

CharitAble for UOG



CharitAble

FOR U-O-G

Submitted By

Qandeel Zulfiqar	19010819-074
Muhammad Awais	19010819-111
Iqra Hassan	19010819-087

Supervisor

Mr. Zaheer Ahmad

2019 / 2021

University of Gujrat
City Campus

CHARITABLE FOR UOG

Project Submitted to
Department of Computer Science
University of Gujrat, City Campus

Submitted By

Qandeel Zulfiqar	19010819-074
Muhammad Awais	19010819-111
Iqra Hassan	19010819-087

In partial fulfillment of the requirements for the degree of
(MSc-CS)

Department of Computer Sciences
University of Gujrat, City Campus

Group Detail

Project ID	
------------	--

CharitAble for UOG	
Group Leader:	Muhammad Awais
Group Members:	3

Member Name	Registration No	Email Address	Class
Qandeel Zulfiqar	19010819-074	19010819-074@uog.edu.pk	MSc-CS
Muhammad Awais	19010819-111	19010819-111@uog.edu.pk	MSc-CS
Iqra Hassan	19010819-087	19010819-087@uog.edu.pk	MSc-CS

Mr. Zaheer Ahmad

Lecturer, UOG

Proofreading Certificate

This is acknowledge that the project entitled

CharitAble for UOG

Has been proofread

Mr. Zaheer Ahmad

Supervisor

Lecturer, UOG

Declaration

We hereby declare that we developed this project (**CharitAble for UOG**) and report entirely on the basis of our personal efforts made under the sincere guidance of our project supervisor. We further declare that, the title project and all associated documents, report are submitted as partial requirements for the degree of “MSc-CS”.

Members' Name	Registration NO	Signature
Qandeel Zulfiqar	19010819-074	_____
Muhammad Awais	19010819-111	_____
Iqra Hassan	19010819-087	_____

TABLE OF CONTENTS

CHAPTER 1: PROJECT FEASIBILITY REPORT **ERROR! BOOKMARK NOT DEFINED.**

1.1. INTRODUCTION.....	ERROR! BOOKMARK NOT DEFINED.
1.2. PROJECT/PRODUCT FEASIBILITY REPORT.....	ERROR! BOOKMARK NOT DEFINED.
1.2.1. Technical Feasibility.....	<i>Error! Bookmark not defined.</i>
1.2.2. Operational Feasibility.....	<i>05Error! Bookmark not defined.</i>
1.2.3. Economic Feasibility	<i>Error! Bookmark not defined.</i>
1.2.4. Schedule Feasibility.....	<i>Error! Bookmark not defined.</i>
1.2.5. Specification Feasibility.....	<i>Error! Bookmark not defined.</i>
1.2.6. Information Feasibility	<i>Error! Bookmark not defined.</i>
1.2.7. Motivational Feasibility.....	<i>Error! Bookmark not defined.</i>
1.2.8. Legal & Ethical Feasibility.....	<i>Error! Bookmark not defined.</i>
1.3. PROJECT/PRODUCT SCOPE	ERROR! BOOKMARK NOT DEFINED.
1.4. PROJECT/PRODUCT COSTING	ERROR! BOOKMARK NOT DEFINED.
1.4.1. Project Cost Estimation By Function Point Analysis	<i>1Error! Bookmark not defined.</i>
1.4.2. Project Cost Estimation by using COCOMO '81 (Constructive Cost Model)	<i>11</i>
1.4.3. Activity Based Costing	<i>18</i>
1.5. TASK DEPENDENCY TABLE	ERROR! BOOKMARK NOT DEFINED.
1.6. CPM - CRITICAL PATH METHOD.....	ERROR! BOOKMARK NOT DEFINED.
1.7. GANTT CHART	ERROR! BOOKMARK NOT DEFINED.
1.8. INTRODUCTION TO TEAM MEMBER AND THEIR SKILL SET ..	ERROR! BOOKMARK NOT DEFINED.
1.9. TASK AND MEMBER ASSIGNMENT TABLE.....	ERROR! BOOKMARK NOT DEFINED.
1.10. TOOLS AND TECHNOLOGY WITH REASONING ..	ERROR! BOOKMARK NOT DEFINED.
1.11. VISION DOCUMENT	ERROR! BOOKMARK NOT DEFINED.
1.12. RISK LIST	ERROR! BOOKMARK NOT DEFINED.

CHAPTER 2: SOFTWARE REQUIREMENT SPECIFICATION **.....ERROR! BOOKMARK NOT DEFINED.**

2.1 Introduction.....	<i>Error! Bookmark not defined.</i>
2.2 Systems Specifications	<i>Error! Bookmark not defined.</i>
2.2.1 organization chart.....	<i>Error! Bookmark not defined.</i>
2.2.2 Existing System	<i>Error! Bookmark not defined.</i>
2.2.3 Scope of the System.....	<i>Error! Bookmark not defined.</i>
2.2.4 Summary of Requirements:(Initial Requirements)	<i>Error! Bookmark not defined.</i>
2.2.5 Identifying External Entities:.....	<i>Error! Bookmark not defined.</i>
2.2.6 Capture "shall" Statements:.....	<i>Error! Bookmark not defined.</i>
2.2.7 Allocate Requirements:.....	<i>Error! Bookmark not defined.</i>
2.2.8 Priorities Requirements:.....	<i>Error! Bookmark not defined.</i>
2.2.9 High Level Usecase Diagram:.....	<i>Error! Bookmark not defined.</i>
2.10. Analysis Level Usecase Diagram:	<i>Error! Bookmark not defined.</i>
2.11. Usecase Description	<i>Error! Bookmark not defined.</i>

CHAPTER 3: DESIGN DOCUMENT (FOR OBJECT ORIENTED APPROACH) **.....ERROR! BOOKMARK NOT DEFINED.**

3.1. INTRODUCTION:	ERROR! BOOKMARK NOT DEFINED.
3.2. SEQUENCE DIAGRAM	ERROR! BOOKMARK NOT DEFINED.
3.3 <i>Defining a Sequence diagram</i>	<i>Error! Bookmark not defined.</i>
3.4. DESIGN CLASS DIAGRAM.....	ERROR! BOOKMARK NOT DEFINED.
3.5. DATA MODEL.....	ERROR! BOOKMARK NOT DEFINED.
3.1. INTRODUCTION:	ERROR! BOOKMARK NOT DEFINED.
3.2. ENTITY RELATIONSHIP DIAGRAM:	ERROR! BOOKMARK NOT DEFINED.
3.3. DATA FLOW DIAGRAM (FUNCTIONAL MODEL).....	40
3.4. STATE TRANSITION DIAGRAM.....	41
3.5. ARCHITECTURAL DESIGN	41
3.6. COMPONENT LEVEL DESIGN	41

CHAPTER 4: USER INTERFACE DESIGNERROR! BOOKMARK NOT DEFINED.

4.1. INTRODUCTION.....	ERROR! BOOKMARK NOT DEFINED.
4.2. SITE MAPS	ERROR! BOOKMARK NOT DEFINED.
4.3. STORY BOARDS	ERROR! BOOKMARK NOT DEFINED.

CHAPTER 5: SOFTWARE TESTINGERROR! BOOKMARK NOT DEFINED.

5.1 INTRODUCTION:	ERROR! BOOKMARK NOT DEFINED.
5.2. TEST PLAN.....	ERROR! BOOKMARK NOT DEFINED.
7.2.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
7.2.2. <i>Outline</i>	<i>Error! Bookmark not defined.</i>
5.3. TEST DESIGN SPECIFICATION	ERROR! BOOKMARK NOT DEFINED.
5.3.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
5.3.2. <i>Outline</i>	<i>Error! Bookmark not defined.</i>
5.4. TEST CASE SPECIFICATION.....	ERROR! BOOKMARK NOT DEFINED.
5.4.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
5.4.2. <i>Outline</i>	<i>Error! Bookmark not defined.</i>
5.5. TEST PROCEDURE SPECIFICATION	ERROR! BOOKMARK NOT DEFINED.
5.5.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
5.5.2 <i>Outline</i>	<i>Error! Bookmark not defined.</i>
5.6. TEST ITEM TRANSMITTAL REPORT	ERROR! BOOKMARK NOT DEFINED.
5.6.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
5.6.2. <i>Outline</i>	<i>Error! Bookmark not defined.</i>
5.7. TEST LOG	ERROR! BOOKMARK NOT DEFINED.
5.7.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
5.7.2. <i>Outline</i>	<i>Error! Bookmark not defined.</i>
5.8. TEST INCIDENT REPORT	ERROR! BOOKMARK NOT DEFINED.
5.8.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
5.8.2. <i>Outline</i>	<i>Error! Bookmark not defined.</i>
5.9. TEST SUMMARY REPORT.....	ERROR! BOOKMARK NOT DEFINED.
5.9.1. <i>Purpose</i>	<i>Error! Bookmark not defined.</i>
5.9.2. <i>Outline</i>	<i>Error! Bookmark not defined.</i>

APPENDIXES:.....ERROR! BOOKMARK NOT DEFINED.

APPENDIX 1: FINAL DOCUMENTATION FORMAT GUIDELINESERROR! BOOKMARK NOT DEFINED.

Chapter 1

Project Feasibility Report

1.1. Introduction:

The existing of “CharitAble for UOG” is handled manually. In the CharitAble for UOG we are providing the charity to the UOG students in this we have three entities which are donor, volunteer and admin. In this donation system donor and volunteer can contact directly to each other for donations.

In this admin approve the volunteer by checking the profile or information which is provided by the volunteer and admin can also check the donor profile and donations history. The CharitAble for UOG is for those students who are seeking for help and for those students who wants to help needy students.

In this system students can help and directly in other words donor may be a student or may be a faculty member can help needy student without any interfere of faculty authorities. The term which is used for needy student that is volunteer

1.2. Project/Product Feasibility Report:

The most basic purpose of that feasibility study to represent the main problem and then we also determine the solution to related problem after completing the feasibility study of that project. It revealed the level/scope, limitations and basic importance, or need of that system. It also represents concern to undertake that project or, not and all level of scheduling, defining, project planning must fulfill here. This study provides complete description also then the feasibility report will provide the following basic contents:

- Technical
- Operational
- Economic
- Schedule
- Specification
- Information
- Motivational
- Legal and Ethical

1.2.1. Technical Feasibility:

The main consideration is to be given the study of the available resources of the organization where the software is to be implemented. Here the system analyst evaluates the technical merits of the system given emphasis on the performance, reliability, maintainability, and productivity.

By taking the consideration before developing the proposed system, the resources availability of the organization was immense computer facilities equipped with sophisticated machine and the software hence this technically feasible.

There are also some core tools and technologies that will be a part of this project.

Language / Technology	Tools	Reason	Hand Side
HTML	Adobe Dreamweaver Or Apache NetBeans	Development of Front end	Front Hand / User Interface
CSS	Adobe Dreamweaver Or Apache NetBeans	Designed User Interface for Iteration.	Front Hand / User Interface
JavaScript for Validation	Adobe Dreamweaver Or Apache NetBeans	Use to validate data which is entered by User	User Validation
PHP	Adobe Dreamweaver Or Apache NetBeans	Back Hand Code to get Data from User and Store in Database	Back Hand Side
MySQL	Apache SQL XAMPP Server	Use as Local Database Server to Store Records	Back Hand Side use to Store Data in Database

1.2.2. Operational Feasibility

operational feasibility contains the concept that the specific team member that completes the project must expert enough and skilled. It is also feasible that the project is able to provide the solution for underlying problem. This Charity system will remove the concept of manual work which means in this donor and volunteer can directly communicate with each another, and must provide responsive and will give efficient performance in its nature. Problem can be removed accurately.

1.2.3. Economic Feasibility:

Economic feasibility is the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better result along with increased efficiency. Cost benefit analysis is usually performed for this purpose. It is the comparative study of the cost verses the benefit and savings that are expected from the proposed system. Since the organization is well equipped with the required hardware was found to be economically.

This system will provide economic benefits to its users, basic two types of it are as follow:

1) Cost Estimates:

This cost estimation provides the total cost related to whole project

Acquisition Cost= laptop =30,000

Maintenance and Operation Cost= 40,000

2) Benefit Estimates:

Benefits estimation provides detail of tangible and in tangible benefits of that project.

➤ Tangible Benefits:

It calculates the monetary detail of the project; benefits are as written:

- It will improve the process of charity.
- There is no Cost for the services
- It will be efficient in time saving term and provide the fast Charity system

➤ Intangible Benefits:

These tangible benefits not contain any monetary term, this project con the following intangible benefits:

- It increases the responsibilities.
- Accurate access on the data,
- Decision making is very efficient and time saving.
- Prefer direct communication of donor and volunteer
- Better response on volunteer request.
- Data entry possible with in suitable time.
- It avoids the miss-utilization of the available resources/services.

1.2.4. Schedule Feasibility:

It is very important that this project must complete in the earliest time. In order to fulfill that condition. We will use the “critical path method” term. This method will divide the relative task, will define the activities and relationship between these activities also dependencies. And finding on critical path that indicate the earliest start and finish time to our project task. This “CPM” most first method to complete the whole project.

1.2.5. Specification Feasibility:

This is a management system, according to this project user needs and requirements are basic of this project we will involve user requirements specification documentation and also an analysis on these user requirements. It will indicate that what type of functionality is the part of this project. It will provide whole detail of functional and Non-functional specifications.

1.2.6. Information Feasibility:

In order to complete the whole project, this system must divide in different related task and various important activities and there would time period for the completion of these activities. It will represent project reliability in all conditions. And then if it will perform all activities and functionality in accurate manners and promote to automate all these activities this will be perfect in projects nature.

1.2.7. Motivational Feasibility:

Evaluation of the client staff regarding the motivation to perform the necessary steps correctly and promptly must occur.

1.2.8. Legal & Ethical Feasibility:

This project totally contains the legal and ethical feasibility term everything of this project is legal and reliable for use, tools that will be used and provided by the university in order to finalize that project are accurate and legal in nature. Then we prefer the use of “Microsoft Visio” and online tools on cruelty.

1.3. Project/Product Scope:

This system is only for the UOG students in the outsider students or those students who are not the UOG students cannot get any help the CharitAble only work in UOG for the UOG students. Students who want to help those students who are financially poor can help easily. The student who wants to get financial help for study can ask for help on the website where he/she can get financial help by a Donor.

1.4. Project/Product Costing:

1.4.1. Project Cost Estimation by Function Point Analysis:

Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. Since ‘functionality’ cannot be measured directly, it must be derived indirectly using other direct measures. Function-oriented metrics were first proposed by Albrecht, who suggested a measure called the function point. Function points are derived using an empirical relationship based on countable (direct) measures of software’s information domain and assessments of software complexity.

Function Point Analysis can provide a mechanism to track and monitor scope creep. Function Point counts at the end of requirements; analysis, design, code, testing and implementation can be compared. The function point count at the end of requirements and/or designs can be compared to function points delivered. If the project has grown, there has been scope creep. The amount of growth is an indication of how well requirements were gathered by and/or communicated to the project team. If the amount of growth of projects declines over time it is a natural assumption that communication with the user has improved.

Function points are computed by completing the table shown in the figure below. Five information domain characteristics are determined and counts are provided in the appropriate table location.

Parameter	Low	Medium	High
User Inputs	3	4	6
User Output	4	5	7
User Inquiries	3	4	6
Internal files	7	10	15
External interface	5	7	10

➤ **Information domain values are defined in the following manner:**

Number of user inputs: 12

Number of user outputs: 3

Number of user inquiries: 3

Number of files: 4

Number of external interfaces: 5

➤ **The formulae are given as follows:**

Cost / FP = labor rate / productivity parameter

Total Project Cost = FP est. * (cost / FP)

Total Estimated Effort = FP est. / productivity parameter

➤ **To Find UFC:**

No. of user inputs = 3(low) + 3 (medium) + 6 (high)

No. of user inquiries = 2(med) + 1(high)

No. of user outputs = 3 (high)

No. of internal files = 2 (med)

No. of external interface files = 0

No. of unadjusted functional points (UFP) = $(3*3) + (3*4) + (6*6) + (2*4) + (1*6) + (3*7) + (2*10) = 112$

Technical COMPLEXITY FACTOR:

SR. NO	FACTORS	DEGREE
1	Data communication	2
2	Distributed data processing	2
3	Performance criteria	3
4	Heavily utilized hardware	4
5	Fast Processing	3
6	Online Data Entry	3
7	Online Updating	1
8	End user efficiency	2
9	Complex computations	1
10	Reusability	3
11	Ease of installation	2
12	Ease of operation	3
13	Probability	2
14	Maintainability	2

TABLE 2:**To Find Technical COMPLEXITY FACTOR:****➤ Technical Complexity Factor (TCF):**

Degree of influence = 33

$$TCF = 0.65 + 0.01 * DI$$

$$TCF = 0.65 + 0.01 * 33$$

$$= 0.98$$

➤ FUNCTION POINTS:

$$\text{Function points (FP)} = \text{UFP} * (\text{TCF})$$

$$= 112 * (0.97)$$

$$= 108.64$$

1.4.2. Project Cost Estimation by using COCOMO'81 (Constructive Cost Model):

The Constructive Cost Model (COCOMO) is a procedural software cost estimation model developed by Barry W. Boehm. The model parameters are derived from fitting a regression formula using data from historical projects.

