward to the donating communities. Then the donor accepts request and donate according to is will. In this system also keep a feedback section for both donor and needy people to avoid, deceive, misbehavior and fraud. This system we will provide services for the donation and for deserving peoples.

Table of Contents

Right Charity.....	2
Final Year Project	2
Session 2019-2023	2
Plagiarism Free Certificate.....	3
Change Record.....	4
Dedication.....	6
Acknowledgements	7
Executive Summary.....	8
List of Figures.....	12
Chapter1	14
Introduction	14
1.1. Background	15
1.2. Motivations and Challenges.....	16
1.3. Goals and Objectives	16
1.4. Literature Review/Existing Solutions.....	16
1.5. Gap Analysis.....	16
1.6. Proposed Solution.....	17
1.7. Project Plan	17
1.7.1. Work Breakdown Structure	17
1.8. Report Outline	19
Chapter 2	20
Software Requirement.....	20
2.1. Introduction	21
2.1.1. Purpose.....	21
2.1.2. Document Conventions	21
2.1.3. Intended Audience and Reading Suggestions	21
2.1.4. Product Scope	21
2.1.5. References	22
2.2. Overall Description.....	22
2.2.1. Product Perspective	22
2.2.2. Product Functions	22
2.2.3. User Classes and Characteristics.....	22
2.2.4. Operating Environment	23
2.2.5. Design and Implementation Constraints.....	23
2.2.6. User Documentation	24
2.2.7. Assumptions and Dependencies.....	24
2.3. External Interface Requirements	24

2.3.1.	User Interfaces.....	24
2.3.3.	Software Interfaces.....	25
☒	Website:26	
☒	Consumer:	26
☒	Donors: 26	
☒	Non-Profit Organizations:.....	27
2.4.1.	System Feature 1Registration of users	27
2.4.2.	System Feature 2Login of Users.....	28
2.4.3.	System Feature 3Logout.....	30
2.4.4.	System Feature 4Search	31
2.4.5.	System Feature 5View transactions	32
2.4.6.	System Feature 6	33
2.4.7.	System Feature 7	34
Chapter 3.....		37
Use Case Analysis		37
3.1.	Use Case Model	38
Fully Dressed Use case:.....		40
Use case model for Consumer:.....		41
3.2.	Use Case Descriptions.....	42
Chapter 4.....		50
System Design		50
4.1.	Architecture Diagram.....	51
4.3.	Class Diagram	53
4.5.	Operation contracts	57
4.6.1	Operation Contracts # 1.....	57
4.6.2	Operation Contracts # 2.....	57
4.7	Activity Diagram.....	59
Admin:		61
4.9	Component Diagram.....	63
4.10	Deployment Diagram	64
Chapter 5.....		66
Implementation		66
5.3.	Deployment Environment.....	69
5.4.	Tools and Techniques.....	69
5.5.	Best Practices / Coding Standards	71
5.6.	Version Control.....	71
Chapter 6.....		72
Testing and Evaluation		72
6.1.1.	73

6.1.3:	75
6.1.4:	Error! Bookmark not defined.
6.1.6:	77
Chapter 7	84
Summary, Conclusion andFuture Enhancements	84
Reference andBibliography	87
Reference and Bibliography	88
Index	89
Index	90

List of Figures

Figure 1: Work break down structure	5
Figure 2: Gantt chart	7
Figure 3: use case of admin	26
Figure 4: use case of non-profit org	27
Figure 5: Fully Dressed Use Case	27
Figure 6: use case of donors	28
Figure 7: use case of consumer	29
Figure 8 : Architecture Diagram	36
Figure 9 : Domain Model	37
Figure 10: ERD diagram	38
Figure 11: Class Diagram	39
Figure 12: sequence diagram of admin	40
Figure 13 : sequence diagram of user	41
Figure 14: sequence diagram of non-profit org	44
Figure 15 : sequence diagram of consumer	44
Figure 16 : Activity diagram	45
Figure 28: state transition diagram	48
Figure 29: Component Diagram	49
Figure 30: Deployment Diagram	50
Figure 31 : DFD	51

List of Tables

Table 1:Role& Responsibility Matrix	6
Table 2 : Hardware Interfaces	17
Table 3 :User case of register	31
Table 4: User case of login	33
Table 5 : use case of consumer	34
Table 6 : tool and technique	54
Table 7 : Use case of Sign up	58
Table 8 : use case of login	59
Table 9 : use case of update password	60
Table 10 : use case of delete data	61
Table 11: use case of search data	62
Table 12 : use case of record to display	62
Table 13: boundary analysis	63

Chapter1

Introduction

Chapter 1: Introduction

Many fake charity organizations pose as genuine and loot money from innocent people in the name of charity. Most people want to donate money to a good cause or charity, but they are unsure if the money is going to reach the right hands of the destitute. The blockchain system can bring transparency to online charity trusts. Contributors can see the journey of the donation in real-time and confirm if it's reaching the deserving hands or not. The reason we are using blockchain is that it is a decentralized database and it is more secure and maintains the integrity of the data. And in this system, there is a function in which needy people can rise a request for themselves.

1.1. Background

There are a lot of charity organizations working in Pakistan to help needy people. And there is also online donation system in Pakistan but people still feel insecure to donate. Because there is no proper record of transaction.

Some donor agencies have lost their trust in NGOs in under-developed countries, as some NGOs have been involved in the misuse of funds. This is evident from irregularities in the records. For instance, in education funds, on some occasions, the same student has appeared in the records of multiple NGOs as a beneficiary, when in fact, a maximum of one NGO could be paying for a particular beneficiary. Therefore, the number of actual beneficiaries would be smaller than the number of claimed beneficiaries. This research proposes a blockchain-based solution to ensure trust between donor agencies from all over the world, and NGOs in under-developed countries. The list of National IDs along with other keys would be available publicly on a blockchain. The distributed software would ensure that the same set of keys are not entered twice in this blockchain, preventing the problem highlighted above.

Blockchain is a decentralized technology that allows different stakeholders to access and replicate a database. The database can only be updated using predetermined rules, and once changed, it is shared with all parties. Each transaction in the blockchain is connected in a chain to make sure that everybody has the most up-to-date version of the ledger.

1.2. Motivations and Challenges

Now a day people use more mobiles for internet for different purpose. It is very easy to use for everyone. It is app only for donation funds. A great new idea to help people through online system. The close gap between fortune and unfortunate people in the society.

1.3. Goals and Objectives

Donation is advanced concept to donate quickly on time. It is a completely social and non-profit app. It brings to close gap between fortunate people and unfortunate people in the society. And also, to keep transactions safe and secure. And to make sure that all the funds in right hands and funds are using for needy people.

1.4. Literature Review/Existing Solutions

We create a social system which is easy in use for everyone. In this system different NGO-TRUST who need funds according to their requirement can send request to our organization and admin will approve their request and provide funds in quick time. Right charity act as a medium that connects Donors who wish to donate with Charities who work with needy, and thus help in diverting the excess to needy to reduce their problems.

There are a lot of ENGO'S working online in Pakistan without using Blockchain technology. That's why people feel insecure to donate because they don't know about all the transactions records.

Some famous charity organizations working online in Pakistan

- Transparent hands
- Shaukat Khanam cancer hospital
- Al Mustafa welfare society
- Al khidmat foundation

1.5. Gap Analysis

There is no online system for online donation that use blockchain technology. This app creates for social welfare and non-profit organization who collect funds according to their demand from different volunteer.

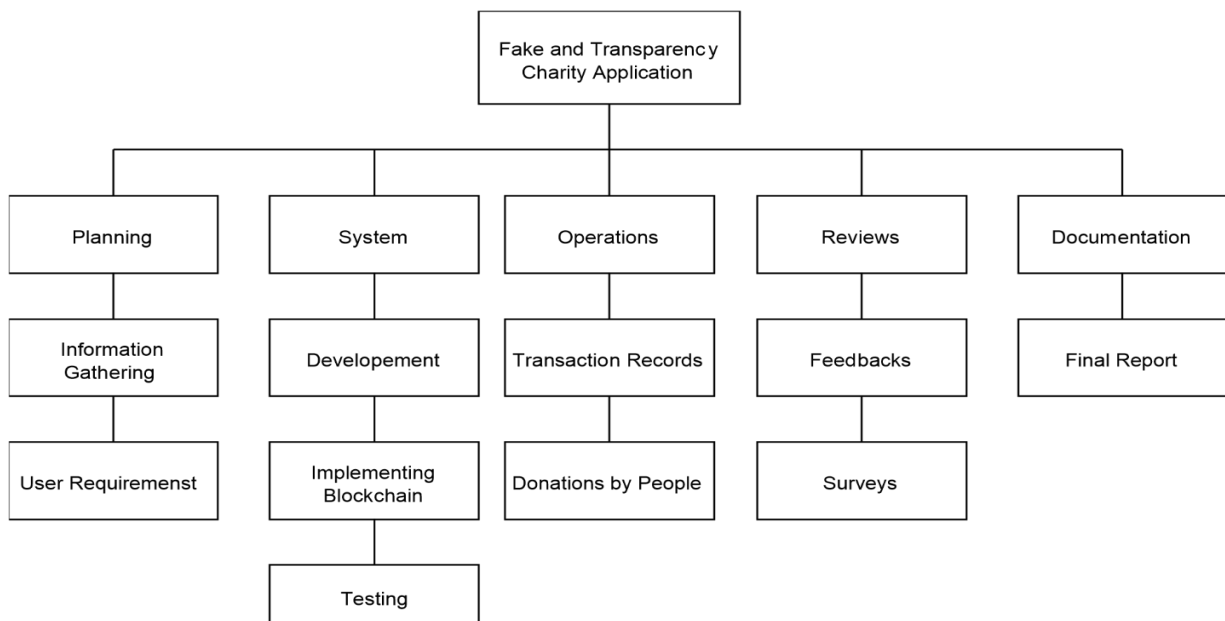
1.6. Proposed Solution

We are providing transparency to the public using blockchain. People can easily see the records of donations and how many donations were collected by a specific foundation. And how they are using donations given by people. People can see all the records with valid proof. We are storing all this information in the blockchain. We are providing a platform through which the public can easily find out which foundation is fake or which one is genuine. Because people are afraid to donate and are not sure about that their donations are in right place or not. So, with the help of this platform donators can easily track records of all the transactions of foundations.

1.7. Project Plan

This is our project plan that tells about how our Android Application operate by and with following responsibility matrix, WBS and Gantt chart.

1.7.1. Work Breakdown Structure



1.7.2. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Requirement Gathering	1	Technical specification Restricted areas Available solutions	25	Muneeb
2	Analysis	2		16	Anas, Jawad
3	Architecture	3		55	Muneeb, Jawad
4	Designing	4		19	Anas
5	Implementation & Development	5		121	Muneeb, Anas, Jawad
6	Database Management	6		23	Muneeb, Jawad
7	Testing	7		11	Muneeb, Anas,
8	Final Documentation	8		20	Muneeb, Anas, Jawad

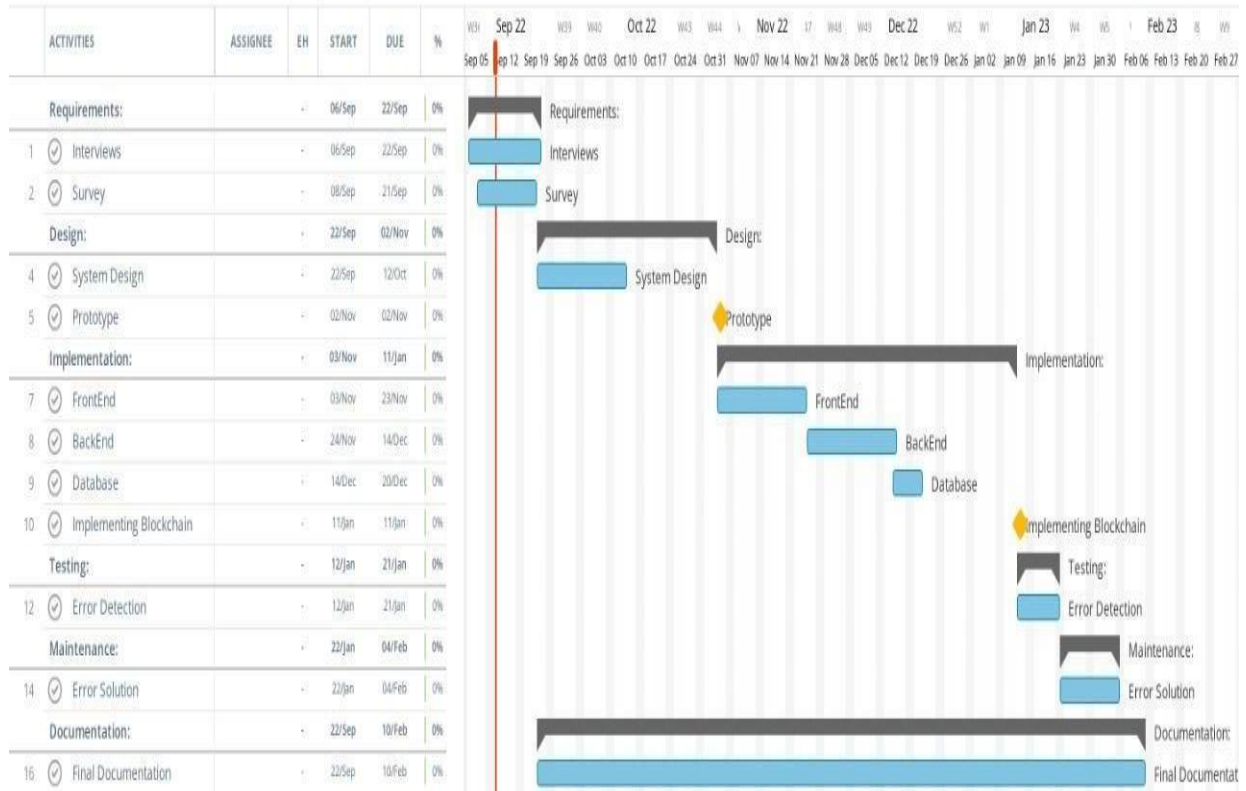
Table 1.1: WBS

1.7.3. Gantt Chart

fyp

Read-only view, generated on 12.Sep.2022

Instagantt



1.8. Report Outline

- The Requirement was gathered in about 15 Days.
- The Documentation was made in a whole year.
- The coding and data were implanted in almost 4 months.
- The Test were made, and errors were debugged after and through the coding period.
- Validation and finalization in added golden days.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

This online system is used for social welfare. The website performance should be optimized, and response time should be minimized. this website is social website used to donate funds which helps need people to fulfil their needs.

