SOFTWARE DESIGN SPECIFICATIONS

for

Crowd Fund Raising

Prepared by
Hassan Berry, Shehryar Naeem,
Saad Ismail

FAST-NUCES Karachi

December 3, 2018

Document History

Version	Name of Person	Date	Description of Change
1	Hassan Berry	01-12-2018	Document Created
2	Hassan Berry	04-12-2018	Document Completed
3	Saad Ismail	06-12-2018	Document Converted to LaTeX
4	Saad Ismail	06-12-2018	Proof-read

Distribution List

Name	Role
Ms. Rubab Jaffar	Supervisor
Mr. Awais Ahmed	Co-Supervisor

Document Information

Category	Information
Customer	FAST-NU Karachi
Project	Crowd Fund Raising
Document	Software Design Specification
Document Version	1.0
Status	Final
Author(s)	Hassan Berry, Saad Ismail, Shehryar Naeem
Approver(s)	
Distribution	Supervisor,Co-Supervisor

Contents

1 Introduction

1.1 Purpose of Document

This document provides explicit information about the requirements of our project and how the project is put together. This document also documents the design methodology that we will be using for the project.

1.2 Intended Audience

Any engineers who might get the job of improving our system to add further functionalities and will need to understand how the system was designed in the first place. Additionally; our supervisors who will grade us based on how correct, concise, and complete this document is.

1.3 Document Convention

Latex document class: scrreprt

1.4 Project Overview

The project is a web application which allows users of different types to interact with the Interface and get their specific requirements fulfilled. The purpose of the software is to allow companies, startups and individuals to interact on a common platform and exchange monetary benefits via sponsoring an idea, or donation to a cause.

1.5 Scope

This project will consist of creating a marketable, and deployable application based on the concept of crowdsourcing and crowdfunding. The modules of the project will in include a mobile application and web based front end where users can interact with the database.

1.5.1 Deliverables

- Functional Database

- Web based application

2 Design Considerations

The following design considerations need to be kept in mind:

1. The User needs to access a website which is both visually appealing, and fast. All other functionalities come later.

2. The database needs to handle multiple users at one time.

3. All the functionalities must be present, in addition to the new features which will make loyal users.

2.1 Assumptions and Dependencies

Assumptions:

- The Users like a floating banner.

- The Users like dynamic websites.

- The User like categories to show up at the top.

3 System Architecture

3.1 System Level Architecture

Elements:

- User

- Administrator

- Database

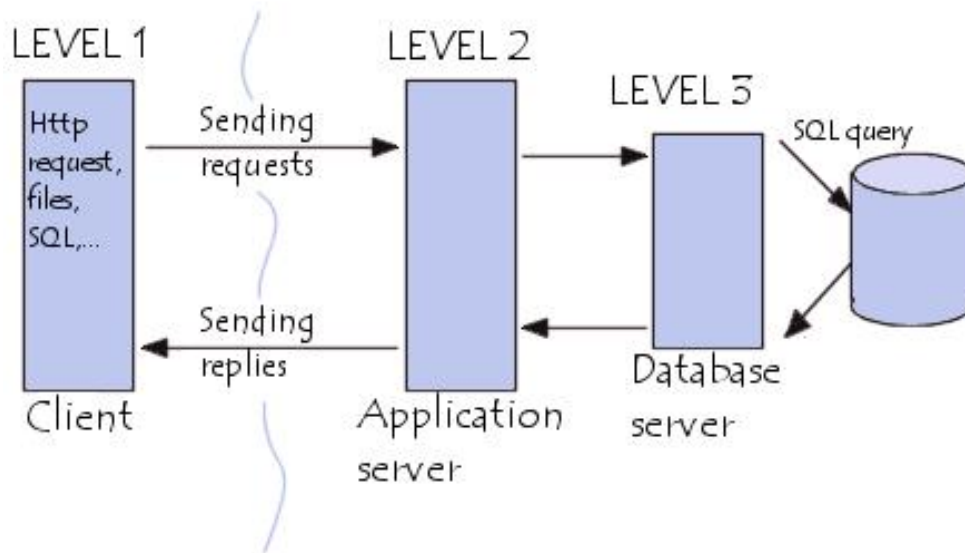- User Interface

Relationships:

- User interacts with User Interface to login, register, add category, post projects etc.

- Administrator interacts with Database to add new categories and check registered users.

User Interface Requirements:

- The User Interface needs to be designed in a way that is functionally capable, visually attractive and is mobile-responsive.

3.1.1 Software Architecture

We are using 3-tier architecture. The user only interacts with the front-end application (Angular 2) which requests back-end application (Node.JS) which then further fetches the data from the database (MySQL).