COCOMO 81 model estimate for the project cost by nominal effort involving set of multipliers and development time for that project by selecting its mode.

For project mode selection, we have to look for table that contains certain features, and all three modes that are organic, semi-detached and embedded. Mode will be selected according to scaling done for these features. These estimation for mode selection is given.

Feature	Organic	Semi detached	Embedded
Organizational understanding of product objectives	Thorough	Considerable	General
Experience in working with related software systems	Extensive	Considerable	Moderate
Need for software conformance with pre-established requirements	Basic	Considerable	Full
Need for software conformance with external interface specifications.	Basic	Considerable	Full
Concurrent development of associated new hardware and operational procedures	Some	Moderate	Extensive
Need for innovative data processing architectures, algorithms	Minimal	Some	Considerable
Premium on early completion	Low	Medium	High
Product Size Range	<50 KDSI	<300 KDSI	All sizes

Organic mode:

$$\text{Nominal Effort} = 3.2 (\text{KDSI})^{1.05} \cdot M$$

$$\text{TDEV} = 2.5 (\text{Effort})^{0.38}$$

We will do product's rating to determine set of effort multipliers on a set of 15 cost driver attributes.

Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
Required Software Reliability	0.75	0.88	1.00	1.15	1.40	-
Database Size Product Complexity	-	0.94	1.00	1.08	1.16	-
	.70	0.85	1.00	1.15	1.30	1.65
Execution time constraints	-	-	1.00	1.11	1.30	1.66
Storage Constraints	-	-	1.00	1.06	1.21	1.56
Virtual Machine Volatility	-	0.87	1.00	1.15	1.30	-
Computer Turnaround Time	-	0.87	1.00	1.07	1.15	-
Analyst Capability	1.46	1.19	1.00	0.86	0.71	-
Application Experience	1.29	1.13	1.00	0.91	0.82	-
Programming Capability	1.42	1.17	1.00	0.86	0.70	-
Virtual machine experience	1.21	1.10	1.00	0.90	-	-
Prog. language experience	1.14	1.07	1.00	0.95	-	-
Modern programming practice	1.24	1.10	1.00	0.91	0.91	0.82
Use of software tools	1.24	1.10	1.00	0.91	0.83	0.83
Development schedule	1.23	-	1.00	1.04	1.10	1.10

Type	Effort	Schedule
Organic	PM= 2.4 (KLOC)1.05	TD= 2.5(PM)0.38
Semi-Detached	PM= 3.0 (KLOC)1.12	TD= 2.5(PM)0.35
Embedded	PM= 2.4 (KLOC)1.20	TD= 2.5(PM)0.32

PM= person-month (effort)

KLOC= lines of code, in thousands

TD= number of months estimated for software development (duration)

People Required = Effort / Duration

As our project is organic, we will use equations of organic type:

Estimated KLOC in our project is 15

KLOC=15

PM= 2.4 (KLOC)1.05

PM=2.4(15)1.05 = 41 person-month

TD= 2.5(PM)0.38 = 2.5(41.21)0.38 = 10

People Required = Effort / Duration = 41/10 = 4

1.4.3. Activity Based Costing:

Sr.	Activities	Resources	Cost Rate	Duration
1	Website Design, Layout, Structure	Bootstrap	5K	4
2	Front End	HTML, CSS, JavaScript	5K	1
3	Development of website components	Bootstrap	Free	8
4	Back End	XAMPP Distribution	5K	7

1.5. Task Dependency Table:

The following are the steps to develop a task dependency table:

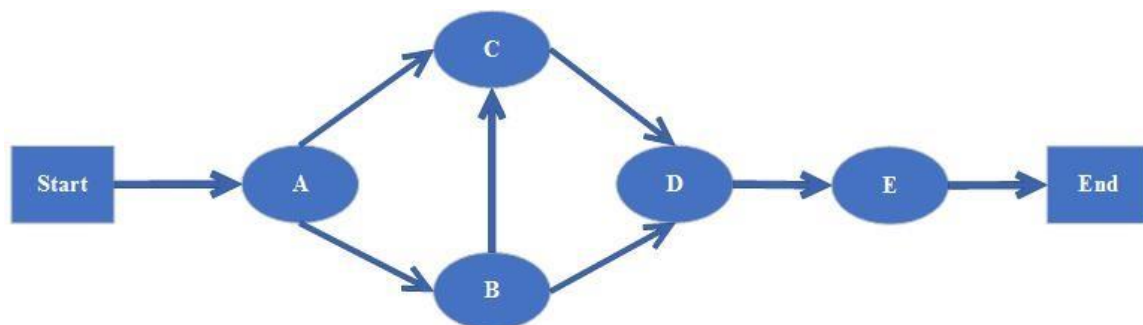
Task No#	Task Name	Dependent No#
T1	Requirement Gathering	
T2	Analysis	T1
T3	Design	T2
T4	Coding	T3
T5	Defining the backend structure	T5
T6	Server-Side Logic	T6
T7	Testing	T4, T5, T6
T8	Deployment	T4, T5, T6, T7

1.6. CPM - Critical Path Method:

1. Determine the Sequence of the Activities:

Task ID	Task Description	Duration (Days)	Dependencies
	Start	-	
A	Communication	7	-
B	Planning	7	A
C	Modeling	14	A, B
D	Construction	21	B, C
E	Deployment	30	D

2. Draw the Network Diagram:



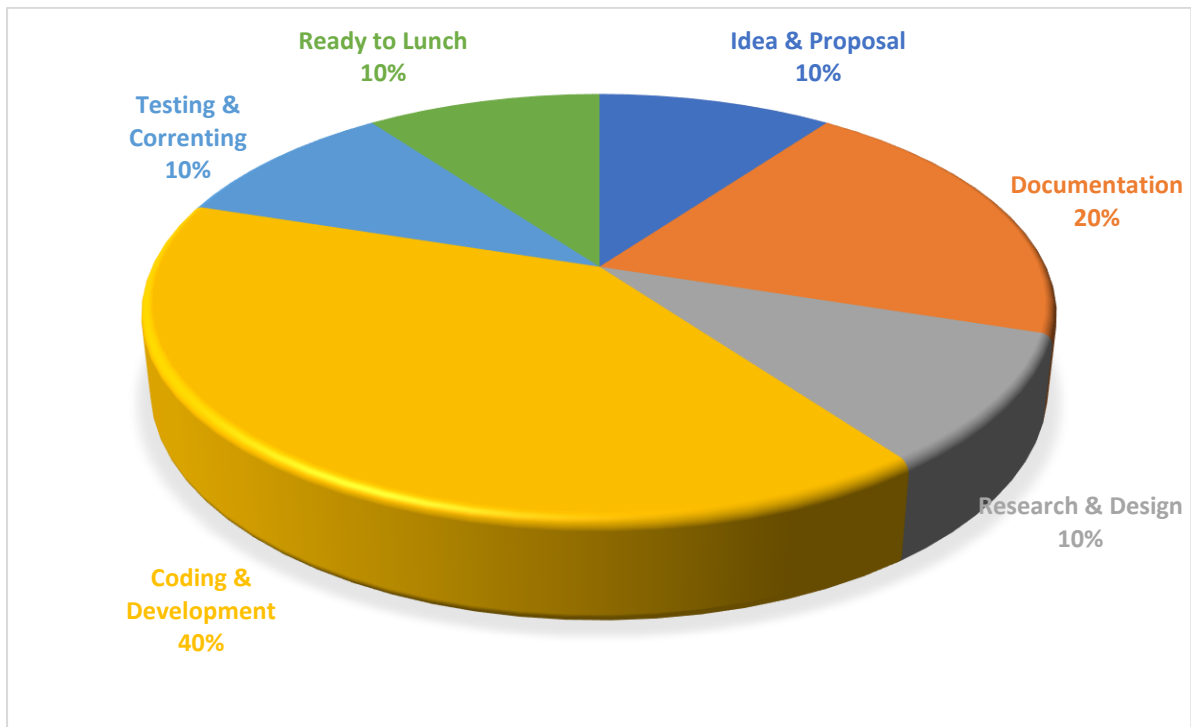
3. Estimate Activity Completion Time:

Task ID	Predecessors	Durations (Days)
A	-	15
B	A	15
C	A, B	30
D	B, C	40
E	D	20

4. Identify the Critical Path

Activity	Duration	ES	EF	LS	LF	TS	FS
A	15	1	16	1	16	0	0
B	15	16	31	16	31	0	0
C	30	16	46	1	31	15	15
D	40	31	71	31	71	0	0
E	20	71	91	71	91	0	

1.7. Gantt Chart:



1.8. Introduction to Team member and their skill set:

The project team consists of 3 members. Each member is technical skilled and perform every task with very efficient way. Every team member performs their task in time due to their technical expertise.

1.9. Task and Member Assignment Table:

Serial No#	Activity	Duration	Deliverable
1	Idea Discussion	7	Proposal Document
2	Requirements Analysis	7	SRS
3	Prototyping	7	Wireframes and UI
4	Design Phase	21	UML Diagrams
5	Implementation	14	Final Product
6	Testing against test cases	15	Testing Document

1.10. Tools and Technology with reasoning:

The tool that is used for front-end Sublime that support web development work. We use query language in SQL server. HTML language is used for web page designing.

- The HTML is used for frontend of our website.
- The CSS is used for styling of the website
- The JavaScript is used scripting
- The development model is used in our project is Prototype model.
- The language that we used in our project is PHP for server-side scripting.
- Platform that we used in our project is PHP.
- The framework that we used in our project is PHP.
- Database that we used in our project is SQL, Xampp.

We use these tools and technology because Sublime support this framework. Moreover, PHP language is very powerful language that also support for developing any type of web application.

1.11. Vision Document:

Provide a modern-day charity platform based on web-based application use to provides about Charity which named as CharitAble for UOG.

Users:

- Admin
- Donor
- Volunteer

Key factor to judge the Quality:

- System is automated and provide solution of problems that occurred in manual auction
- Each user has facility to interact with his area of working.
- Users can register to avail Donation services.
- Implement modern technology.
- Users are notified about updates by notifications on the CharitAble portal.

Financial:

Maintenance & Design Level Risks:

- Health factor of staff (illness issues).
- Effects of changes in requirements of project.
- Effects of restructuring in organization.
- Estimation: The development team might not be able to estimate the work time, preventing customers from deciding priorities effectively. The rate of defects repair is underestimated.
- Reusability: Reusability of modules can affect the functionality of the project.
- Acceptance: Market may accept web platform, if it will provide real time problem solution
- Scope: The total features requested may be beyond what the development team can deliver in the time available.

Software Level Risk:

- Immature technology with rapid changes in standard.
- No internet availability in most northern areas.

Technology Level Risk:

- The project will fail without web browser, internet and desktop.

Development Level Risks:

- Finance: There may be credit risk; the question is whether they have sufficient resources to manage the construction and operation of the project and to efficiently resolve any problems which may arise. To minimize this risk human resources are important (Limitation and resources).
- Time: Limited time available to complete this whole project as this project will.

Chapter 2

Software Requirement Specification

Phase I

It includes the following areas:

- Sign up
- Verification

Phase II

It includes all the remaining aspects which are not included in previous phases. It is the complete and comprehensive application of Online Career Guidance.

2.2.4. Summary of Requirements:(Initial Requirements):

The purposed system must fulfill following requirements as follow:

- Donor Dashboard
- Select Volunteer & Donate
- Donation History
- Profile Updating
- Volunteer Dashboard
- Generate Help Request
- Verification of Received Donation
- Donation History
- Profile Updating
- Administrator Dashboard
- Add New Member
- Check Existing Member
- Check Donor List
- Check Volunteer List
- Check Donations
- Filtered Donation Report (Week, Month, Year)
- Search Specific (Users, Donations)
- Complain Box
- Profile

2.2.5. Identifying External Entities:

The Identification of External Interfaces is done in two phases.

2.2.5.1. Over Specify Entities from Abstract

- Authentication
- View and manage record

2.2.5.2 Perform Refinement:

Now we have refined these entities furthermore. Following entities are related to our project

- Login System
- Generate requests
- Make donations
- Check lists
- Update record

2.2.6. Capture "shall" Statements:

Para #	Initial Requirements
1.0	A volunteer “shall” signup to create or make a profile.
1.0	A donor “shall” signup to create or make a profile.
1.0	The donor and volunteer “shall” login when they have signup once.
1.0	volunteer “shall” make a request for the help
1.0	A donor “shall” make donation by using this after selecting the volunteer.
1.0	Donation make by the donor “shall” approve/accept by the volunteer.
1.0	Volunteer or donor “shall” change their passwords when they want to change it.
1.0	Admin “shall” generate the reports of donations
1.0	Admin “shall” check the complains by the donor or volunteer, and also give the response for the complaint.
1.0	Admin “shall” check the lists or donor and volunteer who are existing in the system.
2.0	Admin “shall” add a new admin when he needs for assistant/partner to divide his work.
2.0	Admin “shall” check the existing admin or assistant admin.

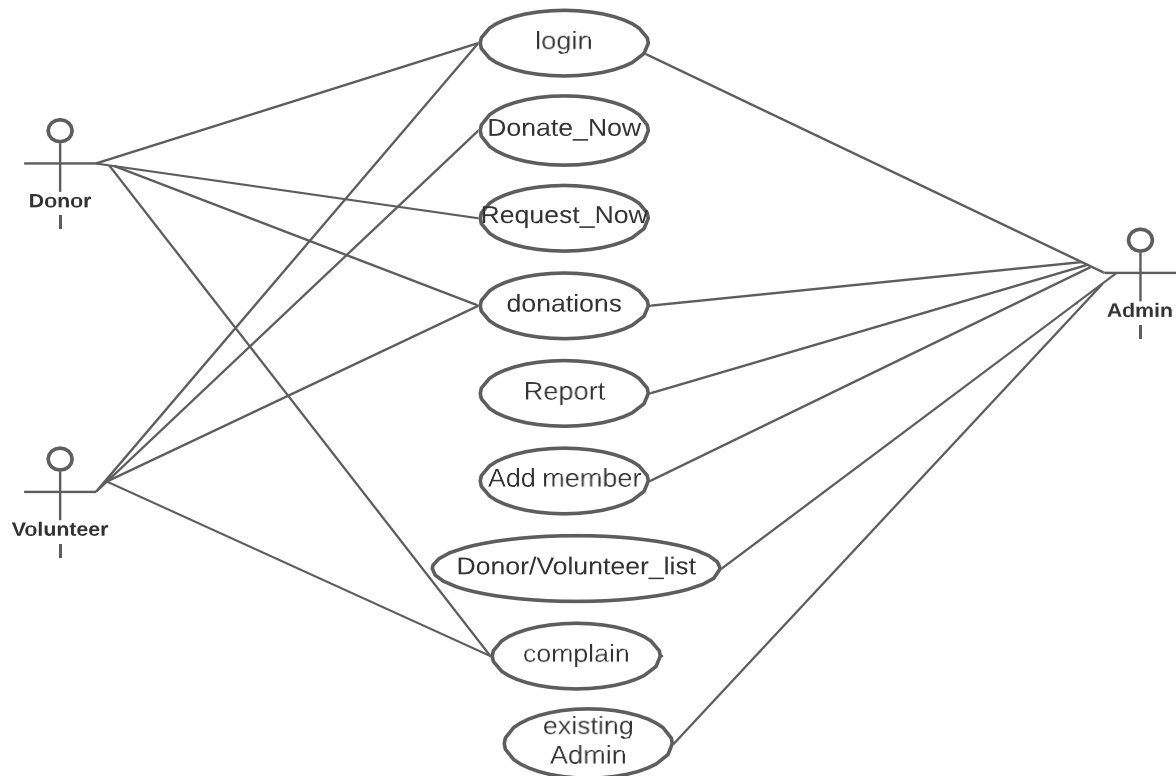
2.2.7 Allocate Requirements:

Para #	Initial Requirements	Use Case Name
1.0	A volunteer “shall” signup to create or make a profile.	UC_Volunteer_signup
1.0	A donor “shall” signup to create or make a profile.	UC_Donor_signup
1.0	The donor and volunteer “shall” login when they have signup once,	UC_login
1.0	volunteer “shall” make a request for the help	UC_Request_Now
1.0	A donor “shall” make donation by using this after selecting the volunteer.	UC_Donate_Now
	Donation make by the donor “shall” approve/accept by the volunteer.	UC_Approve_donation
1.0	Volunteer or donor “shall” change their passwords when they want to change it.	UC_Change_Password
1.0	Admin “shall” generate the reports of donations	UC_Report
1.0	Admin “shall” check the complains by the donor or volunteer, and also give the response for the complaint.	UC_Complain
1.0	Admin “shall” check the lists or donor and volunteer who are existing in the system.	UC_Volunteer/Donor_list
1.0	Admin “shall” add a new member, where member may be a new admin or volunteer.	UC_Add_New_member
2.0	Admin “shall” check the existing admin or assistant admin.	UC_Existing_Admin

