Specific Test Plans

Installation Test

We will follow the installation instructions on BlueHost, where our application is deployed. After the deployment, we will first access the URL to check if the application is correctly loaded and then check basic functionalities of the application. Then we will run all the unit tests to make sure the other functionalities of the application works on the host.

Robert performed this test successfully with no issues. The deployed site loads correctly, with reasonable speed, and passes all unit and requirements tests

Usability Test

We will ask customers to continue using the application and providing us with sufficient feedback.

The Usability testing will first be conducted in our group. We will collect group membersâ concerns with regard to the usability of this application.

2 Members of other groups will also use our application and give us feedbacks based on their technical background and understandings of what usability requirements our application should meet.

We will also find 6 people without technical background to use the application and perform a set of tasks. We will keep track of which task they spend the most of their time, ask them where they think our application is confusing and what improvements we should make.

People are going to test basic functions of the application:

1. creating houses and residents
2. create a new appointment
3. find an appointment
4. find a doctor
5. make new reservation for a car
6. generate a report

People will be asked with the following questions:

1. Which task did you find most difficult?
2. Did you run into any problem performing any of these tasks?
3. Which part of the system do you like the most?
4. Do you have any further opinions or concerns?

We each asked friends and members of other groups to thoroughly test our system. We added more questions and tasks to the list above, asking testers to change passwords and delete car reservations, among other things. All testers were able to complete the required tasks and rated the appoinment creation process either a 4 or a 5 on a usability scale from 1 to 5. Bringing up the details of a newly created appointment was the only repeated confusion, although testers were able to resolve this. We are considering changing our language used to make this more clear. Overall, we are very pleased with the positive feedback from our testers.

Security Testing

Unit Tests

Security testing will be largely carried out through the unit tests, which will make sure that the user must be properly authenticated to access each part of the website. We have multiple levels of authentication, including admin, staff, volunteer, and workstation head. Each of these levels will have permissions specifying which parts of the site they can access, which the unit tests will ensure is possible. As part of our requirements testing, will will also manually ensure that unauthorized users cannot access parts of the site or perform actions that they are restricted from.

All united tests passed to ensure that user levels functioned as desired, preventing access to certain pages and allowing only admistrators to make changes to records that they did not create.

Malformed URLs

Security testing will include making sure that users without access to a specific page or feature cannot navigate to that page as the appropriate links will be hidden to them. Trying to manually navigate there through URLs will return users to the main page and display an error, as with trying to navigate to any non-existent part of the website.

All relevant URLs were tested to make sure that non-administrators could not access pages that they were not authorized to see. We also tested their ability to perform actions (New, Update, etc.) by using the correct URL. All unauthorized attempts were denied access and redirected to the main page with a notification that they did not have the required privileges to access such content.

SQL Injection

We should make sure that there is no way to access the application through SQL injection. Different ways of SQL injection should be tested and any possible security issues should be addressed immediately.

We ensured that form responses were sanatized to prevent SQL Injection attacks.

Password Choice

Users should not be able to create password thatâs less than 8 characters and complicated password combinations are encouraged.

Password length requirements are fully enforced. This is confirmed by testing.

Requirements Testing

The tests will each contribute to testing one of the given requirements. The unit tests will test all code that contributes to meeting these requirements. However, the requirements will also be tested manually, to ensure that the code works together properly and as intended. The requirements can be found under the innisfree github wiki here.

All requirements were carried out on both the course server and the deployment server. All requirements tests passed completely and without issue. In addition, our code is 100% covered by unit tests and all tests are passing.

Compatility Testing

We will execute the requirements test on the most recent version of the major web browsers (Firefox, Chrome, IE, and Safari). If — and only if — the requirements test passes on all of those browsers, then the compatibility test will be considered to have passed.

The following browsers successfully passed compatibility testing: Firefox, Chrome, Safari, and Internet Explorer. All requirements were able to be carried out on each browser.