Software Requirements Specification

for

SmallWorld's Web App

Prepared by Lauren Mcmillen and Junior Garcia

New York University Abu Dhabi

April 12, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Junior Garcia and Lauren Mcmillen	April 12,2020	Creation of the SRS	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the Web Application being built for SmallWorld. SmallWorld is a student-led non-profit start-up whose aim is to facilitate effective, long-term humanitarian engagement among high school students through mentorship and an online platform. In order to achieve the aforementioned, we designed a Web Application that will be used as the main platform for Smallworld to achieve its online functionality.

1.2 Document Conventions

For the sake of simplicity, we adopt the following conventions throughout the document:

- Web Application: A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface.
- SWAP: SmallWorld's Web Application
- highschoolers: high school students
- URl: Uniform Resource Locator
- browser: A web browser is a software application for accessing information on the World Wide Web.
- Github repository: GitHub is a Git repository hosting service, but it adds many of its own features.
- Frontend: Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that data
- Backend: The back end of a website consists of a server, an application, and a database.
- API: application program interface,is a software that interfaces with another software.

1.3 Intended Audience and Reading Suggestions

Although we do our best at describing SWAP in layman's terms, we assume that the readers of this document have an understanding of how Web Applications work. As such, this document is intended for people with experience in coding and developing for the Internet.

1.4 Product Scope

SmallWorlds' Unique Selling Point (USP) over other organizations that help high schoolers achieve long-term humanitarian impact is that aims to connect students and mentors in a frictionless way by breaking down any bureaucratic impediments that exist between them . In order to best achieve the aforementioned, we leverage the Internet by creating a Web Application that allows mentors and students to easily create and join teams, develop an action plan, and track progress on the humanitarian problem aimed to be solved.

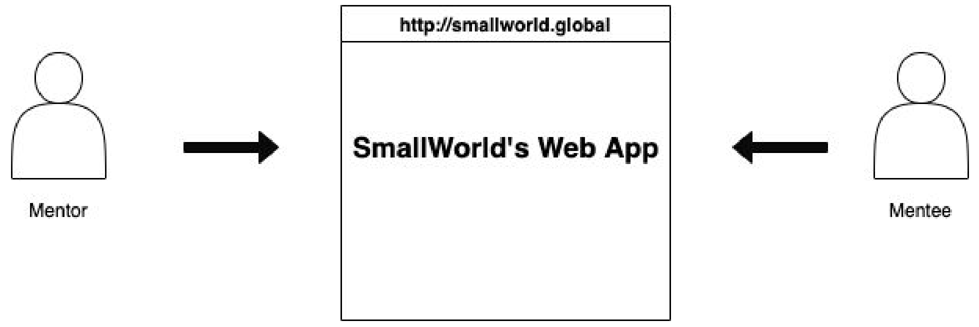
1.5 References

- https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app
- https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/
- https://www.computerhope.com/jargon/b/browser.htm
- https://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html
- https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces

2. Overall Description

2.1 Product Perspective

SWAP aims to be SmallWorld's first attempt at connecting like-minded mentors and high-schoolers through the Internet to solve humanitarian problems. Despite the vast variety of existing options that can be used to achieve the aforementioned, like social media platforms like Facebook or more traditional avenues like Local Volunteering Centers, we noticed the prevalence of high schoolers dropping out from such programs due to a variety of reasons. The main one, we believe, is that there exists a huge disconnect between the mentor and the mentee due to the organizational hurdles the high schooler needs to overcome in order to connect with the mentor and establish a strong relationship that helps the high schooler create positive change. Mentor-mentee relationships are imperative to increase interest in high schoolers to engage in their local community and leverage their skillset to contribute even a grain of salt to an issue of importance. As such, the goal of SWAP is to be a novel interface that facilitates the interactions between a mentor and the mentee(s) through the Internet with the goal of making a tangible impact in the mentee(s)'s community.



