

1.1 Introduction:

The project Welfare foundation is a website for donate to the Welfare foundation by donors. The project “welfare foundation” is developed in web base, which mainly focuses on basic operations like displaying the list of all donors, donating money for foundation and using a payment method razor pay for payment. This foundation is work for a poor child to give them teaching and other facilities.

1.2 Overview:

Welfare foundation is a project which aims is developing a computerized system to maintain the donation and increasing online payment method .This project has many features which are generally not available in normal welfare foundation like online donation accepting and maintaining donors. it also has a facility of admin login through which the admin can monitor the whole system.

It has also a facility where donors after logging in can donate for this foundation and can get donation acceptance email by foundation. And admin can login in this and can see list of all donors who are donated for the foundation and how many money they donate this type of information admin can see.

Problem definition:

- Manual process of keeping donors records, donated money, managing donors data is very difficult.
- Donors will have to come to the foundation and then he can donate
- Donating manually by going there and paying.

Proposed solution:

- Online donation.
- It Reduce manual work.
- Can maintain all data of donors easily.
- Using safe and easy payment gateway for donation.

Request Analysis

The request was studied carefully to access what are exactly required. The focus being on the task of determining precisely what originator wants.

The analysis includes as what will be the fields, tables the tools to be used as it is a detailed study of the various operations performed by a system and their relationship within and outside the system. As it points question of what must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis data are collected on the available files, decision points and transaction handled by the present system.

The project is being analysed with every great care possible.

3. Feasibility Study

After doing the project Welfare Foundation, study and analysing all the existing or required functionalities of the system, the next test is to do the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

Economical Feasibility:

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

Technical Feasibility:

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provide in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platform.

Technical feasibility centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed addition. This phase involves financial considerations to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible. Our project is technical feasible as the technologies required are already available software required are VB, MS Access. Hardware requirement are 2GB RAM, 4GB storage Pentium IV processor correct.

The technical issue usually raised during the feasibility stage of the investigation include the followings:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

Are there technical guarantees of accuracy reliability case of access and data security?

Operational Feasibility:

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management form the users?
- Will the system be used and work properly if it is being developed and implemented ?

Will there be any resistance from the user that will undermine the possible application benefits?

Behavioural Feasibility:

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user is likely to have towards the development of a system.

Time Feasibility:

Time feasibility check whether the project can be completed within the time frame suggest by customer. As per this project was assigned for four months duration and also study suggest that this much time is enough to complete the project therefore the project is also feasible as far as time is considered.

4. Requirement Gathering

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of the system engineering are refined by establishing and complete information description, a detailed functional description, a representation of system behaviour, an indication of performance requirement and design constraints appropriate of validation criteria. The introduction of the software requirement specification states the goal and objectives of the software, describing the context of the computer-based system.

RESOURCE REQUIRED

HARDWARE SPECIFICATION

RAM: 1GB or more.

Processor: i3 processor or above.

Hard-Disk: 2GB or more.

System Type: 64-bit Operating System.

SOFTWARE SPECIFICATION

Operating System: Windows family.

Web Technology: PHP, HTML, CSS.

Database:- MySQL5.0.

Server: Wamp Server.

Design Tool: Sublime Text3.

Documentation Tool: Microsoft PowerPoint.

UML Diagram: Star UML.

TECHNOLOGY USED

PHP:

PHP stands for PHP: Hypertext Pre processor. PHP is a server-side scripting language, like ASP. PHP supports many databases (MySQL, Informix, Mysql, Sybase, Solid, PostgreSQL, Generic ODBC, etc.). PHP is an open source software. PHP is free to download and use. PHP sits between your browser and the web server. When you type in the URL of a PHP website in your browser, your browser sends out a request to the web server. PHP is a server-side, cross-platform, HTML-embedded scripting language. There are over half a million domains running PHP and it is freely available for download.

HTML:

HTML stands for Hyper Text Markup Language, which is most widely used language on web to develop web pages. HTML refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a web page is called Hypertext.

CSS:

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

TOOLS USED:

MySQL is an open source RDBMS that relies on SQL for processing the data in database. MySQL provides APIs for the languages like C, C++, JAVA, PHP and Python. Users can quickly and easily create and access their fir in a secure and easy way. MySQL is most commonly used for web applications and for embedded applications and has become a popular alternative to proprietary database system because of its speed and reliability. MySQL can run on UNIX, Windows and Mac OS.

DATA FLOW DIAGRAMS

A data flow diagram is graphical tool used to describe and analyse movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notation Yourdon, gane and sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose . The development of DFD's is done in several level . Each process in lower level diagram can be broken down into a more detailed DFD in the next level . The lop-level diagram is often called context diagram . It consist a single process bit, which plays vital role in studying the current system . The process in the context level diagram is exploded into other process at the first level DFD to understand the process.

Larry the idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst Constantine first developed the DFD as a way of expressing system requirement in a graphical from this lead to the modular design.

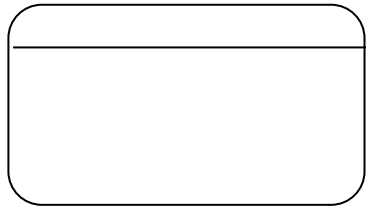
A DFD is also known as a “bubble chart ” has the purpose of clarifying system requirement and identifying major transformation that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consist of a series bubbles joined by data flow in the system.

DFD SYMBOLS:

In the DFD, there are four symbols:

1. A square defines a source or destination of system data.

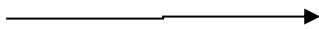
2. An arrow identifies data flow. It is the pipeline through which the information flows.
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data.



Forms data flow



Source or destination of



Data flow

Data Flow

1. Data flow has only direction of flow between symbols . It may flow in both directions between a process and a data store to show read before an update .The latter is usually indicated however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. Data flow cannot go directly back to the same process it leads . There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4. A Data flow to a data store means update (delete or change).
5. A data flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Flow charts are easy to understand diagram showing how steps in a process fit together . This makes them useful tools for communicating how processes work , and for clearly documenting how a particular job is done. Furthermore , the act of mapping a process out in flow chart format helps you cleaify your understanding of the process , and helps you think about where the process can be improved. A flow chat therefore be used to:

- Define and analyse processes.
- Build a step by step picture of the process for analysis , discussion, or communication .
- Define standardize or find areas for improvement in a process.

How to use the tool:

Most flow charts are made up of three main types of symbol:

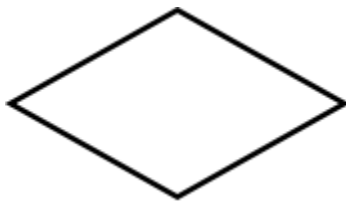
- Elongated circles, which signify the start or end of a process.



