**1.1 Test Coverage**

The following section covers the testing phase of the project. Testing is a key concept when it comes to the delivery of a particular product. The reason being is that allows for the workers to assess the early problems in the software and to ensure that the requirements of top priority are managed, since these functionality may danger the overall software. The section is divided into different test categories:

* Tested Items
* Untested Items
* Unit testing
* Requirement testing

**Tested Items**: Defined as the features that were already tested. It results into two possible outcomes: pass or fail.

**Untested Items**: Defined as the features that were not already tested due to a specific reason which results into not prioritizing that particular matter.

**Unit Testing:** Defined as testing a small part of the software, in other words, a small part of the code is evaluated to see if it works.

**Requirements:** Defined as the essential functionality that a user, student and admin must be able to accomplish with the software at hand.

* + 1. **Tested items**

The two units which were tested on consist of the authentication unit and the preference unit.

**Authentication**: This was immediately tested since the requirement of signing up and logging forms the basis of the whole use of the software. This consists of ensuring that the user is successfully directed to his/her appropriate page. This item is the number one priority because this functionality allows access to the Scheduler.

**Manage Preferences**: This unit was tested because the integration of preferences demands to be of creative thinker when generating the schedule. This is of great importance because the system’s requirement is to take into consideration the things that the student wants and testing solely on effectively containing, changing and deleting the preferences proves to be of big priority.

**Stress and Security**: This is an important key because the project, assuming it will be used by a certain client for a big academic institution, provides access to personal information. These tests are of importance since they provide a safety margin. Without protected and secure access, the software would suffer and, nonetheless the clients, since its functionality is overshadowed by not being secured and properly stressed.

**1.1.2 Untested Items of Interest**

**Remaining server calls:** There are server calls for obtaining all courses, sections, and user information as well as calls for updating all of that information and calls for generating the schedule. These calls can be unit tested the same way the other calls were. They are important to test because the user interface relies on the server calls behaving as expected. Whether they succeed or fail, they must do it in the way the UI would expect.

**The user interface:** This was user tested, but it was not tested programmatically because there is no official React unit testing framework. There are some open source frameworks that we did not look too much into because they are very new and are neither robust nor heavily tested. Unit testing a UI is also much more complicated than testing a single function that handles data. If we were to unit test, we would use one of the open source React unit testing frameworks and edit it as needed. It is important to test the UI because this is the user facing portion of the application. If this does not perform as expected, it is immediately apparent to the user and does not look professional. User testing does not always pick up on all of the bugs the same way a unit test does, so unit testing the UI would thoroughly verify that it performs as expected.

**1.1.3 Requirements**

|  |  |  |
| --- | --- | --- |
| **Student & Admin** | | |
| **Feature** | **Use Case** | **Related Test Cases** |
| **Login/Logout** | **UC1, UC2** |  |
| **Reset Password** | **UC4** |  |
| **Edit account information** | **UC5** |  |

|  |  |  |
| --- | --- | --- |
| **Admin** | | |
| **Feature** | **Use Case** | **Related Test Cases** |
| **Add course to program** | **UC13** |  |
| **Delete course** | **UC14** |  |
| **Edit course** | **UC15** |  |
| **Add section** | **UC16** |  |
| **Edit section** | **UC17** |  |
| **Delete section** | **UC18** |  |

|  |  |  |
| --- | --- | --- |
| **Student** | | |
| **Feature** | **Use Case** | **Related Test Cases** |
| **Sign up** | **UC3** |  |
| **Set preferences** | **UC6** |  |
| **Set needed courses** | **UC7** |  |
| **Delete needed courses** | **UC8** |  |
| **Set taken courses** | **UC9** |  |
| **Delete taken courses** | **UC10** |  |
| **Generate schedule** | **UC11** |  |
| **View auto-generated list of taken courses** | **UC12** |  |

**1.1.4 Other Untested Items**

|  |  |
| --- | --- |
| **Student & Admin** | |
| **Confirm email** | **Too complex** |
| **Delete schedule** | **Unlikely thing for user to do** |
| **Change section** | **Low priority** |
| **View default schedule** | **Other components need to be executed first.** |
| **Save schedule** | **Other components need to be executed first.** |
| **Print schedule** | **Other components need to be executed first. Too complex (limited knowledge to make it work).** |
| **View schedule** | **Other components need to be executed first.** |

**1.1.5 Units**

|  |  |
| --- | --- |
| **Unit** | **Tested?** |
| **Login/Sign up** | **Yes (Dias & Sean)** |
| **Manage courses** | **No** |
| **Manage preferences** | **Yes (Gabriele)** |
| **Manage account information** | **No** |
| **Manage course DB** | **No** |
| **Generate Schedule** | **No** |

**1.2 Test cases**

Insert intro here

**1.2.1 Unit testing**

Explanation

Unit test 1

Unit test 2

Unit test 3

**1.2.2 Requirements testing**

Explanation + black box testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC1 | Login | | | | |
| ID | Description | Expected Output | Result | Bug ID | Comments |
| 1.1 | User inputs valid username and password | User is redirected to student main page |  |  |  |
| 1.2 | User inputs a valid password and "admin" username | User is redirected to admin main page |  |  |  |
| 1.3 | User inputs invalid username and/or password | "Incorrect password and/or username" |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC2 | Logout | | | | |
| ID | Description | Expected Output | Result | Bug ID | Comments |
| 2.1 | User clicks on "logout" button | User is redirected to student main page |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC3 | Sign up | | | | |
| ID | Description | Expected Output | Result | Bug ID | Comments |
| 3.1 | User inputs valid username, email and password. |  |  |  |  |
| 3.2 | User inputs a valid password and username, but an invalid email. |  |  |  |  |
| 3.3 | User inputs a valid password and email, but an invalid username. |  |  |  |  |
| 3.4 | User inputs "admin" as a username, with a valid password and email. |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC4 | Reset Password | | | | |
| ID | Description | Expected Output | Result | Bug ID | Comments |
| 4.1 | User clicks on "reset password" option on the login page, and inputs a valid email. |  |  |  |  |
| 4.2 | User clicks on "reset password" option on the login page, and inputs an invalid email. |  |  |  |  |