

Software Requirements Specification

Tool for Non-Profit Organization

Version: 1.0

Date: 2-10-2019

Team No : D6

Sl.No	Name	USN#
1	MEGHANA	01FB16ECS204
2	P ABHISHIKTA SAI	01FB16ECS242
3	MANJUNATH BHADRANNAVAR	01FB16ECS195
4	MAYANK THAKUR	01FB16ECS200
5	MUDIT JHA	01FB16ECS214
6	MANOJ H C	01FB16ECS197
7	NIKITA UPPALADINNI	01FB16ECS231
8	PANDURANG	01FB16ECS244
9	NIREEKSHA	01FB16ECS721
10	MANTHENA SIVA SAI VARMA	01FB16ECS198

Revision History

Date	Description	Author	Approver	Comments
2-10-2019	Initial Version	<Your Name>	<Approver's Name>	<First Revision>

Table of Contents

REVISION HISTORY	I
DOCUMENT APPROVAL	I
1. INTRODUCTION	1
1.1 PURPOSE	1
1.2 SCOPE	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	1
2. GENERAL DESCRIPTION	1
2.1 PRODUCT PERSPECTIVE	2
2.2 PRODUCT FEATURES	3
2.3 USER CHARACTERISTICS	4
2.4 OPERATING ENVIRONMENT	4
2.5 GENERAL CONSTRAINTS	5
2.6 ASSUMPTIONS AND DEPENDENCIES	5
2.7 DOCUMENTATION	5
3. SPECIFIC REQUIREMENTS	5
3.1 FUNCTIONAL REQUIREMENTS	6
3.2 NON-FUNCTIONAL REQUIREMENTS	6
3.3 EXTERNAL INTERFACE REQUIREMENTS	7
3.3.1 <i>User Interfaces</i>	7
3.3.2 <i>Hardware and Software in terms of how they would interact or how they would be executed</i>	7
3.3.3 <i>Software Interfaces</i>	7
4 APPENDIX A: GLOSSARY	7

1. Introduction

1.1 Purpose and Intended Audience

The purpose of this document is to list out all the requirements for the task of building a website for a non-profit organization along with all the specific functionalities that they require. This document serves to lay out a guideline to follow for the developers by providing a high-level view as well as ensure that all the relevant stakeholders are aware of the scope of the project.

This website which is to be built for an anonymous non-profit organization aims to provide tools which automate and ease the various tasks that would otherwise be done manually. The product to be developed should include an online portal for collecting donations, providing updates on volunteering activities, an administrative side where one can manage expenditure, dashboard among other features, all to be added while adhering to a budget constraint.

The Intended Audience is non-profit management and users who access the website.

Well Written!

1.2 Scope

The web portal for a non-profit organization is a web application which aims to automate the redundant and repetitive tasks that those in-charge have to undertake, provide a convenient method to collect donations from the public all while being intuitive and easy to use. It is to be designed while keeping the security of the organization's data in mind.

The tasks that this product aims to fulfill revolve around making the management and day-to-day running of the non-profit organization easier for those in charge. Among the automation tasks that it performs is an automatic receipt generator which eliminates the need for manual entry, a secure payment portal to accept donations, and some form of an expense management system to keep track of the inflow and outflow of money. The overall administration is to be made easier by including an admin site which includes a dashboard for keeping up-to-date with the various activities, updating the volunteering opportunities and general upkeep of the site's contents for the public.

Well written in terms of what will be done as a part of this project.

Not listed out what will not be done and will not be considered in the project.

1.3 Definitions, Acronyms, and Abbreviations

None.

Defining any kind of acronyms like NPO could be added here

2. General Description

2.1 Product Perspective

NPO software tool is a new system that replaces the current physical transaction model deployed in the organization. The context diagram in Figure 1 illustrates the external entities and system interfaces.

