Software Requirements Specification

for

Crowdfunding site GoGetFunding

**Version 1.0**

**Prepared by**

|  |  |  |
| --- | --- | --- |
|  |  |  |

Group Name: 3.1.B

|  |  |  |
| --- | --- | --- |
| Sergiu Voloc | student 1 | Computers Ro |

|  |  |
| --- | --- |
| **Instructor:** | **Asist. drd. ing. Cătălin SBORA** |
| **Course:** | **Web Application Design** |
| **Date:** | **17.03.2020** |

Table of Table of Contents

**Contents**

[**1** **INTRODUCTION** 3](#_Toc35322403)

[1.1 Document Purpose 3](#_Toc35322404)

[1.2 Product Scope 3](#_Toc35322405)

[1.3 Definitions, Acronyms and Abbreviations 4](#_Toc35322406)

[1.4 References and Acknowledgments 5](#_Toc35322407)

[**2** **OVERALL DESCRIPTION** 6](#_Toc35322408)

[2.1 Product Functionality 6](#_Toc35322409)

[2.2 End Users and Characteristics 6](#_Toc35322410)

[2.3 System Stakeholders 7](#_Toc35322411)

[2.4 Operating Environment 7](#_Toc35322412)

[**3** **SPECIFIC REQUIREMENTS** 9](#_Toc35322413)

[3.1 External interface requirements 9](#_Toc35322414)

[3.1.1 User Interfaces 9](#_Toc35322415)

[3.1.2 Hardware Interfaces 16](#_Toc35322416)

[3.1.3 Software Interfaces 16](#_Toc35322417)

[3.1.4 Communications Interfaces 16](#_Toc35322418)

[3.2 Functional requirements 16](#_Toc35322419)

[3.3 Performance requirements 17](#_Toc35322420)

[3.4 Design constraints 17](#_Toc35322421)

[3.5 Software system attributes 17](#_Toc35322422)

[**4** **Non-functional Requirements** 19](#_Toc35322423)

[4.1 Software Quality Attributes 19](#_Toc35322424)

[**5** **Other Requirements** 20](#_Toc35322425)

[5.1 Safety 20](#_Toc35322426)

# Introduction

GoGetFunding project is about a Crowdfunding platform where anybody can explore and fund projects that interests and beside this to post a personal project(idea) to be funded.

I will present what are the benefits of this platform and that there is a demand and great usability of it.

## Document Purpose

This document is intended to provide a description of crowdfunding platform.

I will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react with the users.

## Product Scope

The purpose of my online product is to ease project funding problems and to create a convenient and easy-to-use application for every user that visit the platform.

This application can be user from personal causes and events to any projects and more. In this way there will be the possibility to help people from all over the world raise millions online and bring their ideas intro reality.

To make this possible I analyzed with other platforms that exists and I want to give users as many capabilities as possible and these are:

- Add video, pictures, links & more

- Accept donations by credit and debit card or through PayPal

- It's free to start a campaign and there is only charge a competitive 4% on the amount raised. Processing fees are an average of 2.9%

- Crowdfund Anything! Causes, People & More!

- Europe Based Platform - Millions of $ Raised

- Ongoing Access to Funds

- Your own personal fundraising coach

- No Penalties for Missing a Goal

- Mobile Friendly

- Safe and Secure

- Easy 4 Minute Setup

- Global, Multi-Currency Platform

## 

## Definitions, Acronyms and Abbreviations

**Definitions:**

**Crowdfunding**- is the practice of funding a project or venture by raising small amounts of money from many people, typically via the Internet. Crowdfunding is a form of crowdsourcing and alternative finance.

**Mobile Friendly / responsive** – working good on mobile devices.

**PayPal**- Holdings, Inc. is an American company operating a worldwide online payments system that supports online money transfers and serves as an electronic alternative to traditional paper methods like checks and money orders.