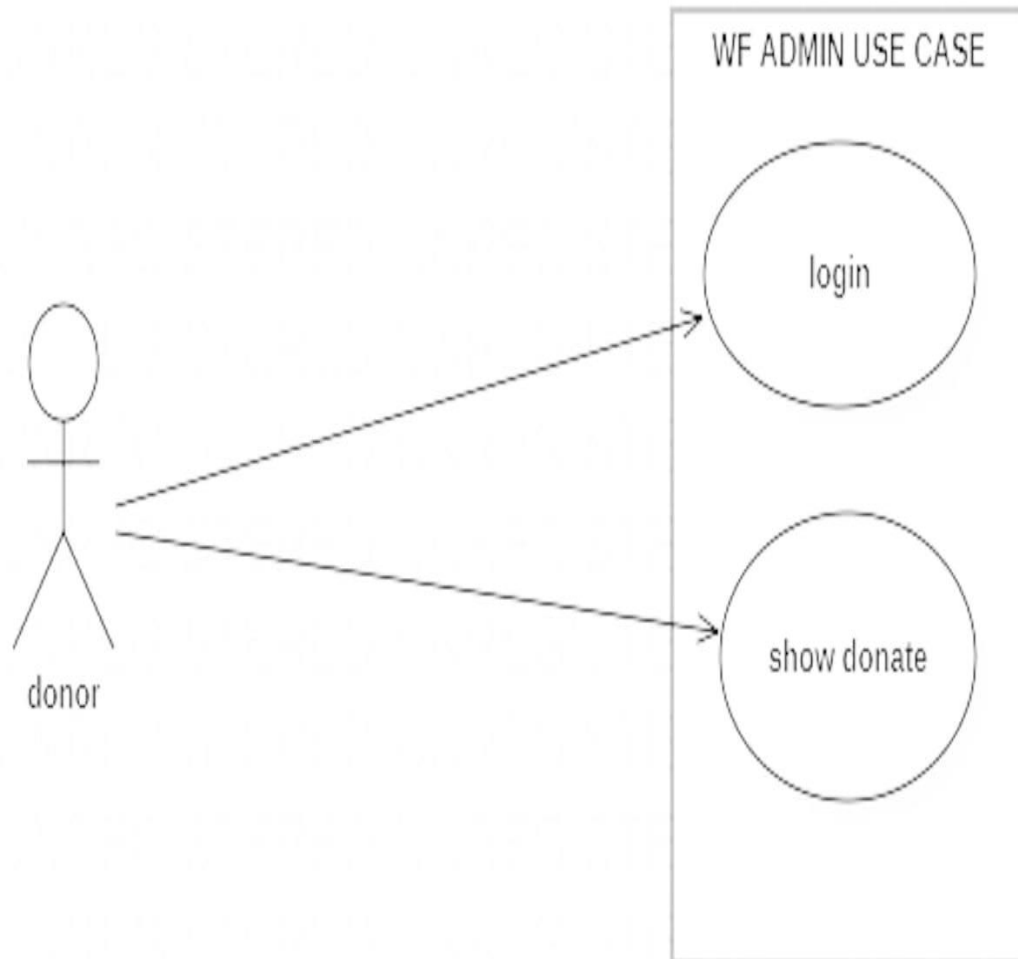
- Rectangles , which show instructions or actions



- Diamonds , which show decision that must be made

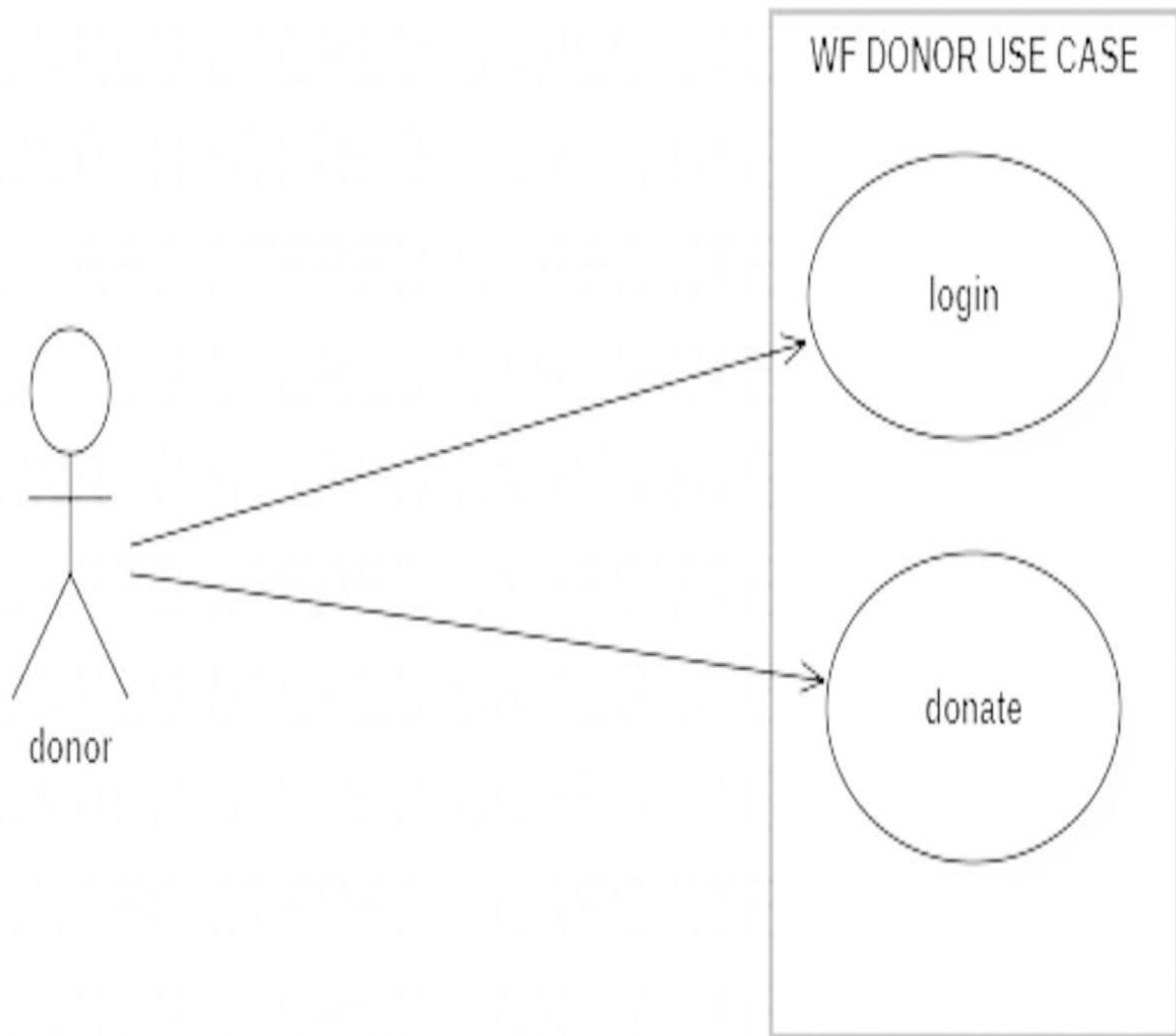


Use Case Model (Admin):



Use case diagram (admin)

Use Case Model (User):



Use case diagram (user)

Use Case diagram description:

A use case diagram at its simplest is a representation of user interaction with the system and depicting the specifications of use case. A use case diagram can portray the different type of user of a system and the various ways that they interact with the system.

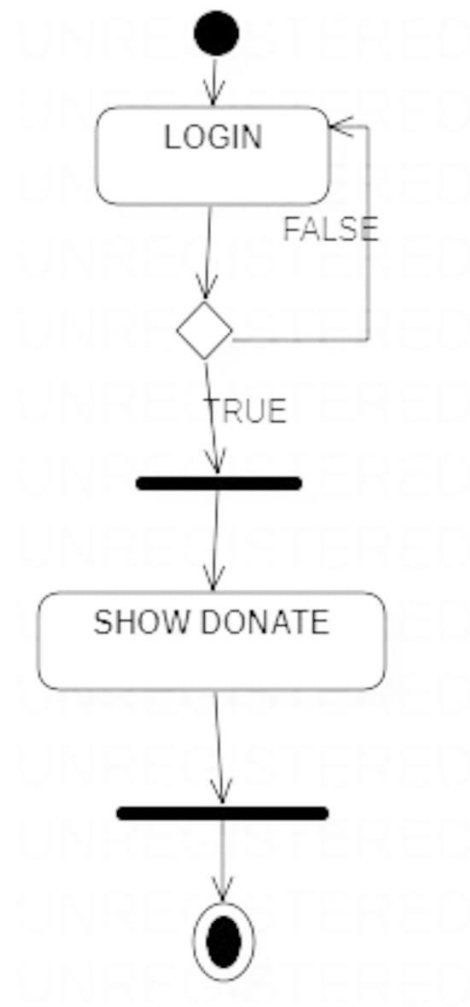
The type of diagram is typically used in conjunction with the textual use case and will often to be accomplished by other type of diagrams as well.