2.1.2. Document Conventions

This record is its degree is to portray the necessities of the Advertisement Management System and its affiliations. This report has been composed after the AMS and consider that future changes to this program ought to be incorporated into this prerequisites determination record for keeping up its helpful part. To the extent the program usage is concerned while it's not been actualized, this record can be utilized as manual for improvement of framework. CSS, VCSS property, descriptor, and pseudo-class names are specify by single quotes. Values are specifying by single quotes. Document language element names are in upper case Letters. Document language attribute names are in lower case letters and specify by Double quotes.

2.1.3. Intended Audience and Reading Suggestions

This website is highly scalable. Its purpose is to enable a sense of solidarity in the community. The app has a societal impact since its aim is to close the gap between the less fortunate people and the more unfortunate in terms of quality of life.

2.1.4. Product Scope

Our projects keep the record of the donation the donor can be sure about his donation he can track the record of the organization and can check the organization's recent money transactions and can keep the track of the donation user will be sure about the organization either the organization is fake or real. Because everyone wants to be sure about his money that his money is in right place or not. Because there are a lot of fake organizations which are scamming people.

So, our community need this type of platform which makes sure donors about their donations. Because this is necessary for donors to make sure that their donations are equally distributed to the needy people.

2.1.5. References

International Journal of Advanced Research in Computer and Communication Engineering, 2022

2.2. Overall Description

2.2.1. Product Perspective

Right Charity will provide a platform for donors and seekers after they successfully register into the system. If a user wishes to donate something, he/she can send a request in application. We are aiming to manage all the transactions and sending all the funds to the needy people without any kind of fraud. We are looking and expecting to update and refine the donation system the application will be beneficial for both donors and seekers.

2.2.2. Product Functions

- There are three different roles in this system (Donor, Receiver and Admin).
- The Donor performs operations like Registration and Login into the System. He can also donate and view all donation requests.
- The Admin and Donor both can view the Receiver's location.
- The admin can also monitor and update the database.
- The Admin and Receiver both can view the Donor's location.
- The Receiver can also perform operations like requesting for items or some kind of help.
- And there is a one more function through which visitor on our system can see the news about major crises happening in country.

2.2.3. User Classes and Characteristics

There are 4 kinds of client of this product

➤ Overall population:

Overall population can utilize the framework to see the major crises currently happening in our country. And how much amount an organization needed to help the needy people.

➤ **Donors:**

Donors can easily register himself and donate funds and view the crises currently happening in country.

➤ **Administrator:**

Administrator can add, alter and erase news and also can add, update information of donors, receiver and also can delete user or blacklist them. Admin can see all the statistics of users and also handle all the transactions.

➤ **Receiver:**

Receiver can ask for a help on our system and receive funds and can also view the location of donors.

2.2.4. Operating Environment

This system will operate in browsers like Mozilla, Firefox, and Opera and for their different version also.

- It can be open on window, android and iPhone.
- The processor should be core 2 duo or above.
- The processor's speed should be 2.8 GHz or greater.
- Ram should be or greater than 512 MB.

2.2.5. Design and Implementation Constraints

There are the following design and implementation constraints that we should keep in our mind during developing of this website.

- This system is working for single server
- Limited to HTTP/HTTPS.
- Figma for prototyping the system
- Using react and html for front-end.
- Attach database to the back-end used to manage the database specification
- Also using solidity
- Operating system required for this application is windows kit Kat 4.0 android or any other higher version

2.2.6. User Documentation

Latterly, the uses of mobile devices and wireless network applications became more and more widespread. Project has solved the issue that people donate food manually by visiting each organization number of times. It provides single platform for donors and consumers to donate food timely and securely to the needy people. The system shall provide a very simple interface, which should be user friendly and easy for them to understand and operate better.

2.2.7. Assumptions and Dependencies

There are assumptions and dependencies that should be fulfilled for use of this application:

- User should have basic knowledge of English language and computer and smart phone usage.
- Admin monitoring the use of Website/application.

User have basic understanding on how to operate computer and computer software.

2.3. External Interface Requirements

2.3.1. User Interfaces

Screen Name	Description
Login	User can login with credentials he/she provided when registered to our provide if wrong username or password then cannot login to account
Search	There is search option which helps donors tosearch consumer and for consumer through which they can search donors.
Home Screen	There is a home screen which shows all the information about our foundation.
	And there is also option to Connect Wallet onhome screen. And there is also option to Donate Now on home screen.

Transaction	And there is also transaction record page.
Donors, Consumer and Organization	This page all donors, consumer and registered organizations.
Payment Method	Through user can donate.
News	This page shows news about crises.

2.3.2. Hardware Interfaces

- Windows.
- A browser which supports HTML & JavaScript.
- Processor
- Ram
- HD (Hard Drive)

Just the prescribed setup (fundamental necessities of a PC framework) no other particular equipments are required to run the product.

2.3.3. Software Interfaces

- **External machine interfaces**

External Machine interface is not required since computation will be done on Windows OS and Android OS.

- **External system interfaces**

This will only include the blockchain attached to the system.

- **Human interface**

The software will use a Graphical User Interface coded in XML.

- **Communications Interfaces**

The e- donation system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

2.4. System Features

The system is going to consist of multiple modules, each separately developed with their own features.

➤ **Website:**

- **Admin:**

Register: admin can register into the System

Login: admin can login into the System

View Detail: admin can check all the detail of the users.

Manage: admin can also manage all the users.

Manage News: admin can manage all the news on website.

Accept Donors Request: admin can accept donors request to donate.

Log out: admin can logout.

- **Consumer:**

Register: User can register into the System

Login: User can login into the System

Search donor: User can also search for donor.

Search organization: consumer can easily search for non-profit organization.

Raise Request: Consumer can raise request for funds.

Receive Funds: Consumer can receive funds from donors and organizations.

Log out: consumer can logout.

- **Donors:**

Register: User can register into the System

Login: User can login into the System

Search Consumer: user can also search for consumers.

Search Organizations: user can also search for organizations

Donate: User can donate funds.

View Transactions: User can view all transactions record.

View Location: Donors can view the location of consumer and organization.

Accept Request: Donors can accept the requests of consumer and organization.

Log out: After managing things donors can logout.

- **Non-Profit Organizations:**

Register: Organization can register into the System

Login: Organizations can login into the System

Search Consumer: Organizations can also search for consumers.

Search Donors: Organizations can also search for donors.

Donate: Organizations can donate funds.

View Transactions: Organizations can view all transactions record.**View**

Location: Organizations can view the location of consumer. **Accept**

Request: Donors can accept the requests of consumer.

Raise Request: Organizations can raise request for funds.

Log out: After managing things donors can logout.

2.4.1. System Feature 1

Registration of users

2.4.1.1. Description and Priority

It is high priority feature of a system because without registration users can't perform anyaction.

2.4.1.2. Stimulus/Response Sequences

Identifier	FR-01
Title	Register user
Requirements	First name, Last Name, Emails, Password, contact no, address etc.
Rational	Register to the system with data integrity
Restriction/risk	User must enter correct information otherwise not registered
Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 2 - Register User Response

2.4.1.3. Functional Requirements

Register
Actor
<ul style="list-style-type: none"> • Un-registered user
Pre conditions
<ul style="list-style-type: none"> • Fill to sign up form for registration • User must open the registration page
Basic flow
<ul style="list-style-type: none"> • User wants to register for the particular application • User must provide all required information
Alternate flows
<ul style="list-style-type: none"> • User don't open the register page
Post conditions
<ul style="list-style-type: none"> • User registered successfully

Table 3 - Functional Requirement Register

2.4.2. System Feature 2

Login of Users

2.4.2.1. Description and Priority

It is high priority feature of a system because without login customer can't buy products and vendor can't go to dashboard.

2.4.2.2. Stimulus/Response Sequences

Identifier	FR-02
Title	Login
Requirements	User id, password, etc.
Rational	Enter data then user might be login
Restriction/risk	User must enter correct information otherwise not registered

Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 4 - Login User Response

2.4.2.3. Functional Requirements

Login
Actor
<ul style="list-style-type: none"> • Consumer • Donor • Admin • Organization
Pre conditions
<ul style="list-style-type: none"> • User must open the application • User must be registered
Basic flow
<ul style="list-style-type: none"> • User wants to register for the particular application • User must provide all required information
Alternate flows
<ul style="list-style-type: none"> • User would not use or registered to your application
Post conditions
<ul style="list-style-type: none"> • User login successfully

Table 5 - Functional Requirement Login

2.4.3. System Feature 3

Logout

2.4.3.1. Description and Priority

It is medium priority feature of a system because without logout anyone can access user's account.

2.4.3.2. Stimulus/Response Sequences

Identifier	FR-03
Title	Logout
Requirements	None
Rational	User must be login
Restriction/risk	User must enter correct information otherwise not registered
Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 6 - Logout User Response

2.4.3.3. Functional Requirements

Identifier	FR-03
Title	Logout
Requirements	None
Rational	User must be login
Restriction/risk	User must enter correct information otherwise not registered
Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 7- Functional Requirement Logout

2.4.4. System Feature 4

Search

2.4.4.1. Description and Priority

It is medium priority feature of a system with which users can search for donors, consumer, organizations.

2.4.4.2. Stimulus/Response Sequences

Identifier	FR-04
Title	Search
Requirements	Donors
Rational	User must go to search bar
Restriction/risk	None
Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 8 - Search Response

2.4.4.3. Functional Requirements

Search
Actor
<ul style="list-style-type: none">• Consumer• Donor• Organization
Pre conditions
<ul style="list-style-type: none">• User must enter product name in search box
Basic flow
<ul style="list-style-type: none">• User wants to search for a specific person
Alternate flows
<ul style="list-style-type: none">• User would not apply for searching
Post conditions
<ul style="list-style-type: none">• User search successfully

Table 9 - Functional Requirement Search

2.4.5. System Feature 5

View transactions

2.4.5.1. Description and Priority

It is medium priority feature of a system with which donors and organizations can view transactions.

2.4.5.2. Stimulus/Response Sequences

Identifier	FR-05
Title	<i>View transactions</i>
Requirements	<i>Transactions information</i>
Rational	User must view all details of <i>transactions</i>
Restriction/risk	None
Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 10 - View transactions Response

2.4.5.3. Functional Requirements

View product
Actor
<ul style="list-style-type: none">• Donors• Organizations
Pre conditions
<ul style="list-style-type: none">• User must view product details
Basic flow
<ul style="list-style-type: none">• User wants to view for product
Alternate flows
<ul style="list-style-type: none">• User don't want to view product

Post conditions
<ul style="list-style-type: none"> User view successfully

Table 11 - Functional View transactions

2.4.6. System Feature 6

Donate

2.4.6.1. Description and Priority

It is medium priority feature of a system with which users can donate.

2.4.6.2. Stimulus/Response Sequences

Identifier	FR-06
Title	Donate
Requirements	Must login into system
Rational	User must go connect with wallet
Restriction/risk	None
Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 12 – Donate Response

2.4.6.3. Functional Requirements

Select product
Actor
<ul style="list-style-type: none"> Organizations Donors
Pre conditions
<ul style="list-style-type: none"> User must select consumer
Basic flow
<ul style="list-style-type: none"> User wants to select a specific consumer
Alternate flows

<ul style="list-style-type: none"> User would not apply for select
Post conditions
<ul style="list-style-type: none"> User donate successfully

Table 13 - Functional Requirement Donate

2.4.7. System Feature 7

Raise Request

2.4.7.1. Description and Priority

It is medium priority feature of a system with which user can raise request.

2.4.7.2. Stimulus/Response Sequences

Identifier	FR-07
Title	Raise Request
Requirements	Must login into the system
Rational	None
Restriction/risk	None
Dependency	Connection with server through PC, mobile laptop etc.
Priority	Quality, security

Table 14 - Raise Request Response

2.4.7.3. Functional Requirements

Edit/update
Actor
<ul style="list-style-type: none"> Organizations Consumer
Pre conditions
<ul style="list-style-type: none"> Organizations or Consumer raise the request.
Basic flow
<ul style="list-style-type: none"> Organizations or Consumer wants to raise the request.
Alternate flows

<ul style="list-style-type: none"> • Organizations or Consumer didn't raise the request.
Post conditions
<ul style="list-style-type: none"> • Organizations or Consumer request successfully.

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

- **Response time:** The home page loaded in 3-5 seconds.
- **Availability:** The site will available 99.99% of the time.
- **Throughput:** Page hits per second or orders per hour.
- **Capacity:** Maximum throughput that can be achieved with acceptable response time and availability.

2.5.2. Safety Requirements

- Once admin delete record or user account the data of that user can't be retrieve. To handle this problem system gives warning before delete the record.
- Safety disclaimers will be given for both students and teachers.

If any user has some problem while using the site, so he can contact via mail to website maintainer

2.5.3. Security Requirements

Security is one of the major concerns of these days. Our system is based on following security parameters which are given below:

- This software keeps the information of students and teachers safe.
- User will be registered after verification of user data.
- Secure login and logout. If renter forget his account credential than he has to get new credential through his sign-up e-mail.
- Admin area will be secured.

2.5.4. Software Quality Attributes

- **Usability:** This website should be easy to use and easy to learn how to operate.

- **Correctness:** When user search something but spellings are not right the system search related to that spelling or give autocomplete search.
- **Portability:** This product can be used on different devices like tablets, mobiles, laptops etc.
- **Robustness:** This software can handle many problems itself like if user forgets his password so system gives it option to re-write name and user password.
- **Flexibility:** The functionality of this software can be increased after development because documentation is available.
- **Performance:** The loading speed of web pages onto a client's web browser is quite good.
- **Reliability:** This software is reliable.
- **Maintainability:** Admin of the website will maintain the working of this software.
- **Security:** This software is secure enough to use. This software keeps the information of users safe.

2.5.5. Business Rules

- If user want to perform some crud operation than he/she has to first sign up.

User has to follow terms and conditions of website.

