

Voice of Charity: Prospecting the Donation Recurrence & Donor Retention in Crowd funding

Submitted in partial fulfillment of the requirements for the award of
Bachelor of science Degree in Computer Science

By

Dinesh Babu.S (Reg. No. 38290016)

Mukesh.K (Reg. No. 38290049)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Dinesh Babu.S RegNo:38290016&Mukesh.K RegNo:38290049** whocarried out the project entitled “**Privacy Protection in Interactive Content Based Image Retrieval**” under our supervision from December 2020 to APRIL 2021.

Internal Guide

Ms. R.Yogitha B.E, M.E.

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination heldon_____

InternalExaminer

ExternalExaminer

DECLARATION

I, **DINESH BABU.S**(Reg. No. **38290016**)hereby declare that the Project Report entitled" **Voice of Charity: Prospecting the Donation Recurrence & Donor Retention in Crowd funding** " done by me under the guidance of**MS.R.YOGITHA** is submitted in partial fulfillment of the requirements for the award of Bachelor of Arts and Science in Computer Science.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI, M.E., Ph.D., and Dr. L. LAKSHMANAN, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.R.Yogitha** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the Completion of the project.

ABSTRACT

Crowdfunding is a practice of raising funds from people to support your project which has brought new life to charity, i.e., making it easy to donate any amount of money to help across the globe. Donation-based crowdfunding is the most preferred mode of fundraising. Crowdfunding through online platforms knows no boundaries, and has the potential to go viral. The problem of high donor attrition i.e., many donors donate only once or very few times within a rather short lifecycle and then leave. Thus, it is an urgent task to analyze the factors of and then further predict the donor behavior. In this paper, we present a focused study on analysis of donation recurrence and donor retention to predict the donor's interest in donation. Specifically, we proposed a model, which has the details of recipient, actual donor and the verifying person. After the donation process, every donor will get a proper donation certificate approved by the government. The experimental results will clearly demonstrate the individual's interest for donation and to appreciate them to donate more in their future with a proper secured transaction with the support of the government

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS	Xiii
1	INTRODUCTION	1
2	SYSTEM ANALYSIS	4
2.1	Existing system	4
2.2	Proposed system	5
2.3	Hardware and Software specification	5
3	SYSTEM DESIGN	7

3.1	Module	8
3.2	Module Explanation	8
4	Project Purpose and Scope	10
4.1	Purpose	10
4.2	Project Scope	10
4.3	Product Perspective	11
4.4	System Features	12
4.5	Design and Implementation Constraints	13
4.6	Other Nonfunctional Requirements	14
	SOURCE CODE	23
	SNAP SHOTS	35
	REFERENCES	45

LIST OF FIGURE

CHAPTER 5

5.1 ARCHITECTURE DESIGN

5.2 SEQUENCE DIAGRAM

5.3 USE CASE DIAGRAM

5.4 ACITIVITY DIAGRAM

5.5 COLLABARTION DIAGRAM

LIST OF ABBREVIATIONS

JDK Java Development Toolkit

DEX Dalvik Executables

TCP Transmission Control Protocol

IP Internet Protocol

HTTP Hyper Text Transfer Protocol

ADT Android Development Tool

CHAPTER 1

INTRODUCTION

CROWDFUNDING is an emerging Internet-based fundraising mechanism soliciting small monetary contributions from crowd donors to help others in trouble or with dreams. Recent years have witnessed the rapid development of crowdfunding platforms among which the donation-based ones are becoming increasingly popular and Donors. Leveraging Internet, crowdfunding has brought new life to charity, i.e., making it easy to donate any amount of money even every penny to help others across the globe.

For example, Kiva.org is an international nonprofit platform, founded in 2005, with a mission to connect people through lending to alleviate poverty. Specifically, Kiva.org enables Field Partners (nonprofit organizations around the world) to screen the needy or suffering, and post requests in the form of projects to Kiva.org for funding. Then the accessing donors crowdfund these projects in increments of \$25 or more. Donors may act as individuals or teams. The critical component for the success of crowdfunding communities is the recruitment and continued engagement of donors. However, because of the non-profit nature, the situation relating to donor retention for donation-based

crowdfunding as well as traditional charities is extremely serious, i.e., usually, the donor

1

attrition rate is above 70% . Actually, customer attrition/churn is crucial and highly focused on in many commercial scenarios, such as E-commerce, finance and services. However, for a quite long time, relevant studies on analyzing donor retention in charity have been rather limited in the literature. Fortunately, with the accumulation of large-scale user behavior data in crowdfunding platforms, many data-driven studies which focus on analyzing the user behaviors have been conducted . For example, Liu,studied the donation motivation classification in Kiva.com. Especially, Althoff explored various factors impacting donor retention in DonorsChoose.org from the statistical perspectives which was inspiring for our research. However, how to comprehensively analyze the heterogeneous factors affecting and then further predict the donor retention or attrition, are still largely unexplored areas, both in the charity and in other domains. In addition to these heterogeneous factors, according to our observation and analysis, donors' own behaviors (i.e., donation recurrence) could particularly reflect their decision on retention. In fact, donation recurrence prediction is an inevitable intermediate goal for predicting the donor retention. Thus, in this paper, we attempt to track this problem by jointly predicting the donor retention and also the

intermediate goal (predicting donation recurrence). Although it is necessary to construct the predictions of donation recurrence and donor

2

retention, as they can alert platforms that they need to do something before they lose donors, this is a very challenging task. First, donor behaviors, e.g., donation recurrence, donor retention or attrition, are influenced by various factors, such as their motives and preferences, their social contacts in crowdfunding communities, and the characteristics of the projects to which they have recently donated. How to comprehensively analyze the heterogeneous features and integrate them for accurate prediction is not a trivial issue. Second, according to our data analysis, the behaviors of donors, especially the donation recurrence, are highly correlated with their retention or attrition. How to model the relations of donation recurrence and donor retention and further synchronously predict these two behavioral events with a joint model are quite open problems. Finally, the presence of a large amount of censored data i.e., the exact attrition outcomes of some donors are unobservable or they

do not perform any behaviors (donation or attrition) during our monitoring periods, imposes significant challenges in relation to this problem. Because many donors may be still in the platform in our data

and most lost donors do not explicitly close their accounts when leaving, the censoring phenomenon is an inescapable concern.

3

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In the existing system, donation recurrence and donor retention was predicted with the help of large-scale behavioral data collected from crowdfunding. It uses the Joint Deep Survival model to integrate the heterogeneous data since they are highly relevant. This system clearly demonstrates the effectiveness for analyzing and predicting the donation recurrence and donor retention in crowdfunding. The retention of the donor is not clearly known because they have not observed the occurrence of donor attrition. The models may lose the abilities of capturing the sequence dependence for such a long time.