With regards to use case diagrams, that are exactly what they are meant to do, while a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that “Use case diagrams are the blueprints for the system”. They provide the simplified and graphical representation of what the system must actually do.

- Always structure and organize the use case diagram from the perspective of the actor.
- Use case should start off simple and at the highest view possible. Only they can be refined and detailed further.
- Use case diagrams are based upon functionality and thus should focus on the “WHAT” and not the “HOW”.

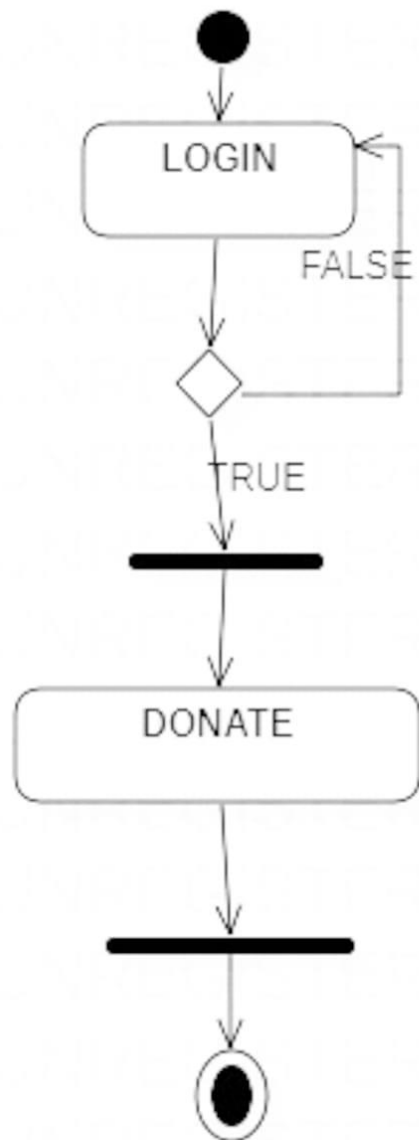


Activity Diagram (Admin):



Activity diagram (admin)

Activity Diagram (User):



Activity Diagram (User):

Activity diagram description:

Activity diagrams are graphical representations of workflows of stepwise activity and actions with support for choice, iteration and concurrency. In the unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow control. The most important shape types:

- Rounded rectangles represent actions.
- Diamonds represent decisions.
- Bars represent the start (split) or end (join) of concurrent activities.
- The black circle represents the start (initial state) of the workflow.
- An encircled black circle represents the end (final state).

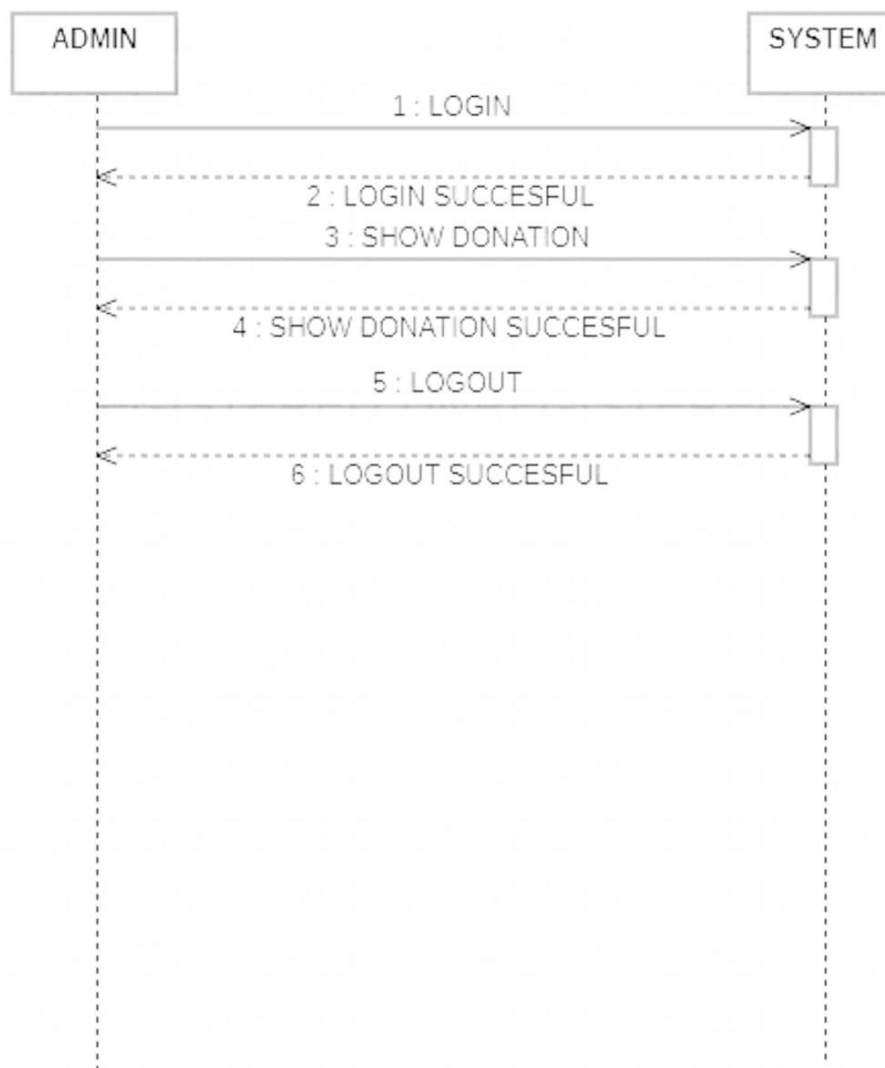
Arrows run from the start towards the end represent the order in which activities happen.

Activity diagrams may be regarded as a form of flow chart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

It was a specialized form of state diagrams, and it was renormalized to be based on Petri net-like semantics, increasing the scope of situations that can be modeled using activity diagrams.

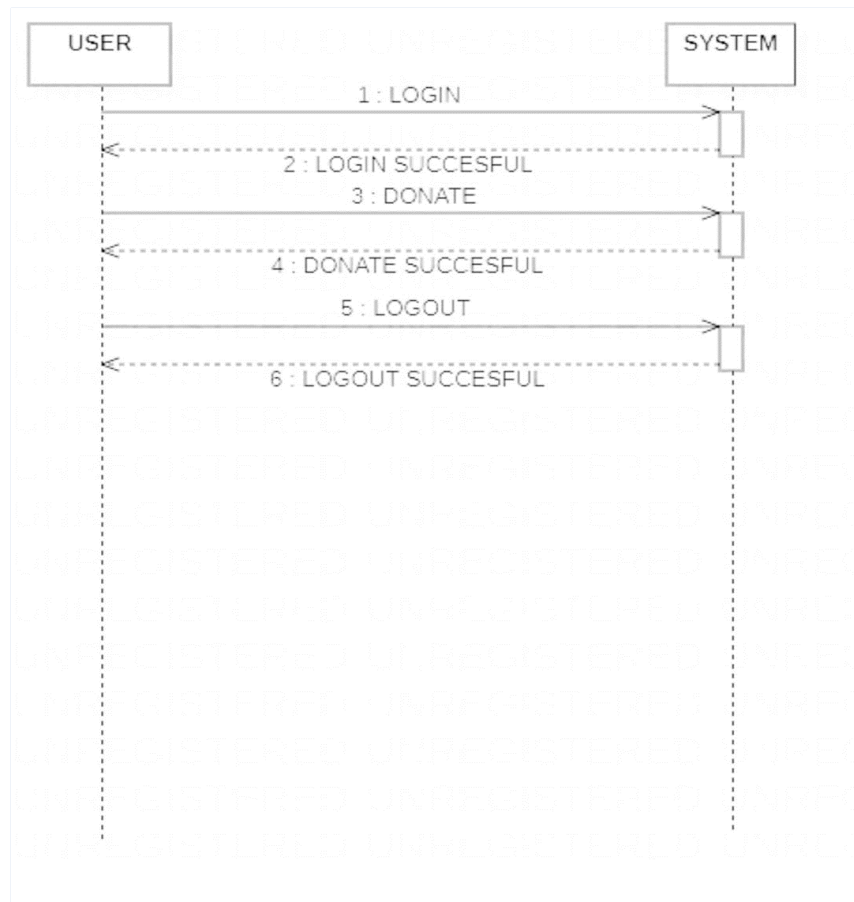
UML activity diagrams can be used in various domains, i.e. in design of embedded systems, it is possible to verify such a specification using model checking technique.