**ASP.NET** - open-source server-side web-application framework designed for web development to produce dynamic web pages developed by Microsoft to allow programmers to build dynamic websites, applications and services.

**C#** - object-oriented programming language used with XML-based Web services on the .NET platform and designed for improving productivity in the development of Web applications.

**Abbreviations** :

|  |  |
| --- | --- |
| FAQ | Frequent asked questions. |
| DB | Database |
| Platform / product / application | the GoGetFunding crowdfunding site. |

**Acronyms** :

|  |  |
| --- | --- |
| CSS | Cascading Style Sheets |
| HTML | Hypertext Mark-up Language |
| MVC | Model-view-controller |
| NA | Not Applicable |
| Signed User | User that entered on platform using login and password |
| Guest User | User that entered on platform without using login and password |

## References and Acknowledgments

1. <https://searchsoftwarequality.techtarget.com/answer/How-to-elicit-performance-requirements>
2. <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
3. <https://fontawesome.com/>
4. <https://www.w3schools.com/>
5. <https://getbootstrap.com/>
6. <https://www.indiegogo.com/>
7. <https://gogetfunding.com/>
8. <https://en.wikipedia.org/>
9. <https://www.kickstarter.com/>
10. <https://docs.microsoft.com/ro-ro/aspnet/core/?view=aspnetcore-3.1#recommended-learning-path>
11. <https://www.pluralsight.com/courses/aspnet-core-fundamentals>
12. Software Requirements Specification and Functional Requirements Document IEEE 830/1998: <http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf>
13. Code Style / Sonar:
14. <https://google.github.io/styleguide/javaguide.html>
15. <https://github.com/google/styleguide>
16. Microsoft's standards - Security : <https://docs.microsoft.com/en-us/dotnet/standard/security/>

# Overall Description

## Product Functionality

This application brings a unified platform for both users who start a project and who donate for one.

Each user can access About page to know better what represent the crowdfunding platform.

There is a link to How it Works page where user can understand better how to place a personal project and how to donate for one.

Also, for most common question I did the FAQ where user will be able to find the answer on almost any general question.

There will be 2 buttons called Discover projects where will be presented most relevant project for user and the second button called Start a Project where the user will be able to insert details of his idea of funded project.

And of course the project page will represent all the details for other users to understand the urge and need of donating to the project.

## End Users and Characteristics

There are two types of users that interact with the system : unsigned user and sing-ed user. Each of these two categories of users has its own requirements and some are shared.

**Signed User:**

* Has the detailed description of how to use our application on the Aboutpage.
* Has the ability to login using a unique login and password.
* Has the possibility to change password.
* Has a list of personal projects placed on platform
* Can donate for the selected project
* Has the ability to comment on each project
* Can Add a project

**Guest user :**

* + The Has the detailed description of how to use our application on the Aboutpage.
* Has the ability to login using a unique login and password
* Has the possibility to change password
* Can donate for the selected project

## System Stakeholders

A stakeholder is a group or individual who is affected by the system, may influence the way this system is shaped. Stakeholders include software development staff, end-users who interact with the system. Other system stakeholders may be engineers who are developing or maintaining related systems, business managers, domain experts.

To maintain and improve the app will be taken care of by the developer team, as well as for bug fixing.

## Operating Environment

As developer, I will be using C# as the main programming Language, and ASP.NET as the framework technology since the app is an online website that requires a web server.