Problem Definition

- ✓ The donor and the recipient are not authenticated so we cannot identify whether they are verified authenticated users.

- ✓ The donor may not know whether the recipient is getting the whole amount of money paid by the donor.
- ✓ Malpractice may occur in the name of donation in the income tax department by some donors.

4

2.2 PROPOSED SYSTEM

In this work, the client and the donor have to fill their personal details which will be verified by the third party, the verifying agent appointed by the government. The verifying agent will accept the details by verifying their details and proceeds the secured transaction from the donors to the clients. This will make the secure transaction only after the secured authentication. So that only the authenticated users and donors can donate or asks for donation. This may avoid the income tax frauds or those who escapes from tax pay by false statements of money lending.

Advantages:

- ✓ By analyzing and predicting the donation recurrence and donor retention, the donors can be predicted easily and are contacted for any need of money from the recipient.
- ✓ The donation recurrence and donor retention are highly correlated and indicates the effectiveness of JDS structure.

2.3 HARDWARE AND SOFTWARE SPECIFICATION

.2.1 HARDWARE REQUIREMENTS

- Hard Disk : 80GB and Above
- RAM : 4GB and Above
- Processor : P IV and Above

5

2.4 SOFTWARE REQUIREMENTS

- Windows 7 and above
- JDK 1.7
- Tomcat 7.0
- MySQL

2.5 TECHNOLOGIES USED

- J2EE (JSP, Servlets), JavaScript, HTML, CSS.

CHAPTER 3

SYSTEM DESIGN

The recipient has to fill the registration form with required mandatory input fields and the personal details completely. These details will get further verified by the government authority or the verifying agent appointed by the government. The donor has to fill the registration form with required mandatory input fields and the personal details completely. These details will get further verified by the government authority or the verifying agent appointed by the government. By using the token, generated after the completion of verifying recipient's details by the government authority. The recipient will login only using the unique token to request money. Their request contains the details of the category for which the recipient needs money. The recipient can view their request details and status of donation for their requests. By using the token, generated after the completion of verifying donor's details by the government authority. The donor will login only using the unique token to donate money. Their request contains the details or the

category for which the donor is willing to donate. The donor can view the history of their donations. After verifying all the details of donor and the recipient, the verifying person will approve the details of recipient and the donor and further allows the process of funding.

7

3.1 MODULES

- ✓ **Donor Verification**
- ✓ **Donor Verification and Donor's Chatbot**
- ✓ **Recipient's Chatbot**
- ✓ **Providing fund from donor to recipient**

3.2 MODULE EXPLANATION:

3.2.1 Donor Verification:

The donor has to fill the registration form with required mandatory input fields and the personal details completely. These details will get verified by the government authority or the verifying agent appointed by the government. After successful verification, the donor will get the registration and the authenticated mail.

Recipient Verification and Chatbot:

The recipient has to fill the registration form with required mandatory input fields and the personal details completely. These details will get verified by the government authority or the verifying agent appointed by the government as same as the donor's verification.

8

After successful verification, the recipient will get the notifications regarding their donation with secured authentication. The recipient's chatbot contains the details or the category for which the recipient needs donation.

Donor's Chatbot:

The donor's chatbot contains the details or the category for which the donor is willing to donate money. The details of the recipients for which the donor request to donate will get displayed.

Providing fund from donor to recipient:

After verifying all the details of donor and the recipient, the verifying person will approve the details of recipient and the donor and

further allows the process of funding. The donor will be displayed with the recipient details in the category for which the donor is willing to donate. The donor will select the recipient and further donates. After the funding process, the donor will be provided by a certificate by the government.

CHAPTER 4

Project Purpose and Scope

4.1 Purpose

The main aim of this project is to clearly demonstrate the effectiveness for analyzing and predicting the donation recurrence and donor retention in crowdfunding and providing a proper trustworthy donation from the donors to the clients.

4.2 Project Scope

In this work, the client and the donor have to fill their personal details which will be verified by the third party, the verifying agent appointed by the government. The verifying agent will accept the details proceeds secured transaction from the donors to the clients. This system uses the clustering algorithm to filter the data from large scale of datasets and uses K-Nearest neighbor algorithm for clustering, which will cluster the

similar data items from large dataset according to the user preference. This system will automatically notifies the donors on any particular day, for example, on their birthday, and appreciate the donors to further improve their sequence of donation.

10

This system will make a secure transaction from the donors to the clients only after verifying whether the donors and the clients are authorized persons. And also this will improve the eagerness of donors for donation.

4.3 Product Perspective

Charity is an organization set up to provide help and raise money for those in need. It is generosity and helpfulness especially toward the needy or suffering. Crowdfunding works through individuals or organizations that invest in (or donate to). Crowdfunding projects in return for a potential profit or reward. Investing this way can be risky, so make sure you know what you're doing. So, we are providing the trustworthy crowdfunding through government, which is the funding of a project by a large number of supporters who each contribute a small amount. Only authenticated recipient and donor can request and donate

money here. This system uses the clustering algorithm to filter the data from large scale of datasets and uses K-Nearest neighbor algorithm for clustering the similar data from large dataset. These systems help in automatically notifying the donors according to their interest in donation on any particular day, for example, on their birthday, and appreciate the donors to further improve their sequence of donation.

11

4.4 SystemFeatures

Crowd funding is a practice of raising funds from people to support your project which has bought new life to charity, i.e., making it easy to donate any amount of money to help across the globe. Donation-based crowd funding is the most preferred mode of fundraising. Crowd funding through online platforms knows no boundaries, and has the potential to go viral. The problem of high donor attrition i.e., many donors donate only once or very few times within a rather short lifecycle and then leave. Thus, it is an urgent task to analyze the factors of and then further predict the donor behavior. In this paper, we present a focused study on analysis of donation recurrence and donor retention to predict the donor's interest in donation. Specifically, we proposed a model, which has the details of recipient, actual donor and the verifying person. After the donation process, every donor will get a proper donation certificate

approved by the government. The experimental results will clearly demonstrate the individual's interest for donation and to appreciate them to donate more in their future with a proper secured transaction with the support of the government.

12

4.5 Design and Implementation Constraints

4.5.1 Constraints in Analysis

- ◆ Constraints as Informal Text
- ◆ Constraints as Operational Restrictions
- ◆ Constraints Integrated in Existing Model Concepts
- ◆ Constraints as a Separate Concept
- ◆ Constraints Implied by the Model Structure

4.5.2 Constraints in Design

- ◆ Determination of the Involved Classes
- ◆ Determination of the Involved Objects
- ◆ Determination of the Involved Actions
- ◆ Determination of the Require Clauses
- ◆ Global actions and Constraint Realization

4.5.3 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one.