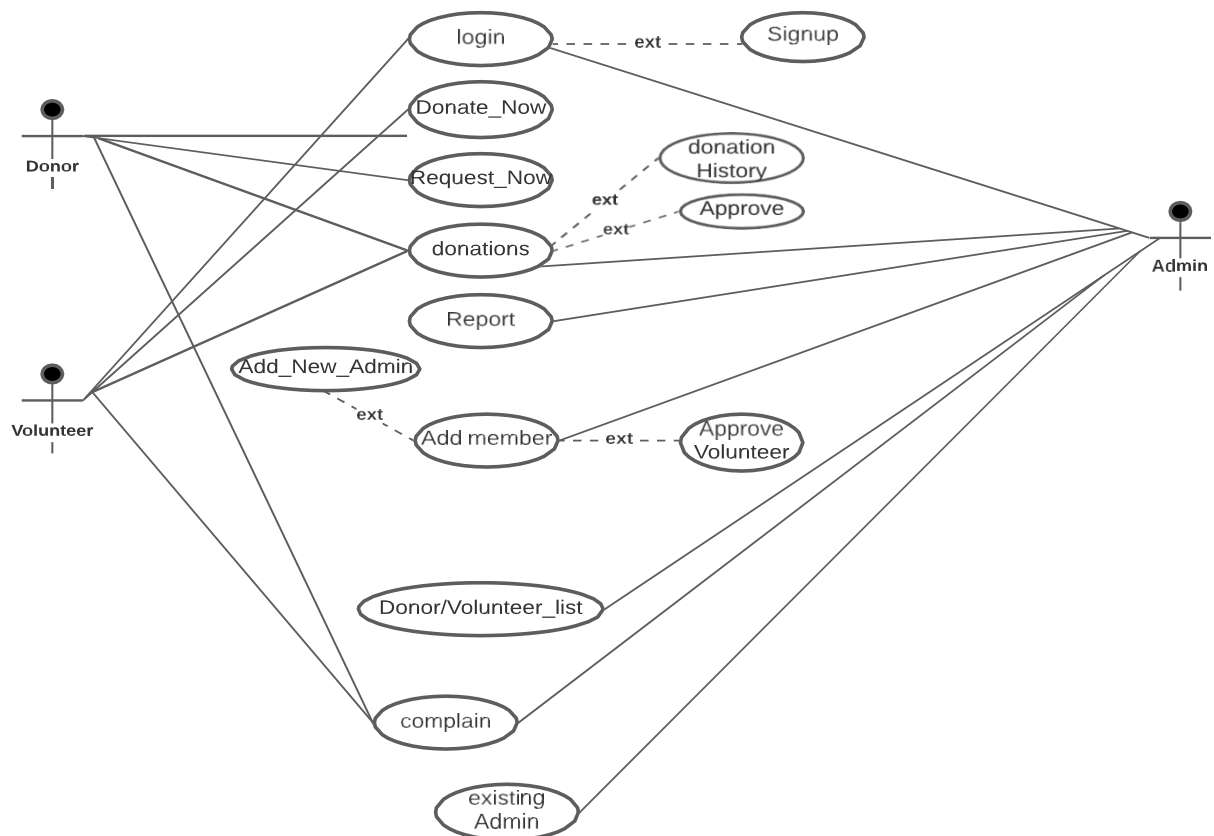
2.2.8. Priorities Requirements:

Para #	Rank	Initial Requirements	Use Case ID	Use Case Name
1.0	Highest	A volunteer “shall” signup to create or make a profile and A donor “shall” signup to create or make a profile.	UC_1	UC_Volunteer/Donor_signup
1.0	Highest	The donor and volunteer “shall” login when they have signup once.	UC_2	UC_login
1.0	Highest	volunteer “shall” make a request for the help	UC_3	UC_Request_Now
1.0	Highest	A donor “shall” make donation by using this after selecting the volunteer.	UC_4	UC_Donate_Now
1.0	Highest	Donation make by the donor “shall” approve/accept by the volunteer.	UC_5	UC_Approve_donation
2.0	Medium	Volunteer or donor “shall” change their passwords when they want to change it.	UC_6	UC_Change_Password
2.0	Medium	Admin “shall” generate the reports of donations	UC_7	UC_Report
2.0	Medium	Admin “shall” check the complains by the donor or volunteer, and also give the response for the complaint.	UC_8	UC_Complain
2.0	Medium	Admin “shall” check the lists or donor and volunteer who are existing in the system.	UC_9	UC_Volunteer/Donor_list
3.0	Low	Admin “shall” add a new member, where member may be a new admin or volunteer.	UC_10	UC_Add_New_member
3.0	Low	Admin “shall” check the existing admin or assistant admin.	UC_11	UC_Existing_Admin

2.2.9. High Level Use Case Diagram:



2.2.10. Analysis Level Use Case Diagram:



2.2.11. Use Case Description:

2.2.11.1 Signup:

Actor: volunteer/donor	Developer: Q-A-H
UC ID: UC_1	Date: 9-July-2021
UC Name: UC_Volunteer/Donor_signup	
Precondition: Actor not have an account.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on signup button 2. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor logs in to the website if provided data is correct.	
Alternatives: Actor should signup first if haves no account already.	

2.2.11.2 User Login:

Actor: volunteer/donor	Developer: Q-A-H
UC ID: UC_2	Date: 9-July-2021
UC Name: UC_login	
Precondition: Actor should have an account.	
Main Scenario: Actor's Action: <ol style="list-style-type: none"> 1. Actor clicks on login button 2. Actor provides required information System Response: <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: actor logs in to the website if provided data is correct.	
Alternatives: actor should signup first if haves no account already.	

2.2.11.3 Generate Request:

Actor: volunteer	Developer: Q-A-H
UC ID: UC_3	Date: 9-July-2021
UC Name: UC_Request_Now	
Precondition: Actor login in the system first.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on Request Now 2. Actor fill the form to make request. 3. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor can make request when actor fill the form correctly.	
Alternatives: Actor should login first.	

2.2.11.4 Make Donation:

Actor: Donor	Developer: Q-A-H
UC ID: UC_4	Date: 9-July-2021
UC Name: UC_Donate_Now	
Precondition: Actor login in the system first.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on Donate Now 2. Actor select the volunteer. 3. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor can make donation when requests are generated and actor choose the volunteer.	
Alternatives: Actor should login first.	

2.2.11.5 Approve Donation:

Actor: volunteer	Developer: Q-A-H
UC ID: UC_5	Date: 9-July-2021
UC Name: UC_Approve_donation	
Precondition: Actor should have received donation offer by donor.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on approve button to complete donation 2. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor convert the donations pending status into complete status.	
Alternatives: Actor should generate request.	

2.2.11.6 Change Password

Actor: volunteer/admin/donor	Developer: Q-A-H
UC ID: UC_6	Date: 9-July-2021
UC Name: UC_Change_Password	
Precondition: Actor should have login.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on change password. 2. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor secure the account.	
Alternatives: Actor should signup.	

2.2.11.7 Reports:

Actor: Admin	Developer: Q-A-H
UC ID: UC_7	Date: 9-July-2021
UC Name: UC_Report	
Precondition: Actor should have login.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on report button to generate report. 2. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor have record report.	
Alternatives: Actor should signup.	

2.2.11.8 Complain:

Actor: Admin	Developer: Q-A-H
UC ID: UC_8	Date: 9-July-2021
UC Name: UC_complain	
Precondition: Actor should have login.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on complain button to check complains or to make response. 2. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor make response to complaints.	
Alternatives: Actor have no complain.	

2.2.11.9 Volunteer / Donor List:

Actor: Admin	Developer: Q-A-H
UC ID: UC_9	Date: 9-July-2021
UC Name: UC_Volunteer / Donor List	
Precondition: Actor should have login.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on Volunteer/Donor_list. 2. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor have volunteers and donors list.	
Alternatives: Actor have no Volunteer / Donor.	

2.2.11.10 Add New member:

Actor: Admin	Developer: Q-A-H
UC ID: UC_10	Date: 9-July-2021
UC Name: UC_Add_New_member	
Precondition: Actor should have login.	
Main Scenario: Actor's Action <ol style="list-style-type: none"> 1. Actor clicks on Add New member button. 2. Actor provides required information System Response <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor added new volunteer/admin.	
Alternatives: Actor must find a admin.	

2.2.11.11 Existing Admin:

Actor: Admin	Developer: Q-A-H
UC ID: UC_11	Date: 9-July-2021
UC Name: UC_Existing_Admin	
Precondition: Actor should have login.	
<p>Main Scenario:</p> <p>Actor's Action</p> <ol style="list-style-type: none"> 1. Actor clicks on Existing admin. 2. Actor provides required information <p>System Response</p> <ol style="list-style-type: none"> 1. System asks for required information. 2. System shows search result. 	
Post Condition: Actor have name of persons who are existing in system with admin authority.	
Alternatives: Actor must add new admin.	

Chapter 3

Design Diagrams Software Requirement

3.1. Introduction:

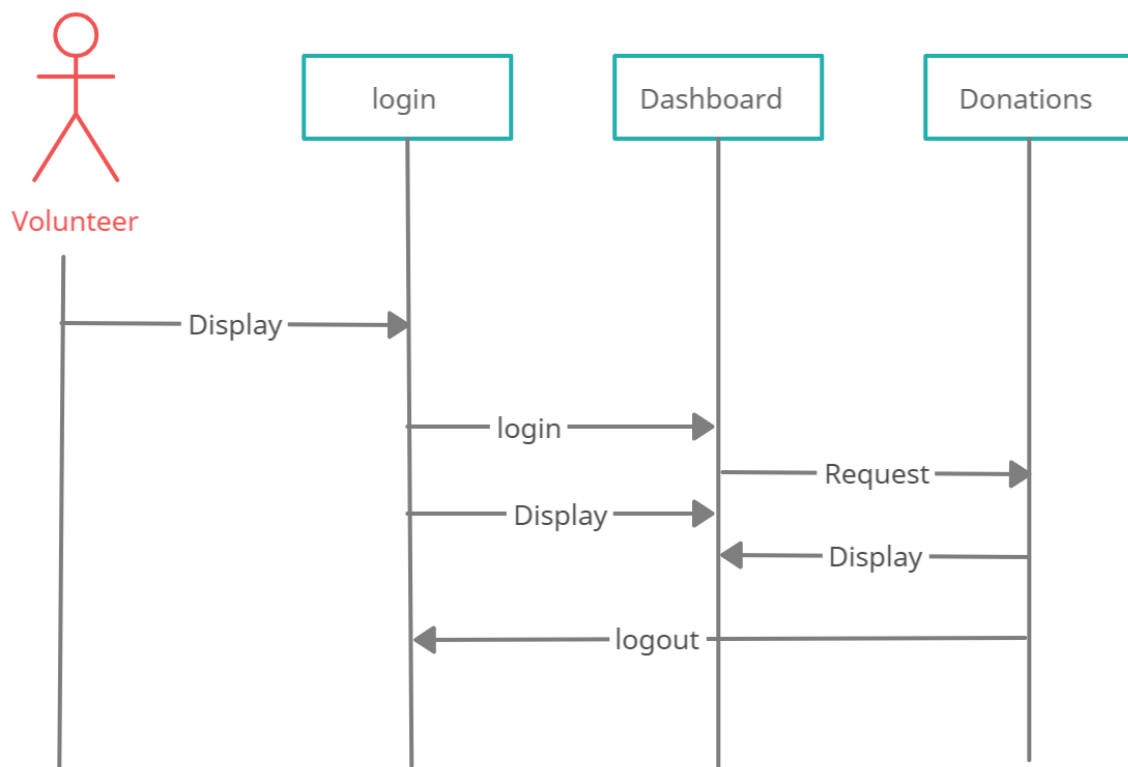
Third deliverable is all about the software design. In the previous deliverable, analysis of the system is completed. So, we understand the current situation of the problem domain. Now we are ready to strive for a solution for the problem domain by using object-oriented approach. Following artifacts must be included in the 3rd deliverable.

1. Sequence Diagram
2. Defining a sequence diagram
3. Design Class Diagram
4. State Transition Diagram
5. Data Model

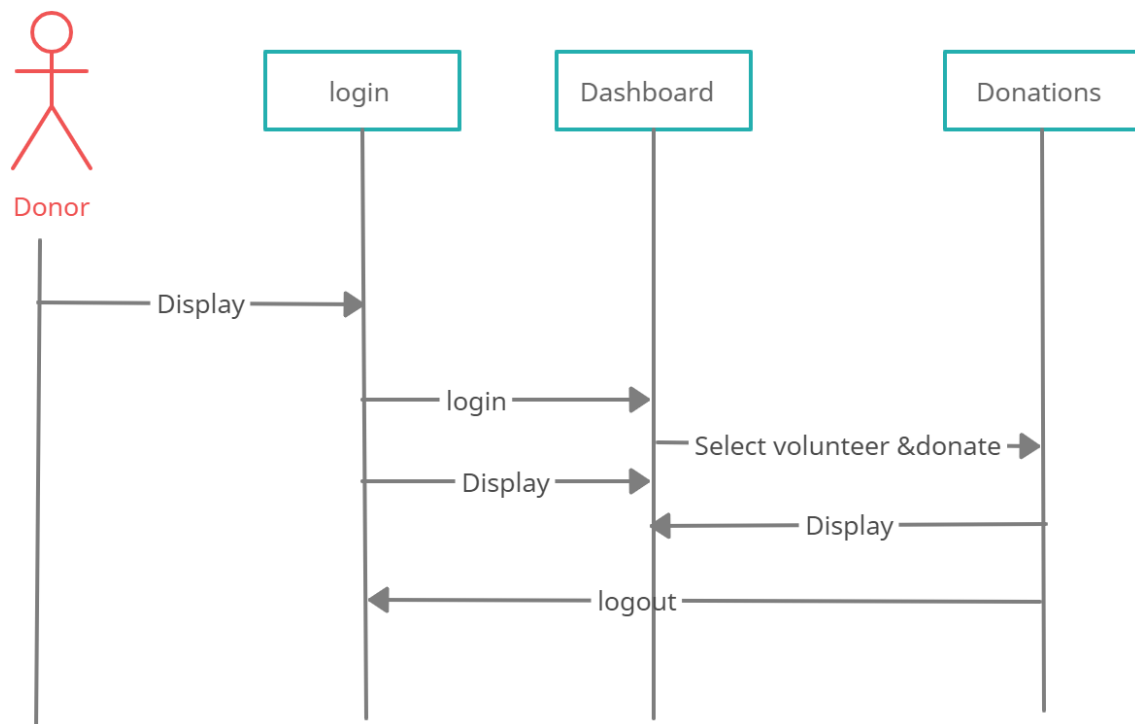
Now we discuss these artifacts one by one as follows:

3.2. Sequence Diagram:

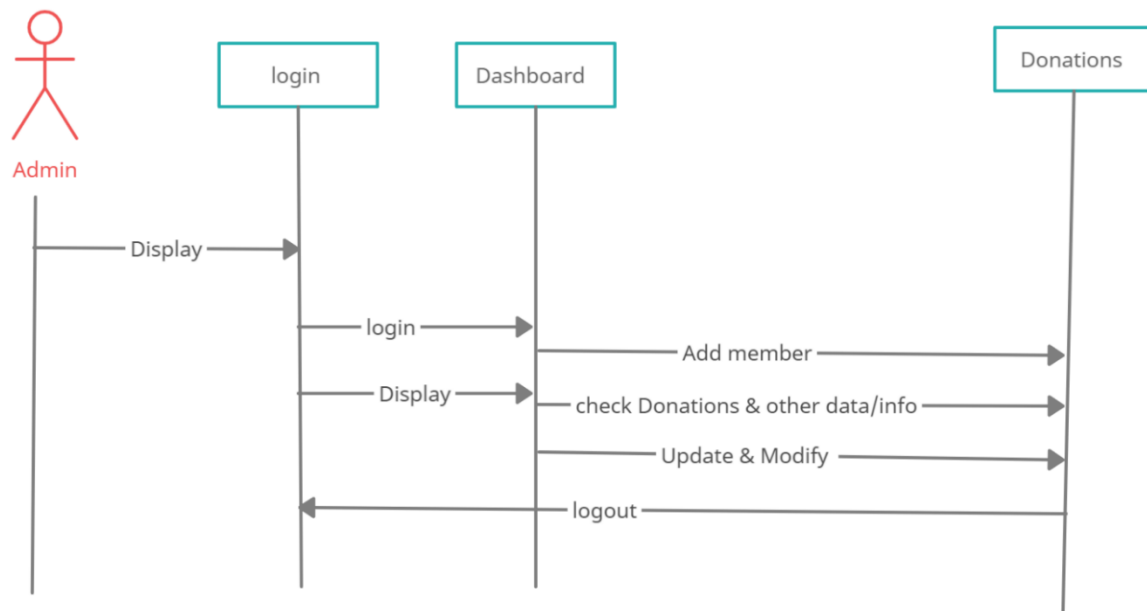
- Sequence diagrams for volunteer



➤ Sequence diagrams For Donor



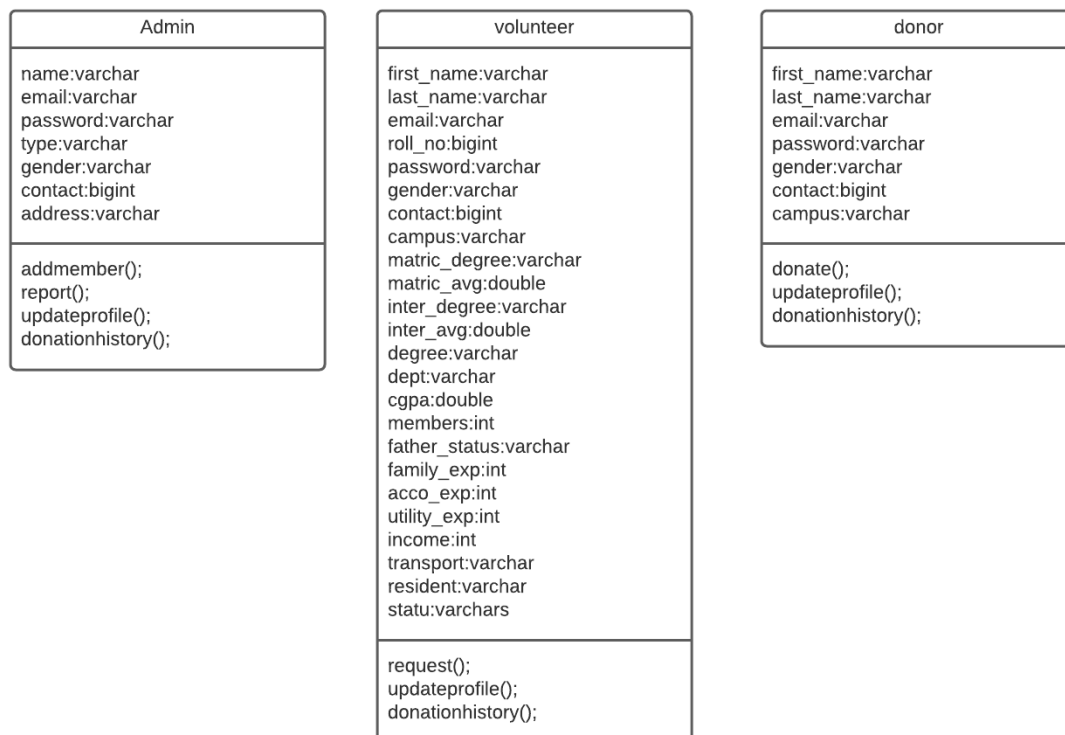
➤ Sequence diagrams For Admin



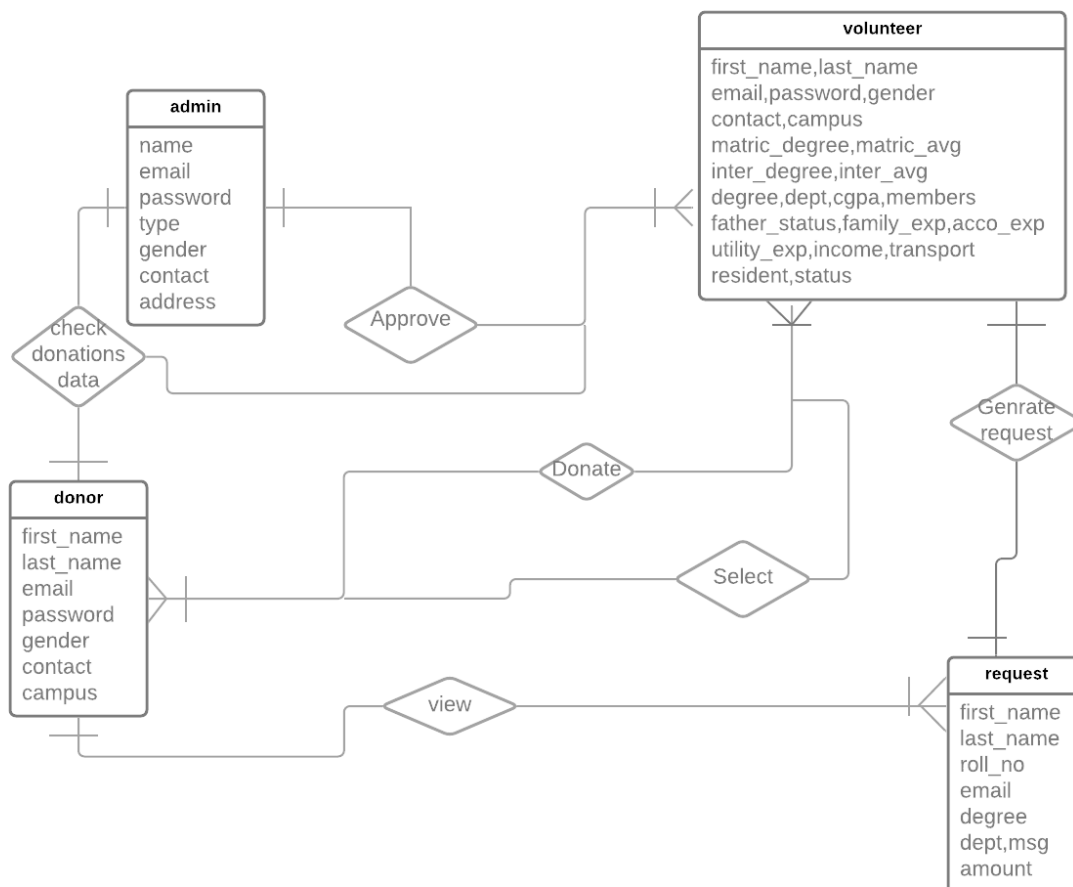
3.3. Defining a Sequence diagram:

- Sequence diagram for Admin: shows that first admin login with his/her account then the admin can perform the activities such as admin can add new member mean admin approve the volunteer accounts, admin can check data about donations or donor or volunteer
- Sequence diagram for donor: show that donor login first with his/her account then donor can check the request of multiple volunteers then donor select a volunteer and make a donation to the volunteer.
- Sequence diagram for volunteer: this shows that to generate a request for help first volunteer login with the account then volunteer can generate a request for help and can approve the donation and can check other info like profile notifications etc.

3.4. Design Class Diagram:



3.5. Data Model:

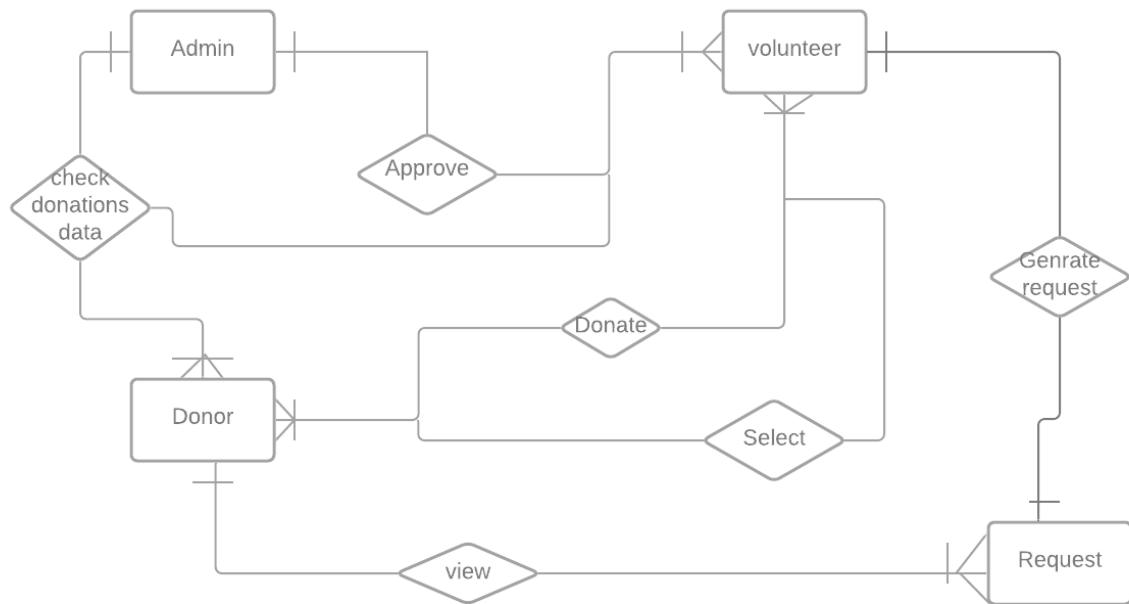


3.6 Introduction:

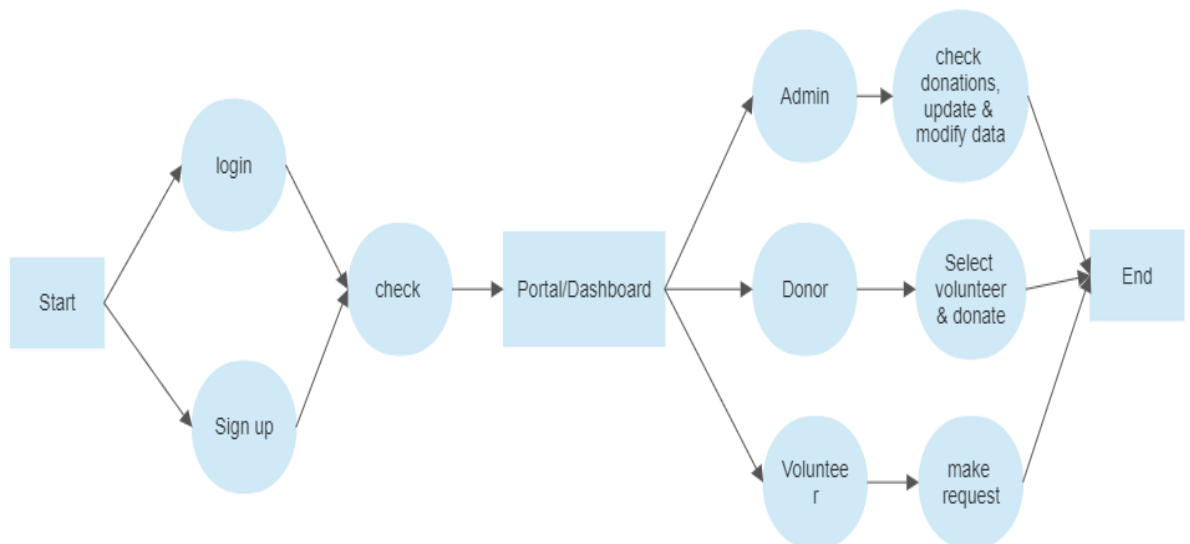
Analysis & Design Model for structured approach must contain following artifacts:

1. Entity Relationship Diagram
2. Data Flow Diagram (Functional Model)
3. State Transition Diagram (Behavioral Model)
4. Architecture Design
5. Component Level Design

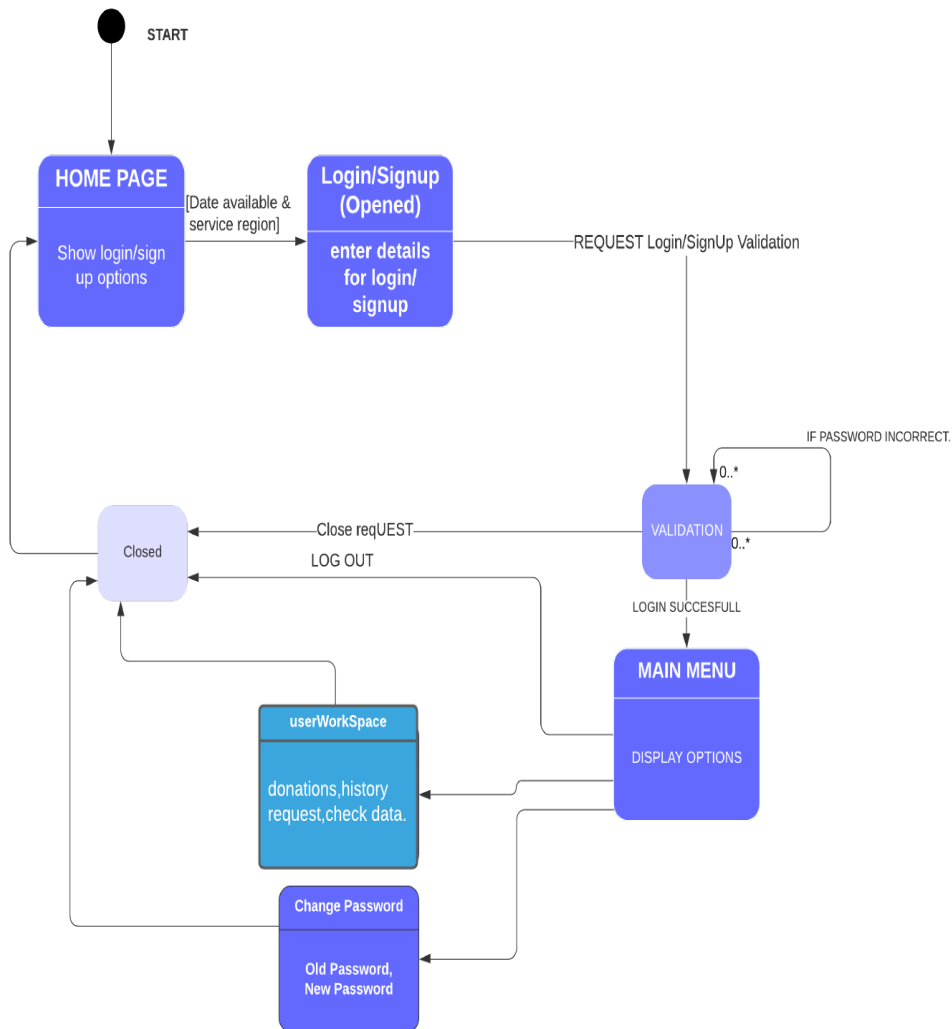
3.7 Entity Relationship Diagram:



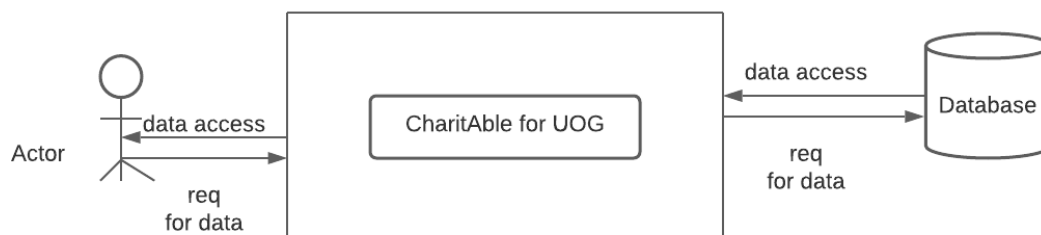
3.8 Data flow diagram (Functional Model)