**Front-end** **:** HTML, CSS, SCSS, JavaScript

**Back-end :** C#

**Framework** **:** MVC

**DB** **:** Microsoft SQL Server

**Version control** **:** I will use GitHub.

**Task distribution** **:** I will use KanbanFlow.

**Designing database relations:** www.lucidchart.com.

**The minimum platform requirements for user system should be:**

* Users require a running browser, supported by the ASP.NET Framework, on any operating system:

Google Chrome (version 50.0 and higher)

Internet Explorer﻿ (version 9.0 and higher)

Apple Safari﻿ (version 10.0 and higher)

Mozilla Firefox﻿ (version 70.0 and higher)

Opera﻿ (version 6.0 and higher)

* Good Ethernet connection is required, at least 3G.
* As for the hardware necessary, the user requires a standard terminal machine, such as a laptop or PC, with functional mouse, keyboard and screen.
* Windows 7/8/10
* Good Ethernet connection is required, at least 3G.
* The operating system needed is a Windows platform that supports ASP.NET.
* As for the hardware necessary, the user requires a standard terminal machine, such as a laptop or PC, with functional mouse, keyboard and screen.

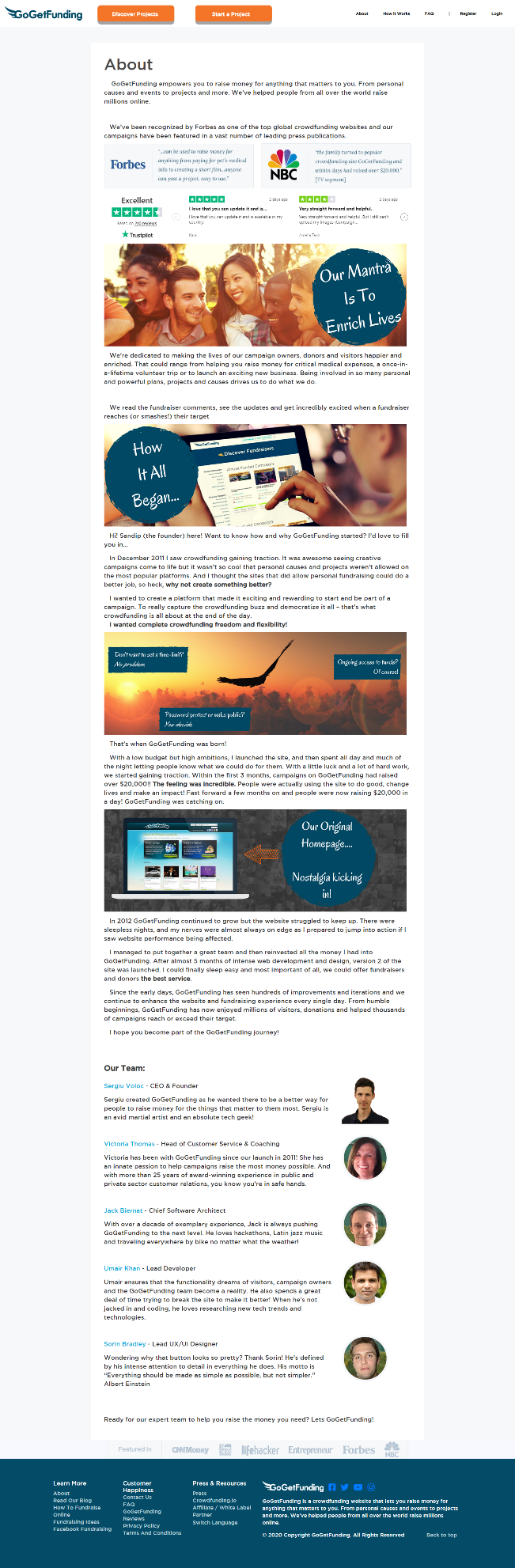
# Specific Requirements

## External interface requirements

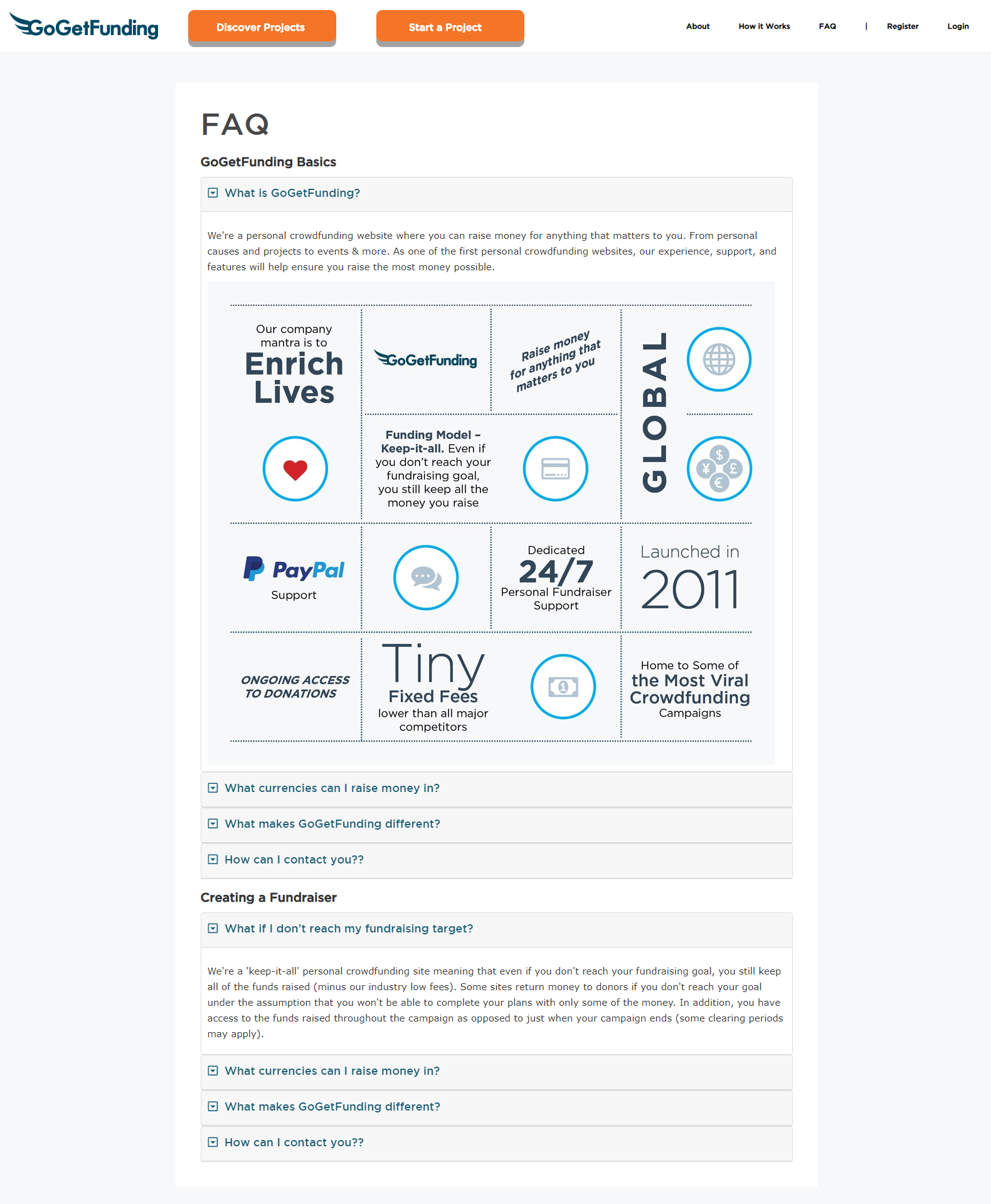
### User Interfaces

The application will recognize whether the username and the password correspond to an admin or user and it will show the right interface. It will have an interactive and easy to use interface.