Chapter 3

Use Case Analysis

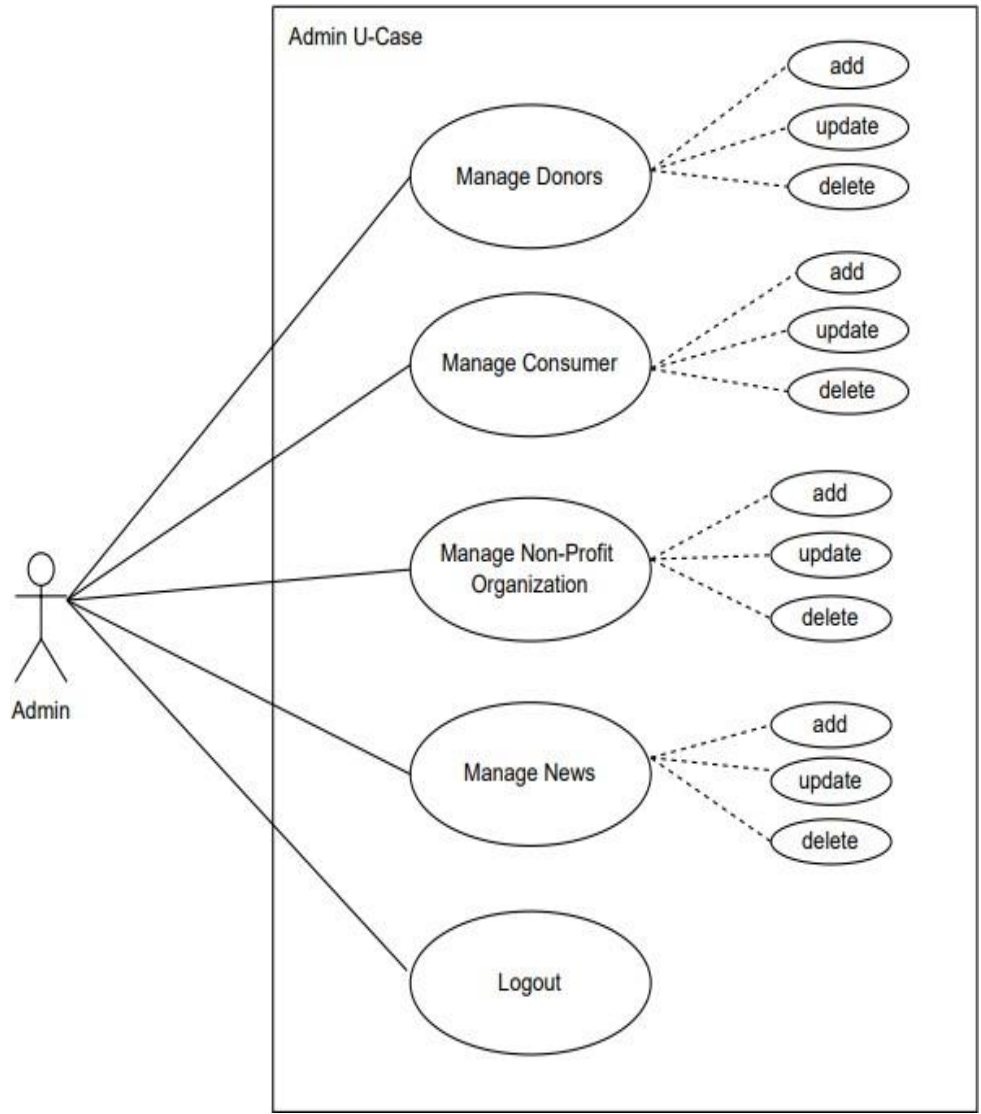
Chapter 3: System Analysis

This chapter describe about system use cases and how the different actors are interacting with the system. This system is all about use case of system and role we assign and categorize the user of solution. We describe the all terms about the use case model and use case diagram of every role.at the end we make a proper dressed use case model of our project.

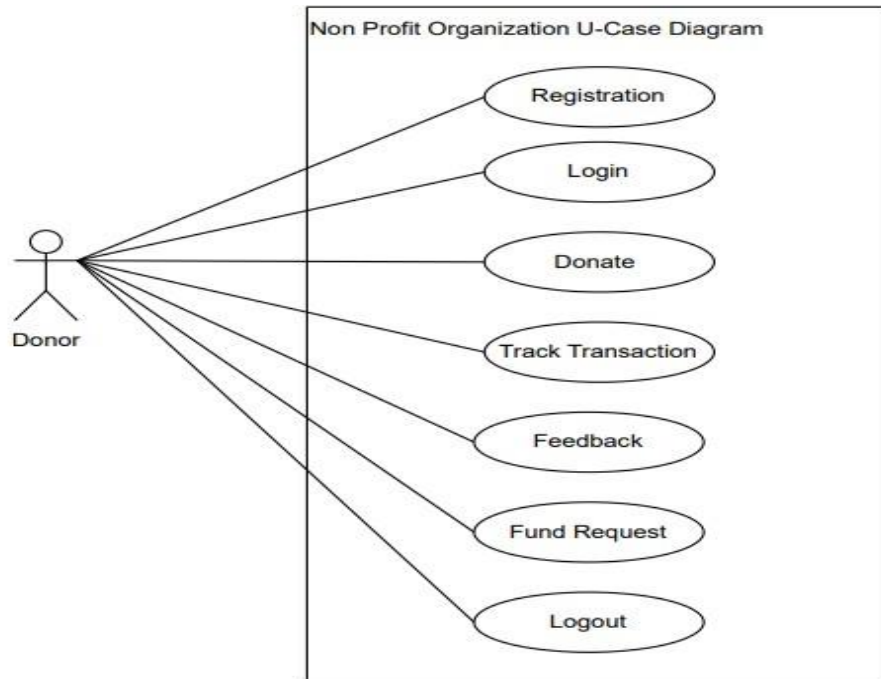
3.1. Use Case Model

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

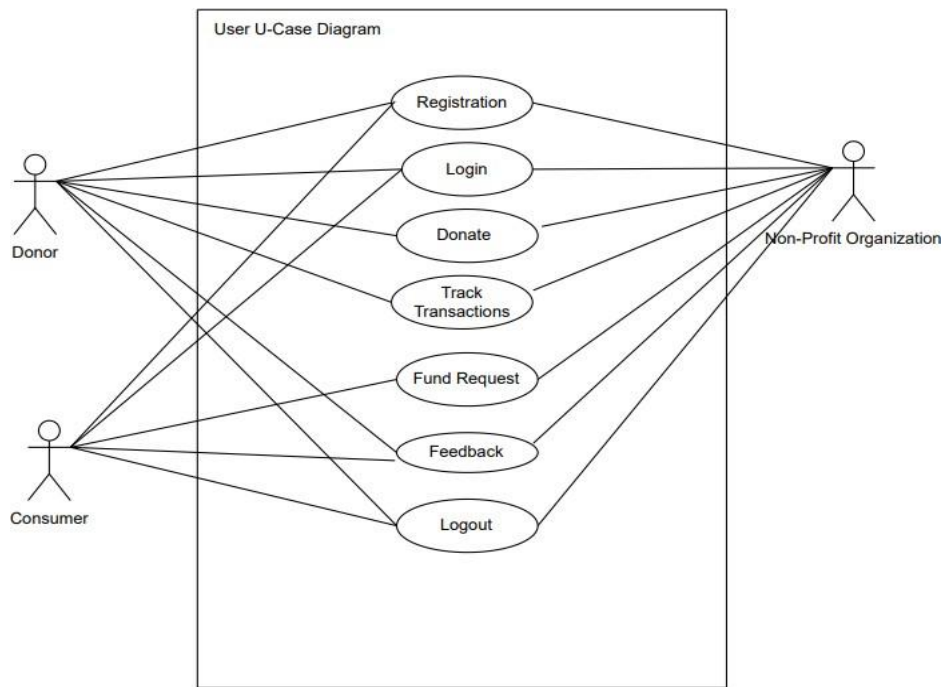
Use case model for Admin:



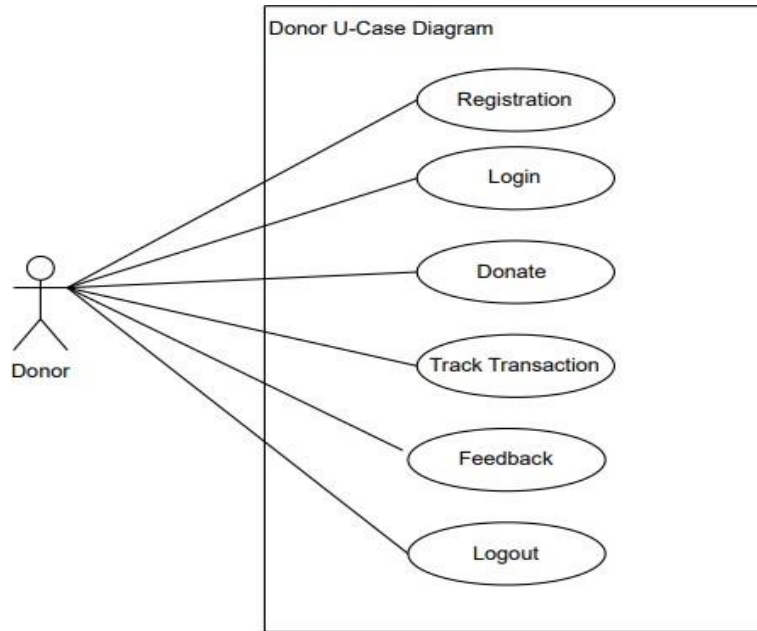
Use case model for Non-Profit Organizations:



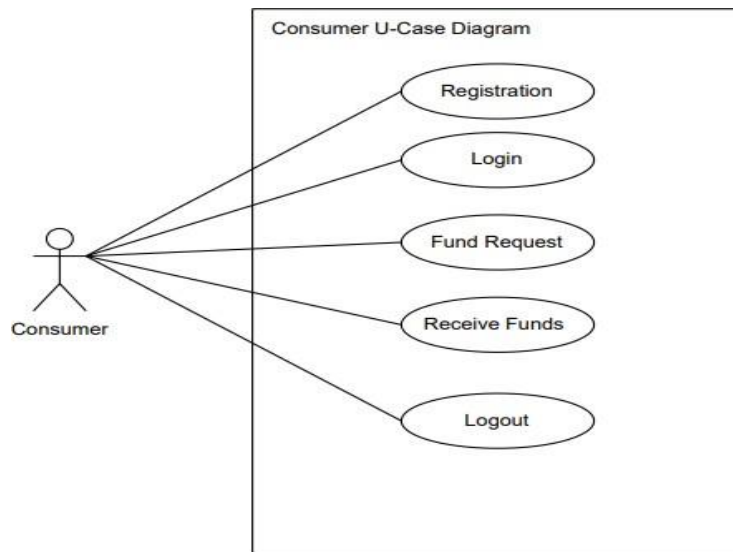
Fully Dressed Use case:



Use case for Donors:



Use case model for Consumer:



3.2. Use Case Descriptions

Registration:

Name	Register
Brief Description	A user of the System creates an account
Actor(s)	Admin
Flow of Events	
Basic Flow	
<p>This use case starts when a system user is not logged in to the system and goes to the loginpage.</p> <ol style="list-style-type: none"> 1. The System prompts the user for a username and password or register new account. 2. The user selects registration option. 3. The System prompts user for registration information, Username, password, etc 4. The user enters in their information. 5. System verifies information and creates account. 6. The use case ends. 	
Alternate Flows	
Title	Description
Cancel Registration	<ol style="list-style-type: none"> 1. The user selects the cancel option. 2. The system returns the user to the home page without the user being logged in and any information entered has been erased.
Invalid Information Entered	<ol style="list-style-type: none"> 1. User clicks submit after entering information system asked for. 2. System displays information with appropriate message to correct invalid information. 3. User re-enters information.
Pre-Conditions	

Title	Description
System	To logged in to the system the user must have the System
Internet	To logged in to the system the user must have the network on theSystem.
Post-Conditions	
Title	Description
Success	The user entered successful information and is returned to thehome page as a Logged in User
Failure	User is unable to log in for one or more reasons and is returned tothe home page as a Guest.
Extension Points	
If the User is unable to log in for one or more reasons the user must once check the internetConnection.	

Login:

Name	Log In
Brief Description	The System user logs in to the System.
Actor(s)	Logged In User
Flow of Events	
Basic Flow	
<p>This use case starts when a system user is not logged in to the system and goes to the loginpage.</p> <ol style="list-style-type: none"> 1. The System prompts the user for a username and password or register new account 2. The user enters his/her username and password. 3. The system validates the entered username and password, making sure that the entered username is a valid username in the System, and that the required password is entered for the entered username. 	

4. The user is signed in and returned to the home page as a Logged In User.	
5. The use case ends.	
Alternate Flows	
Title	Description
User Fails Authentication	<p>If the User entered an invalid username and/or password, the following occurs:</p> <ol style="list-style-type: none"> 1. The system describes the reasons why the User failed authentication. 2. The system presents the User with suggestions for changes necessary to allow the User to pass authentication. 3. The system prompts the User to re-enter the valid information. 4. The Basic Flow continues where the User enters new information.
Pre-Conditions	
Title	Description
Register	A user of the System creates an account
Post-Conditions	
Title	Description
Success	The User is authenticated and the system displays a home page based on the user type.
Failure	User is unable to log in for one or more reasons.
Extension Points	
The user must make sure that the entered username is a valid username in the System or entered password is a valid password.	

Consumer:

Name	Consumer Request for Fund
Brief Description	The consumer may be whole family or single person who request fund.
Actor(s)	Consumer
Flow of Events	
Basic Flow	
<p>This use case starts when a consumer accesses the “Application Form” feature of the system.</p> <ul style="list-style-type: none"> ➤ The system asks the consumer to enter the appropriate information to get fund (e.g.,Name, Father Name, CNIC, phone number, etc...). ➤ Consumer enters appropriate information and clicks submit. ➤ Consumer validates information. ➤ Consumer goes to our donors. ➤ This Use Case ends. 	
Alternate Flows	
Title	Description
Invalid Information Entered	<ul style="list-style-type: none"> ➤ The system displays an “Invalid Information” error after the user submits the information and asks the Consumer to re-enter the information. ➤ Consumer re-enters information and clicks submit.
Cancel Submitted Application Form	<ul style="list-style-type: none"> ➤ Consumer clicks cancel after selecting the “Application Form” feature. ➤ System returns consumer to the consumer panel.
Pre-Conditions	
Title	Description
Register	A user of the System creates an account

Login	The System user logs in to the System.
Post-Conditions	
Title	Description
Success	The new consumer information is added into the database.
Failure	Consumer information is not added for one or more reasons.
Extension Points	
If the consumer information form is not submitted for one or more reasons the consumer must re-entered or fill valid information in all the requisite information and re-submit the applicationform.	

Donor:

Name	Donors
Brief Description	Donors wants to donate funds for needy.
Actor(s)	Donors
Flow of Events	
Basic Flow	
<p>This use case starts when a donor accesses the "Application Form" feature of the system.</p> <ul style="list-style-type: none"> ➤ The system asks the donors to enter the appropriate information to donate (e.g., Name, Father Name, CNIC, phone number, bank account etc...). ➤ Donors enters appropriate information and clicks submit. ➤ Donors validates information. ➤ Donors goes to our donation page to donate. ➤ This Use Case ends. 	
Alternate Flows	
Title	Description

Invalid Information Entered	<ul style="list-style-type: none"> ➤ The system displays an “Invalid Information” error after the user submits the information and asks the donor to re-enter the information. ➤ Donor re-enters information and clicks submit.
Cancel Submitted Application Form	<ul style="list-style-type: none"> ➤ Donors clicks cancel after selecting the “Application Form” feature. ➤ System returns donor to the donor panel.
Pre-Conditions	
Title	Description
Register	A user of the System creates an account
Login	The System user logs in to the System.
Post-Conditions	
Title	Description
Success	The new donor information is added into the database.
Failure	Donors’ information is not added for one or more reasons.
Extension Points	
If the donor’s information form is not submitted for one or more reasons the donor must re-entered or fill valid information in all the requisite information and re-submit the application form.	