LEVEL 1   LEVEL 2

LEVEL 3

Http request, files, SQL,...

Sending requests

SQL query

Sending replies

Client   Application server

Database server

3.2 Design Strategy

Our design strategy is focused on two main aspects:

- Ease of Use

- Aesthetics

We believe that a website that is aesthetically pleasing to look at and easy to navigate will definitely pull customers in and increase traffic, which is the prime goal of creating any website.

The next goal is to be functionally reliable, and we will ensure that happens so that we can build a web base of loyal customers.

3.2.1 Future system extension

1. We will develop a mobile application on Android and iOS.

2. We will add further functionalities to our current system like crowd-sourcing.
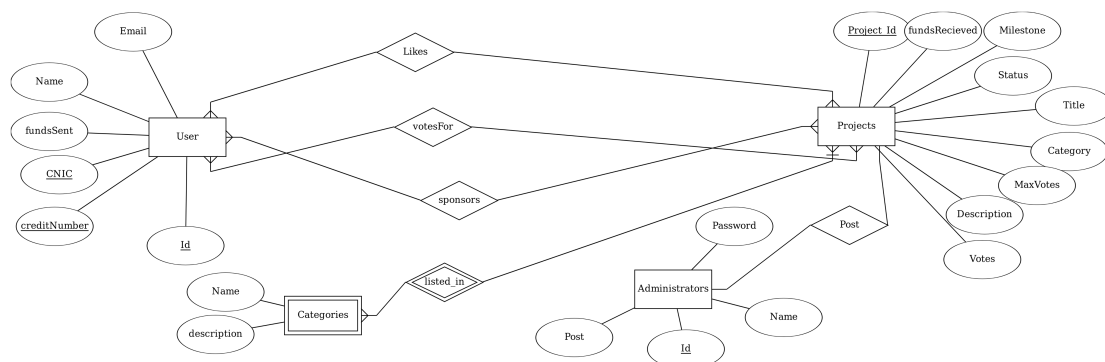
3. We will

3.2.2 System Reuse

Our system is designed in a way that it can be reused again and again on different hosting services and can be connected to different databases.

3.2.3 Data Management

This part of our project is still ongoing.

3.3 Detailed System Design

3.3.1 Database Design

User

Name	User
Alias	
Where-used/how-used	Used to interact with UI and Projects
Content description	

Column Name	Description	Type	Length	Null able	Default Value	Key Type
Email	Provides email address	Varchar	35	no	0	
CNIC	Provides National Identity Number	Integer	13	no	0	PK
ID	Provides Identity number	Integer	10	no	0	FK
Name	Provides username	Varchar	30	no	0	
FundsSent	Provides Funds Sent	Integer	30	Yes	0	
CreditNumber	Provides Credit Number	Integer	19	Yes	0	

Category

Name	Categories
Alias	
Where-used/how-used	Used with projects to give options to user
Content description	

Column Name	Description	Type	Length	Null able	Default Value	Key Type
Name	Provides Category name	Varchar	20	no	0	
Description	Provides category description	Varchar	100			

Administrator

Name	Administrator					
Alias						
Where-used/how-used	Used to manage everything					
Content description						
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
ID	Provides Identity number	Integer	10	no	0	FK
Password	Provides password	Varchar	20	no	0	
Name	Provides name	Varchar	20	no	0	
Post	Provides Post	Varchar	20	no		