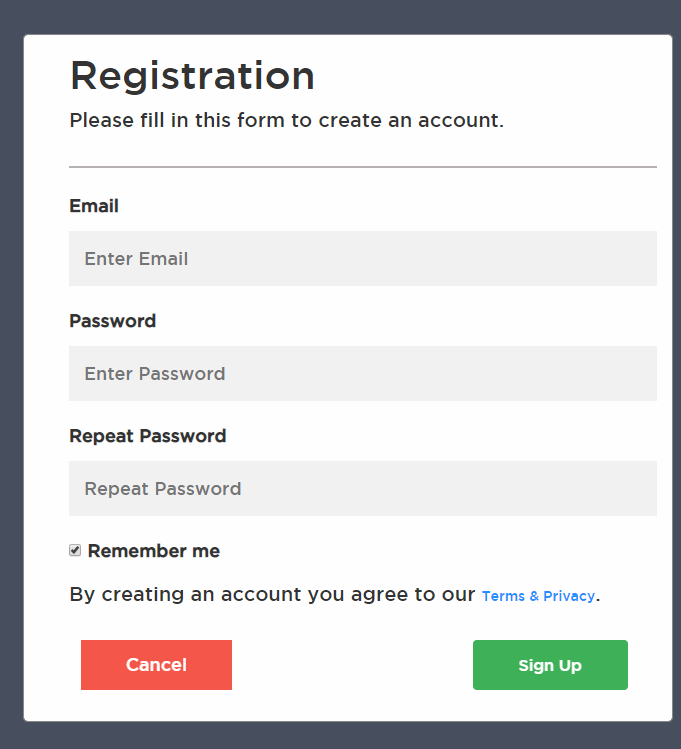
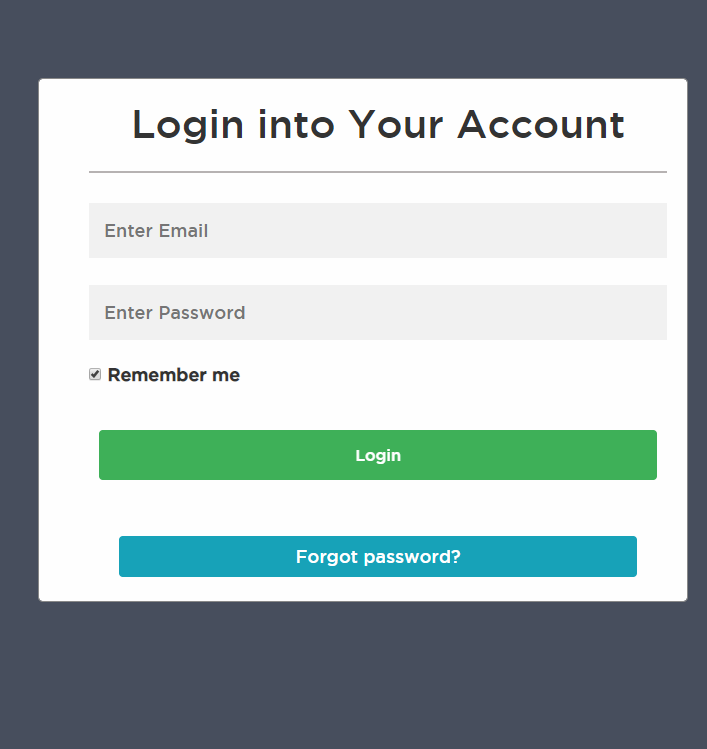
**Fig.1 – Home Page.**