13

It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

4.6 Other Nonfunctional Requirements

4.6.1 Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;

- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

14

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

4.6.2 Safety Requirements

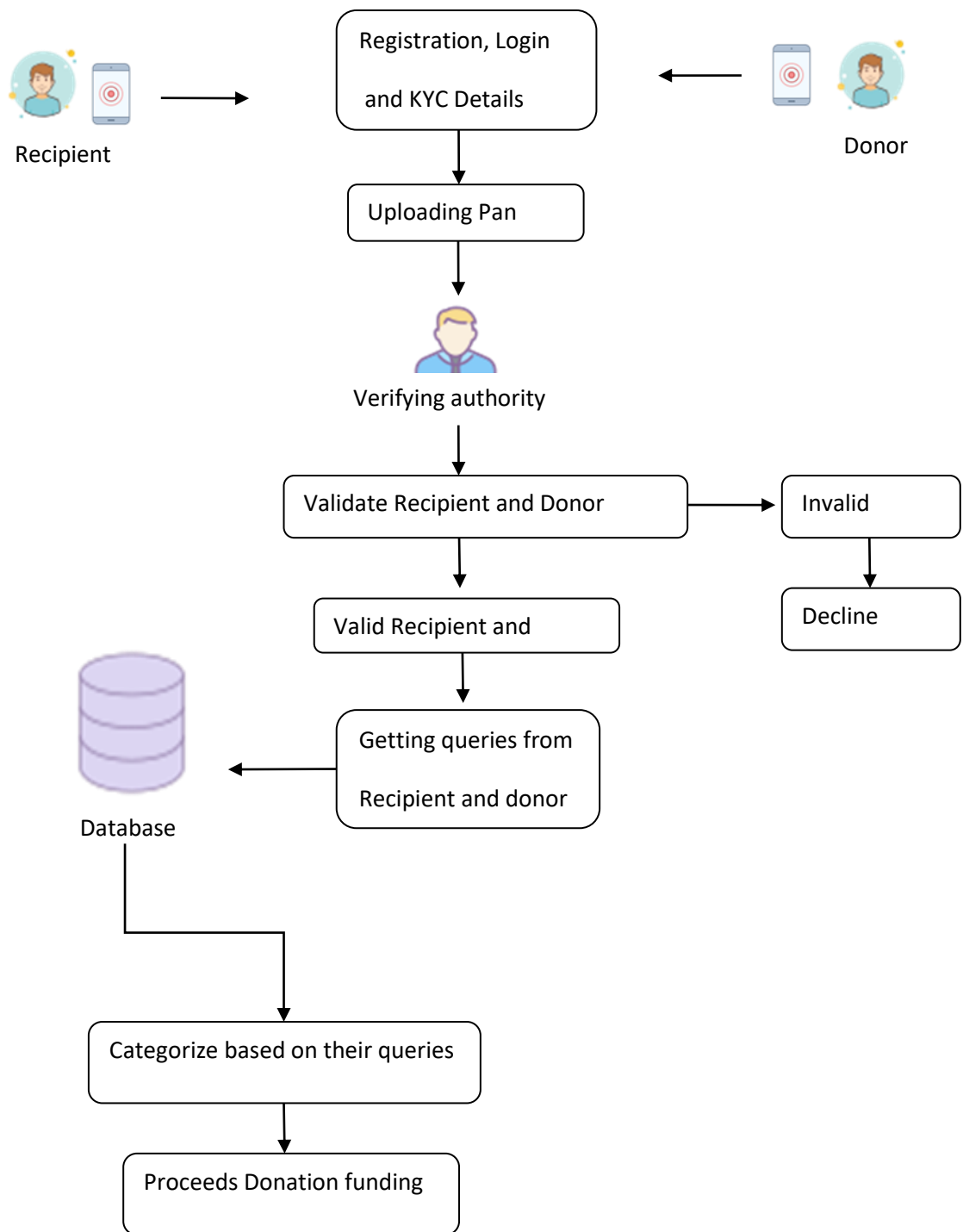
1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.

4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.