Non-Profit Organization:

Name	Organization request for fund or donate.
Brief Description	Organization wants to donate funds for needy or wants fund for organization.
Actor(s)	Non-Profit Organization
Flow of Events	

Basic Flow	
<p>This use case starts when a Non-Profit Organization accesses the "Application Form" feature of the system.</p> <ul style="list-style-type: none"> ➤ The system asks the organization to enter the appropriate information to donate or request for fund (e.g., Name, Father Name, Organization Name, CNIC, phone number, bank account, location etc...). ➤ Non-Profit Organization enters appropriate information and clicks submit. ➤ Non-Profit Organization validates information. ➤ Non-Profit Organization goes to our donation page to donate and also goes to our donors for request. ➤ This Use Case ends. 	
Alternate Flows	
Title	Description
Invalid Information Entered	<ul style="list-style-type: none"> ➤ The system displays an "Invalid Information" error after the user submits the information and asks the organization to re- enter the information. ➤ Organization re-enters information and clicks submit.
Cancel Submitted Application Form	<ul style="list-style-type: none"> ➤ Organization clicks cancel after selecting the "Application Form" feature. ➤ System returns organization to the organization panel.
Pre-Conditions	
Title	Description
Register	A user of the System creates an account
Login	The System user logs in to the System.
Post-Conditions	
Title	Description
Success	The new organization information is added into the database.
Failure	Organization information is not added for one or more reasons.

Extension Points

If the organization information form is not submitted for one or more reasons the organization must re-entered or fill valid information in all the requisite information and re-submit the application form.

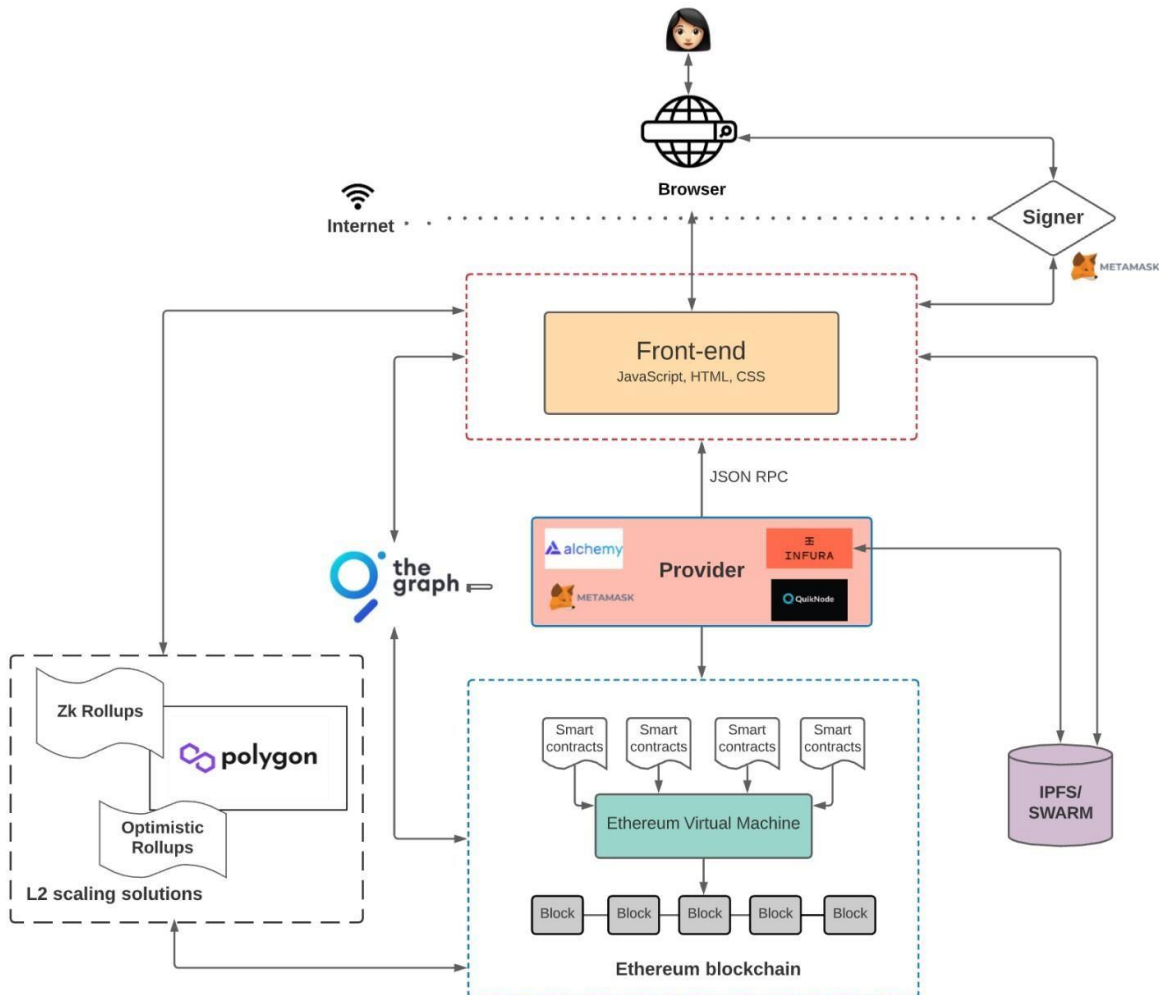
Chapter 4

System Design

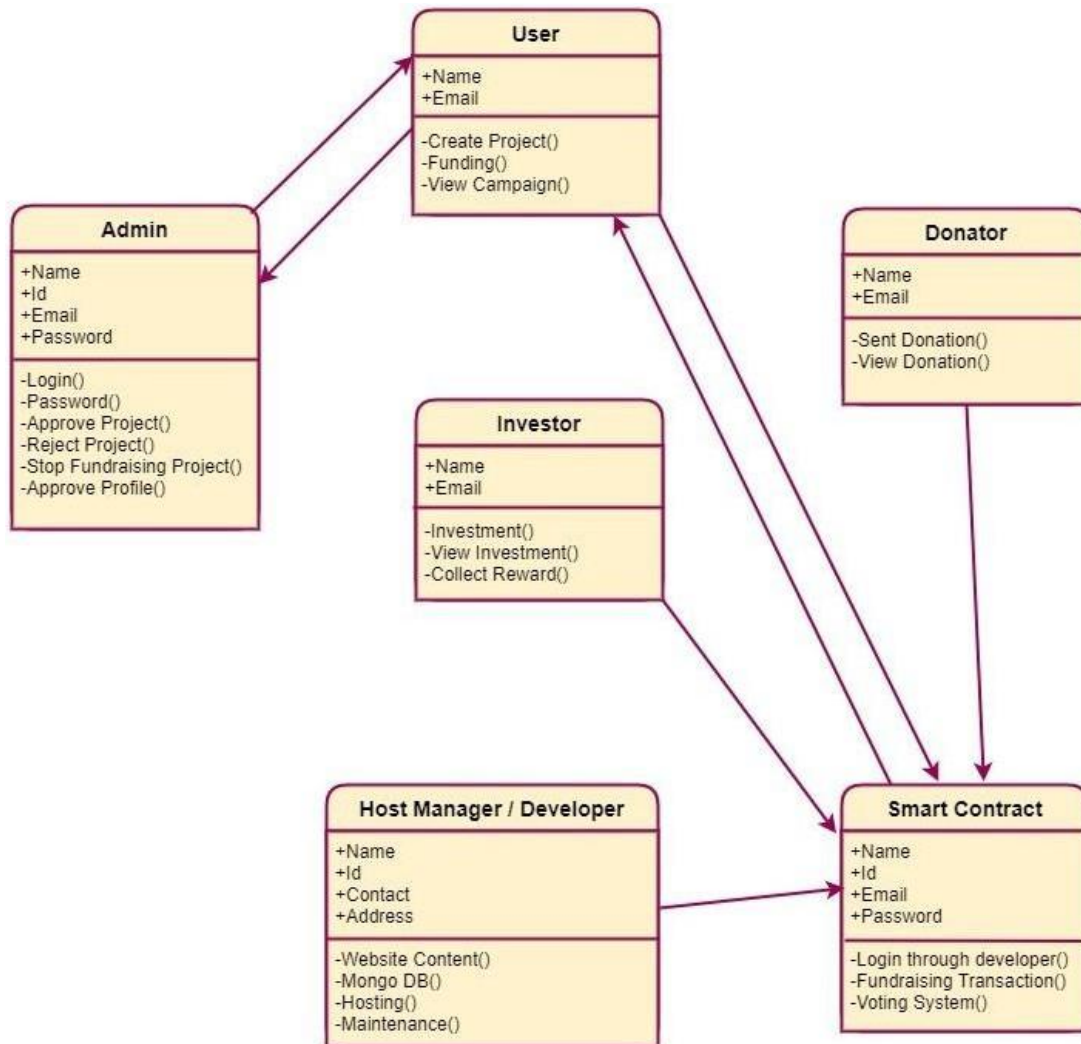
Chapter 4: System Design

The Chapter is all about how the software is going to work and how will the processes be executed as we see we have several diagrams that shows how really is the system performing and what will be the requirements to perform the operations required tasks as well as the diagrams and data clearly describes the process and shows a great help in understanding the system.

