SOFTWARE REQUIREMENTS SPECIFICATION

for

Crowd Fund Raising

Prepared by
Shehryar Naeem, Hassan Berry,
Saad Ismail

FAST-NUCES Karachi

November 22, 2018

Contents

Revision History

Name	Date	Reason For Changes	Version
Hassan Berry	19-11-2018	Initially document created	1
Shehryar Naeem	20-11-2018	Usecase Diagram	2
Hassan Berry	21-11-2018	Proof-read	3
Shehryar Naeem	22-11-2018	Dressed Usecases	4
Saad Ismail	22-11-2018	docx to latex conversion	5
Saad Ismail	22-11-2018	Last minute changes	6

Distribution List

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

1 Introduction

1.1 Purpose

To provide a detailed **overview** of our software product, its parameters and goals. This document describes the project's target audience and its user **interface**, hardware and software requirements.

1.2 Document Conventions

Latex document class: scrreprt

1.3 Intended Audience and Reading Suggestions

Supervisors, and team members.

1.4 References

Not applicable.

2 Overall Description

2.1 Product Perspective

This is a new self-contained product.

2.2 Background

The background of this project is the severe need of a consolidated platform which combines crowd-funding, crowd-sourcing and provides a platform for people interested in philanthropic and social services.

2.3 Project Scope

This project will consist of creating a marketable, and deploy-able application based on the concept of crowd-sourcing and crowd-funding. The modules of the project will in include a mobile application and web based front end where users can interact with the database.

2.3.1 Deliverables

- Functional Database
- Web based application

2.3.2 Out of Scope

- Dynamic Website
- Mobile Application

2.4 Project Objectives

The objective of this project to solve the problem of having a consolidated platform where crowd-sourcing and crowd-funding is combined to be a 'one-stop' place for people seeking funds, companies/individuals looking to invest in great ideas, and individuals/organizations looking to form a team for a specific project.

2.5 Stakeholders

1. Users
2. Administrators

2.6 Operating Environment

The project can be hosted on any PC platforms, regardless of architecture and operating system being used. The development, testing and official support is only for Intel based CPU with Linux Operating System.

2.7 System Constraints

- In order to implement payment system, we will have to partnership with local and international banks.
- The system is open to spam, we are totally dependent on end-users to reduce spam.

2.8 Assumptions and Dependencies

We are totally dependent on banks to handle payments to the projects and we will have to transfer payments manually to the end-users.

3 External Interface Requirements

3.1 User Interfaces

We will provide user-friendly, responsive and easy to use web interfaces for our customers. Those will be based on Bootstrap v4.

3.2 Software Interfaces

There are three main components of the product.

1. **Front-end**
   Powered by Angular 2 (TypeScript)

2. **Back-end**
   Powered by Node.JS and Express.JS

3. **Database**
   Powered by MySQL

3.3 Hardware Interfaces

For initial release, the product will run on a single node but in future, the three software components can be deployed on different nodes to increase concurrency and optimizing the page load speed.

3.4 Communications Interfaces

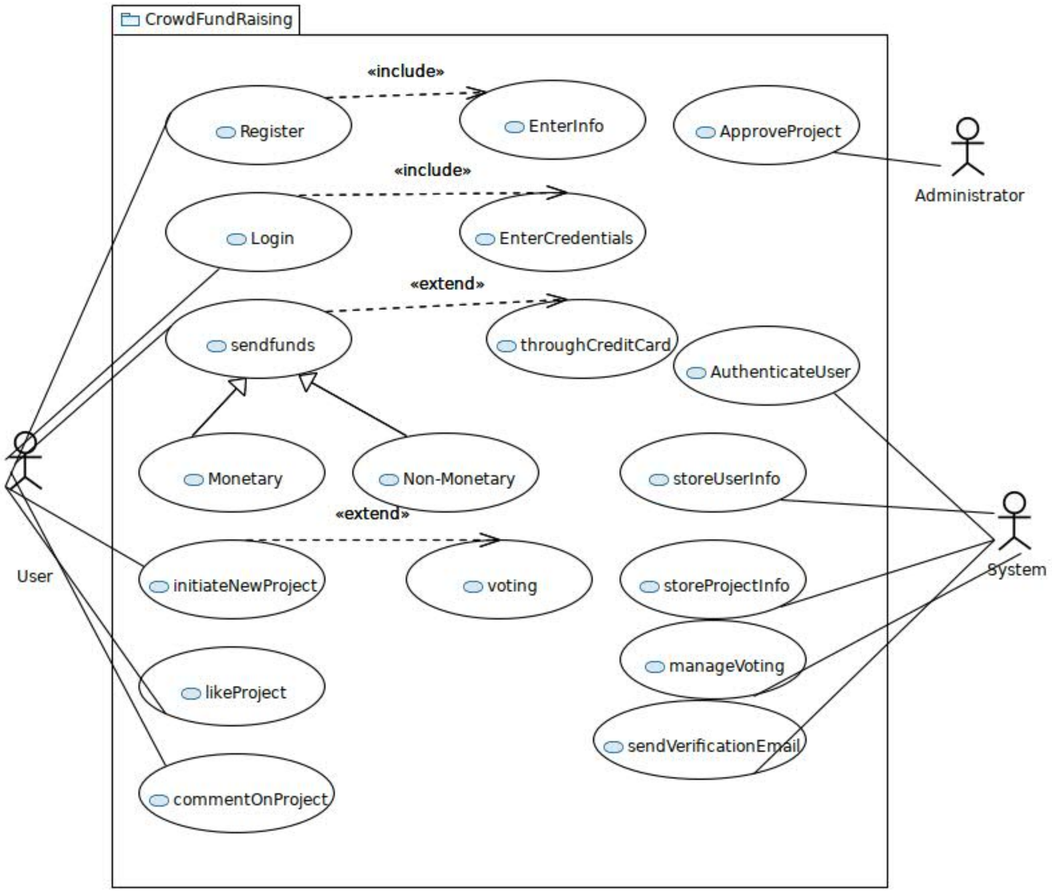
- HTTP(s) (80, 443)
- MySQL (3306)
- SMTP (465 - only outgoing)