Sequence Diagram (Admin):



Sequence diagram (admin)

Sequence Diagram (User):



Sequence Diagram (User):

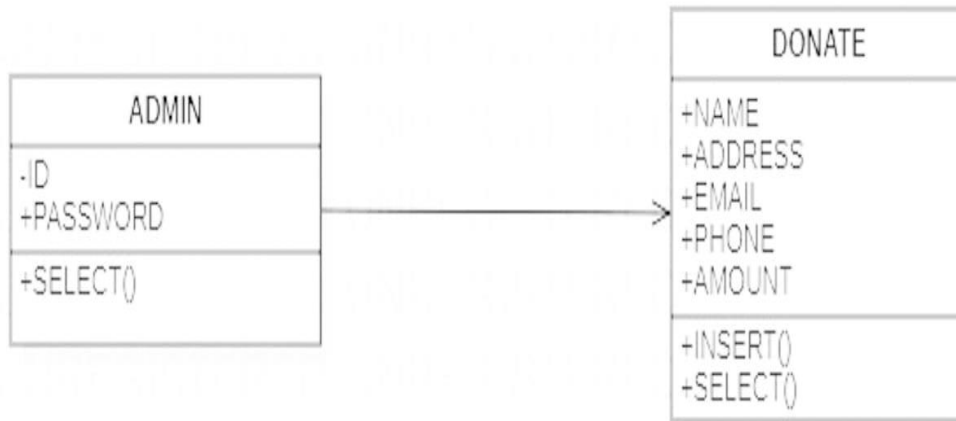
Sequence diagram description:

Sequence diagrams descriptions among classes. These interactions are modeled as exchange of messages. These diagram focus on classes and the messages they exchange to accomplish some described behavior. Sequence diagrams type of interaction diagrams.

- Class roles, which represent roles that objects may play within the interaction.
- Lifelines, which represent the existence of an object over a period of time.
- Activations, which represent the time during which an object is performing an operation.
- Messages, which represent communication between objects.

A sequence diagram shows the participants in an interaction and the sequence of message among them. A sequence diagram shows the interaction of a system with its actor.

Class Diagram

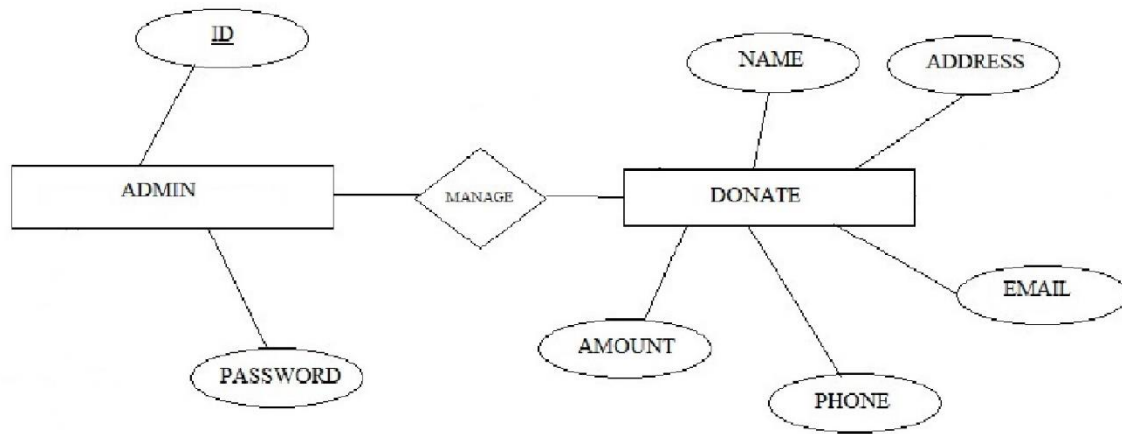


Class Diagram

Class diagram description:

In software engineering a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object orientated modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

ER Diagram



ER diagram

ER diagram description:

In 1976, chain developed the entity relationship (ER) model, a high-level data model that is useful in a developing in a conceptual design for a database, creation of an ER Diagram, which is one of the first steps in designing a database helps the designer to understand and to specify the desired components. An ER model is a diagram containing entities or “items”, relationships among them, and attributes of the entities and relationships. The entity relationship model a high level data model that is useful in developing a conceptual design for a database. The overall logical expressed graphical by an ER-diagram. It consists a set of basic objects entities and of relationship among these objects. An entity is represented by the set of attributes. Attributes are descriptive properties possessed by each member of an entity set. The ER diagram consists of the following major components.

The entity relationship diagram is graphical representation of the database logic and includes a detailed description of all entities, relationship and constraints. The ER diagram is based on perception of a real world that consists of a set of basic object called entities and of relationship among these objects. It is pictorial representation; it is easy to understand such a diagram of following main component-

- Rectangles represent entity set.
- Ellipses represent relationship set.
- Diamonds represent relationship set.
- Double ellipses represent multivalve attribute.

Front Page

```
<?php
?>

<html>

<head>

<style>

body{

background-image:url('im1.jpg');

background-position:center;

background-size:cover;

}

</style>

</head>

<body>

<h1 style="position:absolute;top:130px;left:550px;width:400;font-
family:'algerian'"><b/>welcome to walfare foundation</h1>

<a href="admin login.php"><input type="button"
style="position:absolute;top:300px;left:500px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:yellow" value="LOGIN AS
Admin"></a>

<a href="index.php"><input type="button"
style="position:absolute;top:420px;left:500px;font-family:'algerian';font-
size:20px;height:50px;width:400px;background-color:green" value="donate for
foundation"></a>

</body>

</html>
```


