1. Introduction

This chapter is a part of our software requirement specification for the project "DonorLagbe". In this chapter, we have focused on the intended audience for our project "DonorLagbe"

1.1 Purpose

This document describes the Software Requirement Analysis of **DonorLagbe** application. It contains all the functional, non-functional, supporting requirements and expected requirements. At the same time it establishes a requirement's baseline for the development of the project. The requirements contained in the SRS are independent, uniquely numbered and organised by topics. But as time goes on, our SRS document is expected to evolve as users and developers work together to validate, clarify and expand its contents. The SRS serves as an official medium of expressing users' requirements to the developer and provides a common reference point for both the developer team and the stakeholder community.

1.2 Intended Audience

This SRS report is intended for several audiences which includes the end users as well as the project managers, designers, developers, and testers.

- The work of the developer team means the product which they have created can easily be verified by the customer whether it is acceptable to them or not by using this SRS document of **DonorLagbe**.
- Project Managers will be able to plan milestones, delivery date and at the same time ensure whether the developing team is on track or not during the development of the application by using this SRS document.
- The designers will use this SRS as a basis for creating the application's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfil the customer's needs.
- For developing the system's functionality, this SRS document will be used by the developers. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfil all of the customers' documented requirements.
- The testers will use this SRS in various phases of testing. They will use this SRS to derive test plans and test cases for each documented

requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfils the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.3 Conclusion

This analysis of the audience helped us to focus on the users who will be using our software requirements analysis. This overall document will help each and every person related to this project that includes users, project managers, designers, developers, testers, stakeholders to have a better idea about the project.

2. Inception

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.

To establish the groundwork, the following factors have been worked on to the inception phase:

- Icebreaking
- List of stakeholders.
- Recognizing multiple viewpoints.
- Working towards collaboration.
- Requirements questionnaire.

2.1.1 Icebreaking

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this by talking with them in context-free languages. Their behaviour, responding to our questions or willing to take a change in their shops solely depends on this phase.

2.1.2 List of Stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organisation that might be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”

To identify the stakeholders, we consulted with some receivers and examinees and found some requirements.

We identified the following stakeholders for our “DonorLagbe” project.

- Donor
- Receiver

2.1.3 Recognizing Multiple Viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

Donor’s viewpoint:

- Easy to use UI
- Listing the items easily
- Donating without revealing own identity
- Donating without knowing the receiver’s identity
- Assurance that user is donating to a real needy person ie. person verification.

donor’s viewpoint:

- Receiving without knowing the identity of donor
- Receiving without revealing the identity
- UI should be easy to use

2.1.4 Working towards Collaboration

Every stakeholder has their own requirements. There are some common and conflicting

requirements of our stakeholders. That’s why we followed the following steps to merge these

requirements-

- ❖ Find the common and conflicting requirements.
- ❖ Categorise them.
- ❖ List the requirements based on stakeholder's priority points.
- ❖ Make final decisions about requirements.

Common viewpoints:

- Easy to use UI
- Anonymous donation and receiving

Conflicting viewpoints:

There are no conflicting viewpoints among our stakeholders.

Final requirements:

We finalise the following requirements based on stakeholder's priority point:

- ❖ User friendly system
- ❖ Authentication
- ❖ Error free system (maximum 10% error may be considerable)
- ❖ Easy to maintain software
- ❖ Anonymous donation and receiving

2.2 Conclusion

The Inception phase helped us to establish a basic understanding about the online exam system, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

3. Elicitation of “DonorLagbe”

This chapter specifies the Elicitation phase.

3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organised and systematic manner.

3.2 Eliciting Requirements

The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. The following tasks were done for eliciting requirements-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 Collaborative Requirements Gathering

We have met with some stakeholders in the Inception phase such as the donors and receivers. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with the stakeholders (who are acting a vital role in the whole process) again to
elicit the requirements.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the clients into technical requirements for the software. The prime concern of the QFD is customer satisfaction maximisation. In order to ensure this, QFD enforces an understanding of what customers describe as ‘valuable’ and then deploy these values

throughout the engineering process.

QFD defines three types of requirements:

- Normal Requirements
- Expected Requirements
- Exciting Requirements

3.2.2.1 Normal Requirements

Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfils customers' satisfaction. These are the normal requirements for our project.

- Web based application
- System should include authentication system
- Providing feature to create or update product listing
- View donation history
- Anonymous donation and receiving

3.2.2.2 Expected Requirements

The requirements that are implicit to the system might not be brought up during the meeting because of their fundamental nature. Despite not being explicitly mentioned, their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below:

- Use database to store data
- Login Type (Donor, Receiver)

4. Usage Scenario

Introduction

DonorLagbe is a web based client server application that can be used to make donations anonymously. It will feature donations and receiving of goods hiding the identity of the donors and receivers.

Authentication

There are two types of users: donors and receivers. A user enters the application either by sign up or login.

Sign Up

Both donors and receivers have to provide the following information to sign up to the system:

- Name
- Email
- Address
- Phone No
- Password

Given emails must be verified while signing up. A 6 digit unique verification code will be sent to the emails for verification that the user needs to provide to verify his or her email.

Login

Both donors and receivers can login with their email and password.

Password Recovery

If a user forgets the password, he can recover his passwords by email verification.

Features

The features can be stated in two perspectives: user perspective, system perspective.

User Perspective

With DonorLagbe a **donor** can create, modify or delete a product listing that he wants to donate. He can upload the photo of the products as well.

Listing a product

A product contains Title, Photo, Used condition, Used period, and Location of the product. A user has to provide this info in order to create a listing that he wants to donate.

Update or delete existing listings

A donor can update the product name, start time and duration of the existing products that did not donate already. He can delete the products as well. But the information of the products he has already donated can not be modified or deleted.

A **receiver** will view the list of the available items. He can claim among them which he wants.

System Perspective

The **System** will store the product info and claim info of any listing. System will save the previous records as well.