Charity Management System

Table Of content

Introduction

1.1 Problem Statement:-

The majority of NGOs have experienced difficulties in getting funds or other required things. Getting donors is a very hard task and sometimes dealing with some donor's condition can be a big challenge for NGOs fulfils it. And currently there is no system which connects all NGOs at one place. With the conventional media means such as radio, newspaper or television advertisements sometimes, no donors will reach needy NGOs who are really working for poor peoples. The records of the donor might not be kept safely and there might be missing of the donor's record due to human error or disasters and donors do not know whether it's secure or not.

So with the help of our concept, donors can easily and securely donate anything which they want to donate to NGOs. There is information regarding all NGOs and which type of NGOs in our society on one particular portal.

Also, with our concept one can register new NGOs with necessary government documents which can help donors. We develop a like recent activities type module in that we include all current care funds events.

1.2 Project Scope:-

As we are well- aware of your tight schedules and hectic work atmosphere, System is designed as user-friendly software without adding up further complexities to the operations of your entity. It comes with a fool proof and simple user interface which facilitates easy adoption and smooth functioning of your charity operations. Hence, your organizational members will find System a useful and productive than the outdated manual systems.

Our system is based on relational database with its charity/NGOs management and donations from donors. Our project supports thousands of NGOs and donors around the world. From that we hope to provide a better and reliable experience to ***charity/NGOs, donors and donee (a person who Recipient of a donation or gift)*** *who is main beneficiary of our project.*

1.3 Objective:-

The main objective of the document is to illustrate the requirement of the online charity management system. This gives us a complete multi-locational, multi channel, donation collection and management system for charitable organization.

- *Main beneficiaries are charity/NGOs, donors and donee.*
- *Our main goal is to connect donors who really want to donate and are not aware about NGOs.*
- *Also donors can easily know where their donation goes and NGOs really works for the needy peoples.*
- *In the current situation go out there without necessary work is not safe so that donors can easily donate from home without any difficulty.*

- *“Donate from HOME” is the idea of our project by which donors can donate in those NGOs which are located at different places and facing disaster.*

*So we hand over this problem and decide to make a project called “**Charity Management System**”. Which will cover all problem solutions as discussed above. And our main objective is that more and more charities/NGOs and donors can join and take advantage of this in the near future.*

External interface requirements

2.1 Non-Functional Requirement

Security and Privacy Requirement:

- *Privacy:*
    - *The donation amount can only be seen by the user itself.*

- *Payment:*
    - *The website will not ask users to provide their bank information users can directly donate to respective NGOs via their payment gateway.*

Availability and Portability:

- *This website can be available for all internet users and devices only if they have a valid version of the browser.*

- *This website is portable. So moving from one OS to another OS does not create any problem.*

Performance:

- *At a time many users can browse through the site and for donation separate email will be sent to registered email address and contact number.*

- *At any emergency situation like natural disaster or pandemic there will be specific pages for the different working NGOs to handle the load at one particular page.*

Reliability:

- *For the payment, system generated email will be sent to users to provide many different and secure methods to donate online and the system will continuously update for better security.*

2.2 Functional Requirements

Authentication of Admin:

- Admin can login into the system using username and password.
- Admin can change his/her password he/she forgot.

Authentication of Donor:

- Any new donor can register himself into system.
- A Verification email is sent to the donor's provided recovery email ID.

- Donor will be registered into system by admin.
- After then donor can login using username and password.
- Donor can change his/her password he/she forgot.
- Donor can change personal details and contact details at anytime.

Authentication of NGO:

- NGO can be register by the owner of the NGO.
- A Verification email is sent to the NGO's provided recovery email ID.

- NGO will be registered into system by admin.
- Owner can login into system using username and password.
- Donor can change his/her password he/she forgot.
- Donor can also change the details of the NGO at anytime.

Other Authentication:

- Only the Admin can add or change the description of NGOs, nobody can directly add any fake NGO.
- The unauthenticated user will only be allowed to browse through the site and can see the services provided by the site and NGOs like trending, not allowed to see any contact details of NGOs.
- The security will be based on password.
- There will be a recovery email to the user if the user forgot the password.
- After many unsuccessful attempts the password account will be blocked and the recovery email sent to the registered email address.

2.3 Hardware

- *Web applications do not have any designated hardware, it does not have any direct hardware interface. The physical GPS is managed by the GPS application in the device and database server is managed by the underlying operating system and web server. Payment method is also managed by the underlying operating system; it is also independent of hardware.*

2.4 Software

- *Our system is a web application built using HTML, CSS, bootstrap in front-end and PHP in backend. We will also use*
- *An external payment gateway for donation. We have used MySQL for database management.*

Software Process Model

3.1 Comparison study diff. SDLC Models

Classical waterfall model:-

- It is ideal model
- This model cannot be used in practical project devlopment, since this model does not support any mechanism to correct errors that are committed during any of the phase but detected at a later phase

- This problem is overcome by iterative waterfall model through the inclusion of feedback paths.

Iterative waterfall model:-

- *This model is simple to use and understand*
- *But this model is suitable only for well-understood problems and is not suitable for the development of very large projects and for high risk projects.*

Prototype model:-

- *The Prototyping model is suitable for projects, which either the customer requirements or the technical solutions are not well understood.*
- *In this risks must be identified before the project starts.*
- *This model is especially popular for the development of the user interface part of the project.*

Concurrent development model:-

- *All activities exist concurrently but reside in different states.*
- *Each activities, actions and tasks on the network exists simultaneously with other activities, actions and tasks.*
- *This model requires excellent and current communication between all teams, which is sometimes not easily feasible.*

Incremental model:-

- *The incremental model has same phases that are in waterfall method but it is iterative in nature*
- *Rather than deliver the system as single delivery, the development and delivery is broken down into increments with each increments delivering part of the required functionality*
- *But this model needs a clear and complete definition of the whole system before it can be broken down and built incrementally.*

Summary table:-

Properties of Model	Water-Fall Model	Incremental Model	Spiral Model	Rad Model
Planning in early stage	YES	YES	YES	YES
Returning to an earlier phase	NO	YES	YES	YES
Handle Large-Project	Not Appropriate	Not Appropriate	Appropriate	Not Appropriate
Detailed Documentatio n	Necessary	Yes but not much	Yes	Limited
Cost	Low	Low	Expensive	Low
Flexibility to change	Difficult	Easy	Easy	Easy
User Involvement	Only at beginning	Intermediate	High	Only at the beginning
Maintenance	Least Maintainable	Maintainable	Typical	Easily Maintained
Risk Involvement	High	Low	Medium to high risk	Low
Testing	After completion of coding phase	After every iteration	At the end of the engineering phase	After completion of coding

Team size	Large Team	Not Large Team	Large Team	Small Team
Re-usability	Least possible	To some extent	To some extent	Yes

3.2 Why we choose Incremental model?

- *Initially the project would be of a relatively small scale however after it also work for a large scale application. Therefore, we will be implementing an incremental model for development.*
- *And also we divide our modules by priority like first of we have to implement the admin module which is for administration. After it NGO or donor module and at last payment or trending events module.*
- *And also there is human mind behind this project also there require some changes or occurs error during testing. So this model supports changing requirements, also testing and debugging is made simpler during small iterations.*
- *And also we can make phases like analysis, design, code and test which in incremental model*
- *In analysis we analyses the requirements for that module like what features we have to add on that module*
- *In design phase we view it by relational database point of view*
- *After it we done code section by making front end and back end.*
- *And then test it*
- *This cycle repeat for all increments or iterations.*