McMaster University

Test Plan Revision 0

Capstone Team 14
Ananthan Kanagasabai, Andrei Ciontea, Curran Tam, Joseph Nguyen, Victor Siu

supervised by
Dr.Sarah Khan, Wenbo He

November 2, 2016

Contents

List of Figures

List of Tables

1 Overview

This document will detail the testing that will be done to our website. The document covers:

- Proof of concept in section 2
- Testing system components in section 3
- Testing requirements in section 4
- Timeline of testing in section 5
- Testing survey in section 6

1.1 Test Case Format

The testing throughout the document will be arranged using the following outline:

Test Number: Number of the test

Description: Explains what the test is about

Type: What class of testing does this fall under

Tester(s): Which group will be responsible for testing

Pass: What has to happen for this test to be a pass

1.2 Automated Testing

Automated testing will be used for some of the requirements of the program, such as response time and bandwidth usage. Other tests to be done include entering variables into the form elements (including valid and invalid variables) and detailing their output.

1.2.1 Testing Tools

The testing tools to be used are:

- Microsoft Edge Developer Tools

1.3 Manual Testing

Manual testing will be done to all components of the website where the website cannot be automated or where the time taken to create an automated test would surpass the time it would take to test it manually. The manual tests will be completed by the developers.

Test 1.3.1:

Description: Manually test that clicking on links and buttons will direct the user to the correct pages.

Type: Manual

Tester(s): Developers

Pass: All links successfully redirect the user to the intended page.

Test 1.3.2:

Description: Test the website with several different screen resolutions.

Type: Manual

Tester(s): Developers

Pass: The website must be easily read and accessed on multiple devices such as mobile phones, laptops, and larger monitors.

Test 1.3.3:

Description: Test the website on different operating systems.

Type: Manual

Tester(s): Developers

Pass: The website must be accessible on Windows, Linux and Mac operating systems.

1.3.1 User Experience Testing

The manual testing will be done by humans in order for the developers to get an idea of the user experience. Testing will be done individually by people not involved in the development of the website. Each tester will have to complete a survey detailing their experience with the website. Testers will be used multiple times depending on any additions being made to the website.

Test 1.3.3:

Description: Test that website is usable by the target demographic.

Type: Manual

Tester(s): Testing Group

Pass: The website must be easy and intuitive to use by users of age 10+ (The website is intended for the use of doctors but testing with a younger age group can guarantee ease of use).

2 Proof of Concept Testing

A proof of concept test will be carried out before the website and application will be developed to ensure that it works theoretically. Below are further details of the proof of concept test.

2.1 Significant Risks

Completing the project successfully demands overcoming the following significant risks:

1. The core function of our application requires successful implementation of the required medication found on the medical database, with potential of javascript coding involved.

2. We intend for our website to be compatible with all common browsers such as Microsoft Edge, Firefox, Google Chrome, as well as iOS and Android devices.

3. Users will require internet access in order to use the features of the application.

2.2 Demonstration Plan

For our demonstration we will produce a basic simulation of the website's process, and how it operates. In this website, you will be able to access a public tool which will be able to develop a schedule for taking prescribed medicine.

The user will first be met with a form, which will ask for details on a patients physical and medical attributes, such as age, sex, build, blood type, etc. After filling out the form, a suggested schedule for the medication needed will be displayed for the user in a tabular format by the days of the week and hours. In addition, warnings and side effects of the prescribed medication will be listed and made aware for the user.

2.3 Proof of Concept Test

Below is the test case format for the proof of concept.

Test 2.3.1: Proof of Concept

Description: Tests whether the significant risks presented in the implementation of the operations of our website can be overcome.

Type: Manual

Tester(s): Developers

Pass: A successful demonstration of a simulated process of operating the website and creating a medical schedule.

3 System Testing

The algorithm for our web application will be tested first, as it is the most important feature of the webpage. User experience and UI testing will commence after this.

3.1 Web Application Mechanics Testing

The mechanics of our HIV medication choosing algorithm will be tested in both a manual and automated environment. Manual tests will be performed to mimic the perspective of the users and automated tests will ensure the algorithm's correctness.

3.2 Input Testing

Test 3.1.1.1: Valid input

Description: Tests that valid inputs (in the forms) will return the desired output.

Type: Automated

Tester(s): Developers

Pass: Valid inputs will return the output that is intended for that set of inputs. (Large number of possibilities which is why it should be automated).

Test 3.1.1.2: Warning message from invalid input

Description: Tests that invalid inputs (in the forms) will return a warning message.