15

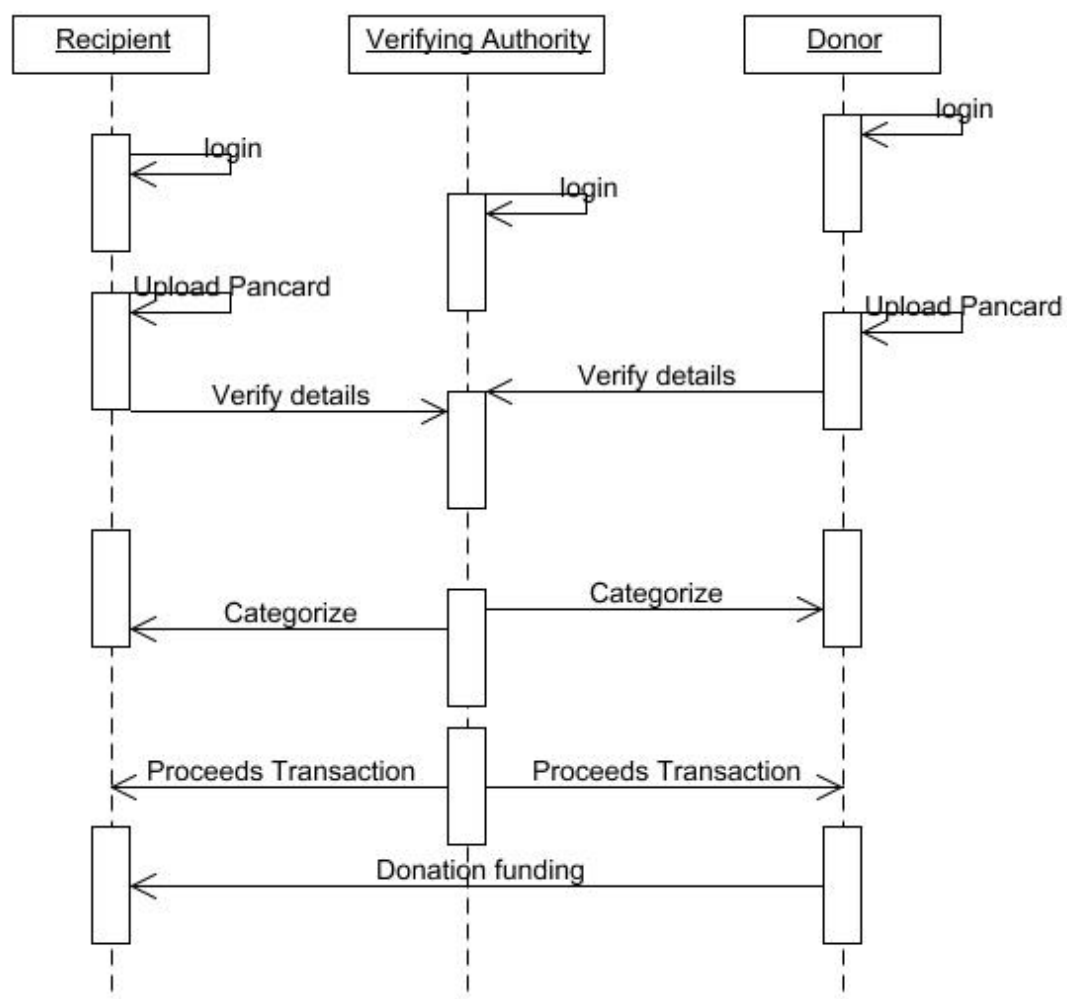
CHAPTER 5

5.1 Architecture Diagram



5.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



5.3 USECASE DIAGRAM

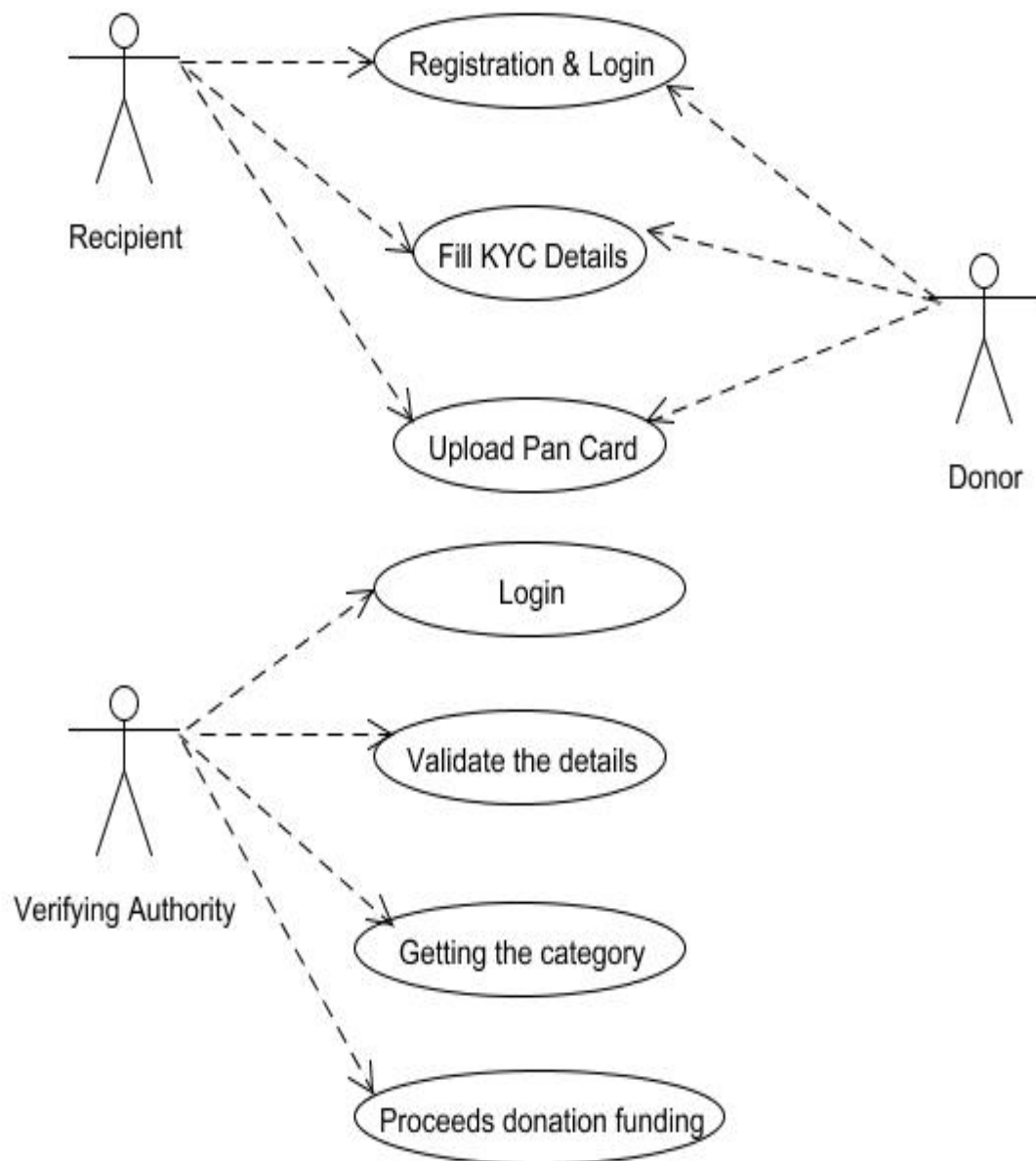
Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