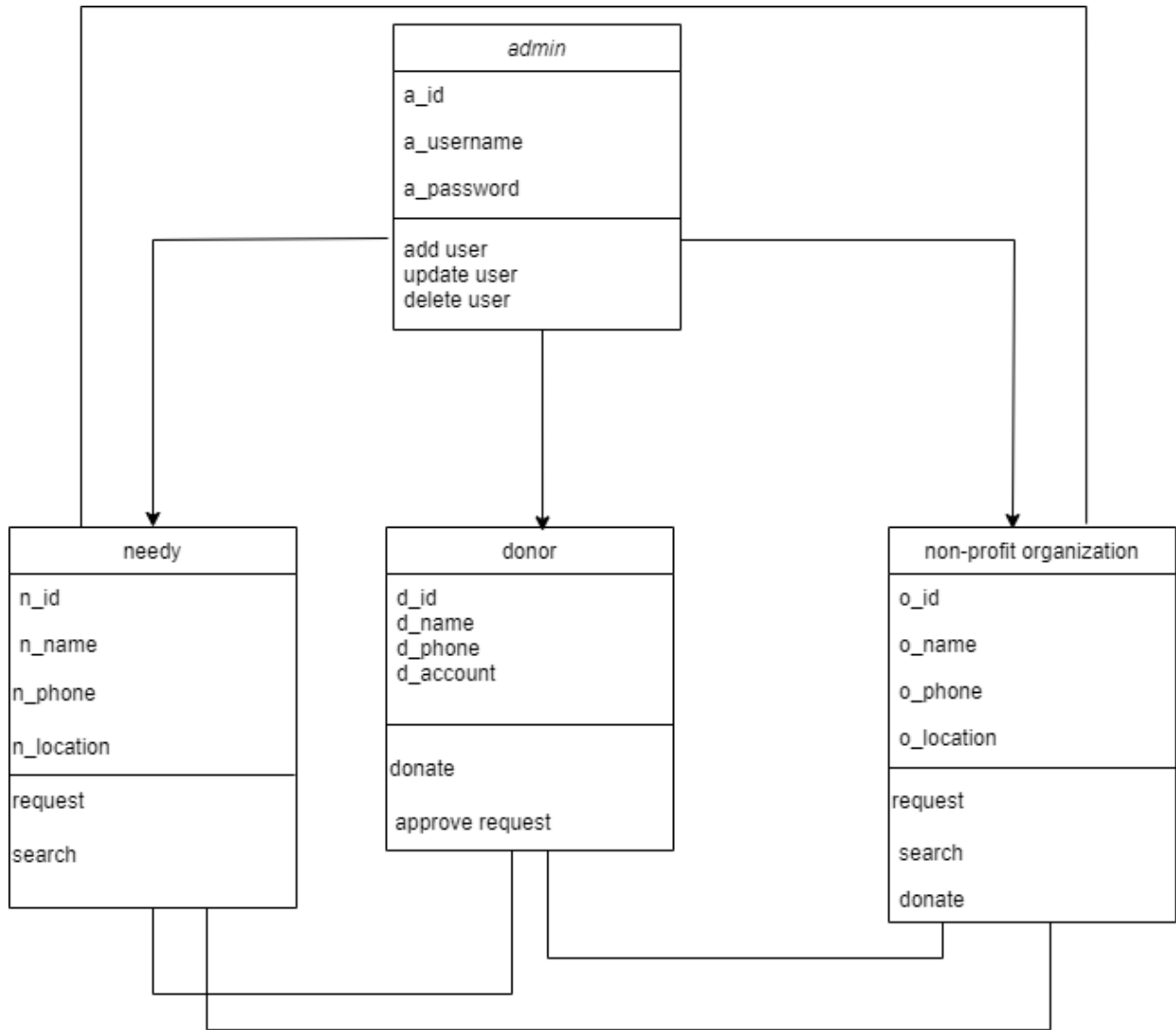
4.1. Architecture Diagram



4.2. Domain Model



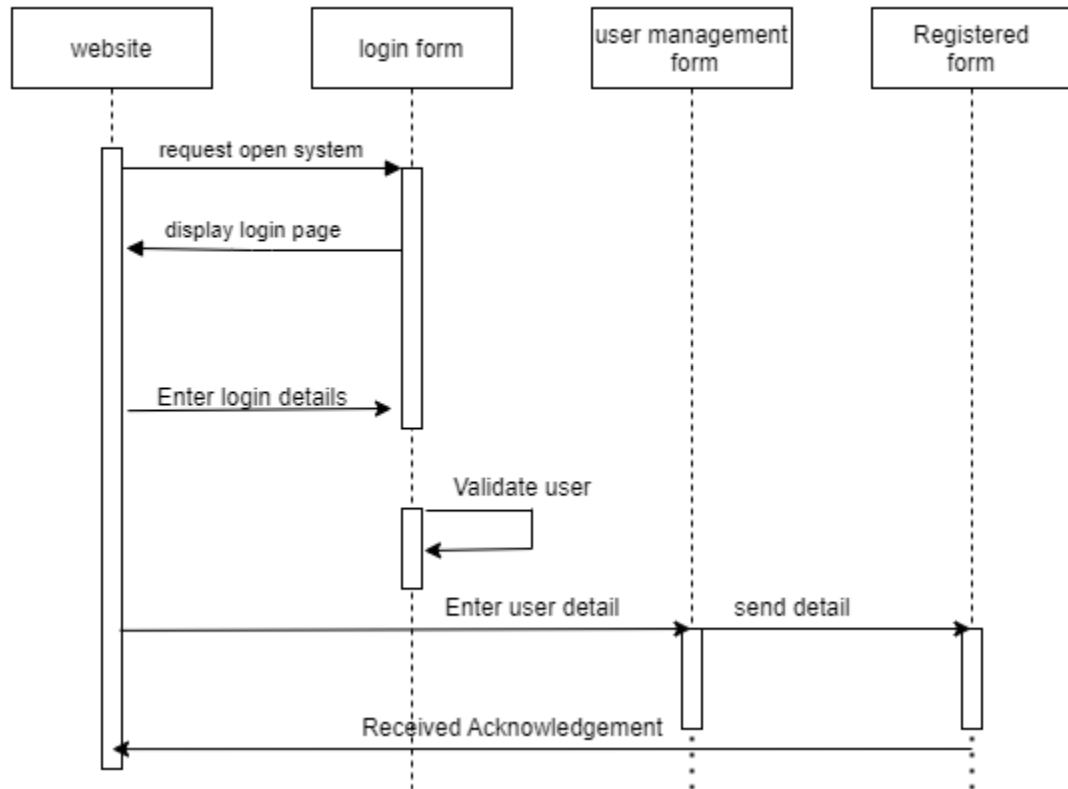
4.3. Class Diagram



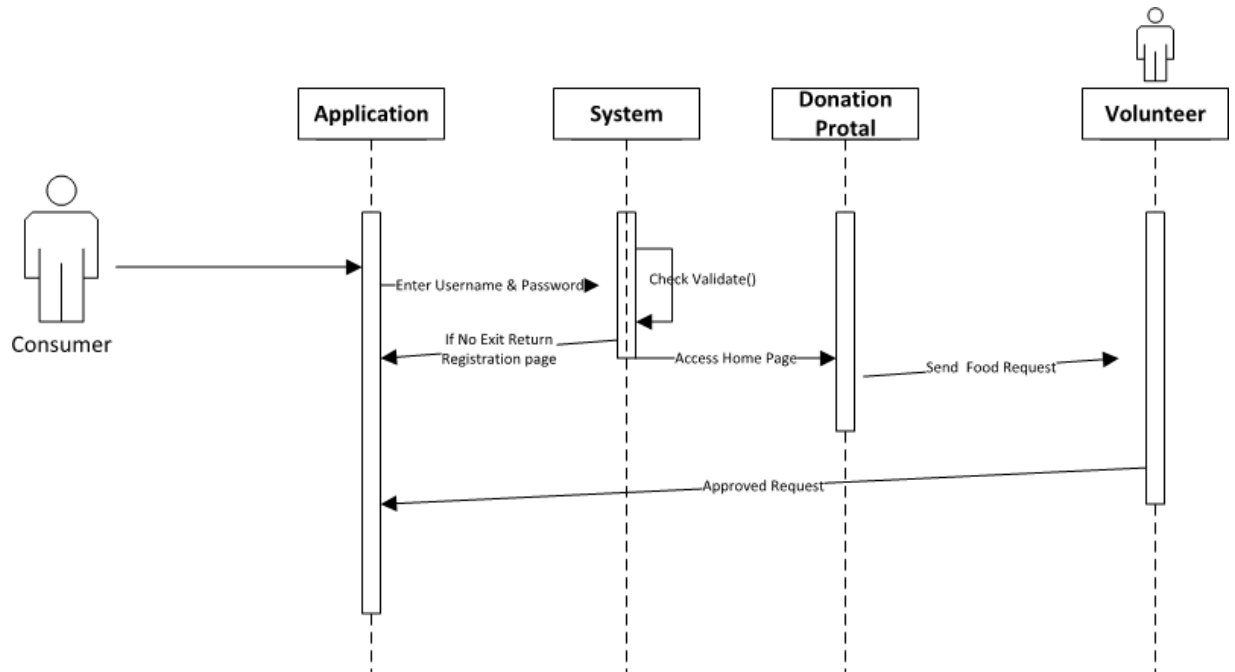
4.4. Sequence / Collaboration Diagram

Sequence Diagram for Admin:

Admin Sequence Diagram



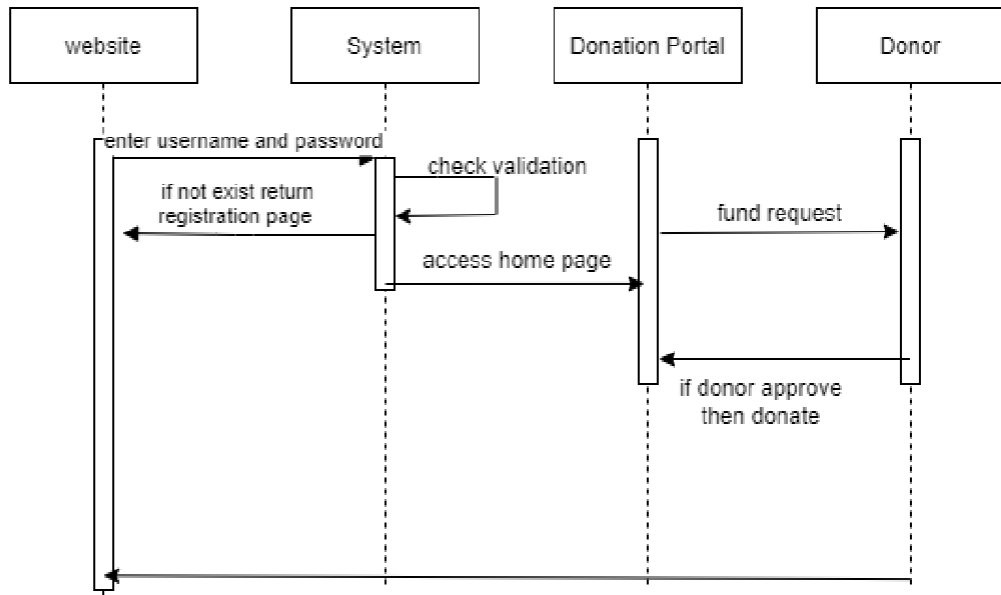
Sequence Diagram for Consumer/Needy:



Consumer Sequence Diagram

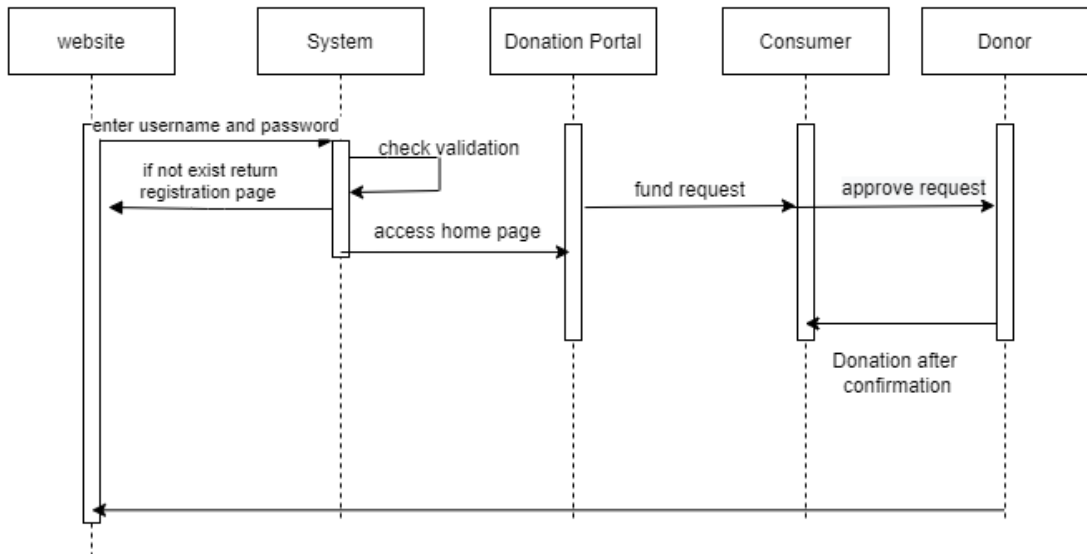
Sequence Diagram for Non-Profit Organization:

Non-Profit Organization Sequence Diagram



Sequence Diagram for Donors:

Donor Sequence Diagram



4.5. Operation contracts

4.6.1 Operation Contracts # 1

Name: Register.

Responsibilities: To register the new user.

Cross References: Use case: Register.

Exceptions: None.

Preconditions: Register interface must be opened and user must enter valid information.

Post conditions: Message pop-ups that you are successfully signed up.

4.6.2 Operation Contracts # 2

Name: Login.

Responsibilities: A user is logged in.

Cross References: Use case: Login.

Exceptions: Invalid username/email or password.

Preconditions: User must have account.

Post conditions: User successfully login.

4.6.3 Operation Contracts # 3

Name: Logout.

Responsibilities: User is logged out.

Cross References: Use case: Logout.

Exceptions: None.

Preconditions: User must be login.

Post conditions: Logout successfully.

4.6.4 Operation Contracts # 4

Name: Donate.

Responsibilities: User must login and donate fund.

Cross References: Use Case: donate.

Exceptions: None.

Preconditions: Must be logged in.

Post conditions: donate successfully.

4.6.5 Operation Contracts # 5

Name: View Consumer.

Responsibilities: donor must View detail of the consumers.

Cross References: Use Case: View consumers.

Exceptions: None.

Preconditions: User must login.

Post conditions: View detail successfully.

4.6.6 Operation Contracts # 6

Name: Consumer Request.

Responsibilities: Consumer can request successfully.

Cross References: Use Case: Consumer Request.

Exceptions: show details to the donors.

Preconditions: User must login.

Post conditions: request successfully.

4.6.7 Operation Contract # 7

Name: Organization Request

Responsibilities: Organization can request successfully.

Cross References: Use Case: Organization request.

Exceptions: show details to the donors.

Preconditions: User must login.

Post conditions: request successfully.

4.6.8 Operation Contract # 8

Name: View Transactions

Responsibilities: must be registered as a donor/as an organization.

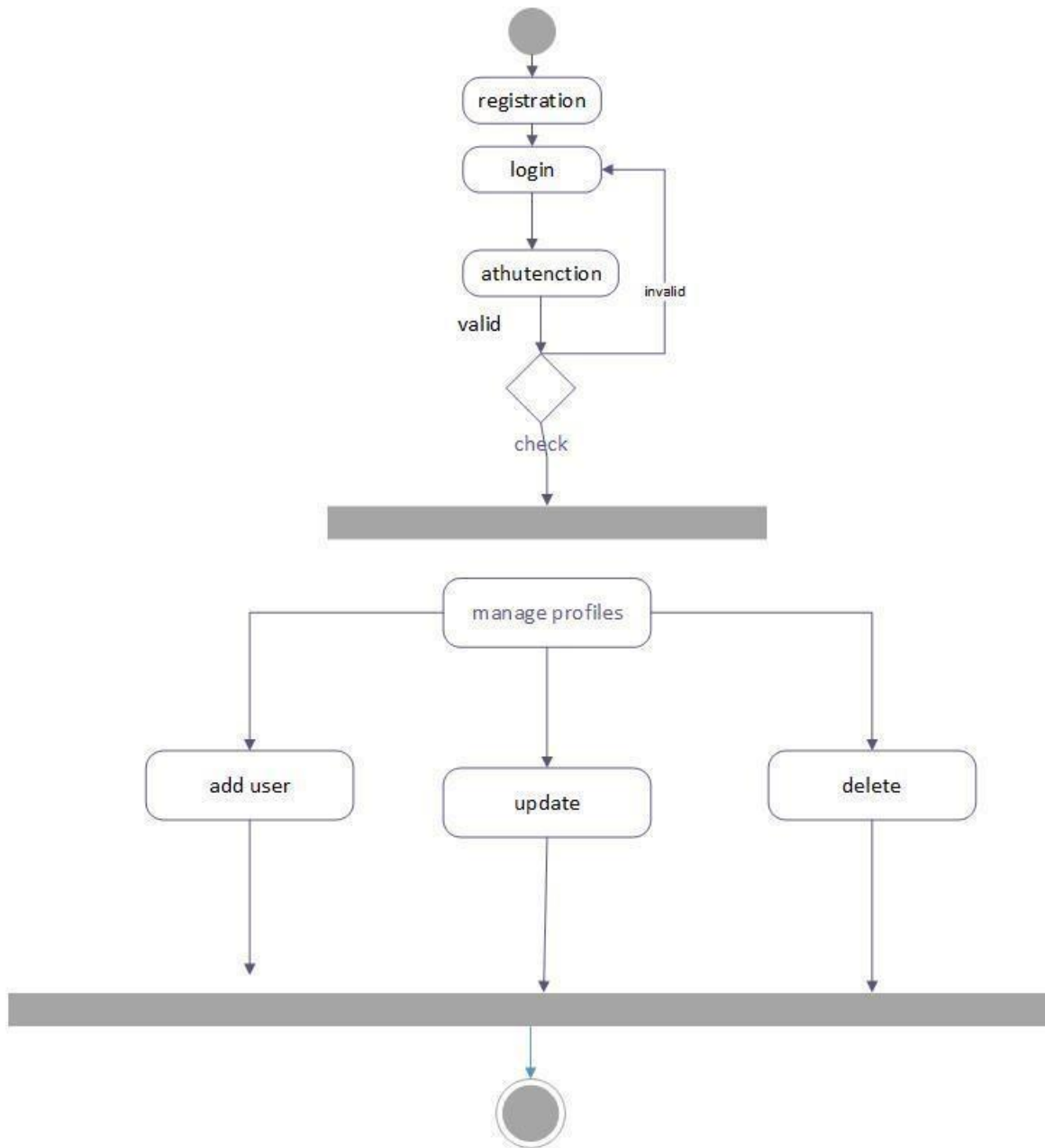
Cross References: Use Case: View Transactions

