



# Software Test Plan

4/1/20

Version 1.0

Project Sponsor:

Dr. Igor Steinmacher

Team Mentor:

Fabio Santos

Team Members:

Adriana Aguilera, Joseph Danciu, Jevin Dement, Ugo Dike, Stavros Triantis

# Table of Contents

Introduction.....	3
Unit Testing.....	4
2.1 Web Server.....	4
2.2 User Profile Miner.....	4
2.3 Project Miner.....	5
2.4 Matching Tool.....	5
2.5 Database.....	6
Integration Testing.....	6
3.1 Adding a New Project.....	8
3.2 Deleting a Project.....	8
3.3 Login.....	8
3.4 Adding Skills via Questionnaire.....	9
3.5 Saving User Skills.....	9
3.6 Matching Users to Projects.....	9
3.7 Displaying Matches.....	10
Usability Testing.....	10
4.1 Questionnaire.....	11
4.2 Project Finder.....	11
4.3 Match History.....	12
Conclusion.....	12

## 1.0 Introduction

Our team, alongside our client Dr. Igor Steinmacher, has been working on a web application that will aid newcomers to Open Source Software (OSS) development on GitHub in finding their next project. The world of open source on GitHub can be difficult to navigate, and presents newcomers with an overwhelming amount of information. These developers often feel discouraged by the sheer amount of projects and often do not know what they are qualified to work on. Our product will provide users with matches to available OSS projects that require skills they already have, ultimately creating a smoother experience for newcomers. This product will take out both the frustrating and confusing aspects of finding a new project, and hopefully motivate more users to contribute to OSS projects.

In this document we outline the tests we will conduct to ensure that this product provides the best possible experience to our end users. Unit tests will ensure that all components of our code will work as expected. Here we will make sure that individual components of our product function properly. Integration testing will ensure that the flow between these components works correctly and flows as expected. As our project is very modular, we need to make sure all parts need to be working together properly. Lastly, usability testing will test if our product is usable for the end user. Usability testing is very important to our product as it is a user driven application, if users are not able to work the product, it would be deemed ineffective. These tests are very important to keep us on track and make sure development is running smoothly, allowing us to fix any content that may not be sufficient.

## **2.0 Unit Testing**

Unit Tests help with the maintenance and changing of code. They help detect changes that may have occurred within specific components, ensuring the components we built are functioning and maintainable in the future. We want to ensure that there is a simple way to identify where the problem is located within our program. We will be using the unittest library to test all of our functions within our main components folder. Our main components consist of the Web Server, User Profile Miner, Project Miner, Matching Tool, and Database. We will get into more detail about how we will test these individually below.

### 2.1 Web Server

The provider for our web server is DigitalOcean, as our client is already subscribed to this service. The web server will be the vessel for our web application on which it will be hosted, as well as provide a space for our database.

We will be testing the web server by running a ping test on the server and returning a 200 or error code if it fails.

### 2.2 User Profile Miner

The User Profile Miner is a program which will allow us to extract unique skills from each user's GitHub profile. The program will check for the java language based projects and download the repositories of those projects. It will only be mining projects that have been contributed to in the past 60 days, to ensure that each developer still has proficient knowledge in these categories.

The project miner will be tested by taking in a test username and returning the proper skills listed within that project.

### 2.3 Project Miner

The Project Miner is a program that will have the ability to manage OSS projects listed in a database. There will be functions to Add projects, and a function for the admin to update projects. Projects added to the manager will be mined using the outsourced project miner. The mined “Skills” from this portion will be considered the required skills to be able to contribute to that specific OSS project. The skills found within the mined projects will be saved into a PostgreSQL database.

The project miner will be tested by taking in an test username and returning the proper skills listed within that project.

### 2.4 Matching Tool

The user’s skills and OSS project skill requirements will be compared. If there is a match between the skills and requirements, we will display it. All matches will be listed for the user, along with other information so they can start contributing. Users can initiate a new match request on their profiles. After matches have been finalized, users will be able to find them in a list once logged into their profile. An email notification will be sent to the individual's email once the mining and matching process has completed. Matches will be stored in a profile indefinitely, and users will be able to access them on their own time. Projects will only be removed if a user no longer meets the skills necessary. The matching tool will have the ability to

pull data from the database, specifically user skills and project requirements to test for a match. It will also have the ability to push its final matches to the database.

For testing purposes, the matching tool will be given a username and a project to match the skills from the database.

## 2.5 Database

We will have two separate sections of our database, one for storing user information and one for storing project information. The database we have chosen to use is Postgresql, as it integrates well with our project and supplemental code provided by the client. As mentioned above, the match tool will be able to pull user skills data and project requirements data, as well as push its final matches. We will be able to query skills from the database that will be used to populate the questionnaire.

We can test the database by having the database return a proper query from each table.

## **3.0 Integration Testing**

Integration Testing focuses mainly on the interfaces and flow of data between the modules. Our product has the following main components: the project miner (provided by client), the user skills miner, a project database, a user database, the matching tool, admin portal, and the user interface. Although they are modular and can stand on their own, the flow and communication between them is necessary for the use of our system.

### 3.1 Adding a New Project

Adding new projects will test the data flow between the admin portal, project miner, and the projects database. Admin will choose a new project to add on the portal, this should trigger the project miner to extract useful information, and then store it in the project database.

