1. Testing Process

Introduce the testing chapter.

* 1. Unit Tests – generate the unit test cases for the classes. You **must** use a unit testing tool e.g., Junit and a code coverage tool EclEmma.
  2. Subsystem Tests – test at least one subsystem using test cases derived from the systems test cases. This will involve the creation of a test driver i.e. a main in the package containing the subsystem and one or more stubs. Use a similar format to that in section 7.1 to document the tests performed. Include the code for the test driver in appendix E. You **must** use a unit testing tool and a code coverage tool.
  3. System Tests – use format described in class. Note the systems test cases should be derived from the use cases (i.e. scenarios). For each use case implemented you should write at least 3 test cases (2 sunny day and one rainy day). **Must** use an automated system testing tool, e.g. Rational Functional Tester or Selenium.

Selenium IDE Chrome Extension has been used for the total of 40 system Tests.

* 1. Evaluation of Tests –
     1. Use a table to show all the test results and comment on how to possibly fix the bugs. Columns of table (*Test Case ID, Pass/Fail, Fix*)

|  |  |  |
| --- | --- | --- |
| Test Case ID | Pass/Fail | Fix |
| SOS32-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS32-System-\*\*-Sunny02\*\* | Pass | N/A |
| SOS32-System-\*\*-Rainy01\*\* | Pass | N/A |
| SOS32-System-\*\*-Rainy02\*\* | Pass | N/A |
| SOS07-Security-\*\*-Sunny01\*\* | Fail | Submit form data has to be fixed |
| SOS07-Security-\*\*-Sunny02\*\* | Fail | Submit form data has to be fixed |
| SOS07-Security-\*\*-Rainy01\*\* | Pass | N/A |
| SOS07-Security-\*\*-Rainy02\*\* | Pass | N/A |
| SOS16-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS16-System-\*\*-Sunny02\*\* | Pass | N/A |
| SOS16-System-\*\*-Rainy01\*\* | Fail | Input validation for the create organization form |
| SOS16-System-\*\*-Rainy02\*\* | Pass | Showing a message about duplicate organization\* |
| SOS01-System-\*\*-Sunny01\*\* | Fail | Time picker message format should be modified to match DB entries |
| SOS01-System-\*\*-Sunny02\*\* | Fail | Time picker message format should be modified to match DB entries |
| SOS01-System-\*\*-Rainy01\*\* | Pass | Showing a message about blank fields\* |
| SOS01-System-\*\*-Rainy02\*\* | Pass | N/A |
| SOS31-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS31-System-\*\*-Sunny02\*\* | Pass | N/A |
| SOS31-System-\*\*-Rainy01\*\* | Pass | Showing an error message\* |
| SOS31-System-\*\*-Rainy02\*\* | Pass | Showing an error message\* |
| SOS22-System-\*\*-Sunny01\*\* | Pass | Does not automatically login right after registration\* |
| SOS22-System-\*\*-Sunny02\*\* | Pass | Does not automatically login right after registration\* |
| SOS22-System-\*\*-Rainy01\*\* | Pass | Showing an error message\* |
| SOS22-System-\*\*-Rainy02\*\* | Pass | Showing an error message\* |
| SOS17-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS17-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS17-System-\*\*-Rainy01\*\* | Pass | N/A |
| SOS17-System-\*\*-Rainy02\*\* | Pass | N/A |
| SOS02-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS02-System-\*\*-Sunny02\*\* | Pass | N/A |
| SOS02-System-\*\*-Rainy01\*\* | Pass | N/A |
| SOS02-System-\*\*-Rainy02\*\* | Pass | N/A |
| SOS04-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS04-System-\*\*-Sunny02\*\* | Pass | N/A |
| SOS04-System-\*\*-Rainy01\*\* | Pass | N/A |
| SOS04-System-\*\*-Rainy02\*\* | Pass | N/A |
| SOS10-System-\*\*-Sunny01\*\* | Pass | N/A |
| SOS10-System-\*\*-Sunny02\*\* | Pass | N/A |
| SOS10-System-\*\*-Rainy01\*\* | Pass | N/A |
| SOS10-System-\*\*-Rainy02\*\* | Pass | N/A |

* + 1. Use a table to show the code coverage for the unit, subsystem and systems testing. Columns of the table (*Type of testing, Number of Test Cases, Number of Bugs, Code Coverage*)