2.2 Product Functions

SWAP will allow mentors and mentees achieve the following:

- Register users as either students or mentors
- Mentors create teams
- Student sign up for a team
- Students and mentors access the curriculum in module-style components
- Students and mentors submit text as part of the curriculum
- Mentors create a final profile of 5 main steps of the project
- This final profile is displayed on a public platform

2.3 User Classes and Characteristics

We have three types of users who will be using SmallWorld

Mentors

Mentees

Anonymous Users

2.4 Operating Environment

The operating environment will be a web application accessed through an url (Uniform-Resource-Locator) that will display the application in the user's Browser.

2.5 Design and Implementation Constraints

The largest consideration is that the mentees will be, in most cases, under the age of 18. As such, security is of heightened importance. First, in order to make an account, parental consent must be granted as per the laws of where the program is implemented. This may involve an email confirmation function in which the parent must confirm the student's involvement and approve that the student may have contact with the mentor. Next, all interactions between the mentor and mentee through the platform must be recorded and available for organizational and parental review. In this first stage of development, the interaction involves shared access to the modules. Both mentor and mentee can add and access information to the modules. Thus, this information must be available for monitoring.

2.6 User Documentation

We plan on developing a user manual for new coming SWAP users. Ideally, this will be embedded within the web application in an interactive tutorial feature. However, we believe this is not a functional requirement and will only be developed if time permits. If not, then we will create a user manual embedded within the README.md file in our Github Repository.

2.7 AssumptionsDependencies,and Software Interfaces

We plan on developing this web application using existing web frameworks. Namely:

- React.js for our frontend, a framework to develop user interfaces developed by Facebook.
- Django, a python web framework that we plan to use as our backend API.
- MySql, for our database.
- Bootstrap, for styling our UI.

3. External Interface Requirements

3.1 User Interfaces

Landing Page - Figure 1: The minimalist landing page of the site will center on a 3D globe with a call to action for the user to "join" and "explore".

Map Interface - Figure 2: A map interface acts as the "database" of completed team projects. A team project will be represented by an icon, pictured here as a different colored dot, and is set at the location at which they are based. Users can filter team problems based on the category of the problem. In this early representation, each problem category is represented as a different color.

Team Profile - Figure 3: After a user clicks on a team's icon on the map interface, they navigate to the page of the team profile. The profile of the teams demonstrates the 5 Steps of their solution in a non-linear, profile-like fashion.

Team Dashboard - Not Featured: Once logged in, a user will be directed automatically to their Team Dashboard. The Team Dashboard will feature a side menu with links to the user's personal profile and the user's team profile; the team profile will be the default screen. The Team Profile consists of the five modules. The title of each module will be displayed in a horizontal bar. When the module is accessible, a "+" button will trigger a drop-down view of the tasks of that module.