Integration Test	Input	Expected Output
Adding a New Project	Admin will enter a new project to be added to the database.	Project will appear in the projects database.

### 3.2 Deleting a Project

This will test the communication between the admin portal and the project database. An admin will delete a project from the portal which will remove it from the database entirely and prevent it from being matched to a user in the future.

Integration Test	Input	Expected Output
Deleting a Project	Admin deletes a project from the database.	Project is removed from the database.

### 3.3 Login

This test will check the connectivity between the user database and the user interface. Once a user enters their credentials they will be redirected to their dashboard where they can see old matches, start a new search, setting, etc.

Integration Test	Input	Expected Output
Login	User inputs their login credentials.	User is successfully redirected to their dashboard.

### 3.4 Adding Skills via Questionnaire

Here we test for the communication between the questionnaire and the user database. Before the profile mining begins, users will be presented with a questionnaire to add skills and that input will be stored in the user database.

Integration Test	Input	Expected Output
Adding Skills via Questionnaire	User adds skills using the questionnaire.	These skills are displayed in the user database.

### 3.5 Saving User Skills

This test involves the user skills miner and the user database. A user inputs their username into the system and begins the mining process for their GitHub profile. Once the user's skills and information have been mined, they should be able to be viewed on the user database.

Integration Test	Input	Expected Output
Saving User Skills	User inputs their username to the interface.	User's skill and information will appear in the database.

### 3.6 Matching Users to Projects

This test checks the interaction between the user skills database, project database and the mining tool. The user will initiate a new project search, after their profile is mined, the matching tool should initiate. If all is correct, the matching tool should proceed without any errors.



Integration Test	Input	Expected Output
Matching Users to Projects	User initiates a new project finder search on their profile.	Matching tool will proceed without errors.

3.7 Displaying Matches

This test checks the communication between the matching tool and the user interface. Once the matching process is finished, matches should be displayed via the user interface on a user’s profile.

Integration Test	Input	Expected Output
Display Matches	User inputs their username to the interface.	Matches will be displayed on a list in a user’s profile.

**4.0 Usability Testing**

Usability Testing refers to an evaluative process for a product by testing it with representative users. Product users are assigned tasks to complete using the product, “while observers watch, listen, and take notes”[1]. By having end users test our product, we will be able to identify the key issues with our User Experience (UX). Our plan for conducting usability testing is having peers in the Computer Science Major use our product. The reason we need people in this field is so we can receive the most accurate feedback, given our targeted audience consists of developers. Both Dr. Igor Steinmacher and team Match Source see the novelty of this product, however these tests will allow us to gain the perspective of end users. We will be able to examine how useful they find this product, including how user friendly it is at the current state. Consequences of bad design are noteworthy, however will not ruin the product. The point of this

testing is to identify each flaw and correct it for the experience of end users. We will primarily consider product viability and user experience in the following testing plan.

#### 4.1 Questionnaire

The user will select the Questionnaire tab on our website. Next they will be taken to a page where they will create a username, password, and insert their email. Lastly, they will be given a filterable search bar to select their skills. Once completed, they will receive an email with all of their selected skills and those skills will be put into the project finder section. During this process, we will be examining how the user navigates through this portion of the site and how pleased they are with the layout. This viewing will allow us to identify any interface confusion and any issues with the questionnaires content.

Usability Test	Input	Expected Output
Questionnaire	Username, Password, Email, Question Answers	Email to user specifying skills determined

#### 4.2 Project Finder

The user will input their GitHub username to start the Project Finder. We will be examining how they interact with our website to access this content. We want them to quickly identify the correct course of action to take, without any confusion. Ideally the product users will have come to our website with the intention of contributing to OSS. We want them to be introduced to the website and quickly lead to the Project Finder. If we notice any confusion with the website layout, we will be able to make appropriate changes accordingly.

Usability Test	Input	Expected Output
Project Finder	GitHub Username	Match history populated with results

### 4.3 Match History

The user will have access to a ‘Match History’ section on our website. This will allow them to view matches and reorder the list with our sorting functionality. We will examine whether or not the variety of sorting options is comprehensive enough to please all users. We will also be seeing how quickly they identify this section and its purpose. This will allow us to identify any site layout flaws or function failures.

Usability Test	Input	Expected Output
Match History	Sorting Selection	Match results sorted accordingly

## **Conclusion**

In summary, our team will be building a web application for Dr. Igor Steinmacher in order to support his efforts of encouraging newcomers to contribute to open source software. The application will be able to mine a user's profile, mine an OSS project for skills required to contribute, and a tool matching a user's skills to available projects. We hope our application will help solve some of the problems newcomers face. In return, more people will contribute and more projects will be recognized with support.

This document described our test plans, that will show anyone that our functional and nonfunctional requirements are not only met, but function as they should with no errors. To do so, our team will be conducting three different types of tests, these include: unit testing, integration testing, and usability testing. Unit testing will focus on specific sections of code and their outcome. Integration testing will be used for our interfaces and the flow of data between our

components. Usability testing will evaluate how a user feels using our product. These tests will ensure that we are delivering the best possible product.

## References

- [1] <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>