3.9 State Transition Diagram:

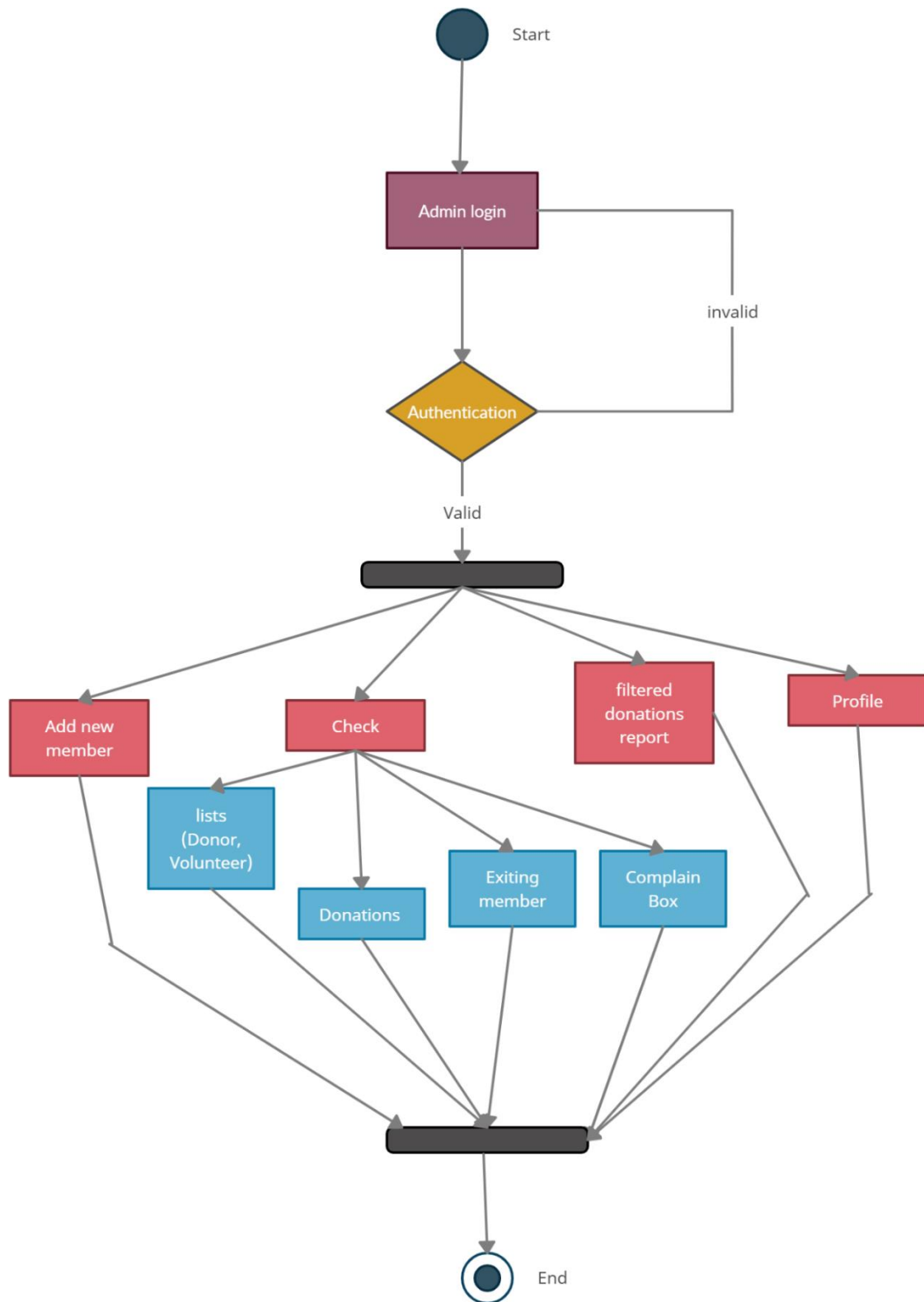


3.10 Architectural design:

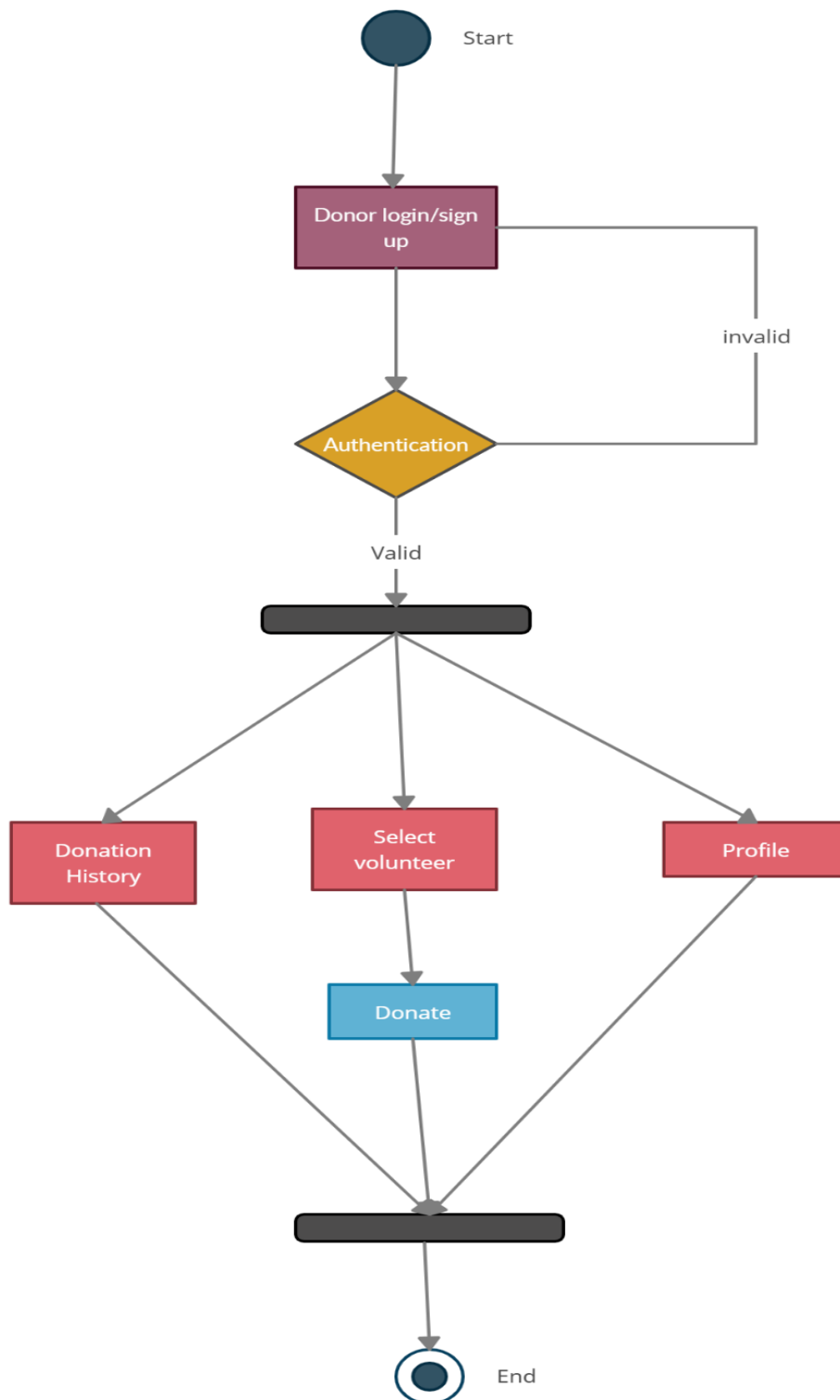


3.11 Component Level Design

➤ Activity Diagram for Admin



➤ Activity Diagram for Donor



Chapter 4

User Interface Design & S.S

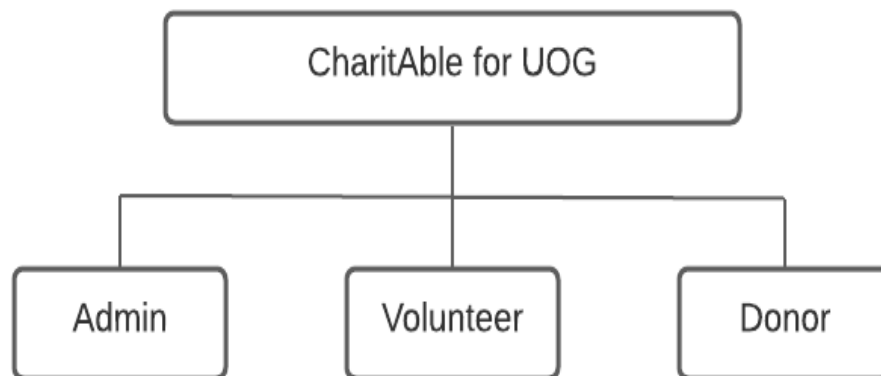
4.1. Introduction:

A user interface design consists of three main parts:

Page elements should be visualized on paper before building them in the computer. Just as you draw a site map to plan the site, use cartoons and storyboards to begin blocking out the site's appearance and navigational scheme.

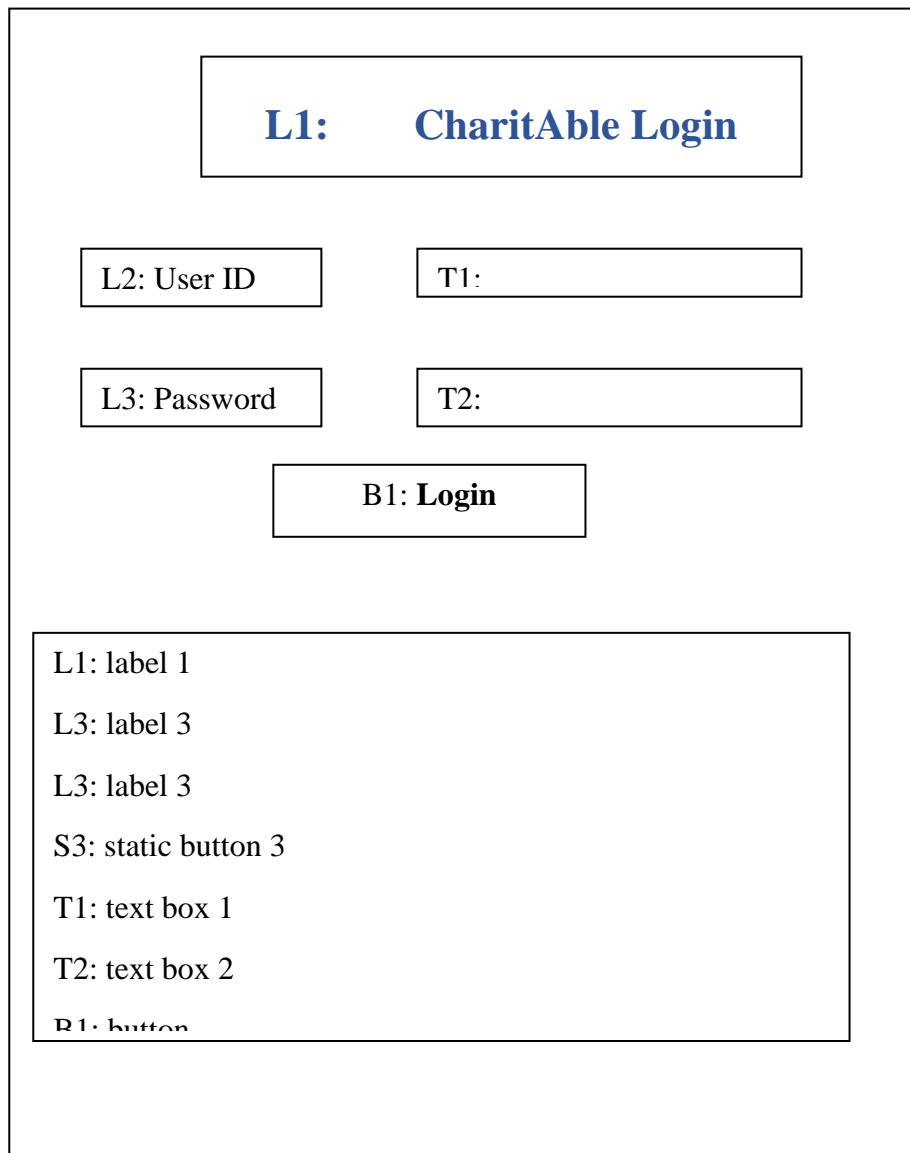
1. Site maps
2. Storyboards
3. Navigational maps
4. Traceability Matrix

4.2. Site Maps:



4.3. Story boards:

A storyboard is a sequence of single images, each of which represents a distinct event or narrative. It is also a visual representation of the script illustrating the interaction between the user and the machine. It can also be imagined as a film in visual-outline form.



4.5 Trace-ability Matrix:

Following columns are involved in the trace-ability matrix.

Trace-ability Matrix 1:

Features	Actor logins/signup the system.
Use Case ID:	UC_01,02
Priority:	Highest
Use Case Cross Ref:	Registration
DB Table Id:	Nil
Elaborated Use-case ID:	UC_01,02
Dependent Classes:	Record Management

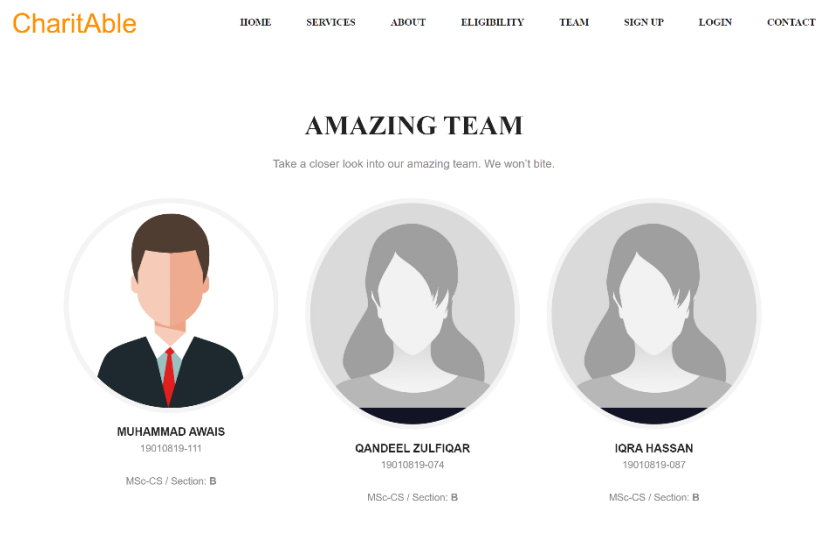
Trace-ability Matrix 2:

Features	Actor's donations, requests, reports, complains; Password changing in system.
Use Case ID:	UC_3,4,5,6,7,8,9
Priority:	Medium
Use Case Cross Ref:	Record Management
DB Table Id:	Nil
Elaborated Use-case ID:	UC_3,4,5,6,7,8,9
Dependent Classes:	Login, logout

Home Screen:



Team Page:



Donor Signup Page:

Notice: Undefined index: submit in C:\xampp\htdocs\Charitable\Signup Volunteer.php on line 10

PERSONAL INFORMATION

WE JUST NEED SOME PERSONAL INFORMATION

First Name *

Last Name *

UOG Email *

UOG Roll No# *

Password *

Gender *

Gender

Contact No# *

Volunteer Signup Page:

Notice: Undefined index: submit in C:\xampp\htdocs\Charitable\Signup Volunteer.php on line 10

Registration Form

Fill all form field to go next step

1

2

3

4

5

USER INFORMATION MATRIC RECORD INTERMEDIATE RECORD CURRENT DEGREE FAMILY DETAILS

MATRIC DETAILS

INCLUDE YOUR MATRIC RECORD IN ALL REQUIRED FIELD.

Matric *

Matric

Total Marks *

Obtained Marks *

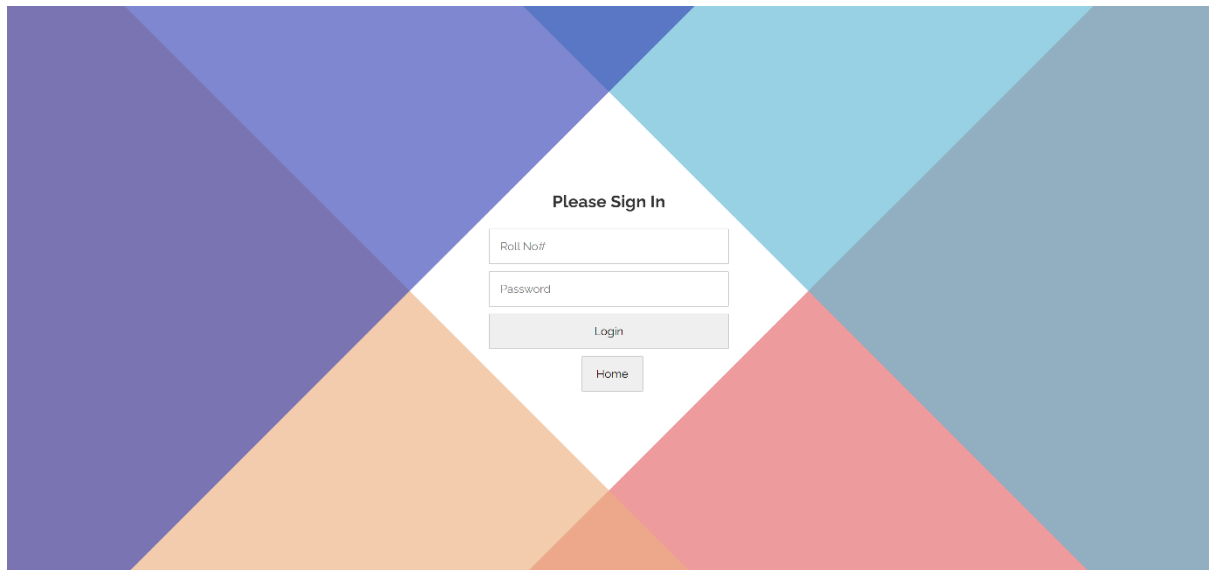
Average Percentage

0%

Previous

Next

Login Page:

A login page mockup with a background of overlapping triangles in shades of purple, blue, orange, and red. In the center, there is a white box containing the text "Please Sign In". Below this text are three input fields: "Roll No/ID", "Password", and a "Login" button. At the bottom of the white box is a "Home" button.

