

# ELEN4010 - Software Development Project

## Testing Framework

Kanaka Babshet (678851) and Ari Croock (718005)  
Daniel Weinberg (547937) and Alice Yang (597609)

2016/04/19

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Testing Layout</b>	<b>2</b>
<b>3</b>	<b>Homepage</b>	<b>2</b>
<b>4</b>	<b>Student and Academic pages</b>	<b>2</b>
4.1	Student . . . . .	2
4.2	Academic . . . . .	2
<b>5</b>	<b>Booking Meetings</b>	<b>2</b>
5.1	Student . . . . .	2
5.2	Academic . . . . .	3
<b>6</b>	<b>Meeting Success</b>	<b>3</b>
<b>7</b>	<b>Test Information</b>	<b>4</b>
<b>8</b>	<b>Acceptance Tests</b>	<b>5</b>
<b>9</b>	<b>Conclusion</b>	<b>6</b>

## **1 Introduction**

Fundamental features are implemented across every webpage, and need to be tested on every microlevel. Therefore, all tests are written from the bottom up, and the system success is dependent on the passing all of these tests.

## **2 Testing Layout**

The front end has been tested using Selenium, a suite of tools to automate web browsers across various platforms.

## **3 Homepage**

The first test ensures that the welcome page can be loaded. The welcome page allows a user to specify whether they are a student or an academic. Should the user be an academic, they can choose their identity from a drop-down menu.

## **4 Student and Academic pages**

### **4.1 Student**

Tests are done to ensure that the correct student page is loaded upon selecting the "Student" option on the homepage. Another test verifies that the student can then submit the academic to book a meeting with, along with the desired week of meeting.

### **4.2 Academic**

Tests are done to ensure that the correct lecturer page is loaded upon submitting the academic identifier. Another test verifies that the academic can then submit the desired week of meeting.

## **5 Booking Meetings**

The loading of the academic's schedule is tested.

### **5.1 Student**

Once the student has selected the preferred meeting time slot in the academic's week schedule, they are directed to a booking form where they can provide their student ID and meeting subject.

## **5.2 Academic**

Once the academic has selected the preferred meeting time slot in their week schedule, they are directed to a booking form where they can provide the meeting subject, meeting type (group or individual), and the student(s) that will be attending the meeting. The academic also has the ability to specify whether a meeting session can be allocated to a group. Once this has happened, the calendar displays "Join" to the students so that they are able to join an existing meeting. The subject of the meeting however, remains the same.

Tests are done to ensure that all the correct elements required for filling the meeting information are available on both booking forms, and that all this information can be submitted successfully.

## **6 Meeting Success**

Once the meeting has been successfully booked (from either the student or the academic), the user can choose to return to the homepage. A simple test is conducted to ensure that the homepage is correctly loaded upon clicking that link.

## 7 Test Information

Table 1: Table demonstrating the test plan and desired outcomes

Test No.	Test	Expected Outcome
1	The correct title is given to the welcome page	It is named "Welcome Page"
2	The student page can be accessed from the welcome page	The user is directed to the student page
3	The correct title is given to the student's page	It is named "Student's Page"
4	All correct elements are found on the student's page	Find the lecturer ID field, and date field for desired week
5	The information on the student's page can be submitted correctly	The user is directed to the lecturer's schedule for the week
6	The lecturer page can be access from the welcome page	The user is directed to the specific lecturer's page
7	The correct title is given to the lecturer's page	It is named "Lecturer's Page"
8	All correct elements are found on the lecturer's page	Find the date field for desired week
9	The information on the lecturer's page can be submitted correctly	The user is directed to the lecturer's schedule for the week
10	The correct elements exist on the student booking form	Find the student ID field and the meeting subject field
11	Submit the student booking form	The user is notified of a successful meeting on a new page
12	Submit the lecturer booking form	The user is notified of a successful meeting on a new page
13	Allow returning to welcome page if meeting is successful	The user is redirected to the welcome page if they click on a link after the meeting has been booked
14	Allow Academic to specify a group booking slot	A group meeting slot is created
15	Allow students to join meetings where academic has specified a group meeting	Calendar replaces "Book" text with "Join" text for students view

## 8 Acceptance Tests

The acceptance tests listed in Table 2 assume the completion of the following preconditioned tests:

- Test that the webpage is running
- Test that the calendar is successfully integrated
- Test that academics and students can access the webpage
- Test that a student can select an academic to meet with
- Test that the student can successfully enter meeting information
- Test that an academic can view their schedule
- Test that the student can view current timeslots booked by themselves
- Test that both students and academics can book meetings
- Test that the system can store all necessary booking information from the students

Table 2: Table detailing the acceptance tests to be run

Test No.	Acceptance Requirement
1	<b>Given</b> that an academic wants to view a schedule <b>When</b> they open the webpage <b>Then</b> ensure that the calendar is visible
2	<b>Given</b> that a student requires a meeting <b>When</b> they do not access the scheduling system <b>Then</b> ensure that the academic can book meetings directly onto the system
3	<b>Given</b> that a student requires a meeting <b>When</b> they have access to the scheduling system <b>Then</b> ensure that they can successfully book a meeting
4	<b>Given</b> that an academic needs to allocate free timeslots <b>When</b> they have access to the scheduling system <b>Then</b> ensure that they can successfully specify free time
5	<b>Given</b> that a student requires a meeting <b>When</b> there is already a meeting booked in that slot <b>Then</b> ensure that the student has the option of requesting to join the meeting
6	<b>Given</b> that the student needs to cancel a timeslot <b>When</b> he can no longer attend a meeting <b>Then</b> ensure that the specified timeslot is available again
7	<b>Given</b> that an academic needs to specify recurring meetings <b>When</b> a student is required to meet an academic periodically <b>Then</b> ensure time intervals and frequency can be specified upon booking meetings
8	<b>Given</b> that an academic would like to keep record <b>When</b> meetings have occurred <b>Then</b> ensure that the system can generate corresponding reports

All acceptance tests will be implemented as user tests.

## 9 Conclusion

The testing framework explains the most efficient and robust method of ensuring the functionality of the code. As mentioned earlier, the most comprehensive way to test the code is to run user tests as they have the ability to fully cover the extent of the system functionality from the perspective of the potential consumer. The system is considered successful if all the tests written pass.