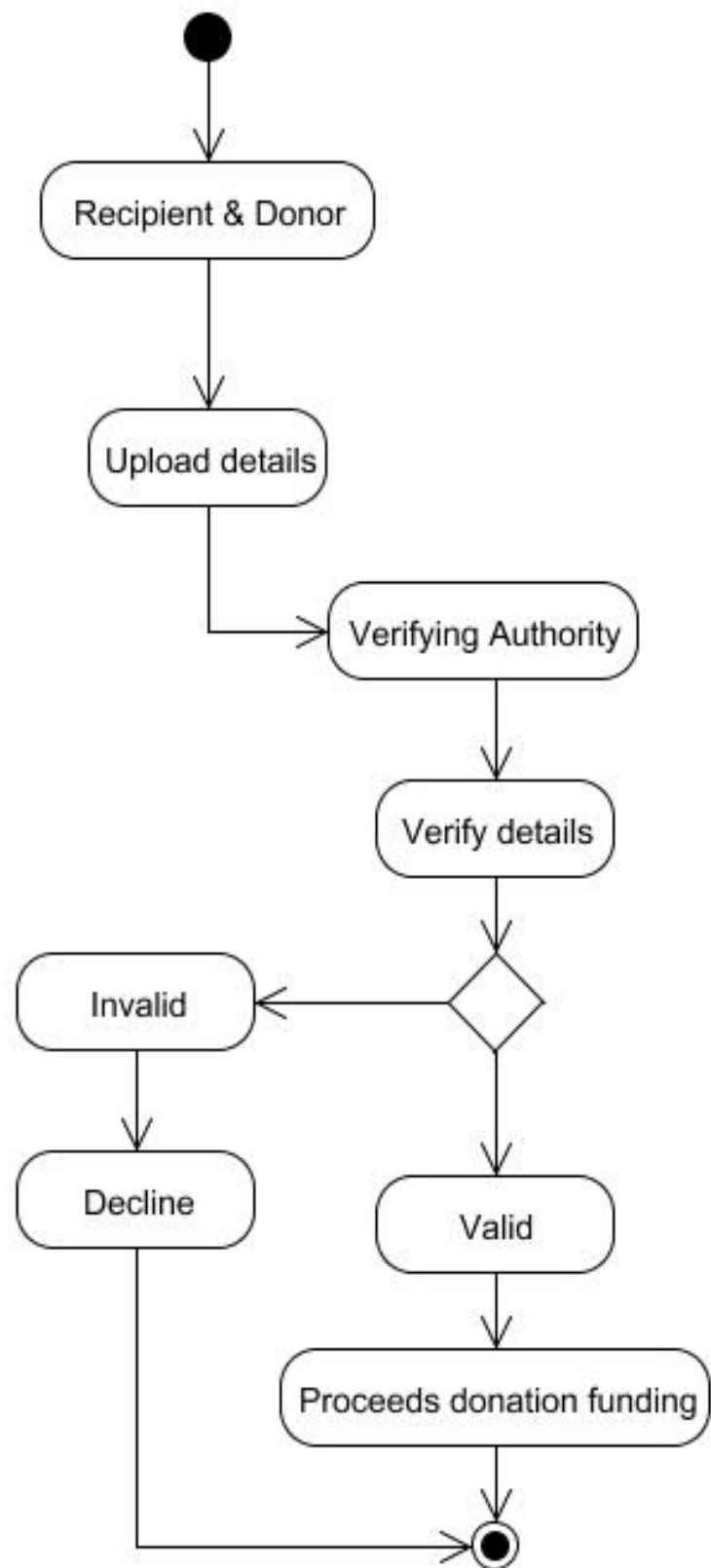


5.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

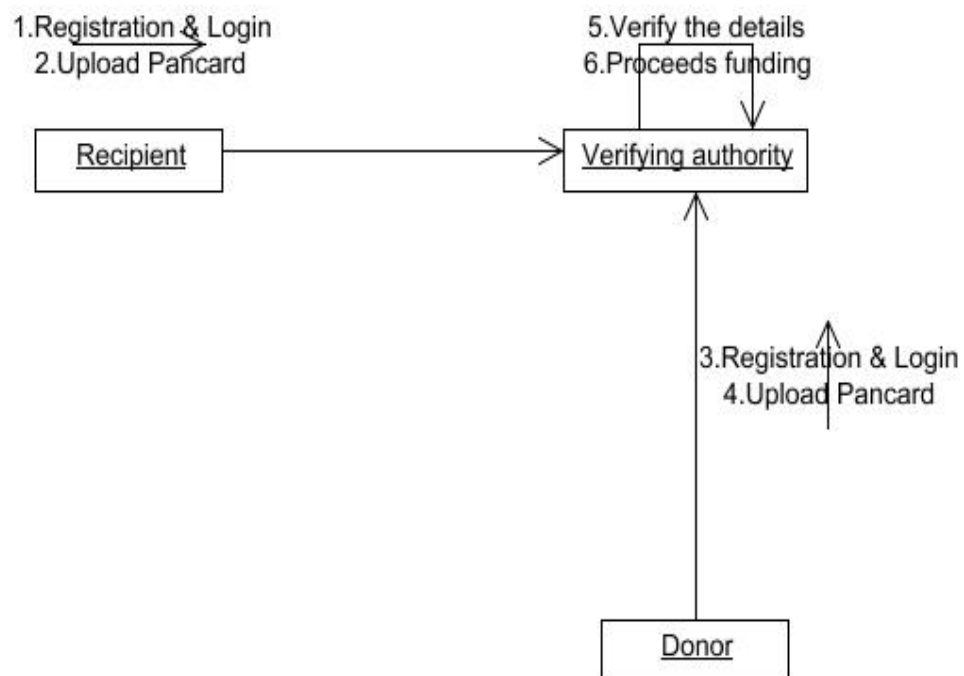
The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



5.5 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.



SOURCE CODE

FOR USER LOG IN

```
package Logic;

import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Date;
import java.util.Properties;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.quartz.JobExecutionContext;
import org.quartz.JobExecutionException;
import org.quartz.SchedulerException;

import com.http.servlet.HttpsServlet;
import Notific.Scheduling;

public class UserLogin extends HttpsServlet{

    boolean stat=false;
    public void execute(JobExecutionContext arg0) throws
    JobExecutionException {
        try
        {
            arg0.getScheduler();

            Date nextRunTim=arg0.getNextFireTime();
            long runTim=arg0.getJobRunTime();
            Connection connection = (Connection)
getServletContext().getAttribute("connection");
```

```

        System.out.println("connection success");
        Statement st = connection.createStatement();
        ResultSet rs;
        //rs = st.executeQuery("select * from donorcard where
dateofbirth= CURRENT_DATE");
        rs = st.executeQuery("SELECT * FROM donorcard
where MONTH(dateofbirth) > MONTH(NOW()) OR( MONTH(dateofbirth)
= MONTH(NOW()) AND DAY(dateofbirth) >= DAY(NOW()))");
        if (rs.next()) {
            String FirstName = rs.getString("firstname");
            String email=rs.getString("email");
            String Dob=rs.getString("dateofbirth");

            System.out.println("=====>" + FirstName + "=====>" +
email + "=====>" + Dob);
            File pf = new File("");

            String proppath = pf.getAbsolutePath();
            String filepath = proppath +
"\webapps\\VoiceOfCharity\\Properties\\MailProperties.properties";
            InputStream pfis =
this.getClass().getResourceAsStream("MailProperties.properties");
            Properties prop = new Properties();
            prop.load(pfis);
            String empuserid = prop.getProperty("adminid");
            String empmailpassword =
prop.getProperty("password");
            String subject = "Online Donation";
            String text = "Hello " + FirstName + "... \n Wishing
you a very happy birthday!!! You have a plenty of categories to donate. If
you are willing to donate on your special day... Login to donate..." +
            "\t Thanks & Regards \t Cloud OTP
Service";

            Mail sme = new Mail(empuserid, email, subject,
text, empuserid,empmailpassword);

        }

    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        public void doPost(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            HttpSession httpsession = request.getSession(true);
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();

            try{
                String Email=request.getParameter("email1");
                String uid=request.getParameter("status1");
                Connection connection = (Connection)
getServletContext().getAttribute("connection");

                System.out.println("connection success");
                Statement st = connection.createStatement();
                ResultSet rs;
                rs = st.executeQuery("select * from kycdetails where
email1='" + Email + "'and status1='" + uid + "' ");

                if (rs.next()) {
                    String Email1 = rs.getString("email1");
                    String status=rs.getString("status1");
                    httpsession.setAttribute("email1", Email1);
                    RequestDispatcher rd =
request.getRequestDispatcher("RChatbot.jsp");
                    rd.forward(request, response);

                } else {
                    request.setAttribute("error", "Invalid Email or
Password or Check that you have entered the password that you get in
your mail after verification.");
                    RequestDispatcher rd =
request.getRequestDispatcher("login.jsp");
                    rd.forward(request, response);
                }

            }catch (Exception e) {
                e.printStackTrace();
            }

        }

        public String pauseAuditing()