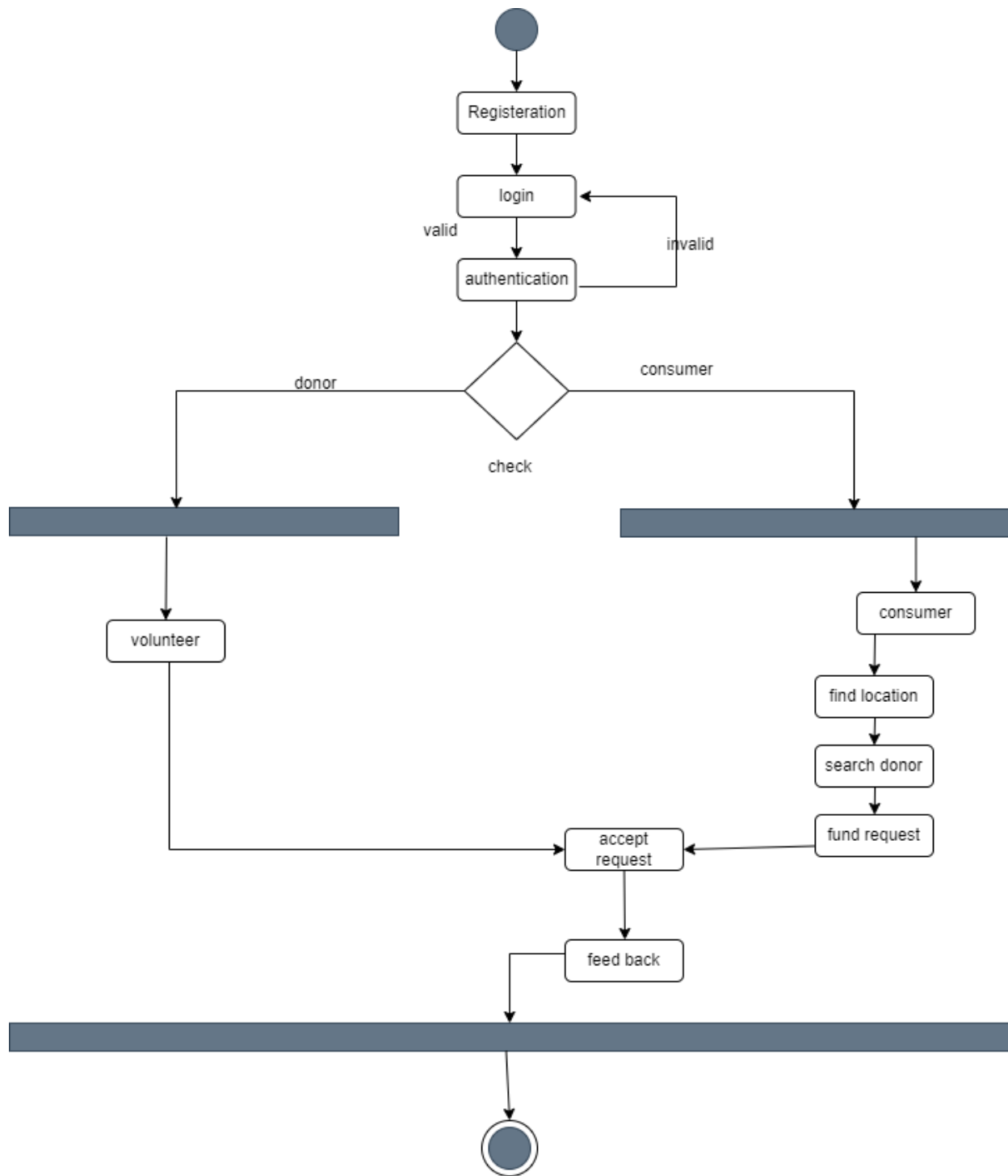
Exceptions: None.

Preconditions: User must login as donor/ as an organization.

Post conditions: View transactions successfully.

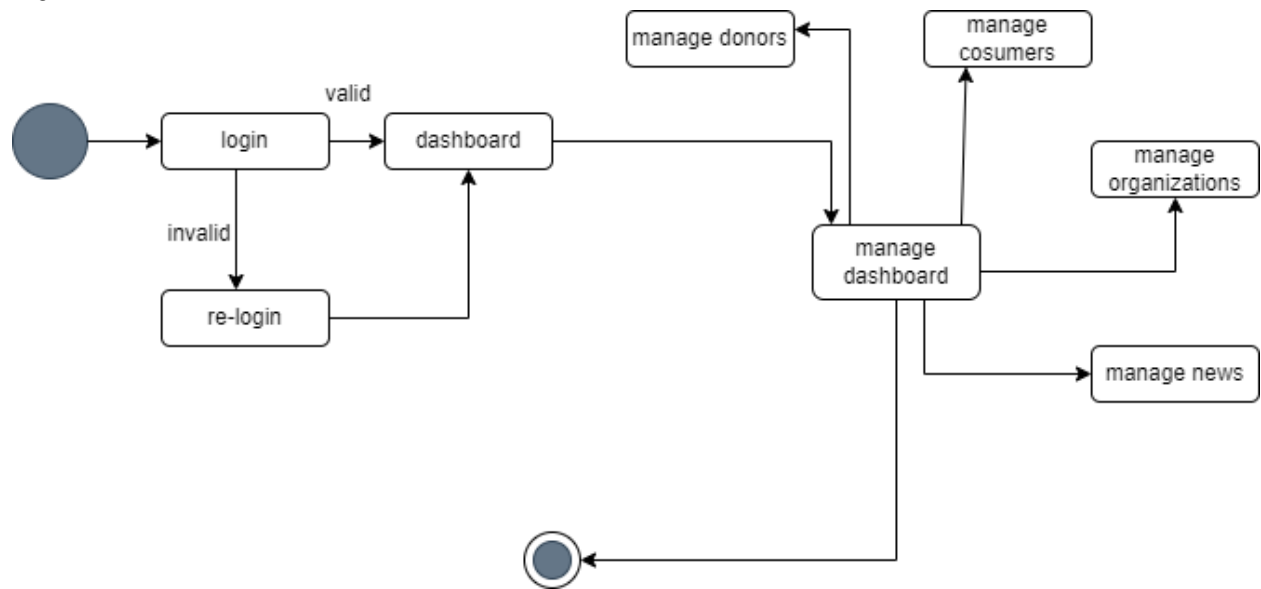
4.7 Activity Diagram



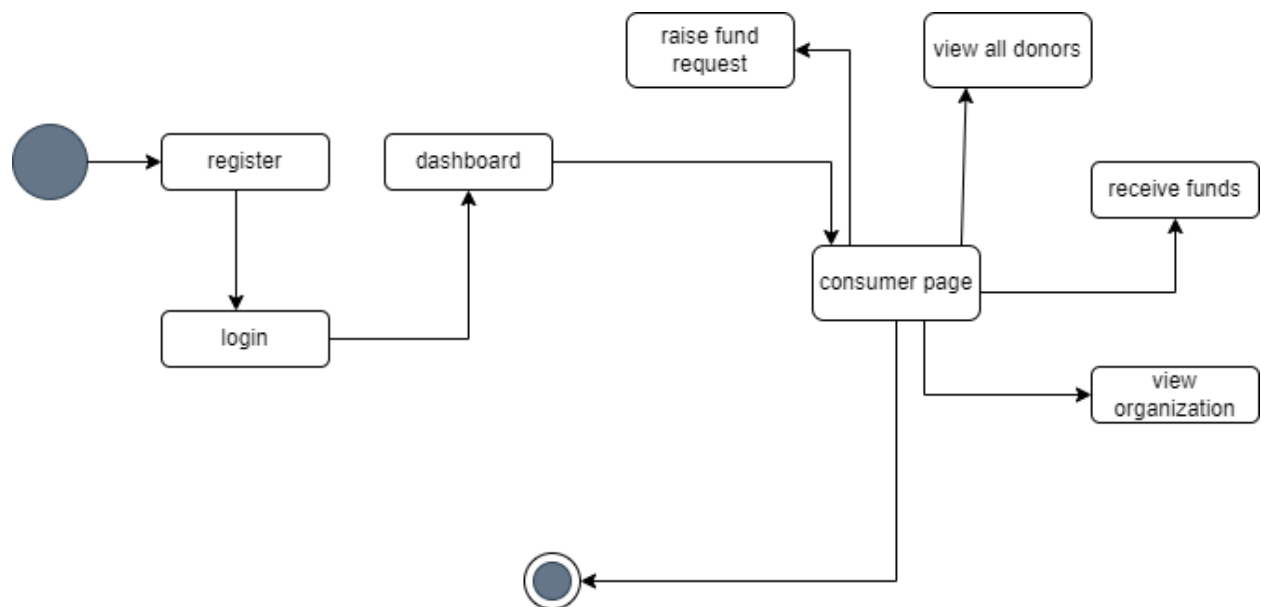


4.8 State Transition Diagram

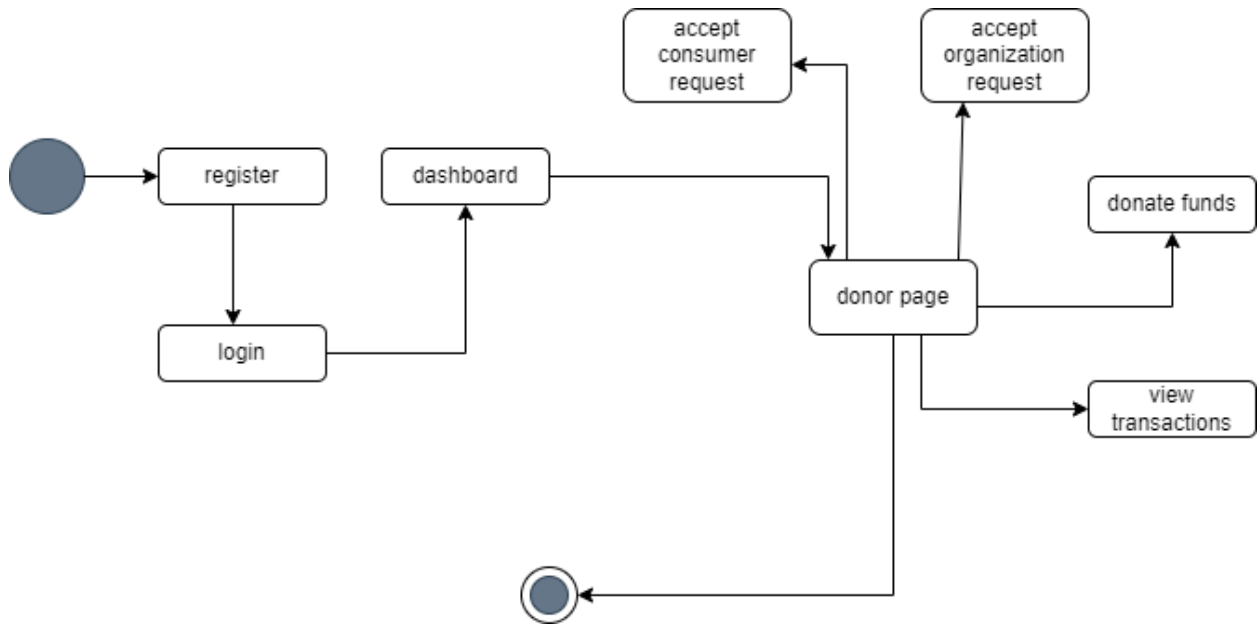
Admin:



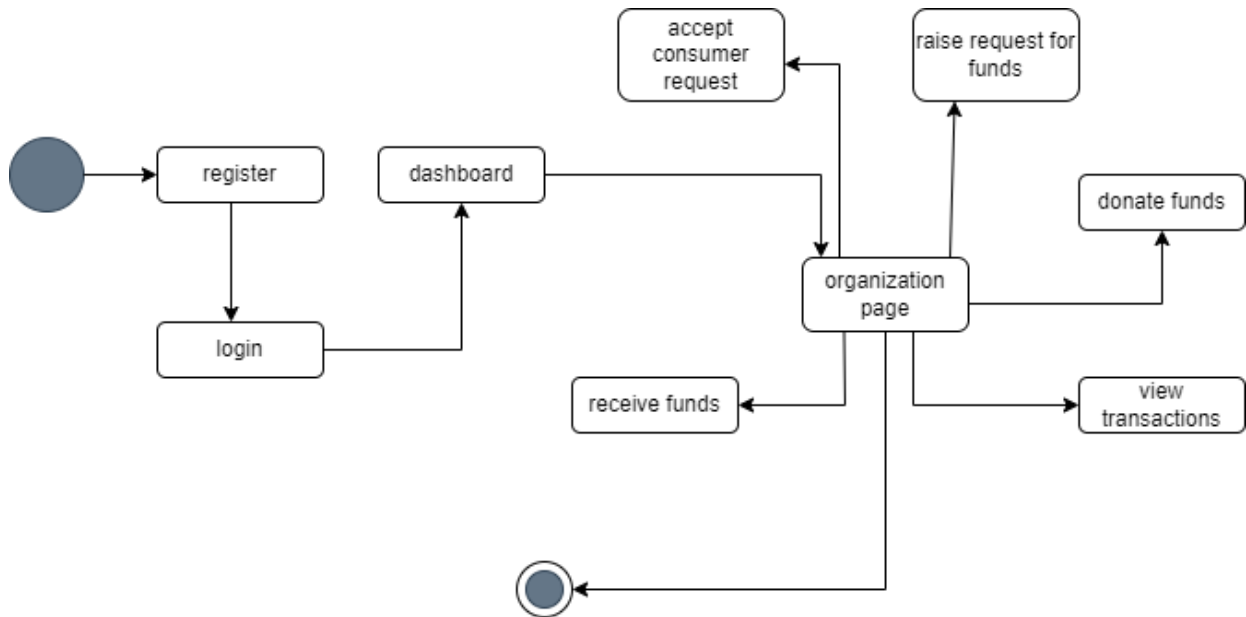
Consumer:



Donor:

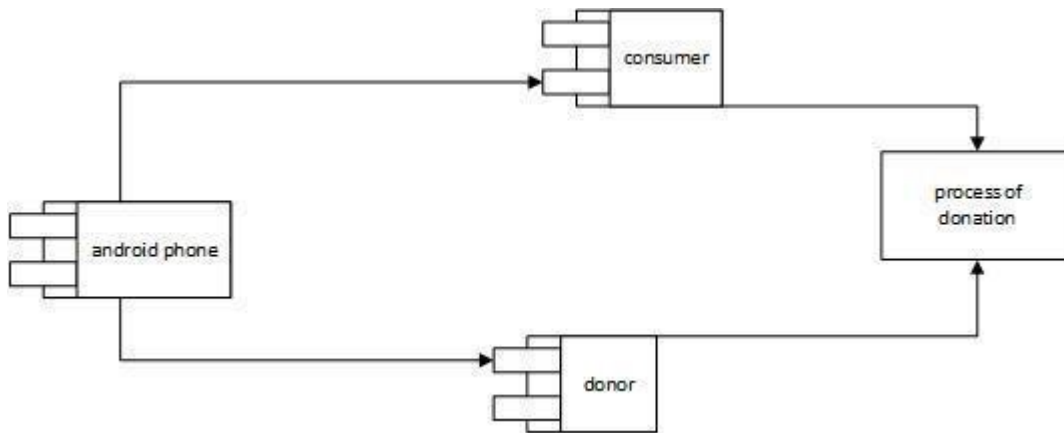


Non-Profit Organization:

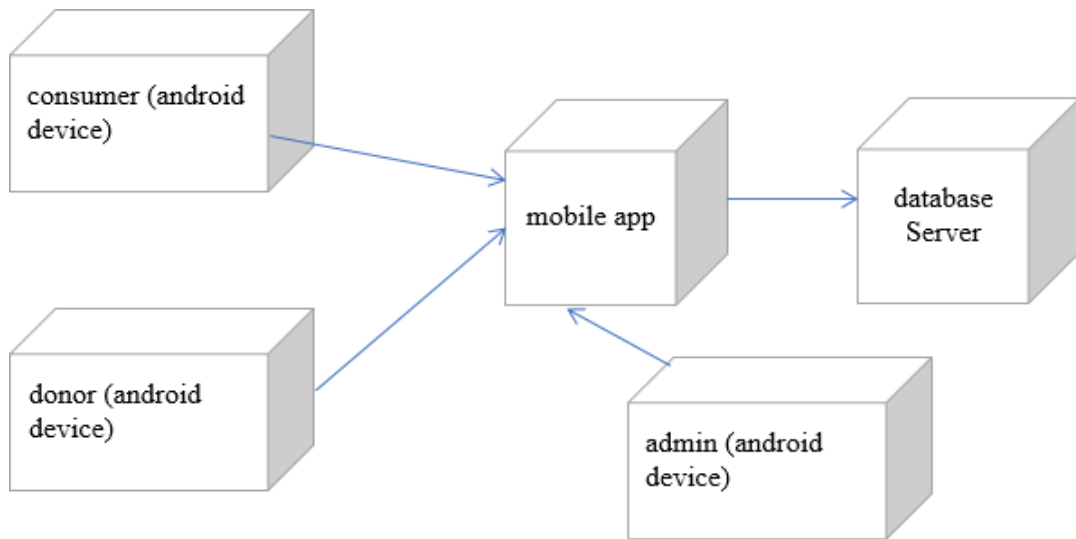


4.9 Component Diagram

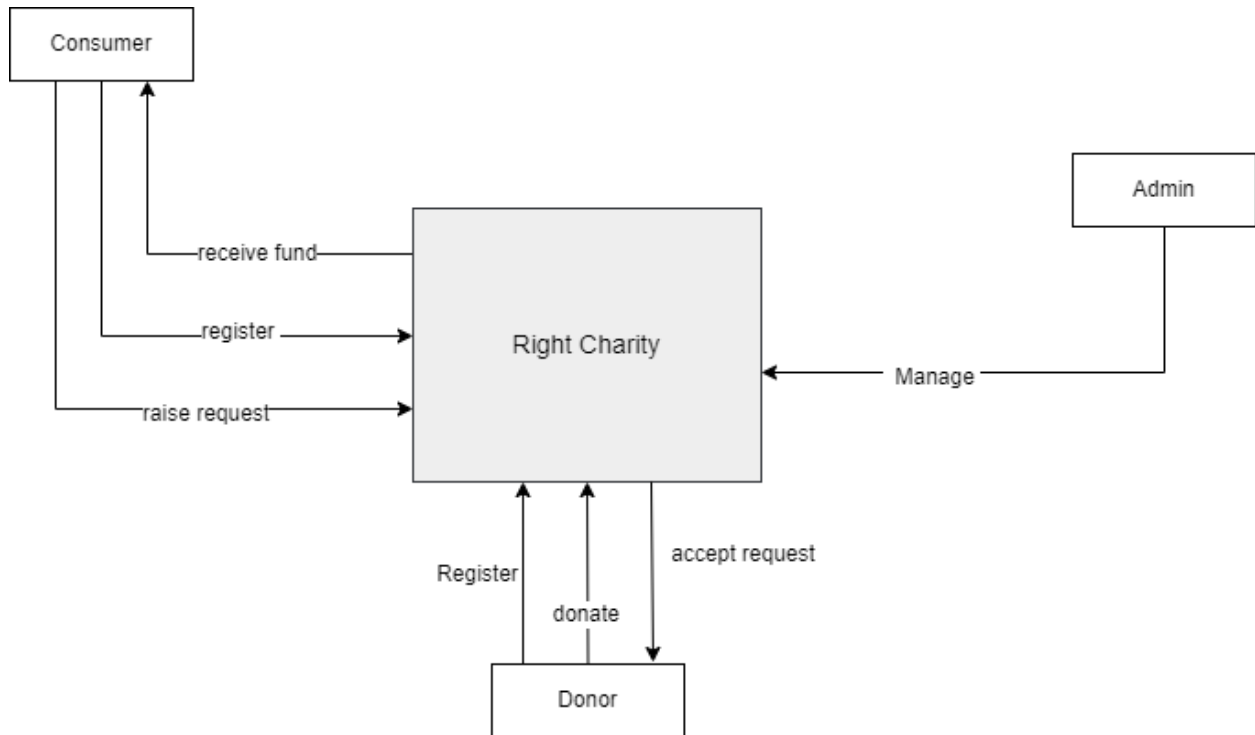
A component diagram allows verification that a system's required functionality is acceptable. These diagrams are presenting the flow and verification system



4.10 Deployment Diagram



4.11 Data Flow diagram



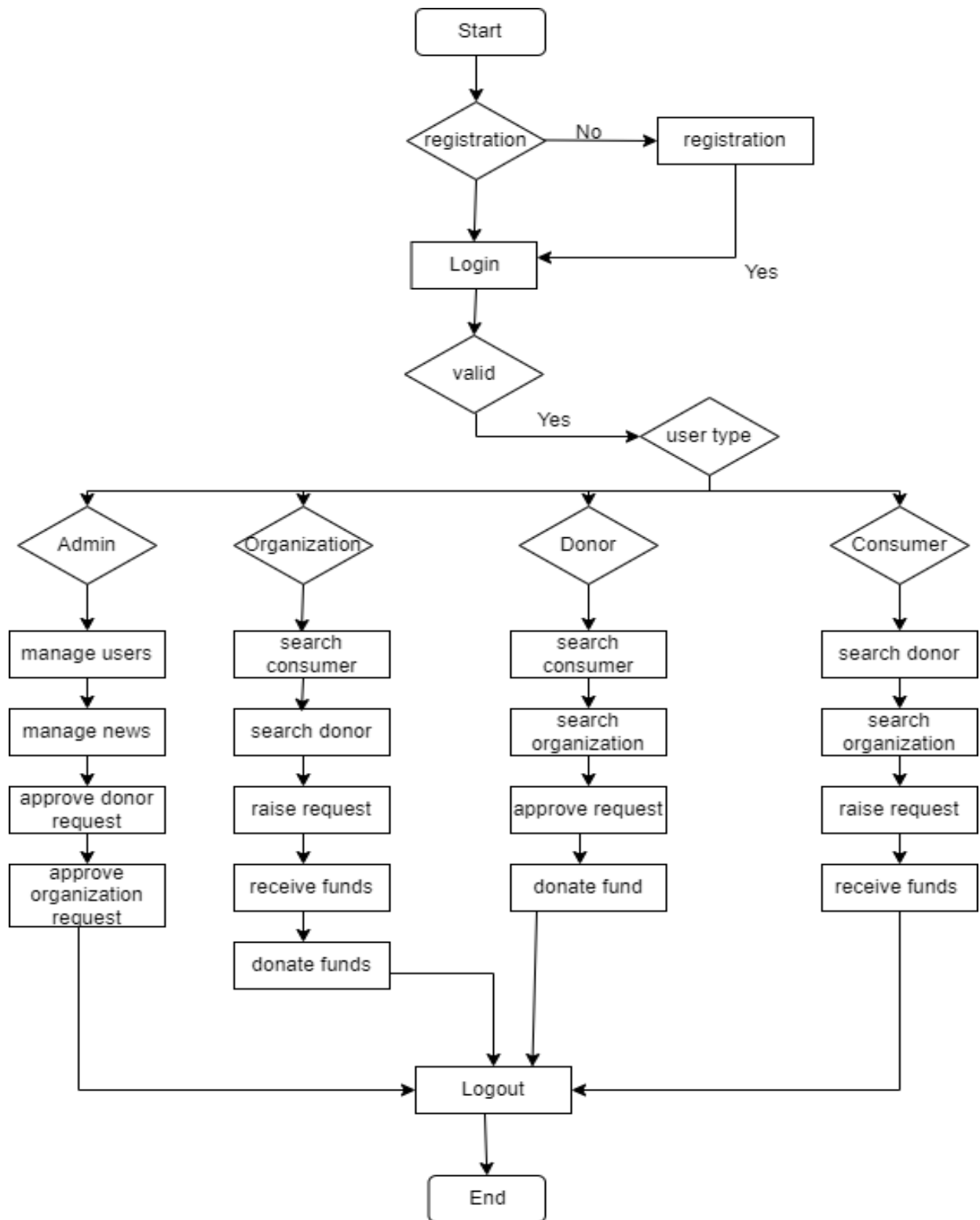
Chapter 5

Implementation

Chapter 5: Implementation

At the end of the system design, it is the beginning for the actual coding to develop the proposed system. During development phase, the table structure of the database will be first built-in order to provide a suitable data type that suit the system back-end development, system connection and data transfer. Next, the computer side client program will be developed and follow by will be the web services that allow web client program to communicate with the server and then we will implement block chain and lastly web side client program will be developed. In testing phase, several test cases will be carried out to test the system in order to determine the system reliability and system accuracy. According to the test cases, a system testing report will be generated for further review to figure out the system weaknesses and made improvement accordingly.

5.1. Important Flow Control/Pseudo codes



5.2. Components, Libraries, Web Services and stubs

- ether.js.
- fs.js.
- frame work
- node.js
- express.js

They are some libraries which we used during development.

Web Services:

- 24-hr access
- Consumer can easily raise request for funds.
- Relation between donors and consumers.
- Security

5.3. Deployment Environment

In Deployment environment we work on Visual Studio Code for development of our website.

5.4. Tools and Techniques

Languages:

Html, CSS, JS. It is used to design the front end of our website.

- **Solidity** Smart contracts are written in this language

- **Node JS:** It is used as backend language that connect blockchain with front interface

Tools

- **Solc:** Solc (Solidity Compiler) is a Solidity command-line compiler written in C++. Its primary purpose is to convert Solidity scripts into a more readable format for the Ethereum Virtual Machine. While Solidity is a slimmed-down, loosely-typed language with a syntax similar to JavaScript, the smart contracts written in it need to be converted to a format that can be easily read and decoded by the EVM. That's where Solc comes into the scene.

- **Remix:** Remix IDE is a browser-based Blockchain tool used for the creation and deployment of smart contracts. Written in Java script (so it can be accessed via any modern browser!), Remix can be used for writing, testing, debugging, and deploying smart contracts written in Solidity. It can be used either locally or in the browser.
- **Metamask:** Metamask is a wallet designed to function that acts as a bridge between Ethereum Blockchain and a browser (Chrome or Firefox). Essentially, it acts as a browser extension. Metamask offers a software platform that allows you to serve Ether and other ERC-20 assets while also letting you interact with Ethereum Dapps. The best part – you can do so right from your browser.
- **Truffle** Truffle is an Ethereum Blockchain framework designed to create a development environment for developing Ethereum-based apps. It comes equipped with a vast library that provides custom deployments for writing new smart contracts, develop complex Ethereum dApps, and help tackle other challenging requirements of Blockchain development.
- **Ganache** Ganache is a Blockchain tool from the Truffle Suite that allows you to create your own private Ethereum blockchain to test dApps, execute commands, and inspect state while taking full control of the operation of the chain.
- **Mist** Mist is the official Ethereum wallet developed by the creators of Ethereum. When it comes to Ethereum, before you can start using the platform, you must have a designated place where you can store your Ether tokens and execute your smart contracts.

Blockchain: We are using a decentralized database known as Blockchain. Blockchain is a shared immutable ledger that contain data of transection being processed by the network. It is a peer to peer, decentralized, without central authority network. Data stores in blocks and each block connect to other and make a chain so that's why it is known as blockchain. No data can be changed once it is created, it works on the consequences of prof of work and prof of the stack. In blockchain the data is verified by each user with the help of asymmetric encryption. Asymmetric encryption work with the concept of private and public keys. Every private key has public key to its corresponding the data which is encrypted with private key is only be decrypted by public key corresponding of that private key. The data in blockchain is maintained and managed by the different distributed groups of computer/ miners. This, along with the different algorithm of security mechanism protect the block of data from alter, modifications and deletions

5.5. Best Practices / Coding Standards

- No coding repetitions
- Function and classes are always commented showing expected input and output.

5.6. Version Control

Version control is a method of tracking changes to documents and files to always know which version is the current iteration. By using project software with version control, or version management, we can efficiently track and control changes to these documents directly within our software. Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. A version control system records all the changes made to a file so a specific version may be called later if needed. And it is initiated version of our website.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Use Case Testing is a functional black box testing technique that helps testers to identify test scenarios that exercise the whole system on each transaction basis from start to finish. And in this chapter, we perform different testing (unit testing, use case testing, equivalence, data flow testing, integration testing) on our system to check the working of our website that our website is performing all the operations or not.

6.1. Use Case Testing

6.1.1.

Test Suite ID	TS001
Test Case ID	TC001
Test Case Summary	To verify that by clicking Signup button username, password, email, phone# and address store in users detail in database.
Related Requirement	RSoo1: User should able to Sign up.
Prerequisites	No
Test Procedure	<ol style="list-style-type: none">1. Select fields in Signup form.2. Enter user data in fields.3. Click Signup button.
Test Data	Valid username: anas, Anas Invalid username: 12 anas ./ anas , anas %^3 Valid password: 123Abc@5, Abc567\$%9 Invalid password: 1_2 anas, _ anas 12 Valid email: anaszaid@gmail.com, anaszaid@yahoo.com Invalid email: ali.com, uzair@yahoo Valid phone# :03116366902

	Invalid phone: @56rfgf7999, 2wstyyA
Expected Result	<ol style="list-style-type: none"> 1. If username, password, email and phone# are valid then store user data by clicking signup button. 2. If given inputs are invalid then display error message. 3. If fields are empty then show warning message.
Actual Result	<ol style="list-style-type: none"> 1. If name is valid, the result is as expected. 2. If name is not valid then invalid message displayed. 3. If fields are empty then warning message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Anas Zaidullah
Date of Creation	01/12/2022
Executed By	Anas Zaidullah
Date of Execution	01/12/2022
Text Environment	Visual Studio Code

Table 13: Use case of Sign up **6.1.2:**

Test Suite ID	TS001
Test Case ID	TC002
Test Case Summary	To verify Authentication or Login.
Related Requirement	RS002: User should able to Login.
Prerequisites	User should Signup first.
Test Procedure	<ol style="list-style-type: none"> 1. Select Username field and enter username. 2. Select Password field and enter password. 3. Click Login button.