Admin Login.php

```
<html>

<head>

<style>

body{

background-image:url("im1.jpg");

background-position:center;

background-size:cover;

}

</style>

</head>

<form method="POST">

<h1 style="position:absolute;top:130px;left:540px;font-family:'algerian';color:blue">LOGIN FORM FOR ADMIN</h1>

<h1 style="position:absolute;top:190px;color:green;left:500px">USER NAME</h1>

<input type="text" name="username" placeholder="enter your user name here"style="position:absolute;top:260px;left:500px;height:35px;width:400px;border-width:2px" required>

<h1 style="position:absolute;top:280px;color:green;left:500px">PASSWORD</h1>

<input type="password" name="pwd" placeholder="enter your password" style="border-width:2px;position:absolute;top:345px;left:500px;height:35px;width:400px">

<input type="submit" value="LOGIN" style="position:absolute;top:410px;left:500px;height:40px;width:400px;color:white;background-color:blue;font-size:20px;font-family:algerian" required>

</form>

<body>

</body>
```

```

</html>

<?php
include("conn.php");

$un=$_POST['username'];

$password=$_POST['pwd'];

if(empty($password) and empty($un))
{
    echo"<h3
style='color:red;position:absolute;left:500px;top:450px'><u>\".\"PLESAE      ENTER
USERNAME AND PASSWORD\".\"</u></h3>";

}

else
{
    $query='select * from admin where username='$un' and password='$password';'

    $q=mysqli_query($conn,$query);

    $num=mysqli_num_rows($q);

    if($num==1)
    {
        header('location:admin.php');
    }

    else
    {
        echo"<h3
style='color:red;position:absolute;left:500px;top:450px'><u>\".\"PLEASE      ENTER
CORRECT USERNAME AND PASSWORD\".\"</u></h3>";

    }

}

```

?>

Admin.php

<html>

<head>

<style>

body{

background-image:url("im1.jpg");

background-position:center;

background-size:cover;

}

table{

position:absolute;

top:250;

left:550;

}

</style>

</head>

<body>

<h1 style="position:absolute;top:120;left:520px;font-family:algerian;color:red;width:380px"><u>here is the list of all doners</u></h1></body></html>

<?php

include("conn.php");

\$q1="select name,email,amount from donate;";

```

$query1=mysqli_query($conn,$q1);

echo"<center/><table
border='1'><tr><th>name</th><th>email</th><th>amount</th></tr>";

while($row=mysqli_fetch_array($query1))
{

echo "<tr><td>".$row['name']. "</td> <br/><td>";

echo $row['email']. "</td><br/><td>";

echo $row['amount']. "</td></tr>";

}

echo"</table>";

?>

index.php

<?php

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>WelFare Foundation</title>


<!-- WEB FONTS -->

```

```
<link
href='http://fonts.googleapis.com/css?family=Roboto:100,300,100italic,400,300italic'
rel='stylesheet' type='text/css'>
```

```
<link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300'
rel='stylesheet' type='text/css'>
```

```
<!-- BOOTSTRAP -->
```

```
<link href="css/bootstrap.min.css" rel="stylesheet">
```

```
<!-- FONT AWESOME -->
```

```
<link rel="stylesheet" href="css/font-awesome.min.css">
```

```
<!-- CUSTOM STYLING -->
```

```
<link href="css/styles.css" rel="stylesheet">
```

```
</head>
```

```
<style type="text/css">
```

```
.h1 {
```

```
font-size: 100px;
```

```
color:white;
```

```
font-weight: bold;
```

```
font-family: Brush Script MT;

}
```

```
</style>
```

```
<body>
```

```
<!-- HEADER -->
```

```
<section id="header">
```

```
<div class="container">
```

```
<div class="row text-center">
```

```
<div class="col-md-8">
```

```
<p class="h1">Welfare Foundation</p>
```

```
<h1 class="white-text">Donate for  
Cause</h1>
```

```
<h5 class="white-text">Your generosity gives  
children a smile, support our work!</h5>
```

```
<div id="countdown" class="white-  
text"></div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="animation">
```

```
<a class="arrow-down-animation" data-scroll href="#about"><i  
class="fa fa-angle-down"></i></a>
```

```
</div>
```

```

</section>

<!-- HEADER ENDS -->

<!-- ABOUT -->

<section id="about">

    <div class="container">

        <div class="row text-center">

            <!-- Section 1 -->

            <div data-sr="enter top over 1s, wait 0.3s, move 24px,
reset" class="col-md-4">

                <i class="fa fa-book"></i>

                <h3>Child Education</h3>

                <div class="colored-line-small-
center"></div>

                <p>According to recent estimates – 25% of
children that are of age to attend primary school in rural zones are not educated versus
16% of children of the same age living in urban areas.</p>

            </div>

            <!-- Section 2 -->

            <div data-sr="enter bottom over 1s, wait 0.3s, move
24px, reset" class="col-md-4 middle">

                <i class="fa fa-heart-o"></i>

                <h3>Health Demographic</h3>

                <div class="colored-line-small-
center"></div>

                <p>Despite these inadequacies, the health of
rural children appears to be deteriorating with regard to crime, substance abuse, and
infection with the human immunodeficiency virus or AIDS.</p>

```

```

</div>

<!-- Section 3 -->

<div data-sr="enter top over 1s, wait 0.3s, move 24px,
reset" class="col-md-4">

    <i class="fa fa-lemon-o"></i>

    <h3>Rural Eating Habits</h3>

    <div class="colored-line-small-
center"></div>

    <p>The proper lifestyle of a child includes
proper eating habits should be monitored properly in the rural areas where economically,
socially and educationally backward for physical developments.</p>

</div>

</div>

</div>

</section>

<!-- ABOUT ENDS -->

<!-- SUBSCRIBE -->

<section id="subscribe">

    <div class="container">

        <i data-sr="ease-in over 1s, reset" class="fa fa-money
icontop"></i>

        <div class="row text-center">

            <div class="col-md-8 col-md-offset-2">

                <h2 data-sr="enter top over 1s, wait 0.3s,
move 24px, reset" class="white-text"><b>Donate Now !!</b></h2>

                <h5 data-sr="enter bottom over 1s, wait 0.3s,
move 24px, reset" class="white-text">Help our organization with sharing the page.</h5>

```



```

<div><a href='http://localhost/Welfare-
Foundation-main/signup.php'><button>donate</button></a>

<script>

(function(){

var d=document; var

x=!d.getElementById('razorpay-embed-btn-js')

if(x){ var

s=d.createElement('script'); s.defer=!0;s.id='razorpay-embed-btn-js';

s.src='https://cdn.razorpay.com/static/embed_btn/bundle.js';d.body.appendChil
d(s);} else{var rzp=window['__rzp__'];

rzp && rzp.init &&

rzp.init()}})();

</script>

</div>

</div>

</div>

</div>

</section>

<!-- SUBSCRIBE ENDS -->

<!-- CONTACT US -->

<section id="contact">

<div class="container">

<div class="row text-center">

<div class="col-md-8 col-md-offset-2">

<!-- CONTACT FORM -->

```


</div>

<!-- AFTER SENDING MESSAGE -->

</div>

</div>

</section>

<!-- CONTACT ENDS -->

<!-- FOOTER -->

<footer id="footer">

<div class="container">

<div class="row text-center">

<p>

© 2021 WelFare Foundation | Design By

KIPS

<i class="fa fa-facebook-square"></i>

</p>

</div>

</div>

</footer>

<!-- FOOTER ENDS -->

```
<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>

<!-- Include all compiled plugins (below), or include individual files as
needed -->

<script src="js/bootstrap.min.js"></script>

<!-- SMOOTH SCROLL -->

<script src="js/smooth-scroll.min.js"></script>

<!-- PARALLAX IMG -->

<script src="js/jquery.parallax-1.1.3.js"></script>

<!-- SCROLL REVEAL -->

<script src="js/scrollReveal.min.js"></script>

<!-- FUNCTIONS -->

<script src="js/functions.js"></script>
```

```
</body>
```

```
</html>
```

signup.php

```
<html>
```

```
<head>
```

```
<style>
```

```
body{
```

```
background-image:url('im1.jpg');
```

```
background-position:center;
```

```
background-size:cover;
```

```

}

</style>

</head>

<form method="POST">

<h1
style="position:absolute;top:120px;left:520px;color:black;font-
family:'algerian'">DONATION FORM</h1>

<input type="text" name="name" placeholder="ENTER YOUR NAME"
style="position:absolute;top:220px;left:500px;height:35px;width:215px" required>

<input type="text" name="address" placeholder="ENTER YOUR ADDRESS"
style="position:absolute;top:220px;left:800px;height:35px;width:215px" required>

<input type="text" name="email" placeholder="ENTER YOUR EMAIL"
style="position:absolute;top:290;left:500px;height:35px;width:215px" required>

<input type="text" name="phone" placeholder="ENTER YOUR NUMBER"
style="position:absolute;top:290;left:800px;height:35px;width:215px">

<input type="text" name="amount" placeholder="ENTER PASSWORD"
style="position:absolute;top:360;left:500px;height:35px;width:215px">

<input
type="submit"
value="donate"
style="position:absolute;top:430;left:500px;height:35px;width:520px;color:white;back
ground-color:blue;font-size:20px;font-family:'algerian'">

</form>

<body>

</body>

</html>

<?php

include("conn.php");

$na=$_POST['name'];

$address=$_POST['address'];

$email=$_POST['email'];

$phone=$_POST['phone'];

