**Tested Items**

The items which were tested were based on the functional requirements listed in Deliverable 1, Section 3.1. However, some of the requirements specified in Deliverable 1 were scoped out due to time constraints, and scoping down the type of users to include only the student user. Priority was thus given to core requirements needed for the application to function as needed.

The following features were maintained within the application and were subsequently tested as well.

|  |  |  |  |
| --- | --- | --- | --- |
| **Login** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC01 | High | This feature was tested as it is necessary for a user to be able to enter the application using their own credentials, else it would be rendered useless as the user cannot enter the system. | 1.1, 1.2, 1.3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Logout** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC02 | High | This was tested as once a user is done utilizing the application, they should be able to log out of the application. If not, there is a security risk as well as tying up resources with the system. | 2.1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Choose Course Preferences** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC06 | High | This feature was tested as a user should be able to set when they would prefer their courses to occur, or if they do not care for the timing. This information is required to generate the appropriate schedules. | 6.1, 6.2, 6.3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Search Courses** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC07 | High | This feature was tested as a user has to absolutely be capable of searching for the course they want in order to add it to their schedule, else making a schedule would be impossible. | 7.1, 7.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **View Course Description** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC08 | Low | This feature was tested as a student should be able to view the description of the course they are taking. However, its priority is not that high as it would not hinder the process of adding a course to the schedule. | 8.1, 8.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Generate Schedule** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC09 | High | This feature was tested as a student should absolutely be able to generate a schedule based on the courses they have selected for the semester in question, else the core function of the application would be rendered useless. | 9.1, 9.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Generate Sequence** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC10 | High | This feature was tested as a student should be able to generate the entire sequence of their program based on the courses they chose per semester, and this is a core functionality of the system. | 10.1, 10.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Save Schedule** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC11 | High | This feature was tested as a user should be able to save all the schedules they have generated, else they would not be able to retain any of the generated schedules. | 11.1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **View Saved Schedule** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC12 | High | This feature was tested as a user should be able to view all the schedules they have saved, and thus be able to choose which schedule is to their preference. | 12.1, 12.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **View Student Schedule** | | | |
| Related Use Case | Priority | Comments | Related Test Cases |
| UC13 | High | This feature was tested as a user should be able to view their current schedule, else it would make choosing courses for the next semesters difficult and compromise the convenience of the system as a result. | 13.1 |

**Untested items of interest**

Below is a list of omitted features that have been scoped out from the functional requirements outlined in Deliverable 1, Section 3.1. Since they were omitted, they were not included in the application, nor were they tested. Had they been included, they would have been subjected to requirements testing as were the included tested items, and tested to see if each feature fails or passes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Related Use Case** | **Related Test Cases** | **Reason for Omission** |
| Update Personal Information | UC03 | 3.1, 3.2, 3.3, 3.4, 3.5 | Omitted due to time constraints |
| Review Course History | UC04 | 4.1 | Omitted due to time constraints |
| Update User | UC20 | 20.1, 20.2 | Omitted due to scoping out System Admin |
| View Degree Audit | UC05 | 5.1 | Omitted due to time constraints |
| Print Schedule | UC14 | 14.1, 14.2 | Omitted due to time constraints |
| Create Course | UC15 | 15.1, 15.2 | Omitted due to scoping out Faculty Admin |
| Remove Course | UC16 | 16.1, 16.2 | Omitted due to scoping out Faculty Admin |
| Update Course | UC17 | 17.1 | Omitted due to scoping out Faculty Admin |
| Create User | UC18 | 18.1 | Omitted due to scoping out System Admin |
| Remove user | UC19 | 19.1, 19.2 | Omitted due to scoping out System Admin |
| View User | UC21 | 21.1, 21.2 | Omitted due to scoping out System Admin |
| Update Course Grades | UC22 | 22.1 | Omitted due to scoping out Proffesor user |
| View Professor Schedule | UC23 | 23.1, 23.2 | Omitted due to scoping out Proffesor user |

Additionally, due to time constraints, we were unable to test some important features of our website, especially in terms of unit testing. For the purposes of this report, unit testing was only carried out on 2 units. However, the application would benefit from writing stubs and drivers for each function, class, and subsystem within the system. As a result, it would become much more apparent where code could be improved or is failing, and could be corrected, providing for a more robust system as a result. The unit tests for all the functions within the application would be carried out in a similar manner to the tests carried out for the two test units contained within this report.

Another test of interest would be supplementary security tests. Currently, the laravel framework used for this application has its own mechanisms to prevent SQL injections from occurring. However, it would be useful to provide additional security testing using other tools so as to ensure the application is completely secure, as the database contains a lot of private information about its users that should not be compromised.