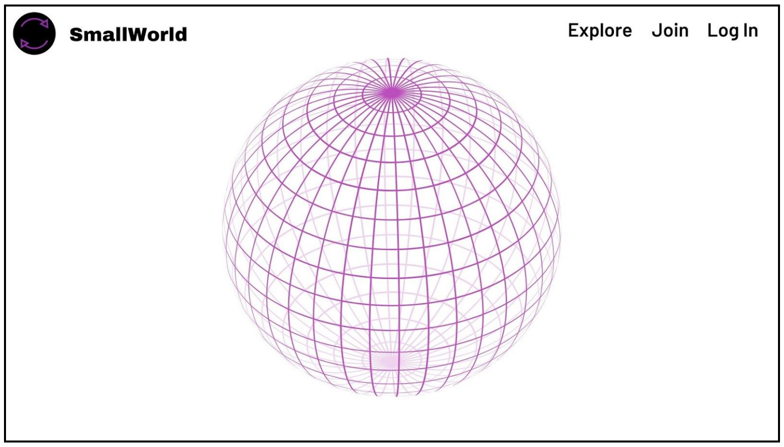


Figure 1 - Landing Page

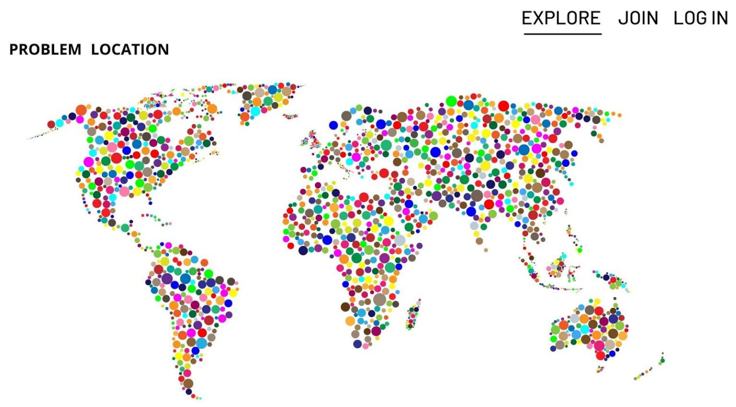


Figure 2 - Map Interface

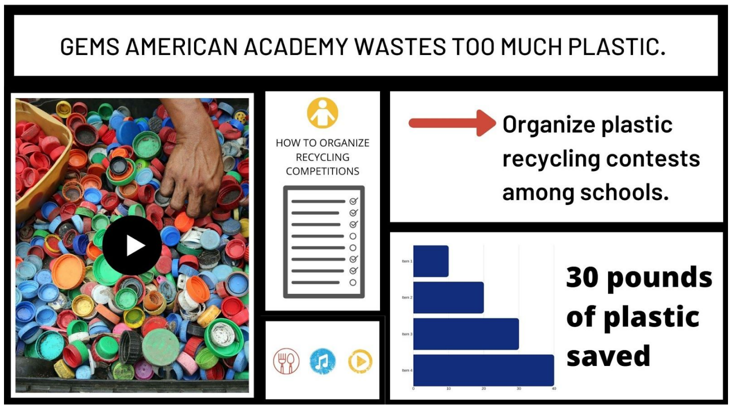

GEMS AMERICAN ACADEMY WASTES TOO MUCH PLASTIC.
HOW TO ORGANIZE RECYCLING COMPETITIONS
Organize plastic recycling contests among schools.
30 pounds of plastic saved

Figure 3 - Team Profile

3.2 Hardware Interfaces

We anticipate that SWAP will be accessible through any Internet-enabled device, including mobile devices and personal computers. We plan on styling our React UI components using Bootstrap, a responsive,mobile first library for front-end development. This will allow our application to run smoothly on any device.

3.3 Communications Interfaces

Given that this is a web application, the communication that exists between the server that will host SWAP and the user's device(client) is HTTP (Hypertext Transfer Protocol), the underlying format that is used to structure requests and responses between a client and a server. As such, our web application will be designed to handle HTTP request methods(mostly GET,PUT,POST,DELETE) from our clients and respond accordingly. We also expect to use HTTP to interface between our backend API (Django) and our frontend(REACT) to deal with data-driven operations that depend on the information stored in our Database.

4. System Features

4.1 Register user

4.1.1 Description and Priority

A user registers as either a student or a mentor and creates a user profile.

High Priority

4.1.2 Stimulus/Response Sequences

- *User* clicks "Register"
- User specifies if Student or Mentor
- User inputs profile information required

4.1.3 Functional Requirements

REQ-1: User registers by selecting "Register as a student" or "Register as a mentor"

REQ-2: User inputs profile information
- Profile Information: First Name, Last Name, City, Username and Password
- Error handling:
    - If the username is the same as another user, the user is prompted to choose another name.

- If password does not satisfy requirements for security, user is prompted to choose another password.

4.2 Mentor creates team

4.2.1 Description and priority

A mentor creates a team, opens the team for users to join, and accepts mentees.
High priority

4.2.2 Stimulus/Response Sequences

- Mentor clicks "Create Team" in Mentor Dashboard
- Mentor inputs required information about team (TBD)
- Mentor clicks "Open to New Members"
- Mentor monitors new member requests through the Mentor Dashboard
- Mentor accepts/declines new member requests

4.2.3 Functional Requirements

REQ-1: Create team
REQ-2: Mentor opens team for members.
REQ-3: Mentor accepts or declines requests.