```

```

$amount=$_POST['amount'];

if(empty($na) or empty($email) or empty($address) or empty($amount))

{

    echo'<h3          style="position:absolute;top:460px;left:480px;color:red;font-
family:algerian"><u><blink>'. "please enter data required data with correct
manner". "</blink></h3>";

}

else

{

    $query="insert into donate values('$na','$address','$email',$phone,$amount)";

    $q=mysqli_query($conn,$query);

    if($q)

    {

        header('location:https://rzp.io/l/XQm6Xxsc');

    }

    else{

        echo'not';

    }

    }

?>

conn.php

<?php

error_reporting(E_ERROR);

$sn="localhost";

$un="root";

```

\$pwd="";

\$db="donation";

\$conn=mysqli_connect(\$sn,\$un,\$pwd,\$db);

?>

5.1 Implementation:

System implementation is the process of making the newly designed systems fully operational. The system is implemented after careful testing. The primary goal of product implementation is development of source code that is easy to read and easy to understand. The term implementation has different meaning, ranging from the conversion of a basic application to a compatible replacement of a computer system. Implementation is used here to mean the process converting a new or a revised system design in to an operational one. During the implementation stage we convert the detailed code in a programming language. Clarity of source code eases debugging, testing and modification of a software product. the difficulties encountered during implementation is successful integration of source code components into a functioning system before a routine can be placed in the evolving system. It may be required that the routine be inspected by an inspection team. Or reviewed or tested to a given level of test coverage.

Testing

Software testing is an important element of s/w quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of s/w as a system element and the costs associated with an s/w failure are motivating forces for well planned, through testing thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing. Testing is a set of activities that can be planned in advance and conducted systematically. Testing is very important stages of a software include unit testing, integration testing and deployment testing.

White box testing:

Is a test case design philosophy that uses the control structure described as part of component-level design to derive test cases.

White box testing is also called glass-box testing or structural testing. It performs close examination of procedural details. They test the software test the software logical path by having test cases exercising specific sets of condition and loops. It examines the program status at various points to determine whether the expected status corresponds to the actual one.

Internal working of the product test can be conducted to ensure that internal operational performs according to specification. And all component have been adequately exercised is called white-box testing.

Using white box testing method it can be ensured that:

- All independent parts within a module have been exercised at least one.
- Exercise all logical decision and their all false sides.
- Exercise all loop at their boundaries and within their operational bound.
- Exercise internal data structure to ensure that validity.

Black box testing:

It is also called functional testing or behavioral testing. It performs tests of software interfaces. They test the operation ability of software functional, acceptance of input and delivery of output and the maintainability of external information integrity.

It is testing without knowledge of the internal working of the item being tested. For example, when black box testing is applied to software engineering, the tester would only know the “legal” inputs and what the expected outputs should be, but now the program actually arrives at those outputs. It is because of this that black box testing can be considered testing with respect to the specifications, no other knowledge of program is necessary.

It examine some of the basic aspects of a system, having little regard for the internal logical software structure.

Black box testing is not an alternative to while box technique. It is complementary approach that is likely to uncover a different class than white box methods.

- Black box testing attempts to find error in the categories.
- Incorrect or missing function.
- Interface error.
- Errors in data structure or database access.
- Performance errors.
- Initialization and termination errors.

10. Reports & Table Figure

Admin Table

```
C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe

mysql> desc admin;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| username | varchar(20) | YES |  | NULL |  |
| password | varchar(30) | YES |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.08 sec)

mysql>
```

donate Table

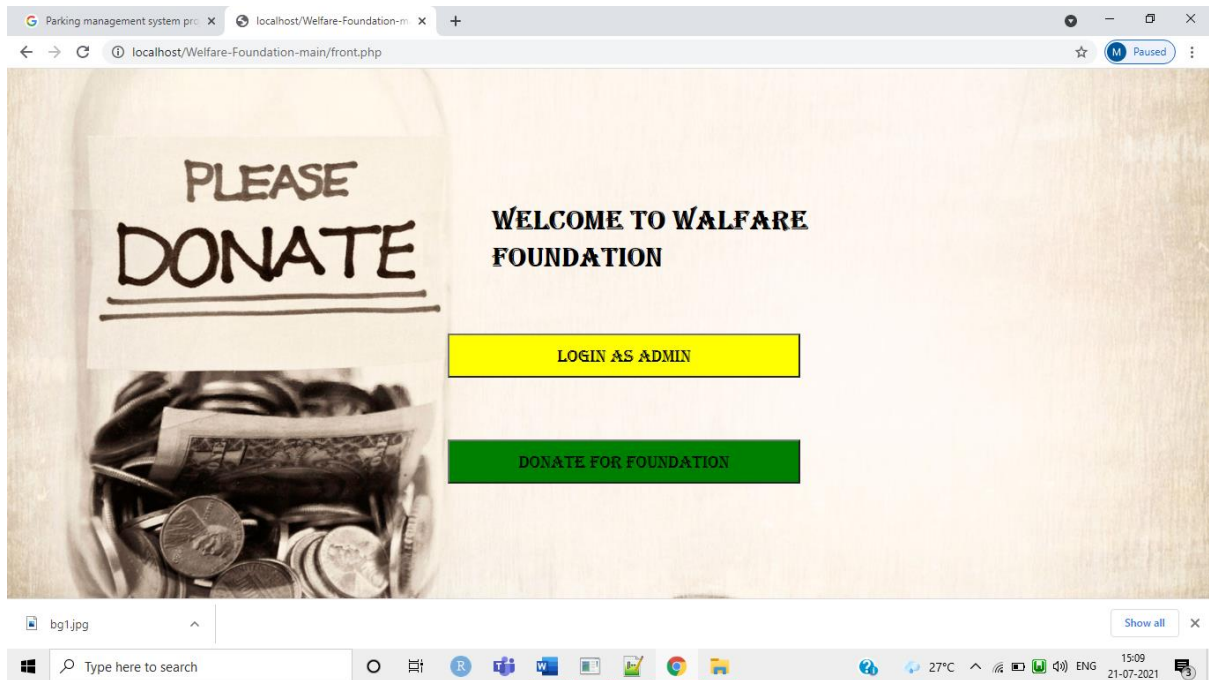
```
C:\wamp\bin\mysql\mysql5.6.12\bin\mysql.exe

mysql> desc donate;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name | varchar(20) | YES |  | NULL |  |
| address | varchar(30) | YES |  | NULL |  |
| email | varchar(30) | YES |  | NULL |  |
| phone | int(11) | YES |  | NULL |  |
| amount | int(20) | YES |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.09 sec)

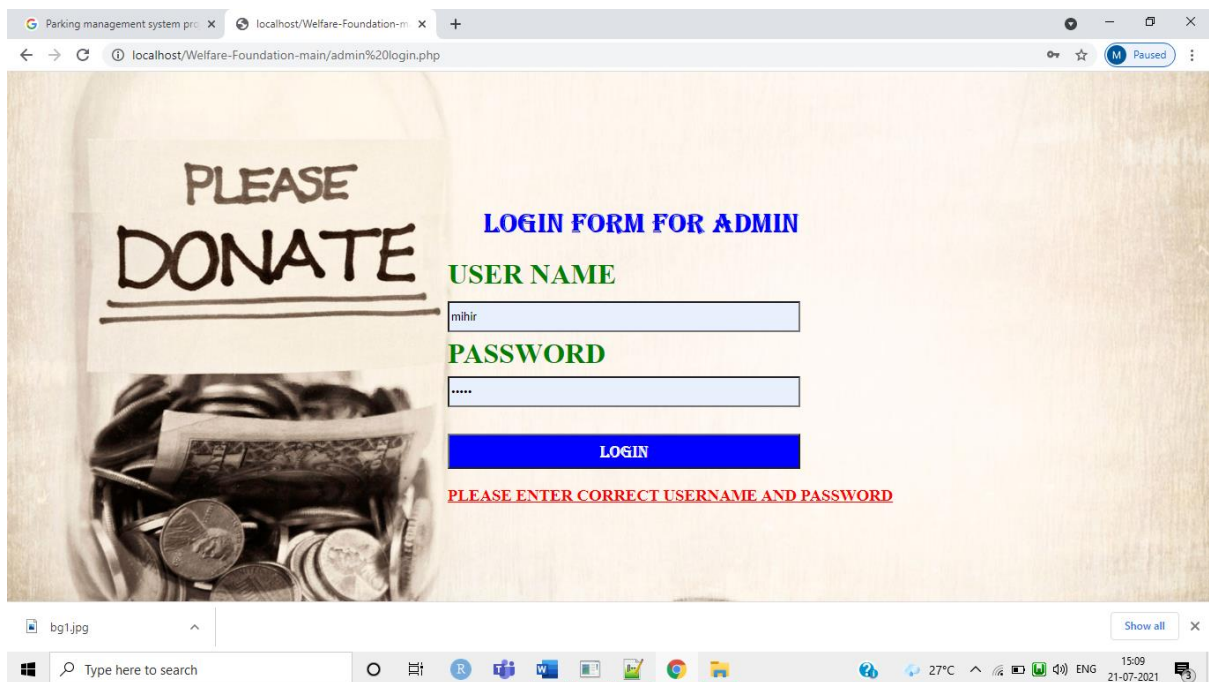
mysql>
```