```

```

{
    try
    {
        Notific.Scheduling sch=new Notific.Scheduling();
        sch.pauseJob();

    } catch (SchedulerException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return "success";
}
public String resumeAuditing()
{
    try
    {
        Notific.Scheduling sch=new Notific.Scheduling();
        sch.resumeJob();

    } catch (SchedulerException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return "success";
}
}

```

BANK LOGIN

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;
import com.opensymphony.xwork2.ActionSupport;

public class BankLogin extends ActionSupport
{
    String adminusername="",adminpassword="";
    Random rand=new Random();
    ArrayList<String> bankacnholdnames=new ArrayList<String>();
    String
    username,phonetext,emailtext,addresstext,citytext,pincodetext,statet
    ext;
    String
    firstselectaccountnotext,firstbankholdernames,firstamounttext,firstphonet
    ext,firstmailtext,firstaddresstext,firstcitytext,firstpincodetext,firststatetext;

    public String getFirstselectaccountnotext() {
        return firstselectaccountnotext;
    }

    public void setFirstselectaccountnotext(String
    firstselectaccountnotext) {
        this.firstselectaccountnotext = firstselectaccountnotext;
    }

    public String getFirstbankholdernames() {
        return firstbankholdernames;
    }

    public void setFirstbankholdernames(String firstbankholdernames)
    {
        this.firstbankholdernames = firstbankholdernames;
    }

    public String getFirstamounttext() {
        return firstamounttext;
    }
}
```

```

    }

    public void setFirstamounttext(String firstamounttext) {
        this.firstamounttext = firstamounttext;
    }

    public String getFirstphonetext() {
        return firstphonetext;
    }

    public void setFirstphonetext(String firstphonetext) {
        this.firstphonetext = firstphonetext;
    }

    public String getFirstmailtext() {
        return firstmailtext;
    }

    public void setFirstmailtext(String firstmailtext) {
        this.firstmailtext = firstmailtext;
    }

    public String getFirstaddresstext() {
        return firstaddresstext;
    }

    public void setFirstaddresstext(String firstaddresstext) {
        this.firstaddresstext = firstaddresstext;
    }

    public String getFirstcitytext() {
        return firstcitytext;
    }

    public void setFirstcitytext(String firstcitytext) {
        this.firstcitytext = firstcitytext;
    }

    public String getFirstpincodetext() {
        return firstpincodetext;
    }

    public void setFirstpincodetext(String firstpincodetext) {

```

```

        this.firstpincodetext = firstpincodetext;
    }

    public String getFirststatetext() {
        return firststatetext;
    }

    public void setFirststatetext(String firststatetext) {
        this.firststatetext = firststatetext;
    }

    public String getEmailtext() {
        return emailtext;
    }

    public void setEmailtext(String emailtext) {
        this.emailtext = emailtext;
    }

    public String getAdminusername() {
        return adminusername;
    }

    public void setAdminusername(String adminusername) {
        this.adminusername = adminusername;
    }

    public String getAdminpassword() {
        return adminpassword;
    }

    public void setAdminpassword(String adminpassword) {
        this.adminpassword = adminpassword;
    }

    public String getUsername() {
        return usernametext;
    }

    public void setUsername(String usernametext) {
        this.usernametext = usernametext;
    }

```

```

}

public String getPhonetext() {
    return phonetext;
}

public void setPhonetext(String phonetext) {
    this.phonetext = phonetext;
}

public String getAddresstext() {
    return addresstext;
}

public void setAddresstext(String addresstext) {
    this.addresstext = addresstext;
}

public String getCitytext() {
    return citytext;
}

public void setCitytext(String citytext) {
    this.citytext = citytext;
}

public String getPincodetext() {
    return pincodetext;
}

public void setPincodetext(String pincodetext) {
    this.pincodetext = pincodetext;
}

public String getStatetext() {
    return statetext;
}

public void setStatetext(String statetext) {
    this.statetext = statetext;
}

```

```

    public ArrayList<String> getBankaccntholdnames() {
        return bankaccntholdnames;
    }

    public void setBankaccntholdnames(ArrayList<String>
bankaccntholdnames) {
        this.bankaccntholdnames = bankaccntholdnames;
    }

    public String execute()
    {
        String status="error";
        try
        {
            if(adminusername.equals("admin") &&
adminpassword.equals("admin"))
            {
                XIApi xlap=new
XIApi("webapps/BankAdmin/Pincode.csv",this);
                commonMetDb();
                status="success";
            }
            else
            {
                addActionMessage("Login Failed!");
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        return status;
    }

    public String setNewHolder()
    {
        String status="success";
        try
        {
            String
accno=rand.nextInt(10)+""+rand.nextInt(10)+""+rand.nextInt(10)+""+rand
.nextInt(10)+""+rand.nextInt(10)+""+rand.nextInt(10)+""+rand.nextInt(10)
+""+rand.nextInt(10); 31

```

```

        String
pinpass=rand.nextInt(10)+""+rand.nextInt(10)+""+rand.nextInt(10)+""+ra
nd.nextInt(10);
        CommonBean comb=new CommonBean();
comb.setAccountnumb(accno);
        comb.setUsername(username text);
        comb.setMobnum(phonetext);
        comb.setEmail(emailtext);
        comb.setAddress(address text);
        comb.setCity(citytext);
        comb.setPincode(pincodetext);
        comb.setState(statetext);
        comb.setPinpass(pinpass);
        comb.setAmount("10000");

        DbQueries db=new DbQueries();
        int dbst=db.insertUserDetails(comb);
        commonMetDb();
        addActionMessage("Added Successfully.");
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    return status;
}
public String updateAccountHold()
{
    String status="success";
    try
    {
        DbQueries db=new DbQueries();
        CommonBean comb=new CommonBean();
        comb.setAccountnumb(firstselectaccountnotext);
        comb.setUsername(firstbankholdernames);
        comb.setMobnum(firstphonetext);
        comb.setEmail(firstmailtext);
        comb.setAddress(firstaddress text);
        comb.setCity(firstcitytext);
        comb.setPincode(firstpincodetext);
        comb.setState(firststatetext);
        db.updateAccountHolderTab(comb);
    }
}

```