Please Sign In

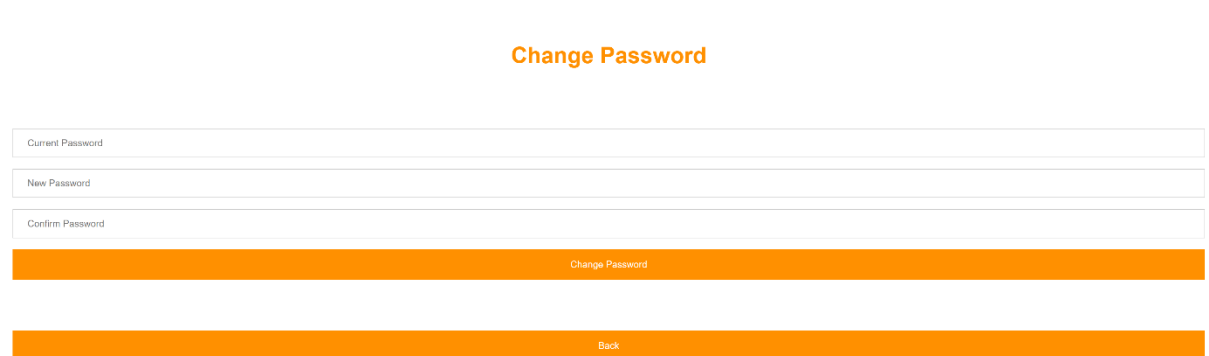
Roll No/ID

Password

Login

Home

Change Password Page:

A change password page mockup. At the top, the text "Change Password" is displayed in orange. Below this are three input fields: "Current Password", "New Password", and "Confirm Password". At the bottom, there are two orange buttons: "Change Password" and "Back".

Change Password

Current Password

New Password

Confirm Password

Change Password

Back

Donor Dashboard Page:



CharitAble

DONATE NOW PENDING COMPLETED PROFILE CHANGE PASSWORD LOGOUT

DONOR

DASHBOARD

19010819111

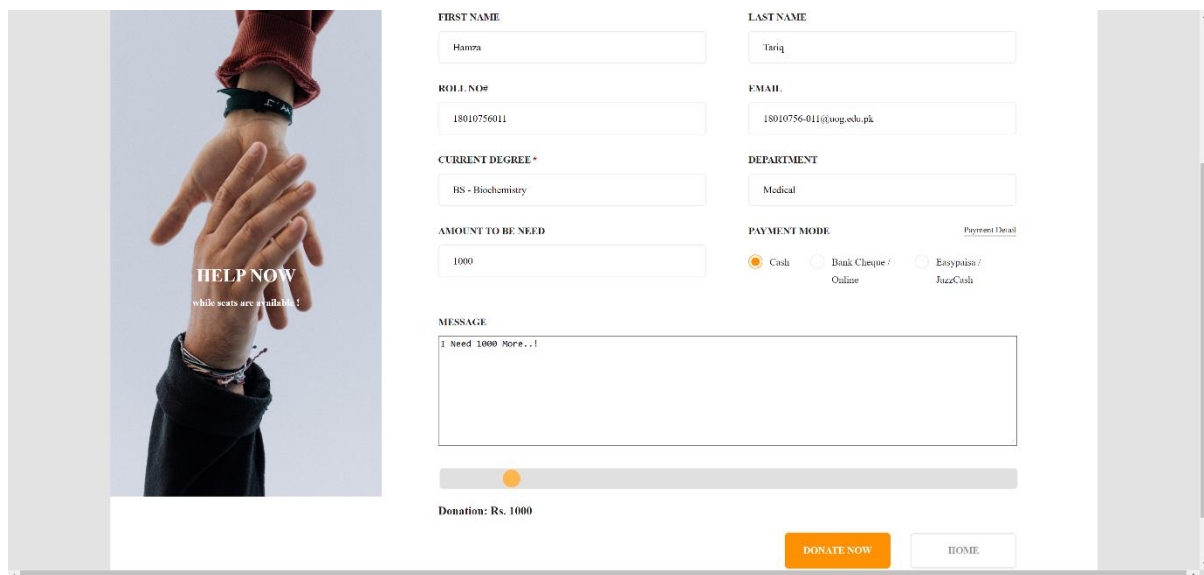
MUHAMMAD AWAIS

Student of GT Road - City Campus
19010819-111@uog.edu.pk

DONATE NOW

CharitAble
FOR U-O-G

Donation Form Page:



HELP NOW
while seats are available!

FIRST NAME
Hamza

LAST NAME
Tariq

ROLL NO#
18010756011

EMAIL
18010756-011@uog.edu.pk

CURRENT DEGREE *
BS - Biochemistry

DEPARTMENT
Medical

AMOUNT TO BE NEEDED
1000

PAYMENT MODE
☒ Cash
 ☐ Bank Cheque / Online
 ☐ Easypaisa / JazzCash

MESSAGE
I need 1000 More...!

Donation: Rs. 1000

DONATE NOW HOME

Pending Donations Page:

CharitAble

DONATE NOW PENDING COMPLETED PROFILE CHANGE PASSWORD LOGOUT

PENDING TRANSECTIONS

NO OF TRANSECTIONS: 3

ID	VOLUNTEER NAME	ROLL NO#	E-MAIL	AMOUNT	STATUS	ACTION
93562068	Qandeel Zulfqar	19010819076	19010819-076@uog.edu.pk	2000	Pending from Volunteer	Check
1223387716	Ali Hassan	19010819121	19010819-121@uog.edu.pk	3000	Pending from Volunteer	Check
1677546070	Khalid Jamshad	17010856048	17010856-048@uog.edu.pk	500	Pending from Volunteer	Check

Completed Donations Page:

CharitAble

DONATE NOW PENDING COMPLETED PROFILE CHANGE PASSWORD LOGOUT

COMPLETED TRANSECTIONS

NO OF TRANSECTIONS: 4

ID	VOLUNTEER NAME	ROLL NO#	E-MAIL	AMOUNT	STATUS	ACTION	
299663410	Hamza Tariq	18010756011	18010756-011@uog.edu.pk	3000	Completed from Volunteer	Check	Delete
332489614	Ali Hassan	19010819121	19010819-121@uog.edu.pk	1500	Completed from Volunteer	Check	Delete
436058862	Qandeel Zulfqar	19010819076	19010819-076@uog.edu.pk	1000	Completed from Volunteer	Check	Delete

Donor Profile Page:

CharitAble

[DONATE NOW](#) [PENDING](#) [COMPLETED](#) [PROFILE](#) [CHANGE PASSWORD](#) [LOGOUT](#)

Donor Profile

PERSONAL INFORMATION

WE JUST NEED SOME PERSONAL INFORMATION

First Name *

Last Name *

UOG Email *

UOG Roll No# *

Password *

Volunteer Dashboard Page:

CharitAble

[REQUEST NOW](#) [NOTIFICATIONS](#) [COMPLETED](#) [PROFILE](#) [CHANGE PASSWORD](#) [LOGOUT](#)

VOLUNTEER

DASHBOARD

17010856048

KHALID JAMSHAD

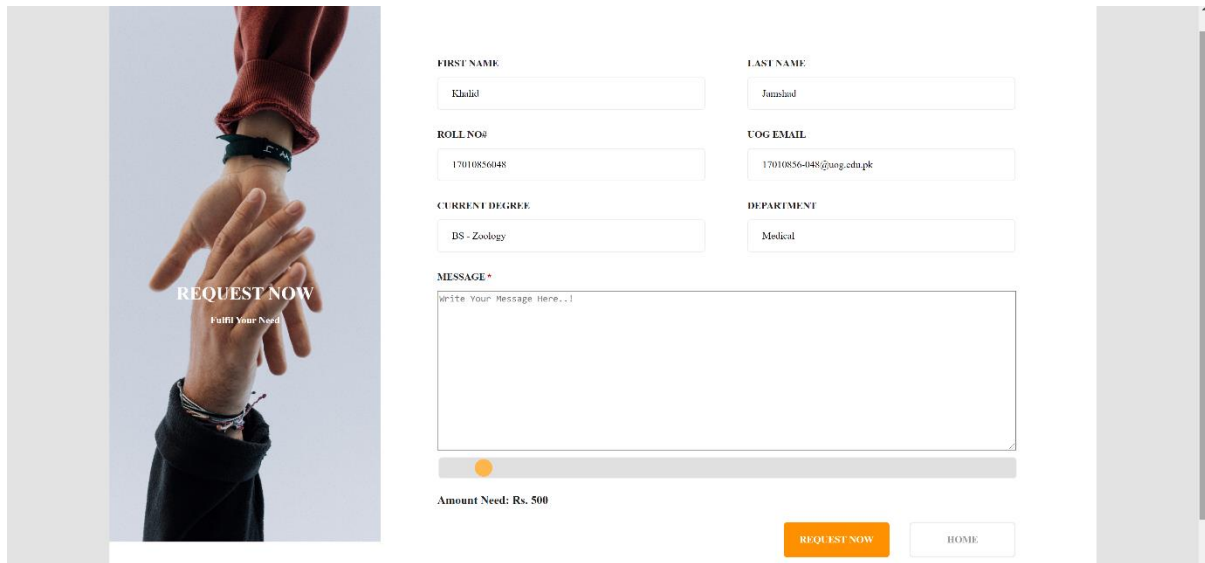
BS - Zoology
Hafiz Hayat Campus
17010856-048@uog.edu.pk
Registered User

[REQUEST NOW](#)



CharitAble
FOR U-O-G

Request Form Page:

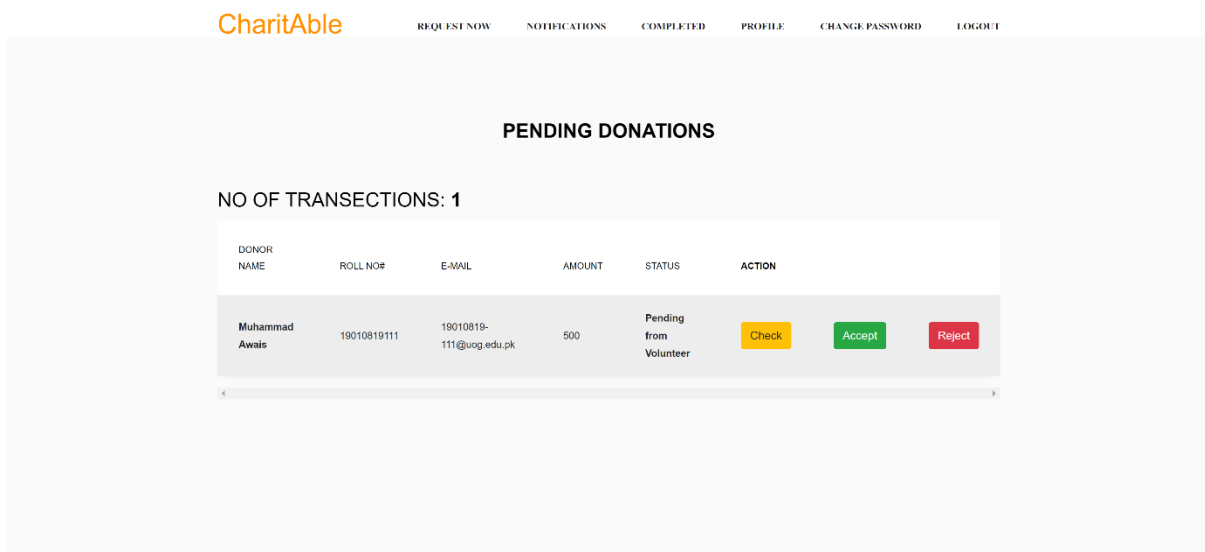


The Request Form Page features a hero image on the left showing two hands clasped together, with the text "REQUEST NOW" and "Fulfill Your Need" overlaid. The form on the right includes the following fields:

- FIRST NAME:** Input field containing "Khalid".
- LAST NAME:** Input field containing "Jamsilad".
- ROLL NO#:** Input field containing "17010856048".
- UOG EMAIL:** Input field containing "17010856-048@uog.edu.pk".
- CURRENT DEGREE:** Input field containing "BS - Zoology".
- DEPARTMENT:** Input field containing "Medical".
- MESSAGE*:** A large text area with the placeholder "Write Your Message Here...".

Below the message field, a progress bar shows a single orange dot. The text "Amount Need: Rs. 500" is displayed. At the bottom right, there are two buttons: "REQUEST NOW" (orange) and "HOME" (white).

Notification Page:



The Notification Page has a header with the "CharitAble" logo and navigation links: REQUEST NOW, NOTIFICATIONS, COMPLETED, PROFILE, CHANGE PASSWORD, and LOGOUT. The main content area is titled "PENDING DONATIONS" and shows "NO OF TRANSECTIONS: 1".

DONOR NAME	ROLL NO#	E-MAIL	AMOUNT	STATUS	ACTION
Muhammad Awaiz	19010819111	19010819-111@uog.edu.pk	500	Pending from Volunteer	Check Accept Reject

Completed Donations Page:

CharitAble

REQUEST NOW NOTIFICATIONS COMPLETED PROFILE CHANGE PASSWORD LOGOUT

COMPLETED TRANSECTIONS

NO OF TRANSECTIONS: 1

ID	VOLUNTEER NAME	ROLL NO#	E-MAIL	AMOUNT	STATUS	ACTION
1053469449	Muhammad Awas	19010819111	19010819-111@uog.edu.pk	1500	Completed from Volunteer	Check

Volunteer Profile Page:

Add New Administrator

Fill all form field to go next step

Notice: Undefined index: submit in C:\xampp\htdocs\CharitAble\Admin\Admin Signup.php

PERSONAL INFORMATION

WE JUST NEED SOME PERSONAL INFORMATION

Type Fixed

Administrator

Full Name *

Email *

Password *

Gender *

Gender

Contact No# *

Admin Dashboard Page:



CharitAble

ADMIN APPROVALS USERS DONATIONS REPORTS COMPLAIN BOX LOGOUT

ADMIN

DASHBOARD

ADMINISTRATOR

MUHAMMAD AWAIS

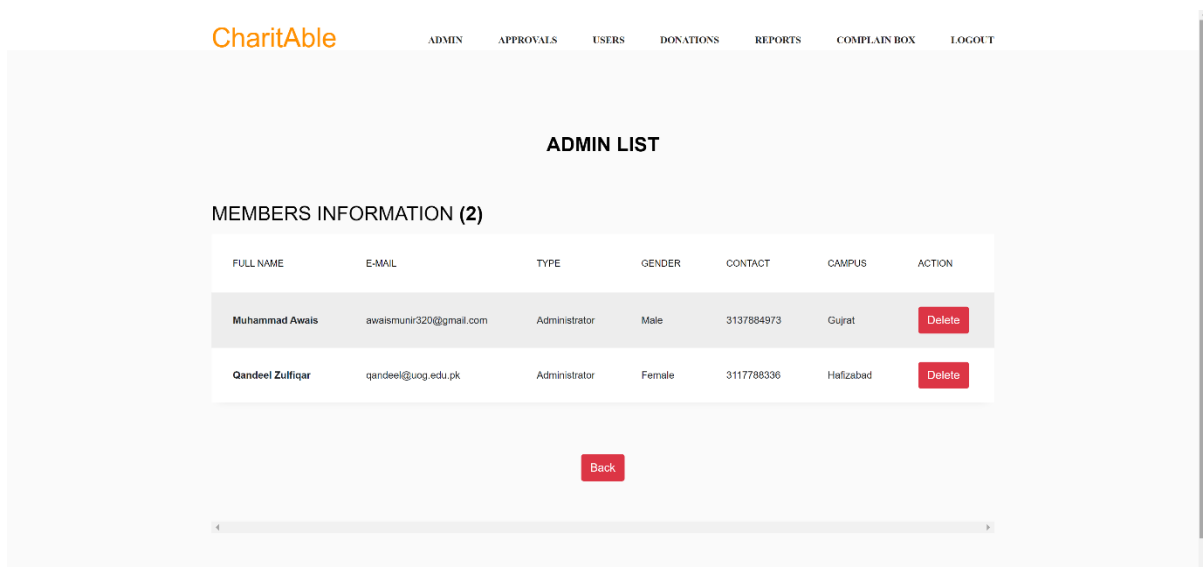
Administrator
awaismunir320@gmail.com

PROFILE



CharitAble
FOR U-O-G

Existing Admin Page:



CharitAble

ADMIN APPROVALS USERS DONATIONS REPORTS COMPLAIN BOX LOGOUT

ADMIN LIST

MEMBERS INFORMATION (2)

FULL NAME	E-MAIL	TYPE	GENDER	CONTACT	CAMPUS	ACTION
Muhammad Awais	awaismunir320@gmail.com	Administrator	Male	3137884973	Gujrat	Delete
Qandeel Zulfiqar	qandeel@uog.edu.pk	Administrator	Female	3117788336	Hafizabad	Delete

Back

Add New Admin Page:

Add New Administrator
Fill all form field to go next step

Notice: Undefined index: submit in C:\xampp\htdocs\CharitAble\Admin\Admin Signup.php