11. Screenshots of Project

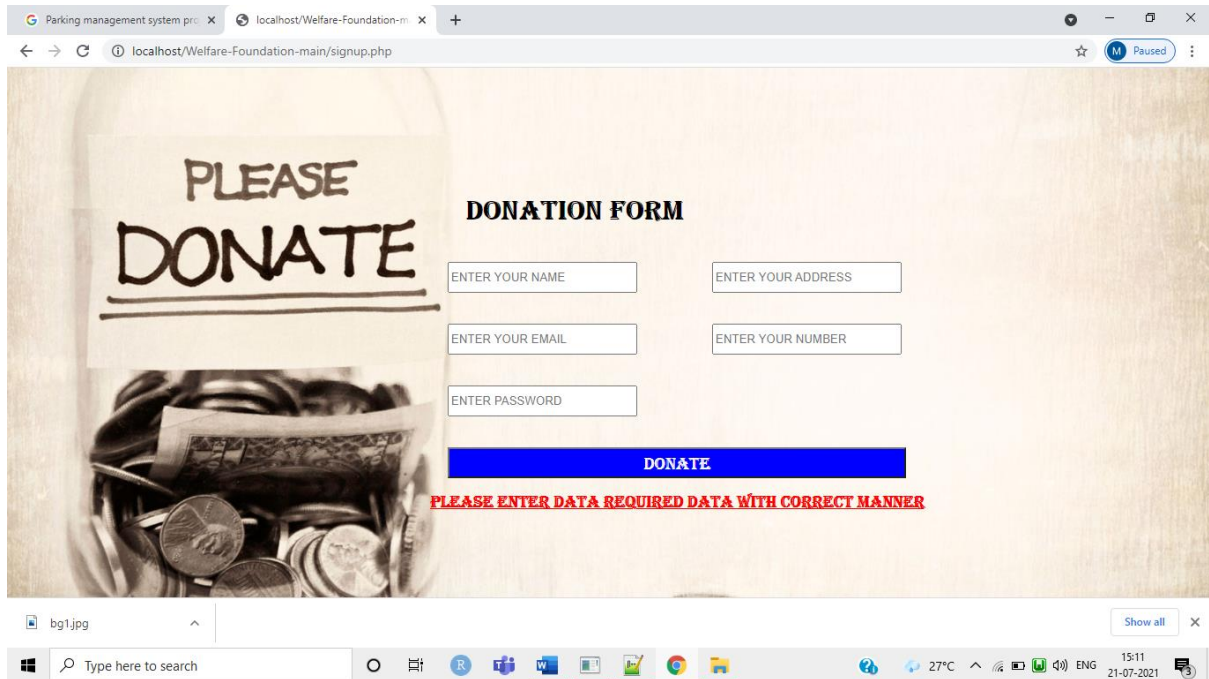
Home Page Page



admin login Page



User signup page



Parking management system pro X localhost/Welfare-Foundation-m X +

localhost/Welfare-Foundation-main/signup.php

PLEASE DONATE

DONATION FORM

ENTER YOUR NAME

ENTER YOUR ADDRESS

ENTER YOUR EMAIL

ENTER YOUR NUMBER

ENTER PASSWORD

DONATE

PLEASE ENTER DATA REQUIRED DATA WITH CORRECT MANNER

bg1.jpg

Show all X

Type here to search

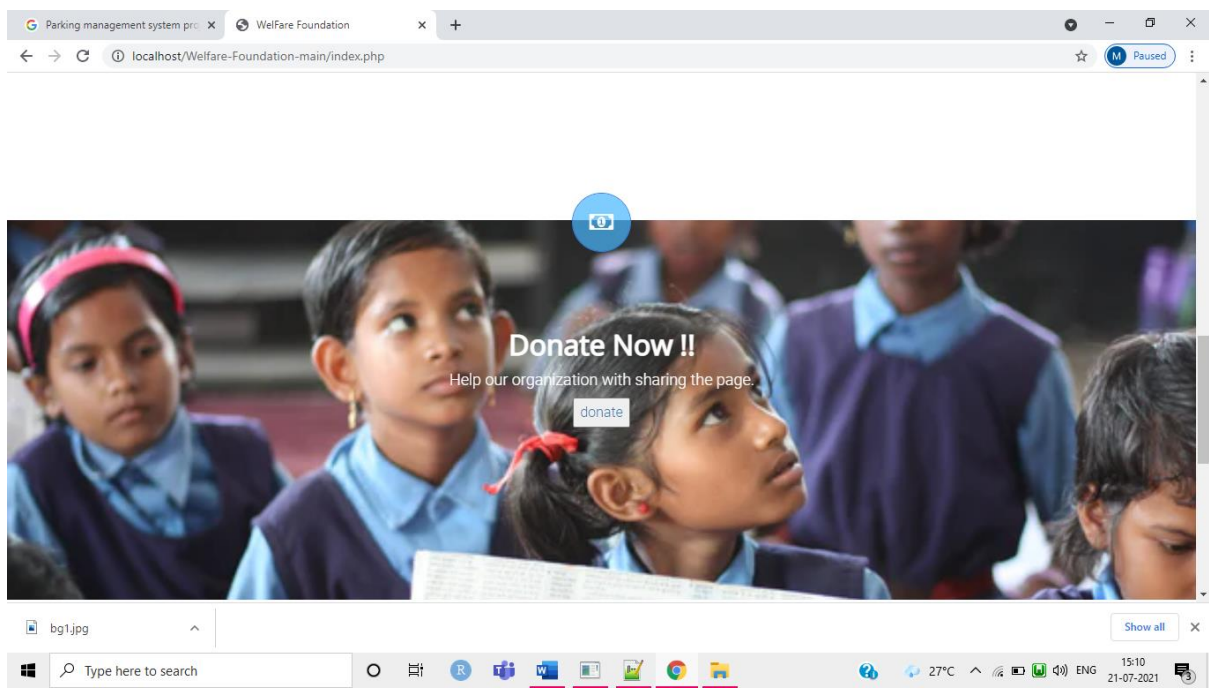
27°C

ENG

15:11

21-07-2021

User work list page



Parking management system pro X Welfare Foundation X +

localhost/Welfare-Foundation-main/index.php

Donate Now !!

Help our organization with sharing the page.

donate

bg1.jpg

Show all X

Type here to search

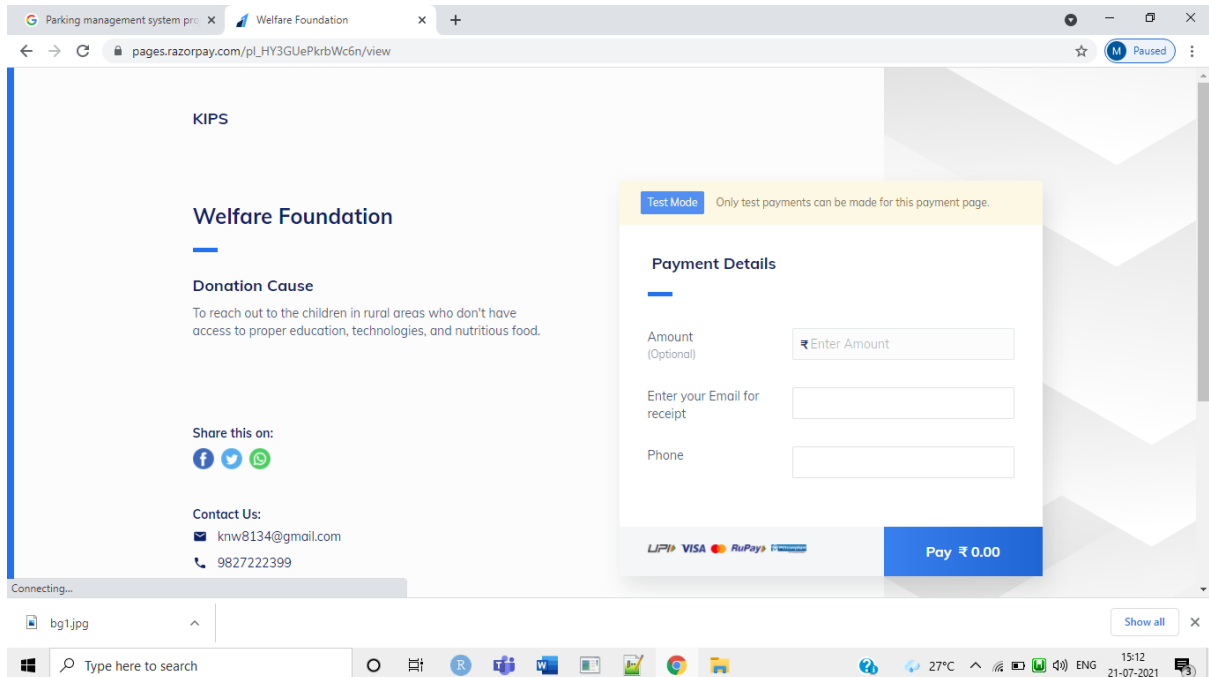
27°C

ENG

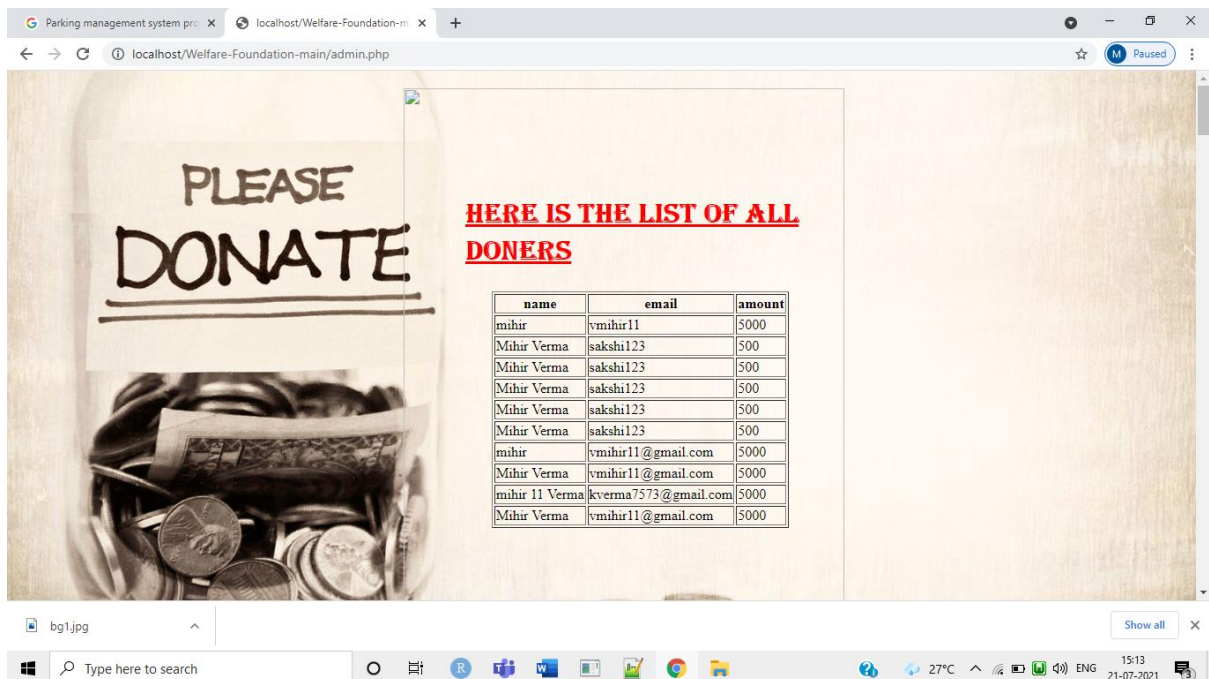
15:10

21-07-2021

Payment gateway page



donors list page



12. Conclusion

The Welfare foundation needs to be computerized to reduce human errors and to increase the efficiency. The welfare foundation in this proposal will be a computerize donation system developed to maintain the donations. Welfare foundation is designed to store donors all data and donating through online payment system.

Overall this project of ours is being developed to help the foundation as well as the best way possible and also reduce the human effort. the main focus of this project is to lessen human effort and encourage efficient record keeping and donating online.

13. Future Scope & Further Enhancement of Project

Main feature of the existing system this system (Welfare foundation) can hold limited number of information and data regarding donators, and dates when they donate. As mentioned earlier this system was a manual system which manages all this information about donors in the paper based book known a register.

14. References

- www.w3school.com
- www.3pdf.com
- www.project.com
- www.ukessay.com
- www.elarninglarning.com