```

        commonMetDb();
        addActionMessage("Updated Successfully.");
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    return status;
}

public String logOut()
{
    return "success";
}

public String toAdminPage()
{
    commonMetDb();
    return "success";
}

public String addAmount()
{
    commonMetDb();
    return "success";
}

public String addAcctAmountHolder()
{
    String status="success";
    try
    {
        DbQueries db=new DbQueries();
        CommonBean comb=new CommonBean();
        comb.setAccountnumb(firstselectaccountnotext);
        comb.setUsername(firstbankholdernames);
        comb.setAmount(firstamounttext);
        db.addAmountAccount(comb);
        commonMetDb();
        addActionMessage("Added Successfully.");
    }
}
}33

```

```

        catch(Exception e)
        {
            e.printStackTrace();
        }
        return status;
    }

    public void commonMetDb()
    {
        DbQueries db=new DbQueries();
        ArrayList arra=db.getUsersAccountld();
        bankacn্থoldnames.addAll(arra);
        Collections.sort(bankacn্থoldnames);
        if(bankacn্থoldnames.isEmpty())
        {
            bankacn্থoldnames.add("--Select--");
        }
    }
}

```

SNAPSHOT



Home / Recipient Register

Register Here

First name	Last name
Mobile Number	Email

Activate Windows
Go to Settings to activate Windows.

KYC Form

Name	Email
Bairavi	bairavi@gmail.com
Date Of Birth	Age
06-10-1997	Less than 30
Father Name	Mother Name
Nagarajan	Seetha
Gender	Marital Status
Female	☒ Single ☐ Married
Annual Income	Aadhar Card Number
\$10000 to \$50000	3895748398763789
Pan Card Number	Contact Number
BNZPM2557D	9898989898
Address Line1	District
Ashok Nagar	Chennai
State	Country
Tamilnadu	India
Pincode	
694832	

Activate Windows
Go to Settings to activate Windows.

Welcome :- Bairavi ...!!

Upload the document*

Upload Income Certificate No file chosen

Activate Windows
Go to Settings to activate Windows.

Recipient Identity



Activate Windows
Go to Settings to activate Windows.



[Home](#) / [Recipient Login](#)

Login Here

*Please enter the ID that you get after registration as your password

Email

Password

Login

Activate Windows
Go to Settings to activate Windows.

Central Board

 [LOGOUT](#)

User

Activities



RECIPIENT REQUESTS



DONOR REQUESTS

Central Board

HOME

 LOGOUT

Recipient

Details

Search...



S.no	Name	Date of Birth	Father Name	Mother Name	Email ID	Marital Status	Aadhar Card Number	Pin Card Number	Contact Number	Address Line 1	Address Line 2	State	Country	Pincode	Image	Pin Card Image	Income Certificate	Status
1	Debjoyoti	03/03/1994	Roy	YASHILA	Debjoyoti@gmail.com	Married	7549385437536546	HF9KG2327G	7373099998	1 Annagirao Nagar Street,Nespapakkam	Coimbatore	Tamilnadu	India	600020				Viewed
2	Kaustavi	09/09/1997	Saha	WALIEHA	Kaustavi@gmail.com	Married	6747812483998788	HJFRJ7345H	7402000000	No.2/41, Rangaliya Street, 1st Floor, Ayandavaram,	Thanjavur	Tamilnadu	India	600082				Viewed
3	Debobrata	01/01/1993	Podder	RUMAIZA	Debobrata@gmail.com	Single	8362545367249379	RJLHJ5757G	7402099999	B-10, Dev Apartments, 77/32, Kalakshetra Road,	Tiruchirappalli	Tamilnadu	India	600029				Viewed
4	Anurag	22/07/1996	Mark Topno	VISHNU	Anurag@gmail.com	Married	7234623746298439	RYWHDE343J	7502000000	T-2, Prathyeka Apartments, 12, First Trust Link Street,	Tirunelveli	Tamilnadu	India	600040				Viewed
5	Tarun	23/02/1994	Wiro	PRIVA	Tarun@gmail.com	Married	8348365673463897	NHJHJ7384G	7502099999	No.10/21, Shanmuga Rasa Street, Gandhi Nagar,	Salem	Tamilnadu	India	600102				Viewed
6	Praveen	03/03/1994	Rao Rakkam	VJAJALAKSHMI	Praveen@gmail.com	Married	7472346923847387	NHJHJ9245V	8012000000	Plot No. 160, 1 st Cross Street, Nagappa Nagar,	Madurai	Tamilnadu	India	600101				Viewed
7	Avinand	10/10/1995	Sharma	VASANTHI	Avinand@gmail.com	Single	9973463793787476	DGHYH343BJ	8012099999	No.10, Soundara Arival Street, Lakshminagar,	Chennai	Tamilnadu	India	600002				Viewed



Home / Recipient Login

Login Here

*Please enter the ID that you get after registration as your password

Email

bai@gmail.com

Password

.....

Login

Click on the box(es) to select the category to get funded.

Select the sub categories in which you need fund

- ☒ Categories
 - ☒ Education
 - ☒ Educate the children
 - ☒ To pay fees
 - ☒ Pay the kindergarten fees
 - ☒ Child with no parents
 - ☐ Child with single parent
 - ☐ Pay the school tuition fees
 - ☐ Pay the exam fees
 - ☐ To buy school uniforms
 - ☐ To buy stationaries
 - ☐ Educate the girls
 - ☐ Educate the adults
 - ☐ Agriculture
 - ☐ Children
 - ☐ Health
 - ☐ Women
 - ☐ Elderly
 - ☐ Physically Challenged People

Submit

Home View Details

Email *

bai@gmail.com

Expected Date for Money Need *

15-01-2021



Required amount of Money *

20000

Click on the box(es) to select the category to get funded.

Select the sub categories in which you need fund

- ☒ Categories



×

Home

Recipient SignUp

Recipient SignIn

Donor SignUp

Donor SignIn

Activate Windows
Go to Settings to activate Windows.

Home / Donor Register

Register Here

First name

Abi

Last name

R

Mobile Number

9876543215

Email

abi@gmail.com|

Register

Reset

Activate Windows
Go to Settings to activate Windows.



[Home](#) / [Donor Login](#)

Donor Login

*Please enter the ID that you get after registration as your password

Email

Password

Login

Donor Card



Donor Registered Successfully!!!



[Home](#) / [Donor Card](#)

Donor Card

To print Donor Certificate with all your personal details after your donation. Please fill the form

Central Board

LOGOUT

User

Activities

RECIPIENT REQUESTS

DONOR REQUESTS

Central Board

[HOME](#) LOGOUT

Donor

Details

Search...

S.no	First Name	Last Name	Date of Birth	Gender	Email ID	Pan Card Number	Contact Number	Address	City	State	Country	Pincode	Image	Pan Card Image	Income Certificate	Status	Donation Chart