Project

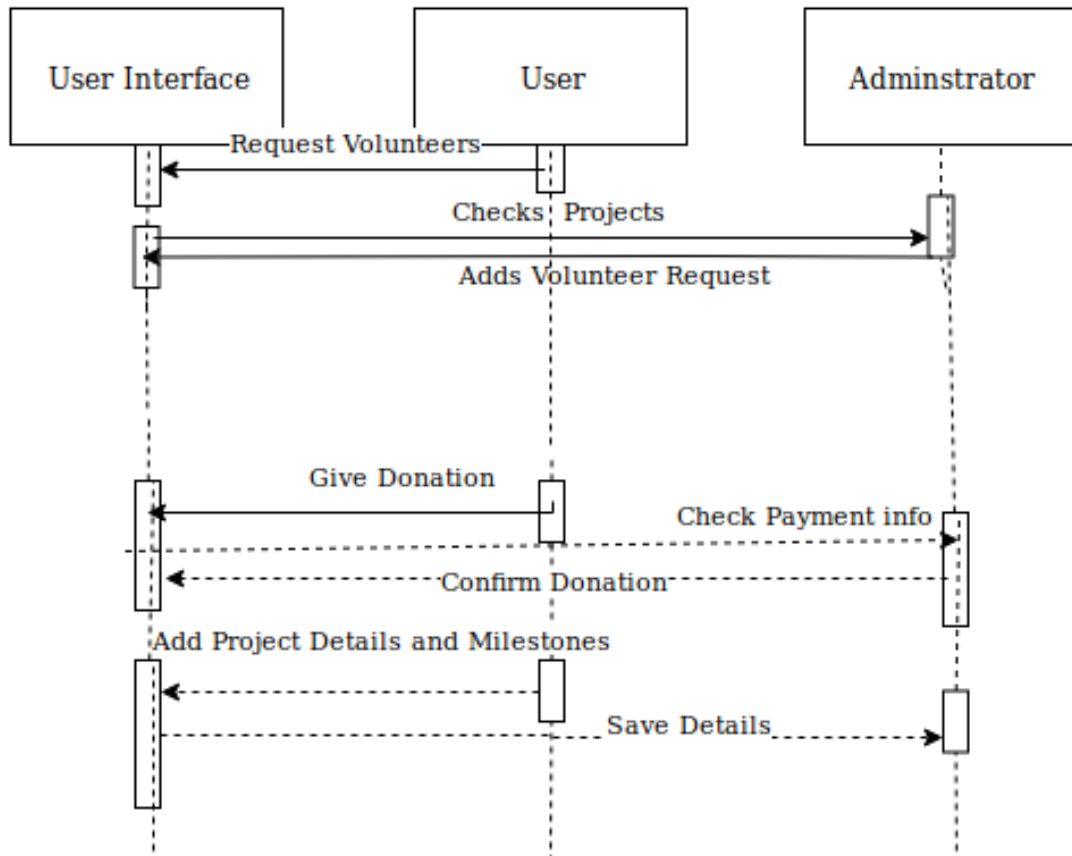| Name	Project					
Alias						
Where-used/how-used	Used with Admin					
Content description						
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
Project ID	Provides Project ID	Integer	10	no	0	PK
Funds Received	Provides funds received	integer	20	yes	0	
Milestone	Describes milestones	integer	20	yes	0	
Status	Describes current status	char	20	no	Started	
Votes	Describes votes	integer	10	no	0	

3.3.2 Application Design

Sequence Diagram
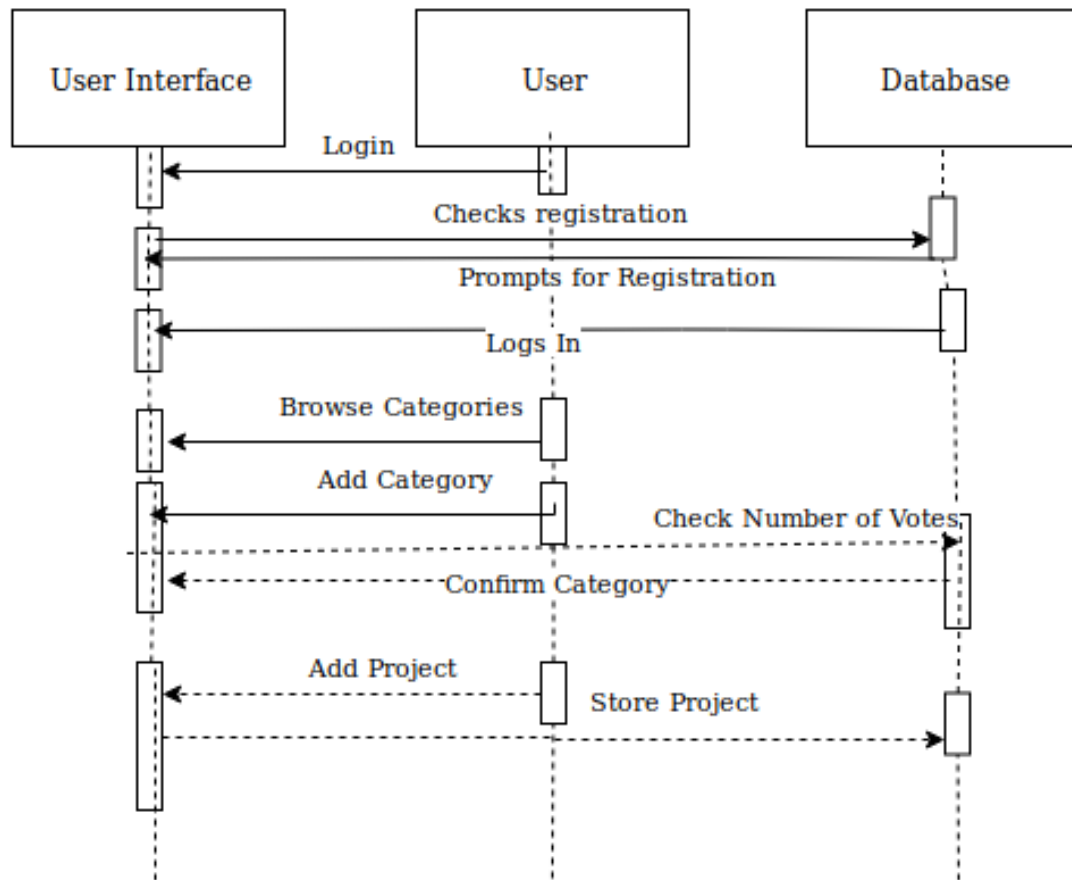
In this sequence Diagram we have covered three main aspects of our project:

1. Login and Register: User clicks login, and enters details. The UI checks with the database if the user is registered or not and prompts if not.

2. Add Category: In the add category function the Database checks the number of votes on the category and compares to other categories and if it fulfills or exceeds a set criteria then it is added.

3. Add Project: The User requests a project to be added and this request is then sent to the Database which stores it.

In this sequence Diagram we have covered three main aspects of our project:

1. Request for Volunteers: This is the social service and philanthropic aspect of our project. The user requests for volunteers and the project is checked by the Administrator, then the request is approved.

2. Give Donation: This is the most integral part of our project. A user requests to donate money to a cause or project, the payment info is confirmed and then the amount is confirmed and sent back.

3. Add project details: Here the user adds details of their project including progress, milestones achieved etc.

State Diagram
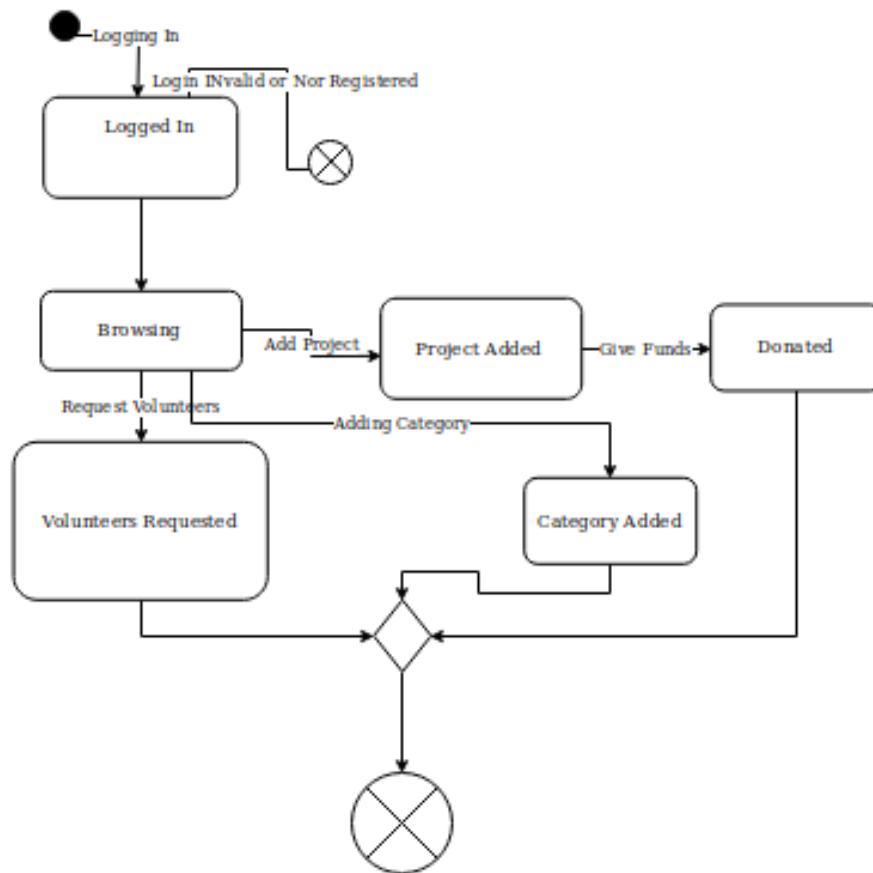
Explanation: This diagram contains all our states and starts when a user enters the logged in state. After that he enters the browsing state and can then proceed to three alternative states which are volunteers requested, category added and project added.

3.4 References

- UML 2 Sequence Diagrams.

- What is the general format of a Software Design Specification?