4 Functional Requirements

4.1 Functional Hierarchy

1. Register
    - Enter User Information
    - Account Verification
2. Login
    - Enter Credentials
    - Authenticate and Redirect to related Webpage
3. Initiate New Project
    - Enter Description and Milestones
    - Select Related Category
4. Send Funds (Mocked)
    - Project Selection
    - Select Fund Type
    - Send Funds
    - Authorize and Confirm Transaction
5. Vote projects

4.2 Use cases


CrowdFundRaising
«include»
Register
EnterInfo
ApproveProject
Administrator
«include»
Login
EnterCredentials
«extend»
sendfunds
throughCreditCard
AuthenticateUser
User
Monetary
Non-Monetary
storeUserInfo
«extend»
System
initiateNewProject
voting
storeProjectInfo
manageVoting
likeProject
sendVerificationEmail
commentOnProject

Use Case 1	**Login**
Primary Actor:	End-User
Preconditions:	• User is already registered.
Postconditions:	• The user can view his/her profile and can access other features.
Main Success Scenario: 1. Enter email 2. Enter password 3. Authenticate	
Alternate Scenarios: 1.a Invalid login details: 1. System shows failure message	

Use Case 2	**Register**
Primary Actor:	End-User
Preconditions:	• User is not already registered.
Postconditions:	• The user can login using his/her account.
Main Success Scenario: 1. Enter fullname 2. Enter email 3. Enter password 4. Enter confirm password 5. System registers the users and sends verification email. 6. User verifies the account by clicking the link received in the email.	
Alternate Scenarios: 1.a User not verified: 1. User will not be allowed to post new projects/vote on existing projects.	

Use Case 3	**Send Funds**
Primary Actor:	End-User
Preconditions:	• User is already logged in.
Postconditions:	• The funds will be scheduled to be sent.

Main Success Scenario:

1. Enter amount
2. Confirm

Alternate Scenarios:

1.a Amount not valid:

   1. System shows failure message.
   2. Transaction is cancelled.

Use Case 4	**Create New Project**
Primary Actor:	End-User
Preconditions:	• User is already logged in.
Postconditions:	• The project stored in the database and available to vote.
Main Success Scenario: 1. Enter project details 2. Enter milestones 3. Confirm	

Use Case 5	**Approve Project**
Primary Actor:	Administrator
Preconditions:	• The project should have reached the voting milestone.
Postconditions:	• The new project will be included in the category.

Main Success Scenario:

1. Verify project details and users authenticity
2. Put the project in suitable category.

Alternate Scenarios:

1.a Project not suitable:

   1. Delete the project.

Use Case 6	**Authenticate User**
Secondary Actor:	System
Preconditions:	• User is not already logged in.
Postconditions:	• The user can view his/her profile and can access other features.
Main Success Scenario: 1. Verify user details 2. Redirect user to suitable webpage	
Alternate Scenarios: 1.a Account details incorrect: 1. System shows failure message 2.a Account not verified: 1. System shows warning message	

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The project must be easily accessible and fast to the end-users. The page load time must be less than 10 seconds, even on slow internet connections. The interface must be user friendly and easy to use so that the user doesn't have any confusion or difficulty. The page rank of the website must be prepared according to the Google Page Rank Algorithm which will allow it to show up easily in Google Search Results.

5.2 Safety Requirements

Users, their payment information and other sensitive information must be handled safely. Moreover, the use of bots must be eliminated and great care will be taken to ensure that the personal details of the Users are not compromised in any way.

A two factor authentication system can be utilized to ensure that the login security of users is protected and no malicious attack will be made successful.

5.3 Security Requirements

The product must not be vulnerable to any known issues and other security bugs like SQL injection, XSS etc.

The possibility of bots forming usernames and accounts must be eliminated and to do this, we will utilize Google Recaptcha which will ensure that the user is a human being.

5.4 Software Quality Attributes

Code quality must be up-to the mark. Unit and Integration tests should be written after the initial release.

All the stages of the Software Development Life Cycle must be followed. Moreover homogenization must be done in order to add clarity and consistency to our models and ensuring that all the stake holders clearly follow the process and can check that their requirements are being followed.