1	Abirami	R	01-12-1890	Female	abirami.guru@gmail.com	F4SRGRW4SGR45	7373099998	GnanaGanesa, Plot No 13-A, 4th Street, Rajiv Nagar,	chennai	Tamilnadu	India	600015				Viewed	
2	Aarthi	A	1990-03-21	Female	aarthi@gmail.com	F4SRGRW4SGR45	7402000000	32/1215, Bobbliraja Street, Opp to Corporation Office,	Vellore	Tamilnadu	India	600104				Viewed	
3	Abarna	M	12-05-1997	Female	abarna@gmail.com	F4SRGRW4SGR45	7402099999	Old No. 28/2, New No.59, Soorammai Street	Thanjavur	Tamilnadu	India	600017				Viewed	
4	Thiru	G	10-03-1888	Male	thiru@gmail.com	F4SRGRW4SGR45	7502000000	No 1, Shanti Street, 1st Cross Street,	Salem	Tamilnadu	India	600038				Viewed	
5	Maran	F	02-11-1890	Male	maran@gmail.com	F4SRGRW4SGR45	7502099999	No.5, Durga Heretage, No.30A, 4th Main Road,	Thoothukudi	Tamilnadu	India	600036				Viewed	


PUBLIC BANK

[HOME](#)
[CONTACT](#)

Payment

Account No:
Atm Pin:
OPass:

©Google zone. All right reserved.

History						
Search.. 						
Recipient Requests Recipient's in Education category requests						
ID	Name	Image	Email	Expected date for money need	Required amount of money	Pay
UID_850	Debjyoti		Debjyoti1@gmail.com	2020-04-22	24000 to go	 Lend Now
UID_1918	Debobrata		Debobrata@gmail.com	2020-06-17	48000 to go	 Lend Now
UID_1392	Arindam		Arindam@gmail.com	2020-09-02	36000 to go	 Lend Now

REFERENCES

- [1] H. Zhao, Y. Ge, Q. Liu, G. Wang, E. Chen, and H. Zhang, "P2p lending survey: Platforms, recent advances and prospects," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 6, p. 72, 2017.
- [2] H. Zhao, H. Zhang, Y. Ge, Q. Liu, E. Chen, H. Li, and L. Wu, "Tracking the dynamics in crowdfunding," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 625–634.
- [3] J. Choo, D. Lee, B. Dilkina, H. Zha, and H. Park, "To gather together for a better world: Understanding and leveraging communities in micro-lending recommendation," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 249–260.
- [4] T. Althoff and J. Leskovec, "Donor retention in online crowdfunding communities: A case study of donorschoose. org," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 34–44.
- [5] L. J. Rosenberg and J. A. Czepiel, "A marketing approach for on Knowledge and Data Engineering, vol. 28, no. 4, pp. 901–911, 2016.
- [12] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced data

management: Overview and challenges,” in Proceedings of the 2017 ACM

45

International Conference on Management of Data. ACM, 2017, pp. 1711–1716.

[13] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, “A survey of general-purpose crowdsourcing techniques,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 9, pp. 2246–2266, 2016.

[14] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, “Crowdsourced data management: A survey,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 9, pp. 2296–2319, 2016.

[15] G. Li, J. Wang, Y. Zheng, J. Fan, and M. J. Franklin, Crowdsourced Data Management: Hybrid Human-Machine Data Management. Springer, 2018.

[16] Y. Zheng, R. Cheng, S. Maniu, and L. Mo, “On optimality of jury selection in crowdsourcing,” in Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015. Open-Proceedings. org., 2015.

[17] Y. Zheng, G. Li, and R. Cheng, “Docs: a domain-aware crowdsourcing system using knowledge bases,” Proceedings of the VLDB Endowment, vol. 10, no. 4, pp. 361–372, 2016.

[18] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" *Proceedings of the VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.

46

[19] H. Hu, Y. Zheng, Z. Bao, G. Li, J. Feng, and R. Cheng, "Crowdsourced poi labelling: Location-aware result inference and task assignment," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 61–72.

[20] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng, "Qasca: A qualityaware task assignment system for crowdsourcing applications," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. ACM, 2015, pp. 1031–1046.

[21] C.-T. Lu, S. Xie, X. Kong, and P. S. Yu, "Inferring the impacts of social media on crowdfunding," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 573–582.

[22] H. Zhang, H. Zhao, Q. Liu, T. Xu, E. Chen, and X. Huang, "Finding potential lenders in p2p lending: A hybrid random walk approach," *Information Sciences*, vol. 432, pp. 376–391, 2018.

[23] H. Zhao, Q. Liu, G. Wang, Y. Ge, and E. Chen, "Portfolio selections in p2p lending: A multi-objective perspective," in *Proceedings of the*

22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016, pp. 2075–2084.

[24] V. Rakesh, W.-C. Lee, and C. K. Reddy, “Probabilistic group recommendation model for crowdfunding domains,” in Proceedings of the Ninth ACM International Conference on Web Search and Data Mining.

47

ACM, 2016, pp. 257–266.

[25] Y. Li, V. Rakesh, and C. K. Reddy, “Project success prediction in crowdfunding environments,” in Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 2016, pp. 247–256.

[26] H. Zhao, Q. Liu, H. Zhu, Y. Ge, E. Chen, Y. Zhu, and J. Du, “A sequential approach to market state modeling and analysis in online p2p lending,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 1, pp. 21–33, 2018.

[27] J. An, D. Quercia, and J. Crowcroft, “Recommending investors for crowdfunding projects,” in Proceedings of the 23rd international conference on World wide web. ACM, 2014, pp. 261–270.

[28] Q. Liu, G. Wang, H. Zhao, C. Liu, T. Xu, and E. Chen, “Enhancing campaign design in crowdfunding: A product supply optimization

perspective,” in Proceedings of the 26th International Joint Conference on Artificial Intelligence, ser. IJCAI’17. AAAI Press, 2017, pp. 695–702.

[29] J. Choo, C. Lee, D. Lee, H. Zha, and H. Park, “Understanding and promoting micro-finance activities in kiva. org,” in Proceedings of the 7th ACM international conference on Web search and data mining. ACM, 2014, pp. 583–592.

48

[30] T. Iwata, K. Saito, and T. Yamada, “Recommendation method for improving customer lifetime value,” IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 9, pp. 1254–1263, 2008.