** Fig.2 – About Page.**

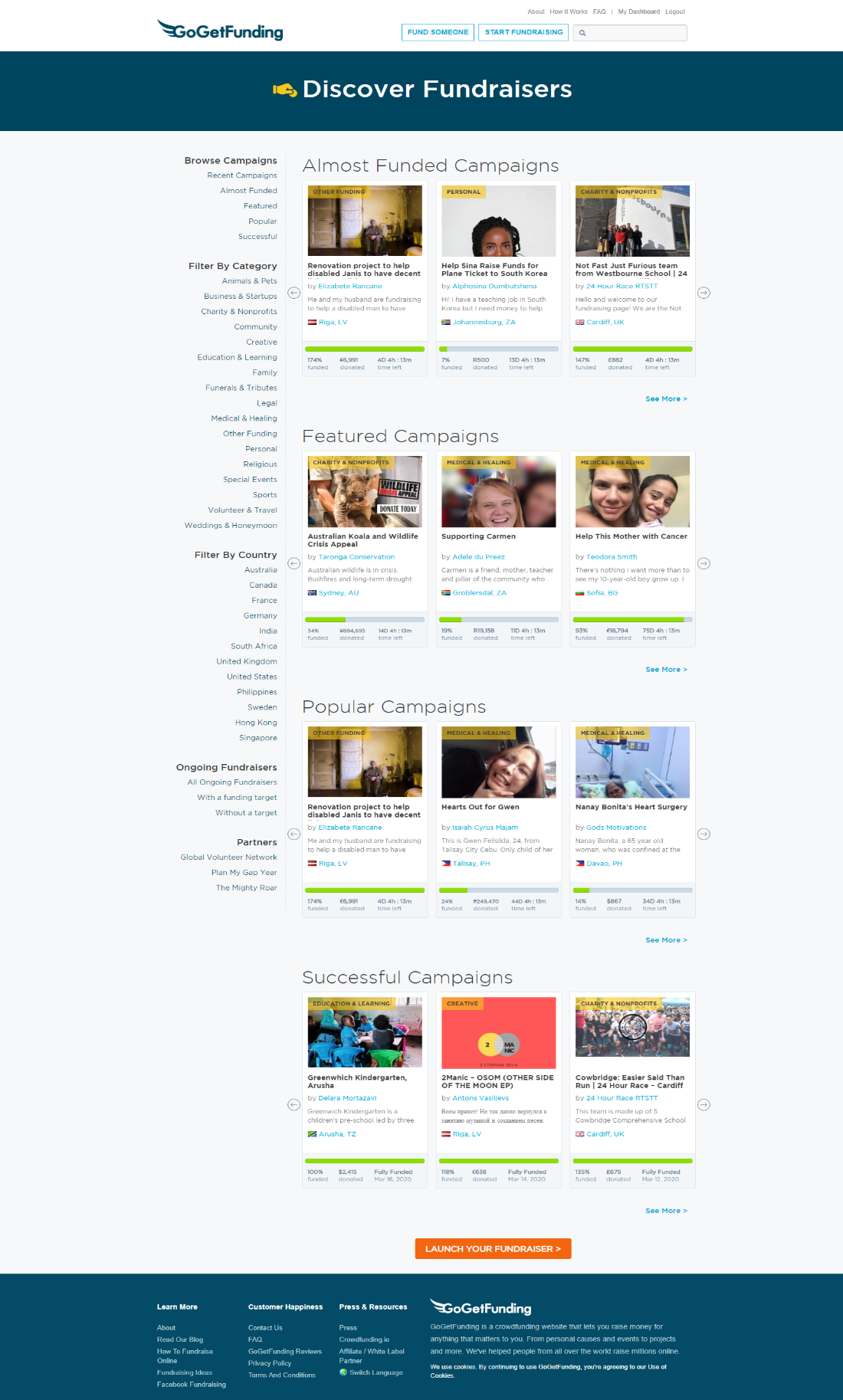
**Fig.3 – How it Works Page.**



**Fig.4 – FAQ Page**



**Fig.5 – Registration Fig.6 – Login**



**Fig.6 – Discover Projects page**



**Fig.6 – Discover Projects page**

### Hardware Interfaces

As for the hardware necessary, the user requires a standard terminal machine, such as a laptop or PC, with functional mouse, keyboard and screen.

### Software Interfaces

The application is connected to a database. The database will communicate with the server, and the server will send the information to the client application.

### Communications Interfaces

The communication protocol that this application will use will be HTTP.