PERSONAL INFORMATION
WE JUST NEED SOME PERSONAL INFORMATION

Type Fixed

Full Name *

Email *

Password *

Gender *

Contact No# *

New Volunteer Approval Page:

CharitAble

ADMIN APPROVALS USERS DONATIONS REPORTS COMPLAIN BOX LOGOUT

NEW VOLUNTEER APPROVALS

FULL NAME	ROLL NO#	DEGREE	CAMPUS	STATUS	ACTION		
Shameen Bisharat	17020544045	BS - Software Engineering	Margazar Campus	Un-Registered	Check	Approve	Reject

Volunteer Users Page:

CharitAble						
ADMIN APPROVALS USERS DONATIONS REPORTS COMPLAIN BOX LOGOUT						
VOLUNTEER LIST						
TOTAL VOLUNTEERS (4)						
FULL NAME	ROLL NO#	E-MAIL	CONTACT	CAMPUS	STATUS	ACTION
Khalid Jamshad	17010856048	17010856-048@uog.edu.pk	3006243007	Hafiz Hayat Campus	Registered	Check Delete
Shameen Bisharat	17020544045	17020544-045@uog.edu.pk	3437782617	Margazar Campus	Un-Registered	Check Delete
Hamza Tariq	18010756011	18010756-011@uog.edu.pk	3006243007	Hafiz Hayat Campus	Registered	Check Delete
Ali Hassan	19010819121	19010819-121@uog.edu.pk	3337282511	Hafiz Hayat Campus	Registered	Check Delete

Donor Users Page:

CharitAble						
ADMIN APPROVALS USERS DONATIONS REPORTS COMPLAIN BOX LOGOUT						
DONOR LIST						
TOTAL DONORS (2)						
FULL NAME	ROLL NO#	E-MAIL	GENDER	CONTACT	CAMPUS	ACTION
Qandeel Zufrigar	19010819074	19010819-074@uog.edu.pk	Female	3211234567	Hafiz Hayat Campus	Delete
Muhammad Awais	19010819111	19010819-111@uog.edu.pk	Male	3137884973	GT Road - City Campus	Delete
Back						

Donations List Page:

DONATIONS LIST						
COMPLETED DONATIONS						
TRSN. ID	VALUNTEER FULL NAME	ROLL NO#	DONOR FULL NAME	ROLL NO#	STATUS	ACTION
299663410	Hamza Tariq	18010756011	Muhammad Awais	19010819111	Completed from Volunteer	Check
332489614	Ali Hassan	19010819121	Muhammad Awais	19010819111	Completed from Volunteer	Check
436058862	Gandeel Zulfiqar	19010819076	Muhammad Awais	19010819111	Completed from Volunteer	Check
1053469449	Khalid Jamshad	17010856048	Muhammad Awais	19010819111	Completed from Volunteer	Check
PENDING DONATIONS						

Reports Generating Page:

CharitAble
ADMIN
APPROVALS
USERS
DONATIONS
REPORTS
COMPLAIN BOX
LOGOUT

REPORT DATES

REPORT FROM	REPORT TO	ACTION
<input type="text" value="mm/dd/yyyy"/>	<input type="text" value="mm/dd/yyyy"/>	Report

Back

Complain Box Page:

CharitAble

ADMINAPPROVALSUSERSDONATIONSMETHODSREPORTS

COMPLAIN BOXLOGOUT

COMPLAIN BOX

TOTAL COMPLAINS (3)

COMPLAIN ID	FULL NAME	E-MAIL	SUBJECT	MESSAGE	ACTION
471382799	Hakim	19010856-002@uog.edu.pk	Dome 2	afdsfd	<div>ReplyDelete</div>
995036271	Muhammad Awais	19010819-111@uog.edu.pk	Dome	Hii..!	<div>ReplyDelete</div>
1514463703	Ali Hassan	19010819-121@uog.edu.pk	Error	sdfthbsnfm	<div>ReplyDelete</div>

Back

Chapter 5

Software Testing

5.1 Introduction:

This deliverable is based on the IEEE standard of software testing i.e., IEEE SOFTWARE TEST DOCUMENTATION Std 829-1998. This standard describes a set of basic test documents that are associated with the dynamic aspects of software testing (i.e., the execution of procedures and code). The standard defines the purpose, outline, and content of each basic document. While the documents described in the standard focus on dynamic testing, several of them may be applicable to other testing activities (e.g., the test plan and test incident report may be used for design and code reviews). This standard may be applied to commercial, scientific, or military software that runs on any digital computer. Applicability is not restricted by the size, complexity, or criticality of the software. However, the standard does not specify any class of software to which it must be applied. The standard addresses the documentation of both initial development testing and the testing of subsequent software releases. For a particular software release, it may be applied to all phases of testing from module testing through user acceptance. However, since all of the basic test documents may not be useful in each test phase, the particular documents to be used in a phase are not specified. Each organization using the standard will need to specify the classes of software to which it applies and the specific documents required for a particular test phase.

The standard does not call for specific testing methodologies, approaches, techniques, facilities, or tools, and does not specify the documentation of their use. Additional test documentation may be required (e.g., code inspection checklists and reports). The standard also does not imply or impose specific methodologies for documentation control, configuration management, or quality assurance. Additional documentation (e.g., a quality assurance plan) may be needed depending on the particular methodologies used.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

5.2. Test Plan:

5.2.1. Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.2.2. Outline:

A test plan shall have the following structure:

1. Test plan identifier
2. Introduction
3. Test items
4. Features to be tested
5. Features not to be tested
6. Approach
7. Item pass/fail criteria
8. Suspension criteria and resumption requirements
9. Test deliverables
10. Testing tasks
11. Environmental needs
12. Responsibilities
13. Staffing and training needs
14. Schedule
15. Risks and contingencies
16. Approvals

The sections shall be ordered in the specified sequence. Additional sections may be included immediately prior to Approvals. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test plan or available to users of the plan.

Details on the content of each section are contained in the following sub-clauses.

5.2.2.1. Test Plan Identifier:

Specify the unique identifier assigned to this test plan.

5.2.2.2. Introduction:

Summarize the software items and software features to be tested. The need for each item and its history may be included. References to the following documents, when they exist, are required in the highest-level test plan:

1. Project authorization;
2. Project plan;
3. Quality assurance plan;
4. Configuration management plan;
5. Relevant policies;
6. Relevant standards.

In multilevel test plans, each lower-level plan must reference the next higher-level plan.

5.2.2.3. Test items:

The items to be include e.g., Contact/About us, Login, Logout, Change Password, Successful Signup, Account Approval by Admin., Generate Report, Request Now, Donations History, Donate Now, Complain Box, Lists update, Update profile

5.2.2.4. Features to be Tested:

The following list describes the features to be tested:

1. Volunteer / Donor signup
2. Login
3. Request Now
4. Donate Now
5. Approve Donation
6. Change Password
7. Report
8. Complain
9. Volunteer / Donor List
10. Add new member
11. Existing admin

5.2.2.5. Features not to be Tested:

- System Performance

This is non-functional requirement. Performance of system under heavy load could not be tested due to lack of time and resources.

- System goes down

This is non-functional feature. It cannot be tested as; we cannot stimulate an actual server crash and try to recover it.

5.2.2.6. Approach:

Structured approach is used:

1. Entity relationship diagram
2. Data flow diagram
3. State transition diagram
4. Architectural design
5. Component level design

5.2.2.7. Item Pass / Fail Criteria:

All core functionality of the systems should function as expected and outlined in the individual test cases. The test cases executed on the Web based CharitAble Application pass if they meet the specific requirements mentioned of the project. A test case is said to fail if the desired functionality is not satisfied by the system.

Item Pass Criteria:

- Any Operating System which supports Browser.
- Ram 4GB
- Minimum Intel 4GB Graphics Card (Slow Processing)
- (Optional) NVIDIA 4 GB Graphics Card (For Good performance)
- Valid Inputs and Fulfill Define Conditions.
- Good Internet Connection.
- Item Fail Criteria:
- Using an operating system which is too old or not support Browser
- Not Ram 4GB
- Not Valid Inputs and Not Fulfill Define condition
- Less than 4GB Graphic Card.

5.2.2.8. Suspension Criteria and Resumption Requirements:

Testing work is performed by the individual to ensure that all the functional activities are working well in the system properly. If some of these functions are not working properly in the system, the further test procedure should not be continued to the next level.

So here are some criteria for which we will pause the test work for the application.

- If a test case fails, then testing for all the dependent features will be suspended. The failed test case will be logged onto the test log which contains the description for the error.
- If system will be to handle lots of users at once.
- The testing will be resumed once the identified failure or issue is resolved.

5.2.2.9. Test Deliverables:

If the CharitAble Donations and request generating functions are successfully working in the web app then it will be considered pass otherwise it will be failed.

1. Test plan
2. Test design specifications
3. Test case specifications
4. Test logs
5. Test summary reports

5.2.2.10. Testing Tasks:

1. Authorization Module testing
2. Login testing
3. Registration testing
4. Performance testing
5. Account Approval by Admin testing
6. Changing Password testing
7. Contact to Admin Module testing.

5.2.2.11. Environmental Needs:

The environment requirements are as follow:

Hardware:

- Laptop and Android mobile.
- 8GB-100 GB Hard 1-4 GB Ram.
- Core i3 and later for better performance.
- Internet connectivity.

Server-side Hardware:

- Core i5 and later for better performance
- 16-32 GB RAM
- 2 - 8TB Hard

Software:

- Windows 8 OS and Later
- Browser i.e., Google Chrome, internet explorer, Mozilla Firefox etc.
- Python Web Server (Linux, Windows, Unix).

5.2.2.12. Responsibilities:

Each member is responsible for requirements, designing, preparing documentation, developing and executing the system properly. The test manager is responsible for testing, managing and resolving the issues of the project

5.2.2.13 Staffing and Training Needs:

Testers should conduct testing on each system. The testers should have basic knowledge about following terms before they proceed testing:

- HTML, CSS, BOOTSTRAP, JavaScript
- Python (Django Framework)
- SQL Database

5.2.2.14. Schedule:

The testing may take about one and half week to complete. Rather Some of its components and features had been tested in a two day.

5.2.2.15. Risks and Contingencies:

If the first component testing is not completed within a day it can be delay. After bug fixes the testing will be performed. If the tester does not have the basic understanding About HTML, CSS, BOOTSTRAP, JavaScript, SQLite Database testing will be delayed.

5.2.2.16 Approvals

The team member who manages the testing process and the team leader who manages the whole product must approve this plan. The approval of supervisor is required here:

Name: Mr. Zaheer Ahmed

Title: CharitAble for U-O-G.

Signature:

Date:

5.3. Test design specification

5.3.1. Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.3.2. Outline

A test plan shall have the following structure:

1. Test plan identifier;
2. Introduction;
3. Test items;
4. Features to be tested;
5. Features not to be tested;
6. Approach;
7. Item pass/fail criteria;
8. Suspension criteria and resumption requirements;
9. Test deliverables;

10. Testing tasks;
11. Environmental needs;
12. Responsibilities;
13. Staffing and training needs;
14. Schedule;
15. Risks and contingencies;
16. Approvals.

5.3.2.1 Test Plan Identifier:

Project Test plan CharitAble for U-O-G ChA_02.

5.3.2.2. Introduction:

The main purpose of the test plan is to discuss the testing details of use cases of System. The test plan for system indicates the personnel responsible for each task and specifies the risks associated with the test plan. The goal of this document is to develop a test plan for Web-based CharitAble for U-O-G.

This document defines all procedures and activities required to prepare for testing of functions of the system which are specified in Vision document. The objectives of the test plan are to define the activities to perform testing, define test deliverable documents and to identify the various risks and contingencies involved in testing.

This test plan for Web based CharitAble for U-O-G Application testing supports the following objectives:

- To identify the features of the system that will be tested.
- To identify the features of the system that will not be tested.
- Pass and fail criteria of the testing items.
- To communicate to the responsible parties the items to be tested, set expectations around schedule, and define environmental needs.
- To define how the tests will be conducted
- In multilevel test plans, each lower-level plan must reference the next higher-level plan.

5.3.2.3. Test Items:

The items to be include e.g., Contact/About us, Login, Logout, Change Password, Successful Signup, Account Approval by Admin., Generate Report, Request Now, Donations History, Donate Now, Complain Box, Lists update, Update profile

5.3.2.4. Features to be Tested:

The following list describes the features to be tested:

1. Volunteer / Donor signup
2. Login
3. Request Now
4. Donate Now
5. Approve Donation
6. Change Password
7. Report
8. Complain
9. Volunteer / Donor List
10. Add new member
11. Existing admin

5.3.2.5. Features not to be tested

- System Performance

This is non-functional requirement. Performance of system under heavy load could not be tested due to lack of time and resources.

- System goes down

This is non-functional feature. It cannot be tested as; we cannot stimulate an actual server crash and try to recover it.

5.3.2.6. Approach:

1. Structured approach is used:
2. Entity relationship diagram
3. Data flow diagram
4. State transition diagram
5. Architectural design
6. Component level design

5.3.2.7. Item Pass / Fail Criteria:

All core functionality of the systems should function as expected and outlined in the individual test cases. The test cases executed on the Web based CharitAble Application pass if they meet the specific requirements mentioned of the project. A test case is said to fail if the desired functionality is not satisfied by the system.

Item Pass Criteria:

- Any Operating System which supports Browser.
- Ram 4GB
- Minimum Intel 4GB Graphics Card (Slow Processing)
- Valid Inputs and Fulfill Define Conditions.
- Good Internet Connection.
- Item Fail Criteria:
- Using an operating system which is too old or not support Browser
- Not Ram 4GB
- Not Valid Inputs and Not Fulfill Define condition
- Less than 4GB Graphic Card.

5.3.2.8. Suspension Criteria and Resumption Requirements:

Testing work is performed by the individual to ensure that all the functional activities are working well in the system properly. If some of these functions are not working properly in the system, the further test procedure should not be continued to the next level.

- So here are some criteria for which we will pause the test work for the application.
- If a test case fails, then testing for all the dependent features will be suspended. The failed test case will be logged onto the test log which contains the description for the error.
- If system will be to handle lots of users at once.
- The testing will be resumed once the identified failure or issue is resolved.

5.3.2.9. Test Deliverables:

If the CharitAble Donations and request generating functions are successfully working in the web app then it will be considered pass otherwise it will be failed.

Following is list of test deliverables:

1. Test plan
2. Test design specifications
3. Test case specifications
4. Test logs
5. Test summary reports
 - Test input data and test output data should be identified as deliverable.
 - Test tools may also be included.

5.3.2.10. Testing Tasks:

1. Authorization Module testing
2. Login testing
3. Registration testing
4. Performance testing
5. Account Approval by Admin testing
6. Changing Password testing
7. Contact to Admin Module testing.

5.3.2.11. Environmental Needs:

The environment requirements are as follow:

Hardware:

- Laptop and Android mobile.
- 8GB-100 GB Hard 1-4 GB Ram.
- Core i3 and later for better performance.
- Internet connectivity.

Server-side hardware:

- Core i5 and later for better performance
- 16-32 GB RAM
- 2 - 8TB Hard

Software:

- Windows 8 OS and Later
- Browser i.e., Google Chrome, internet explorer, Mozilla Firefox etc.
- Python Web Server (Linux, Windows, Unix).

5.3.2.12. Responsibilities:

Each member is responsible for requirements, designing, preparing documentation, developing and executing the system properly. The test manager is responsible for testing, managing and resolving the issues of the project

5.3.2.13. Staffing and Training Needs:

Testers should conduct testing on each system. The testers should have basic knowledge about following terms before they proceed testing:

- HTML, CSS, BOOTSTRAP, JavaScript
- Python (Django Framework)
- SQL Database

5.3.2.14. Schedule:

The testing may take about one and half week to complete. Rather Some of its components and features had been tested in a two day.

5.3.2.15. Risks and Contingencies:

If the first component testing is not completed within a day it can be delay. After bug fixes the testing will be performed. If the tester does not have the basic understanding About HTML, CSS, BOOTSTRAP, JavaScript, SQLite Database testing will be delayed.

5.3.2.16. Approvals:

The team member who manages the testing process and the team leader who manages the whole product must approve this plan. The approval of supervisor is required here:
Name: Mr. Zaheer Ahmed

Title: CharitAble for U-O-G.

Signature:

Date:

5.4 Inter Case Dependencies:

Test case Id	Dependency
TC-1	None
TC-2	TC-1
TC-3	TC-2
TC-4	TC-2,3
TC-5	TC-4
TC-6	TC-2
TC-7	TC-2
TC-8	TC-2
TC-9	TC-2
TC-10	TC-2

5.5. Test procedure specification:

5.5.1. Purpose:

To specify the steps for executing a set of test cases or, more generally, the steps used to analyze a software item in order to evaluate a set of features.

5.5.2 Outline:

A test procedure specification shall have the following structure:

1. Test procedure specification identifier
2. Purpose
3. Special requirements
4. Procedure steps

The sections shall be ordered in the specified sequence. Additional sections, if required, may be included at the end. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test procedure specification or available to users of the procedure specification.