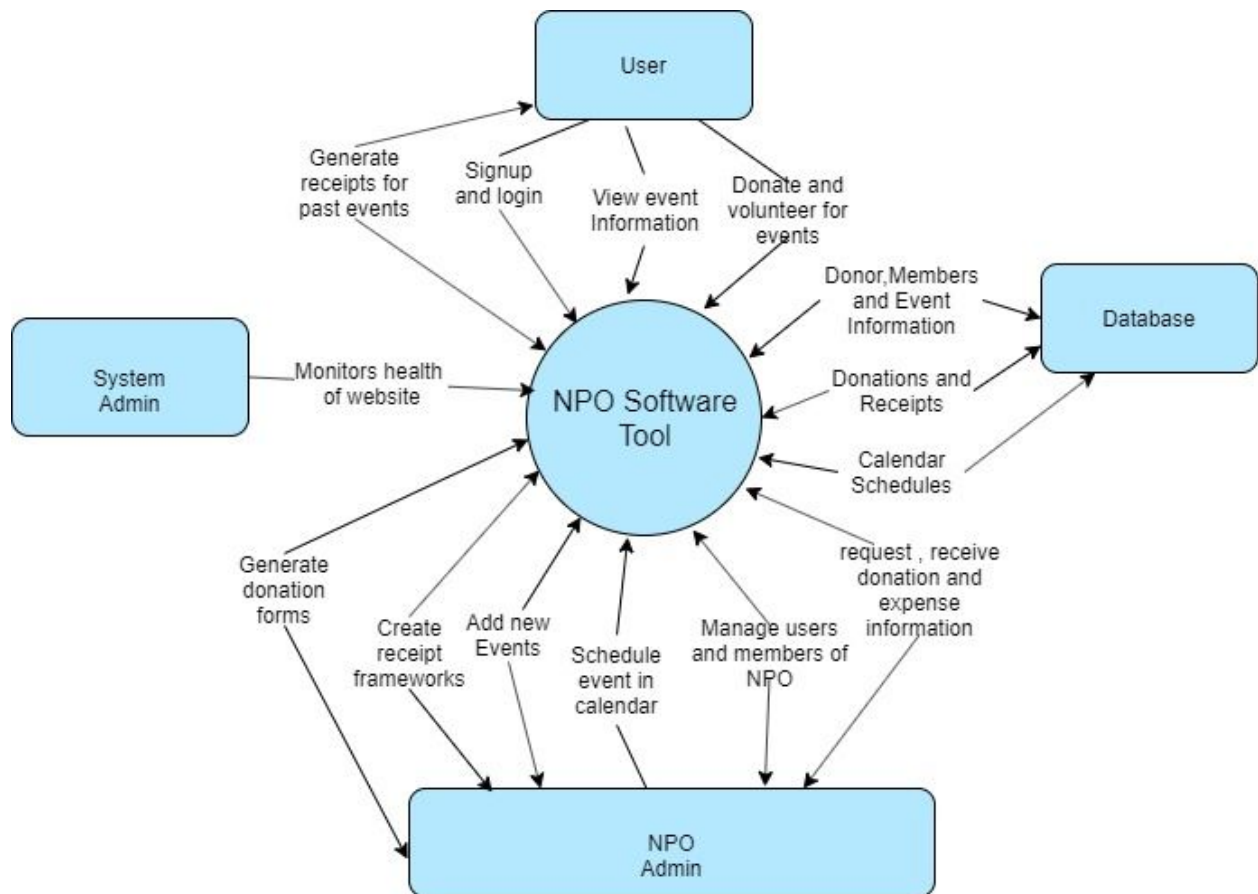


Figure 1

Well done.

2.2 Product Features

See Figure 2 for the relationship between these features. Please note that this document covers all features.