## Functional requirements

* + 1. **User Class 1 – The Signed userser**
       1. **Registration**
* The user should be able to register through the web application.
* The user must an email, password and confirm password.
* The information will be stored into the database.
  + - 1. **Log in**
* A registered user should be able to log in to the web application.
* He must provide correct email and password.
  + - 1. **Retrieve password**
* In case if user forgot the password, he has the possibility to make a new one by <Forgot Password?> button.
  + - 1. **View services**
* A signed user can view everything that can unsigned plus Start a project page, personal details and comment under a project.

* + - 1. **Modify data**
* A signed user should be able to modify his/her personal data from his/her account.
* Add / update / delete a project to be funded.
* Has the possibility to fund a selected project.
  + - 1. **Log out**
* A signed user should be able to log out from his/her account.
  + 1. **User Class 2 – The Guest user**
       1. **Registration**
* The user should be able to register through the web application.
  + - 1. **View services**
* A not signed user should be able to view following pages :
  + About page
  + How it works page
  + FAQ page
  + Homepage page
  + Explrore Projects page
* Has the possibility to fund a selected project.
  + - 1. **Modify data**
* Has the possibility to fund a selected project.

## Performance requirements

The product shall take initial load time depending on the internet connection strength which also depends on the media-device from which the product is running.

If the system loses the connection with the internet or the system gets some strange input the user must be informed.

## Design constraints

Since the user uses the application directly from the browser there are no limitations due to hardware, because the application is responsive and can adapt to any devices. The user must have only a good internet connection.

## Software system attributes

### Availability : the system is available 24 hours a day.

### Reliability : the system can be used by multiple users concurrently. Any user can access the system using even a medium performance PC or a smartphone connected to internet.

The application covers general **security** sides that can affect user experience and data protection, such as :

1. The access in the personal account is made based on the Email and Password.
2. The password of each user is saved in the database being Encrypted, to ensure user confidentiality.
3. Security for personal data will follow a thorough and continuously monitored framework of controls, both technical and organizational, appropriate to the nature of the data processing and the associated risks.
4. The input fields such as email, password etc. will be protected by SQL Injection.

### Maintainability : The application will be easy to extend. The code will be written in a way that it favors implementation of new functions.

### Portability : a user can log in to the system at any time and any place in the world.

# Non-functional Requirements

## Software Quality Attributes

The Quality Attributes of the project involves important and necessarily coding style rules that I consider valuable to respect and implement, thus emphasizing my professionalism. The following rules apply:

1. Use tab (four spaces) of indentation.
2. Use *Snake Case* (my\_var) variables  in C# language.
3. Use *Pascal Casing* (MyClass) for class names and method names.
4. Do not use Screaming Caps for constants or readonly variables, because Caps grab too much attention.
5. Avoid this. unless absolutely necessary.
6. Open-braces ({) go on a new line in C# language.
7. At least one blank line should be between method definitions and property definitions.
8. Begin comment text with an uppercase letter and end with a period.
9. Insert one space between the comment delimiter (//) and the comment text.
10. Avoid using Abbreviations. Exceptions: abbreviations commonly used as names, such as Id, Xml, Ftp, Uri and others.
11. Prefix interfaces with the letter **I**.  Interface names are nouns (phrases) or adjectives.
12. Name source files according to their main classes.

My platform provides a huge performance and user experience because it is way more convenient to raise funds for an idea or project online instead of offline, this way will be bigger impact and amount of backers.

My solution provides a clearly and robust structure of use, every user has these possibilities under control and at ease to use place. Beside this, after a few times of use, the platform became very friendly as it’s easy to learn, thus easy to remember how to interact.

The process of placing a project on platform is very easy and flexible, as any login-ed user can create his project in just four easy steps.

The site will run properly from almost every popular browser.

As it’s a cloud base project, the user will not encounter any trouble with personal data, once it’s uploaded on site, the project will be stored on server.

Processing of personal data will be respected as the Legislative structure indicates.

Privacy of user’s data will be securely hidden from other users.

# Other Requirements

## Safety - NA

The User’s life is not in danger while using GoGetFunding platform.