Test Data	Valid username: Anas,anas Invalid username: 1 anas, / anas, anas %^3 Valid password: 123Aabc@, abMc567\$% Invalid password: 1_2 anas, _ anas 12
Expected Result	1. If username and password are valid then clicking the Login button user successfully login. 2. If username and password are invalid then clicking the Login button invalid message display. 3. If fields are empty then display warning message
Actual Result	1. If fields are valid, the result is as expected. 2. If fields are not valid then invalid message displayed. 3. If fields are empty then warning message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Anas Zaidullah
Date of Creation	01/12/2020
Executed By	Anas Zaidullah
Date of Execution	01/12/2022
Text Environment	Visual Studio Code

Table 14: use case of login

6.1.3:

Test Suite ID	TS001
Test Case ID	TC003
Test Case Summary	To verify that wedding halls, restaurants, volunteer, consumer and non-profit organization once signup if they are forgot password then they can update it by email.
Related Requirement	RS003: user should able to update data.
Prerequisites	user should login.
Test Procedure	<ol style="list-style-type: none"> 1. Select fields. 2. Enter data in fields. 3. Click forgot password button.
Test Data	01/12/2022
Expected Result	<ol style="list-style-type: none"> 1. If enter data is valid then update data by clicking update button 2. If given inputs are invalid then display error message. 3. If fields are empty then display warning message.
Actual Result	<ol style="list-style-type: none"> 1. If fields are valid, the result is as expected. 2. If fields are not valid then invalid message displayed. 3. If fields are empty then warning message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Anas Zaidullah
Date of Creation	01/12/2022
Executed By	Anas Zaidullah
Date of Execution	01/12/2022
Text Environment	Visual Studio Code

Test Suite ID	TS002
Test Case ID	TC004
Test Case Summary	To verify that donors, consumer and non-profit organization once signup if they are deleted in database by performing crud operation by clicking Delete button.
Related Requirement	RS004: Admin can delete data.
Prerequisites	Admin should login.
Test Procedure	Click delete button.
Expected Result	If data is deleted by click button, then message will appear
Actual Result	If action is valid, the result is as expected.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Anas Zaidullah
Date of Creation	01/12/2022
Executed By	Anas Zaidullah
Date of Execution	01/12/2022
Text Environment	Visual Studio Code

Table 16: use case of delete data

6.1.6:

Test Case ID	TC006
Test Case Summary	To verify that record is display.

Related Requirement	RS006: User should be able to view data.
Prerequisites	User should login.
Test Procedure	<ol style="list-style-type: none"> 1. Enter name in fields. 2. Click View button.
Expected Result	<ol style="list-style-type: none"> 1. If name is valid then show detail. 2. If name is invalid then display error message. 3. If field is empty then display warning message.
Actual Result	<ol style="list-style-type: none"> 1. If name is valid, the result is as expected. 2. If name is not valid then invalid message displayed. 3. If fields are empty then warning message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Anas Zaidullah
Date of Creation	01/12/2022
Executed By	Anas Zaidullah
Date of Execution	01/12/2022
Text Environment	Visual Studio Code

Table 18: use case of record to display

6.1.7:

Test Suite ID	TS002
Test Case ID	TC007
Test Case Summary	To verify the transactions flow
Related Requirement	RS007: User should be able to view data.
Prerequisites	User should connect to the wallet.
Test Procedure	<ol style="list-style-type: none">1. Enter name in fields.2. Click donate.
Expected Result	<ol style="list-style-type: none">1. If record is valid then transaction will be successful.2. If record is invalid then display error message.3. If field is empty then display warning message.
Actual Result	<ol style="list-style-type: none">1. If record is valid, the result is as expected.2. If record is not valid then invalid message displayed.3. If fields are empty then warning message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Anas Zaidullah
Date of Creation	01/12/2022
Executed By	Anas Zaidullah
Date of Execution	01/12/2022
Text Environment	Visual Studio Code

Table 19: use case of transactions

6.2. Equivalence partitioning

Registration

Input	Invalid Equivalence class Partitioning	Valid Equivalence class Partitioning
Email	Email without proper format not allowed i.e., user123	Email with proper format is allowed i.e., anas123@gmail.com

Contact	Contact would not be in alphabetic form i.e., user123	It must be in No. 03116366902
Password	Less than 3 not allow	More than 3 or equal 3 allowed

Login

Input	Invalid Equivalence class Partitioning	Valid Equivalence class Partitioning
Email	Email without proper format not allowed i.e., user123	Email with proper format is allowed i.e., anaszaid123@gmail.com
Password	Less than 3 not allow	More than 3 or equal 3 allowed

6.3. Boundary value analysis

Register/LoginEmail

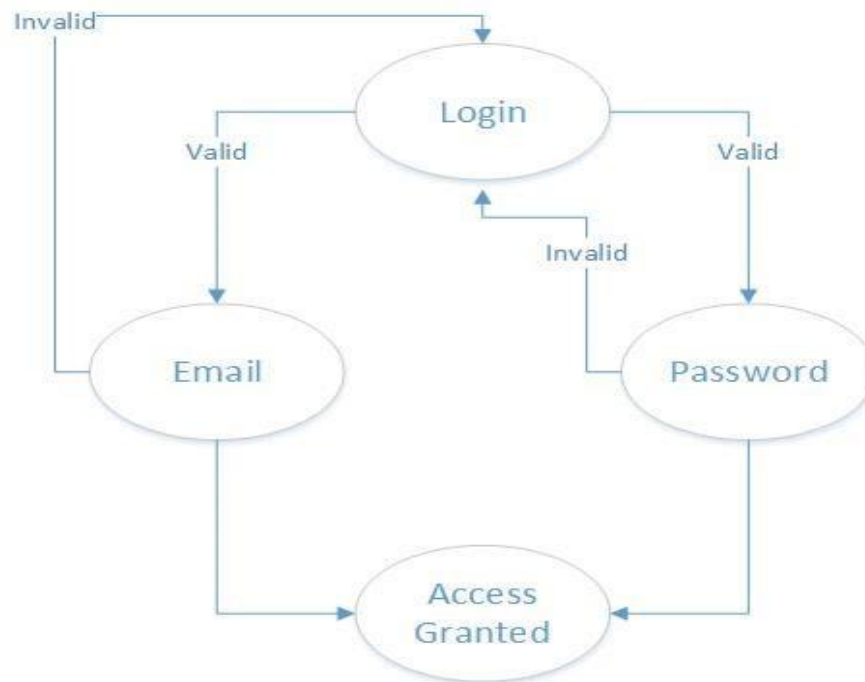
Invalid (Min-1)	Valid (Min, +max, -min, max)	Invalid (max+1)
10	10 to 50	50

Password

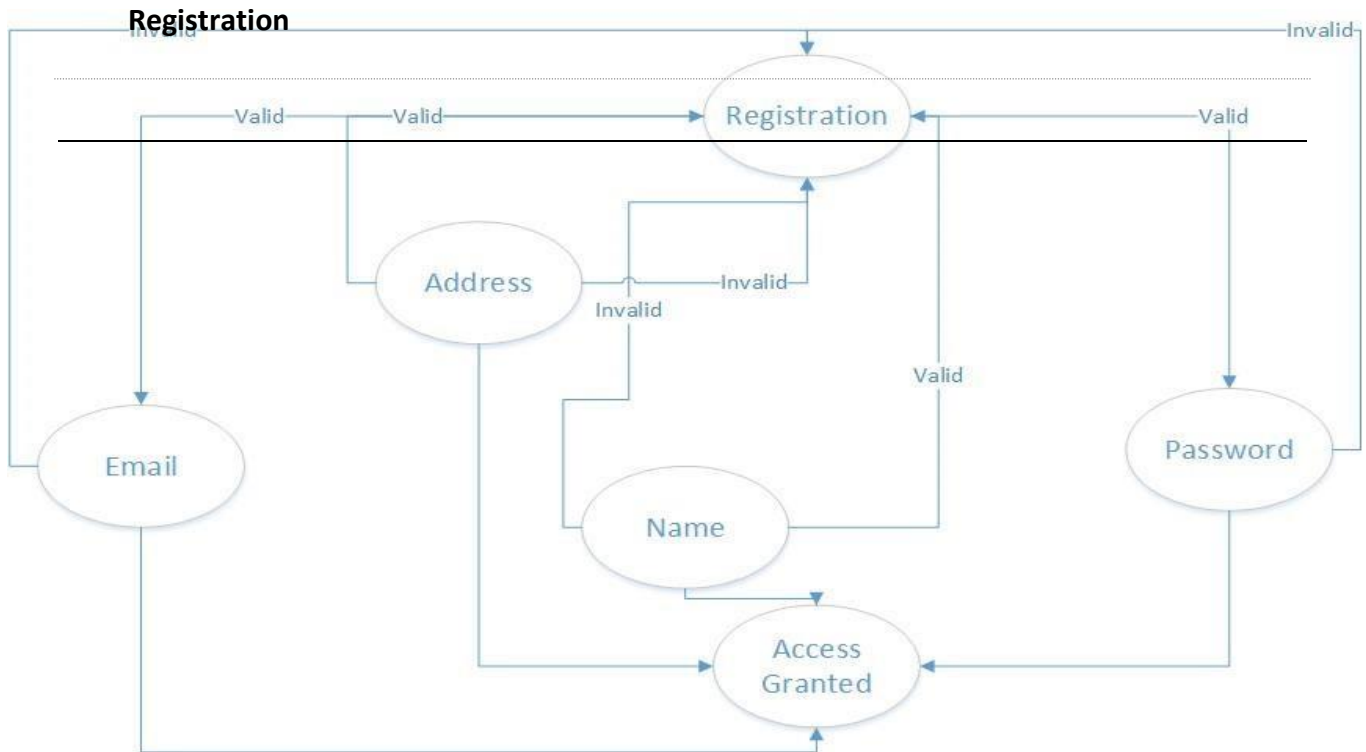
Invalid (Min-1)	Valid (Min, +max, -min, max)	Invalid (max+1)
8	8 to 25	25

6.4. Data flow testing

Login



Data Flow Login



Data Flow Registration

6.5. Unit testing

Unit testing is a white-box testing technique. The main consideration in this test is verification of all modules of the software system. Each module is unit tested, as follows:

- Unit testing for the module User Management is tested by valid user or invalid user.
- Unit testing for the module User Account management is provided by testing Login, Logout and creating new or deleting updating existing user.
- Unit testing for the module Detail of users give detail accurately.
- Unit testing for the donors, organization, consumer portal is performed by doing different transactions, requests, approve request, accept funds etc.
- Unit testing for the module update/delete data.
- Unit testing for the module customer is performed by searching donors and organizations and vice versa.

6.6. Integration testing

Integration testing is the technique for testing the interfaces of software components. Each software components in unit tested, and all the components are integrated to perform together. The tests are conducted to ensure that the components are working properly after interfacing.

All modules are integrated by an incremental approach, and integration testing of the system is performed as follows:

- Integrate login module and test that the software is properly connected to the database.
- Integrate all modules and test by inserting updating, and deleting records.

6.7. Performance testing

Performance testing uses criteria to check that the system functions according to the specifications. All unit modules have been tested successfully. The integration of these unit modules produces reports in the required format.

6.8. Stress Testing

A system evaluated under normal conditions in all previous testing techniques, and no conditions where system can fail are tested. Stress testing evaluates the system under abnormal conditions. The system is tested for various costs and number to check accuracy and to ensure that the system performs accurately under all input conditions.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future

Enhancements

7.1. Project Summary

Now that the project Right Charity is completed so that I can explain the purpose of the website. The main purpose of this website is to control all transactions and provide security and ease to user.

This system will overcome the distance between donors and consumer. Our software would generate the interface and its associated functionalities according to the requirement of donor and consumer. This system will provide many features in a singleplatform.

That said, there is always room for improvement.

7.2. Achievements and Improvements

The biggest achievements here are that we were able to enhance our skills to the professional extend that we learned in four years of studying and apply it to this project. We learned software architecture design techniques, UML modeling, project management, testing and much more, and were able to apply it all in this project.

The next big achievement is the things we learnt during this project. New languages, frameworks, libraries, different software's for diagrams Database. All that will be useful for us in our futures. We learn Creative thinking.

7.3. Critical Review

The critical element of our system is: Is our system is fast moving, give user friendly interface, fulfill the specific requirements of the users at a runtime, has all the features which any foundation can require. To manage all these aspects, it took lot of time and hard work.

7.4. Lessons Learnt

We learn very much from this project. This project sharpens our skills in web development and block chain and many other tools and many management concepts as well as how to deal with a problem and how to stick for finding the solution of any problem until you find. As well as technical skills this project also enhances our personal development skills such as team working, dedication.

7.5 Future Enhancements/Recommendations

As it has been already said, there is always room for further improvement. And since we plan to launch this website as our own startup, enhancements will keep coming.

The system has been built on react native, but at some point, we work on its efficiency, accessibility, flexibility, we are definitely going to scale it up which we believe ours is eventually going to become. We make it more user-friendly interface and can deliver to the different foundations.

Reference and Bibliography

Reference and Bibliography

- [1] C. L. G. C. H. H. R. W. a. M. M. Yunjo Lee, "Face Processing Changes in Normal Aging Revealed by fMRI Adaptation," 2011.
- [2] A. M. Abu Nada, "Age and Gender Prediction and Validation Through Single User Images Using CNN," 2020.
- [3] M. Afifi, *11K Hands: Gender recognition and biometric identification using a large dataset of hand images.*, 2019.
- [4] A. P. S. e. a. Kataria, "Reliability of digitized cheiloscopy, dactyloscopy and combination approach for gender identification—A comparative study.," 2020.
- [5] H. a. P. W. Zhao, "A short review of age and gender recognition based on speech.," 2020.
- [6] M. S. Z. Rizvi, 18 Feb 2020. [Online]. Available: <https://www.analyticsvidhya.com/blog/2020/>.
- [7] 07 March 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8659655/>.
- [8] A. R. N. & D. R. Rattani. [Online]. Available: <https://doi.org/10.1049/iet-bmt.2017.0171>.
- [9] R. S. S. C. C. D. & C. R. Ranjan, 2017. [Online]. Available: <https://doi.org/10.1109/AIMS.2013.13>.
- [10] G. & H. T. Levi, 2015. [Online]. Available: <https://doi.org/10.1109/CVPRW.2015>.

Index

Index

[A]

[B]

[