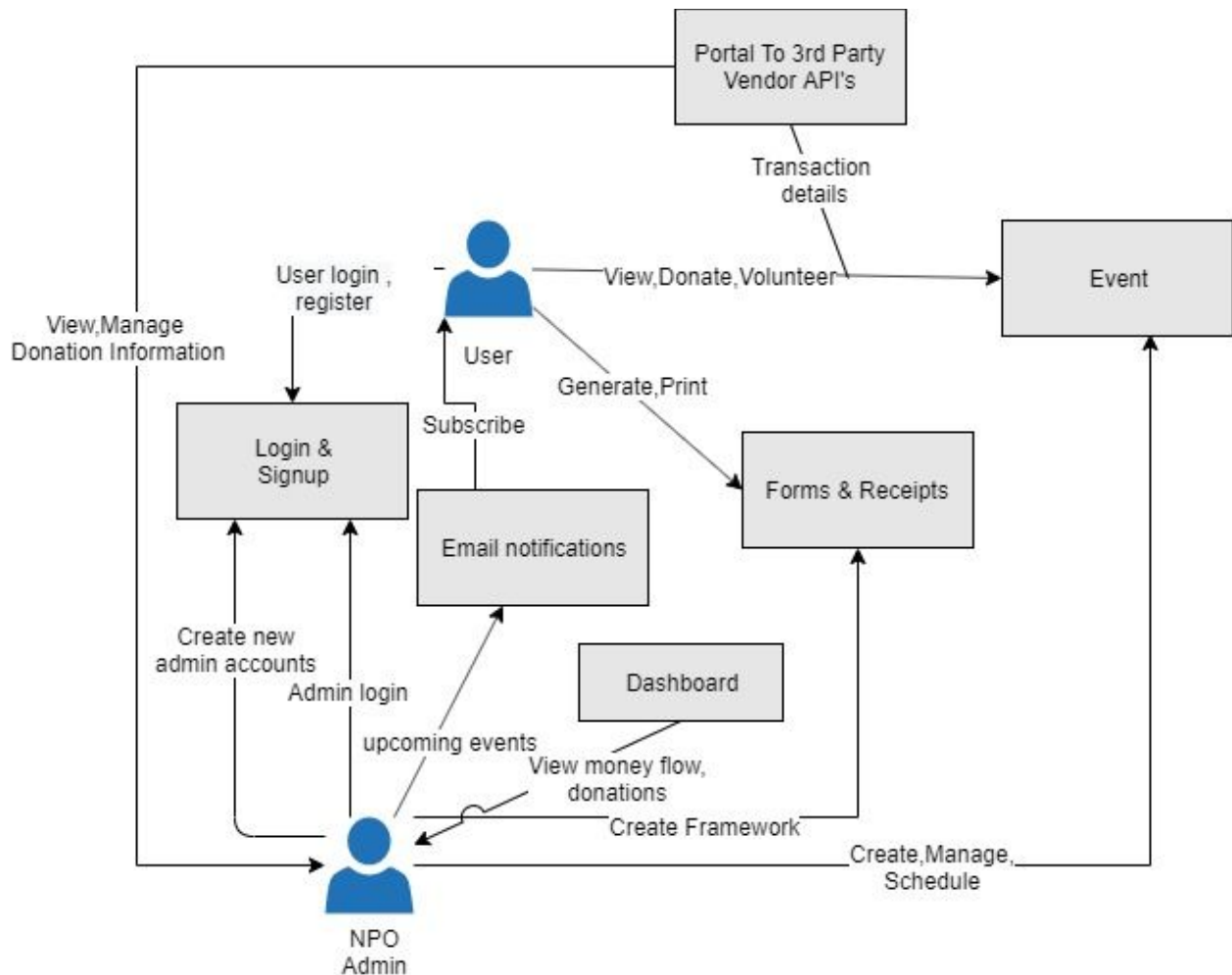


Figure 2

The main features of the product are:

FE-1	An in-built admin site to be used for maintenance and management of the entire site.
FE-2	Dashboard to keep track of monetary flow, latest donations, etc
FE-3	A receipt generator to reduce manual labor.
FE-4	Display information about upcoming events.

FE-5	Dynamic forms creation.
FE-6	Generate and print receipts, financial documents etc.
FE-7	An option to download the donations data in .csv format
FE-8	Calendar service
FE-9	A secure payment portal to interact with third-party payment vendor APIs

Difference between FE-3 and FE-6?

2.3 User Characteristics

System Administrator	<ul style="list-style-type: none"> Monitors the health of the website
NPO Administrator	<ul style="list-style-type: none"> The Administrator will have any and all the privileges of all other user types: <ul style="list-style-type: none"> Update information on upcoming evenCreate new admin accounts Create new forms Add events to the NPO's calendar Send out emails to the interested users. Monitor the accounts. Get donations data.
User	<ul style="list-style-type: none"> Can get information about past and upcoming events. Can donate if they wish to. Can access the website using their login id. Can register online if they would like to volunteer at an event being organized. Can generate receipts for past events.

2.4 Operating Environment

OE-1:	Continuous service is preferred, but as long as there is no data loss, minor service interruptions can be tolerated.
OE-2:	System shall operate in newest versions of all web browsers.
OE-3:	There should be no constraint on users being able to access the system at a given time.