Type: Automated

Tester(s): Developers

Pass: Invalid inputs must return a detailed warning to the user to correct their values. Invalid inputs include using numbers, letters and special characters where it is specified not to.

Test 3.1.1.3: Recommended drug combination

Description: Example inputs: 14 yrs old (textbox), can only take medication orally (check box), has history of liver problems (checkbox), resistance to primary class of drugs (dropdown) (Further research must be done to find all combinations and they will be included in revision 1 of the test plan).

Type: Automated

Tester(s): Developers

Pass: Example Output: The recommended drug combination is tenofovir + lamivudine (or emtricitabine) + efavirenz.

Test 3.1.1.4: Submit button

Description: Testing that submit button links to accurate page.

Type: Automatic

Tester(s): Developers

Pass: The website must link to an online page along with another local page.

Test 3.1.1.5: Color of valid input

Description: Testing responsiveness for valid inputs.

Type: Manual

Tester(s): Developers

Pass: The website will change the colour of a form element when the user provides a valid input.

Test 3.1.1.6: Reset button

Description: Testing the reset button.

Type: Manual

Tester(s): Developers

Pass: The website must clear all form element inputs when the button is pressed.

Test 3.1.1.7: Confirm button

Description: Testing the form confirmation button.

Type: Manual

Tester(s): Developers

Pass: The user's information will not be evaluated/submit unless the terms and conditions checkbox is checked.

Test 3.1.1.8: Print button

Description: Testing the print button.

Type: Manual

Tester(s): Developers

Pass: The website must print the results page when the button is pressed.

4 Requirements Testing

After completing the implemention of the system, the developers will perform testing to ensure that the application will fulfill all requirements mentioned in the Requirements Document.

4.1 Functional Requirements Testing

All of the functional requirements in the Requirements Document should be implemented in the final vision of the application. In order to be readable for the testing, the testers will use the testing checklist to ensure that the functional requirements would be fulfilled.

Test 4.1.1: Functional requirements

Description: Compare the end product with the software requirements document and make sure all features have been implemented.

Type: Functional

Tester(s): Developers

Pass: This test passes if all listed functional requirements from the software requirements documents are present in the end product.

4.2 Non-Functional Requirements Testing

The following tests display the non-functional requirements in the Requirements Document that should be fulfilled in the final version of the application.

Test 4.2.1: Web Browser Support

Description: The website runs on all major web browsers (i.e. Firefox, Google Chrome, etc.).

Type: Functional

Tester(s): Functional

Pass: This test passes when the website is accessed and used on all listed web browsers.

Test 4.2.2: Language, Spelling and Grammar

Description: The language in used in the website is English and there are no spelling errors.

Type: Functional

Tester(s): Developers

Pass: Autocorrect must not detect any spelling errors on the website.

Test 4.2.3: Hardware

Description: The website can run on a mobile device(i.e. Laptop, cellphone, etc).

Type: Functional

Tester(s): Developers

Pass: The user can access the website by using laptop and cellphone.

4.2.1 User Experience Testing

The testing group will test the website to ensure that the following non-functional requirements will be fulfilled.

Test 4.2..1.1: Location

Description: The user can access the website anywhere.

Type: Functional

Tester(s): Testing group

Pass: The users can access the website if their mobile device can connect to the wifi.

Test 4.2.1.2: Safety-Critical

Description: Private information about patients is only viewed by the users.

Type: Functional

Tester(s): Testing group

Pass: The private information about patients can be viewed only if the user login. Only user can be access and modify the medical information of patients.

5 Testing Timeline

This document roughly outlines the testing timeline that will be followed.
All items are arranged in chronological order.

Completion Date	**Responsible Party**	**Task**
21/11/2017 - 25/11/2017	Developers	Complete proof of concept demo
Late February 2017 Tentative	Developers	Automated test cases
Late February 2017 Tentative	Developers	Manual test cases
Early March 2017 Tentative	Testers	Complete user experience survey part 1
Late March 2017 Tentative	Testers	Complete user experience survey part 2

Table 1: Testing Timeline

6 Appendix A: Testing Survey

The testing survey will be completed by two groups. The first group will give us any feedback regarding the website. The second group will give any feedback regarding the updates.

This survey will be completed after using the application for 5 minutes.
Provide a rank between 0 and 5 for the following categories:
Ease of use: 0 1 2 3 4 5 (5 being the easiest)
Navigation: 0 1 2 3 4 5 (5 being the easiest)
Visual Appeal: 0 1 2 3 4 5 (5 being very appealing)
Utility: 0 1 2 3 4 5 (5 being very useful)