4.3 Mentee joins team

4.3.1 Description and priority

A mentee locates an open team through the public platform, requests to join, and joins the team.
High priority

4.3.2 Stimulus/Response Sequences

- Mentee browses public platform for open teams
- Mentee clicks "Request to Join" on the new team's Team Profile
- Mentee monitors request on Student Dashboard.
- Mentee is notified of acceptance/denial of Dashboard.

4.3.3 Functional Requirements

REQ-1: Mentee gains access to Team Dashboard after acceptance.
REQ-2: User can only request to join a team if authenticated as student.
REQ-3: Student can filter teams based on location on the map interface.

4.4 Team members complete tasks

4.4.1 Description and priority

Team members complete tasks in the Team Dashboard which culminate in submitting a "Step" of the profile.
High priority

4.4.2 Stimulus/Response Sequences

- Team members begin working on first task in Team Dashboard by clicking "Start making progress" on Module 1
- After all tasks are submitted for one module, the next module's "Next" button becomes accessible.

4.4.3 Functional Requirements

REQ-1: Team members submit text to forms in modules
REQ-2: Team members submit multimedia files to modules.
REQ-3: The subsequent module becomes accessible to team members after the previous module is submitted.
REQ-4: There are five modules which each correspond to a "Step" in the profile.
REQ-5: Each module contains one or more tasks, which are forms that teams fill out and submit.
REQ-6: All tasks in a module are accessible while the module is active and after the module is completed.
REQ-7: A module is accessible only if the previous module has been fully completed.

TBD: The specific content of the tasks

4.5 Team profile displayed on public platform

4.5.1 Description and priority

After all five modules are submitted, the team's public profile is displayed on the public platform. The Team Profile consists of 5 Steps: 1) Problem, 2) Become Educated, 3) Solution, 4) Action Blueprint, and 5) Impact. The problem is one of 15 categories, and solution is one of 3 solution types.
High priority

4.5.2 Stimulus/Response Sequences

- Mentor submits 5th module.
- Mentor clicks "Publish to Platform"

4.5.3 Functional Requirements

REQ-1: Profile is displayed at the location of the team on the map.
REQ-2: User can search platform using "problem" filter or "solution" filter.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The Web Application being built needs to have a responsive user interface and support the data from all of our users and retrieve it when necessary. As such, we believe that using existing responsive technologies will help us in achieving this goal.

5.2 Safety Requirements

The interactions between student and mentor must be limited to sharing work on the modules until further checks may be implemented to monitor the relationships between an adult and a minor. As stated before, these modules must be accessible to the parent, and this access must be communicated through email notifications and parent access to the student account information.

5.3 Security Requirements

Users must confirm their account by email address validation. In the future, we may introduce a two-factor authentication system to ensure the identities of users are verified and to further safeguard the minors involved.

5.4 Software Quality Attributes

The highest software priority is flexibility. As an early stage non-profit, SmallWorld has many novel ideas for apps and functionalities for its platform. The ability to tweak design, as well as add new functionalities which are compatible with the current platform, are crucial to SmallWorld's development as an innovative non-profit.

5.5 Business Rules

- A user needs an account, consisting of a username, password, email, and personal information, to join a team.
- A mentee under the age of 18 must gain permission from a parent to join a team.
- A mentee's parent must have access to the mentee's account information.
-

Appendix A: Glossary

Team Dashboard: the "home base" of the mentor and the students on a team; features the 5 modules and the tasks within each module

Map Interface: the "database" of SmallWorld; users can search for teams by location, problem, and solution on the interface

5 Steps: the core of the SmallWorld program and the 5 tenets of the team's profile; consists of 1) Problem, 2) Become Educated, 3) Solution, 4) Action Blueprint (which is a detailed step-by-step plan), and 5) Impact;

Team Profile: the output of a team's project; consists of a web-page layout and features each of the 5 steps that the team completed

Appendix B: To Be Determined List

1. *Content of the Tasks in Each Module*
    a. *The number of tasks per each module; the language of the task description; the type of file each task requires are set to be determined by the organization at a later date. For the development of the web app, we will be using dummy tasks.*