OE-4:	Data is generated by online forms and stored in the database.
OE-5:	System is not dependent on geographical areas
OE-6:	Personal data will be stored in the database, so the database must be secure.
OE-7:	Angular javascript library will be needed to create the user interface for the system.
OE-8	Django python framework will be used to create the REST APIs for the backend of the system

OE-1 and OE-3 are non functional requirements. Reliability not mentioned in section 3.2. What type of database are you going to use?

2.5 Constraints

Following are the items that could limit the developer's options for designing the system.

CO-1:	The backend is to be implemented using the Django web framework while the frontend should use Angular.
CO-2:	The data will be stored in the local servers located in the non-profit organization.
CO-3:	Will need to support interaction with third-party payment gateways for donations.

No comments

2.6 Assumptions and Dependencies

Given below are the assumptions and dependencies:

AS-1:	It is assumed that there won't be high traffic (Not more than 500 concurrent users at any point in time)
DE-1:	Payments to be done using PayTM API
DE-2:	Django
DE-3:	Angular 7 (for user interface).

Why 500?

2.7 Documentation

No user documentation information at this time.

3. Specific Requirements

This section will give the requirements that are used to guide the project's software design, implementation, and testing.

3.1 Functional Requirements

Given below are the functional requirements of the website.

The requirements are sorted as per priority, going from highest to lowest in terms of priority.

FR1	Separate login for user and admin	The system shall have separate logins for the user and for the admin, leading them to two different interfaces.
FR2	Online Donations	The system shall offer online modes of payment for donations.
FR3	Update feed	The admin would given the privilege of updating the feed by adding upcoming events.
FR4	Form addition	The system shall allow the administrator to create forms dynamically using scanned images..
FR5	Receipt Generation	The system shall allow user to automatically generate the receipts of past donations and events that they participated in.
FR6	Email services	The system shall keep the supporters updated on the latest events using the email services.
FR7	Calendar	The system shall have a calendar, which can be edited by the admin and can be synced with the users' personal calendars

FR4- Do you mean OCR?

FR7- Calender services can be edited only by the admin?

3.2 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors

NF1	Performance	The system must be able to handle the load of at least 500 concurrent users. The website must also load within 500 milliseconds.
NF2	Security	Since the system stores personal data like PAN and Aadhar number, it must be secure. Security must be implemented both over the network and in the database.
NF3	Availability	The website (along with the database) should be highly available with 97% availability, i.e at most 1 day of downtime per month.
NF4	Maintainability	Since the maintainers of the system would presumably have limited knowledge of the software stack, the system should require minimal maintenance and should be easy to maintain.

Well written

4 External Interface Requirements

4.1 User Interfaces

The website shall have two user interfaces, one for the visitors and the other for the admin.

- UI-1 (Generic Interface): this interface would be present for the outsiders. The interface would primarily be mouse interactable along with the need for keyboard input for form filling. The interface would be user-friendly and straightforward to use.
- UI-2 (Admin Interface): The admin would be provided with an interface composed of tools for analysis and bookkeeping. Since the staff is said to belong to the class of senior citizens with minimal experience of computers, the interface would have to be extremely simple and self-explanatory. This interface would include rich visualizations for easy analysis and convenient means of adding new forms.

The UI can be more detailed with respect to the different tabs/sections that offer a particular functionality.

4.2 Hardware and Software in terms of how they would interact or how they would be executed

4.2.1 Software Interfaces

SI-1: Payment Interface - Secure interface for money transfer must be provided for the purpose of donations. We shall be using PayTM API for this purpose.

SI-2: Software interface for DB communication -

- SI-2.1 : To update donor information
- SI-2.2 : To maintain admin and user account information
- SI-2.3 : To get the data for visualizations
- SI-2.4 : To add/update donations

REST APIs would be used for communication between the frontend and backend.

What about the calender service? Will it not have any db communication?

4.2.2 Hardware Interfaces

None.

4.2.3 Communication Interfaces

CI-1: Email services will be provided to notify the members of the organization and users of upcoming events and fundraisers.

CI-2: Calendar alerts will be used to remind users about the latest events.

No comments

5 Appendix A: Glossary

No glossary terms were available at this time.