Details on the content of each section are contained in the following sub clauses.

5.5.2.1. Test procedure specification identifier:

Test procedure specification identifier CharitAble for U-O-G ChA_04.

5.5.2.2. Purpose:

This module describes the testing procedures and the whole testing process. By looking at this module, a person will learn how to execute the testing procedures of a management system. This procedure executes following test cases TC-1, 2, 6..., 11.

5.5.2.3. Special requirements:

Front-end and Back-end testing would be on a manual. None of special requirements required.

5.5.2.4. Procedure steps:

1. Working WebApp
2. Database Connectivity
3. Login
4. Logout
5. Change Password
6. Successful Signup
7. Account Approval by Admin.
8. Generate Report
9. Request Now
10. Donations History
11. Donate Now
12. Complain Box
13. List's update
14. Update profile

5.5.2.4.1. Log:

It describes any special methods or formats for logging the results of test execution, the incidents observed, and any other events pertinent to the test that are already described in previous section.

5.5.2.4.2. Set up:

1. Install tools for development
2. Verify whether these are installed properly
3. Start Testing.

5.5.2.4.3. Start:

We need to start php, CSS, HTML, JavaScripting for front and for backend end and create a database to store data.

5.5.2.4.4. Proceed:

Access the website of Deep Fake App. Register, Login. Select user workspace from nav bar and Run test cases in dependency order.

Test cases with no dependencies or solved dependencies first.

Test case Id	Dependency
TC-1	None
TC-2	TC-1
TC-3	TC-2
TC-4	TC-2,3
TC-5	TC-4
TC-6	TC-2
TC-7	TC-2
TC-8	TC-2
TC-9	TC-2
TC-10	TC-2

5.5.2.4.5. Measure:

Measurement system is based on human observation mostly, so log what we see on the screen, if something goes wrong.

5.5.2.4.6. Shut down:

Stop CharitAble by closing Database Access and by blocking the accounts.

5.5.2.4.7. Restart:

Refresh the Browser to Restart the process. For example, After new user registration to system when admin wants to revise the user record for modification, test procedure began to restart.

5.5.2.4.8. Stop:

After running all test cases according to inter case dependency it will stop the process by Stopping data access.

5.5.2.4.9. Wrap up:

Report hours and check that test log is written. e.g., If any information is missed or update any record, it gives user to move on this task revise as restore point.

5.5.2.4.10. Contingencies:

We need to submit bug reports. Test manager will use that information to create incident reports.

5.6. Test item transmittal report:

5.6.1. Purpose:

To identify the test items being transmitted for testing. It includes the person responsible for each item, its physical location, and its status. Any variations from the current item requirements and designs are noted in this report.

5.6.2. Outline:

A test item transmittal report shall have the following structure:

1. Transmittal report identifier
2. Transmitted items
3. Location
4. Status
5. Approvals

The sections shall be ordered in the specified sequence. Additional sections may be included just prior to Approvals. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test item transmittal report or available to users of the transmittal report.

Details on the content of each section are contained in the following sub clauses.

5.6.2.1. Transmittal Report Identifier:

Transmittal report identifier Charitable for U-O-G ChA_05.

5.6.2.2. Transmitted items:

The transmitted items refer to Test Plan.

List of Items being transmitted to the test plan.

1. Volunteer / Donor Signup
2. Login
3. Request Now
4. Donate Now
5. Approve Donation
6. Change Password
7. Report
8. Complain
9. Volunteer / Donor List
10. Add new member
11. Existing admin

5.6.2.3. Location:

All transmitted test case design referenced in the report are located

Transmitted items in test case design	Location
Volunteer / Donor Signup	2.2.11.1
Login	2.2.11.2
Request Now	2.2.11.3
Donate Now	2.2.11.4
Approve Donation	2.2.11.5
Change Password	2.2.11.6
Report	2.2.11.7
Complain	2.2.11.8
Volunteer / Donor List	2.2.11.9
Add New Member	2.2.11.10
Existing Admin	2.2.11.11

Transmitted items in test case design	Test Case ID	Location	Status
Volunteer / Donor Signup	TC-1	2.2.11.1	pass
Login	TC-2	2.2.11.2	Pass
Request Now	TC-3	2.2.11.3	Pass
Donate Now	TC-4	2.2.11.4	Pass
Approve Donation	TC-5	2.2.11.5	Pass
Change Password	TC-6	2.2.11.6	Pass
Report	TC-7	2.2.11.7	Pass
Complain	TC-8	2.2.11.8	Pass
Volunteer / Donor List	TC-9	2.2.11.9	Pass
Add New Member	TC-10	2.2.11.10	Pass
Existing Admin	TC-11	2.2.11.11	pass

5.6.2.4. Status:

5.6.2.5. Approvals:

The team member who manages the testing process and the team leader who manages the whole product must approve this plan. The approval of supervisor is required here:

Name: **Mr. Zaheer Ahmed.**

Title: CharitAble for U-O-G.

Signature:

Date:

5.7. Test log:

5.7.1. Purpose:

To provide a chronological record of relevant details about the execution of tests.

5.7.2. Outline:

A test log shall have the following structure:

1. Test log identifier;
2. Description;
3. Activity and event entries.

The sections shall be ordered in the specified sequence. Additional sections may be included at the end. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test log or available to users of the log. Details on the content of each section are contained in the following sub clauses.

5.7.2.1. Test log Identifier:

Test log identifier CharitAble for U-O-G ChA_06.

Test Case Id	Test Case Name	Tester Name	Date	Software	Hardware
TC-1	Volunteer / Donor Signup	Awais	10-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-2	Login	Qandeel	10-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-3	Request Now	Iqra	11-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-4	Donate Now	Awais	12-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-5	<i>Approve donation</i>	Qandeel	13-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-6	<i>Change Password</i>	Iqra	14-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-7	<i>Report</i>	Awais	15-6-20	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-8	<i>Complain</i>	Qandeel	16-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-9	<i>Volunteer/Donor_list</i>	Iqra	17-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-10	<i>Add_New_member</i>	Awais	18-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended
TC-11	<i>Existing_Admin</i>	Qandeel	19-6-21	Window 7 or above	Core i3 Processor and 2 GB Ram Recommended

5.7.2.3. Activity and Event Entries:

Activity	Start Date	End Date
Volunteer/Donor_signup	10-6-21	10-6-21
Login	10-6-21	10-6-21
Request Now	11-6-21	11-6-21
Donate Now	12-6-21	12-6-21
<i>Approve donation</i>	13-6-21	13-6-21
<i>Change Password</i>	14-6-21	14-6-21
<i>Report</i>	15-6-21	15-6-21
<i>Complain</i>	16-6-21	16-6-21
<i>Volunteer/Donor_list</i>	17-6-21	17-6-21
<i>Add_New_member</i>	18-6-21	18-6-21
<i>Existing_Admin</i>	19-6-21	19-6-21

5.7.2.3.1. Execution Description:

Activity	Description
Volunteer/Donor_signup	Enter first Name, last Name, Email, username, password, confirm password, and your purpose to use this app.
	Then click sign up
	System will welcome you.
Log in	Enter username and password.
	Click login.
	Click on request now
Request Now	Fill the form
Donate Now	Click on donate now
	Check request list and select volunteer.
Approve donation	Click accept donation
Change password	Enter username current password, new password and confirm new password
Report	click on report to generate a donations report.
Complain	Click on complain to response on complains.
Volunteer / Donor List	Actor can check the list of volunteers and donors
Add new member	Actor can add new member
Existing admin	Actor can check another admin list who are added by the admin

5.7.2.3.2. Procedure Results:

Transmitted items in test case design	Test Case ID	Location	Status
Volunteer / Donor Signup	TC-1	2.2.11.1	pass
Login	TC-2	2.2.11.2	Pass
Request Now	TC-3	2.2.11.3	Pass
Donate Now	TC-4	2.2.11.4	Pass
Approve Donation	TC-5	2.2.11.5	Pass
Change Password	TC-6	2.2.11.6	Pass
Report	TC-7	2.2.11.7	Pass
Complain	TC-8	2.2.11.8	Pass
Volunteer / Donor List	TC-9	2.2.11.9	Pass
Add New Member	TC-10	2.2.11.10	Pass
Existing Admin	TC-11	2.2.11.11	pass

5.7.2.3.3. Environmental Information:

The environment requirements are as follow:

Hardware:

- Laptop and Android mobile.
- 8GB-100 GB Hard 1-4 GB Ram.
- Core i3 and later for better performance.
- Internet connectivity.

Server-side hardware:

- Core i5 and later for better performance
- 16-32 GB RAM
- 2 - 8TB Hard

Software:

- Windows 8 OS and Later
- Browser i.e., Google Chrome, internet explorer, Mozilla Firefox etc.
- Python Web Server (Linux, Windows, Unix).

5.7.2.3.4. Anomalous Events:

None of Anomalous events.

5.7.2.3.5. Incident Report Identifiers:

Test incident report identifier CharitAble for U-O-G ChA_07.

5.8. Test Incident Report:

5.8.1. Purpose:

To document any event that occurs during the testing process that requires investigation.

5.8.2. Outline:

A test incident report shall have the following structure:

1. Test incident report identifier
2. Summary
3. Incident description
4. Impact

The sections shall be ordered in the specified sequence. Additional sections may be included at the end. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test incident report or available to users of the incident report.

Details on the content of each section are contained in the following sub clauses.

5.8.2.1. Test Incident Report Identifier:

Test incident report identifier CharitAble for U-O-G ChA_08.

5.8.2.2. Summary:

All the items are tested for 'CharitAble for U-O-G'. The testing of this system is conducted by user and admin and testing of front end is conducted manually. When testing the system's features, there was an anomalous event occurred. But later that was fixed.

5.8.2.3. Incident Description:

Provide a description of the incident. This description should include the following items:

1. Inputs
2. Expected results
3. Actual results
4. Anomalies
5. Date and time
6. Procedure step
7. Environment
8. Attempts to repeat
9. Testers
10. Observers

Inputs	Expected result	Actual result	Anomaly	Procedure Step	Environment	Attempts to repeat	Testers	Observers
Enter username and password for Login	Login should be Successful	Login was Successful	Use of local API	Login system should be working properly	Core i3 Processor, 2 Gb Ram and active Internet connection Recommend	2 times repeated	1 tester	1 observer

5.8.2.4. Impact:

If known, indicate what impact this incident will have on test plans, test design specifications, test procedure specifications, or test case specifications.

5.9. Test summary Report:

5.9.1. Purpose:

To summarize the results of the designated testing activities and to provide evaluations based on these results.

5.9.2. Outline:

A test summary report shall have the following structure:

1. Test summary report identifier
2. Summary
3. Variances
4. Comprehensive assessment
5. Summary of results
6. Evaluation
7. Summary of activities
8. Approvals

The sections shall be ordered in the specified sequence. Additional sections may be included just prior to Approvals. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test summary report or available to users of the summary report.

Details on the content of each section are contained in the following sub clauses.

5.9.2.1. Test Summary Report Identifier:

Test summary report identifier CharitAble for U-O-G ChA_09.

5.9.2.2. Summary:

All components were tested, no such big issue is found that effects the system. And now, each component is working properly.

5.9.2.3. Variances:

Variance of links occurred.

5.9.2.4. Comprehensiveness Assessment:

The testing process is done exactly like described in the test plan. All the workable features described in the test plan are tested. The testing process is done exactly like described in the test plan. All the features described in the test plan are tested.

5.9.2.5. Summary of Results:

All results according to project requirements and functioning properly.

5.9.2.6. Evaluation:

Testing process was simple and sufficient for every phase. We reached a conclusion that all functionality of system is working Properly.

5.9.2.7. Summary of Activities:

Unit testing was completed on following dates for major module:

Volunteer / Donor Signup	1-07-2021
Login	1-07-2021
Request Now	1-07-2021
Donate Now	2-07-2021
Approve Donation	2-07-2021
Change Password	2-07-2021
Report	3-07-2021
Complain	3-07-2021
Volunteer / Donor List	3-07-2021
Add new member	4-07-2021
Existing admin	4-07-2021

5.9.2.8. Approvals:

Name	Title	Signature	Date
Mr. Zaheer Ahmed	CharitAble for U-O-G		14-07-2021

Chapter 6

Appendix

Final Documentation Format Guidelines:

Typographical Format and Binding

Page Format:

- Page size: A4
- Top margin: 1.00 inch
- Bottom margin: 1.00 inch
- Left margin: 1.25 inch
- Right margin: 1.00 inch

Page numbering: Top right - part of the Header

Title page not numbered

All other pages before the page of chapter one numbered in lower roman numerals (i, ii, iii, ...)

All other pages starting from first page of chapter one to last page of the report numbered in integers (1, 2, 3, ...)

Footer:

Each page shall have a footnote “Division of Science & Technology, University of Education, Lahore”

- Left aligned
- In case of long titles shorter versions should be used.
- There shall be a line over the footnote.

Header:

- Each page shall have a header “Project Name”
- Left aligned
- In case of long titles shorter versions should be used.
- There shall be a line under the footnote.

Chapter Startup:

Each chapter shall be numbered as Chapter 1, Chapter 2, etc. The name of the chapter shall be written immediately below. Both shall be centered horizontally as well as vertically.

Text:

- Only one side of the paper shall be used.
- The other side shall be blank.
- When a report is opened the right side would contain text, figures, or tables and the left side would be blank.

Tables and Figures:

- Tables and figures shall be placed on one side only
- Separate pages shall be used for figures and tables.
- One page may contain more than one figure or table but text will not be combined or interlaced with figure or table.
- Each table / figure shall be numbered.
- For example, "Table 1.2: Population distribution in Asia" or "Figure 3.2: Temperature distribution"

The table number or figure number shall be placed as normal text centered at the bottom of the table or figure or sideways with table / figure title coming on the opening side of the paper and note on the binding side.

Paragraph:

- Single-spaced.
- Line entered paragraph.
- DONOT put indents at the beginning of the paragraph.
- Left aligned or justified.

Text Format:

- Normal and plane text:
- Font Type: Times New Roman
- Font Size: 12
- Headings:
- Chapter Heading: Times New Roman Bold Size 16 Title Case normal
- Heading 1: Times New Roman Bold Size 14 Title Case normal
- Heading 2: Times New Roman Bold Size 12 Title Case normal
- Heading 3: Times New Roman Bold Size 12 Title Case italic

Sections and Subsections:

In case of sections and subsections follow this format:

1 Section

1.1 Sub Section

1.1.1 Nested Sub Section

a

b

i

ii

The subsequent reference to a any section shall be made using the section and its number. For example, section 2.1.3 means chapter 2 section 1 subsection 3.

Mathematical Equations:

The following numbering scheme should be used to number the equations:

$$f(x) = x+3 \quad (XX:YY)$$

Where XX is the chapter number and YY is the sequence number of that equation in that chapter.

If an equation is previously quoted in an earlier chapter, say as equation 4:5 and need to be re-quoted in chapter 5, its number will remain as equation 4:5.

References:

References are to be placed in square brackets and interlaced in the text. For example, "A comprehensive detail of how to prevent accidents and losses caused by technology can be found in the literature [1]. A project report / thesis cannot be accepted without proper references. The references shall be quoted in the following format:

The articles from journals, books, and magazines are written as:

[1] Abe, M., S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. Journal of the Acoustical Society of Japan, April 1990, E-11 pp 71-76.

[2] Hermansky, H. Perceptual linear predictive (PLP) analysis for speech. Journal of the Acoustical Society of America, January 1990, pp 1738-1752.

The books are written as:

- [1] Nancy G. Leveson, Safeware System Safety and Computers, A guide to preventing accidents and losses caused by technology, Addison-Wesley Publishing Company, Inc. America, 1995.
- [2] Richard R. Brooks, S. S. Iyengar, Multi-Sensor Fusion Fundamentals and Applications with Software, The Prentice-Hall Inc. London, 1998.

Binding

All reports shall be bounded with an appropriate print on the backbone.

Two copies should be submitted.

Color of the binding:

- BSc project / thesis reports: black
- MSc project / thesis reports: blue

Contents of the CD Attached

All reports / theses must accompany a CD whose contents will have the following:

Top-level directories:

Doc All documents related to the project
 Instructions how to access the CD to the point to running the project

All reports already submitted

The final project report in thesis form
 Installation instructions
 Trouble shooting instructions in case of problems
 User manual
 Research material including URLs
 Papers consulted / referred to
 Slides of the presentations

Source All source files that will be needed to compile the project.

Further subdirectories can be used.

This must include sample data files as well.

Project The running project including sample data files as well as sample output.

This should be in a form that if copied to a machine runs without errors.

This may be an exe file of an entire project, an installer depending on the project or simply a running project.

You can have sub directories with appropriate names.

Length

The length of your dissertation depends on the type of project you have selected. An excellent dissertation will often be brief but effective (its author will have said a lot in a small amount of space). Voluminous data can be submitted electronically on CD.