

# Test Report

Bengezi, Mohamed  
400021279

Binu, Amit  
400023175

Samarasinghe, Sachin  
001430998

Group 2: Genzter

December 6, 2017

## Contents

<b>1</b>	<b>Revision History</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
2.1	Summary . . . . .	4
2.2	Background . . . . .	4
2.3	Test Objectives . . . . .	4
<b>3</b>	<b>Results of Testing</b>	<b>5</b>
3.1	Changes Made . . . . .	5
3.2	Use of Automated Testing . . . . .	5
3.3	Results of System Tests . . . . .	5
3.3.1	Test Area 1	
	Conflict Detection Test . . . . .	5
3.3.2	Test Area 2: Output Timetable . . . . .	6
3.3.3	Tests for Non-functional Requirements . . . . .	8
3.3.4	Test Area 4: UI . . . . .	9
3.3.5	Test Area 5: Maintainability and Robustness . . . . .	10
3.3.6	Test Area 6: Browser Compatibility . . . . .	10
<b>4</b>	<b>Traceability</b>	<b>11</b>
<b>5</b>	<b>Code Coverage Metrics</b>	<b>12</b>
5.1	Equivalence Testing . . . . .	12
5.2	Boundary Testing . . . . .	12
5.3	Control-flow Testing . . . . .	12

## List of Figures

1	Example Conflict Error . . . . .	6
2	Conflict Test Results . . . . .	7
3	Example Output . . . . .	8
4	Output Test Results . . . . .	9

## List of Tables

1	Revision History . . . . .	3
2	Traceability Matrix to Requirements . . . . .	11
3	Module Labels . . . . .	11

4	Traceability Matrix to Modules . . . . .	11
---	--	----

## 1 Revision History

Date	Version	Description	Author
01/DEC/17	0.0	Created Test Report	Mohamed Bengezi, Amit Binu, Sachin Samarasinghe
06/DEC/17	1.0	Updated Test Plan, minor fixes	Mohamed Bengezi, Amit Binu

Table 1: Revision History

## 2 Introduction

### 2.1 Summary

The purpose of this document is to provide a full description of the results of the testing that was outlined and detailed in the Test Plan document. In this document, there is a section detailing the input, expected out, and actual output of all test cases specified in the Test Plan. This testing is for the Timetable Generator Project.

### 2.2 Background

The Timetable Generator is a mix of a redevelopment of [this](#) and [this](#), as well as additions of our own.

---

The testing has been conducted by the testing team specified in the Test Plan and has been conducted on the team's local machines.

---

Testing occurred concurrently with the end of the development and implementation phase of the project. This was to ensure that the implementation works as expected, and satisfies all the stated requirements.

---

### 2.3 Test Objectives

The purpose of the testing of this project is to, as stated above, ensure the presence and integrity of the stated requirements. This includes finding faults and bugs in the program, and remove them as to solidify the system, and make it as robust as possible.

## 3 Results of Testing

### 3.1 Changes Made

In regard to the results of the tests detailed in the Test Plan, there were not many changes made to the system because of the testing. There were very minor bugs that were discovered, such as bugs in the UI, as well as a fault in the main scheduling algorithm. The bug was not major enough to disrupt the system in a significant way, but it was there nevertheless.

### 3.2 Use of Automated Testing

Automated testing was used for a large part of the testing phase. We used [Mocha](#) to test conflict detection, which is when two chosen courses conflict, and there are no alternatives possible. Mocha was also used to test the output of various desired courses. For example, the input might be an Earth Science student's required courses, and Mocha would check if the courses are scheduled in the correct spots.

### 3.3 Results of System Tests

#### 3.3.1 Test Area 1

##### Conflict Detection Test

1. Type: Dynamic  
Initial State: Homepage  
Input: SFWRENG 3RA3, EARTHSC 2EI3  
Expected Output: Conflict error page  
Actual Output: Conflict error page
  
2. Type: Dynamic  
Initial State: Homepage  
Input: RELIGST-2TA3, SFWRENG 3BB4  
Expected Output: Conflict error page  
Actual Output: Conflict error page
  
3. Type: Dynamic  
Initial State: Homepage  
Input: BIOLOGY 2F03, SFWRENG 3MX3  
Expected Output: Conflict error page  
Actual Output: Conflict error page

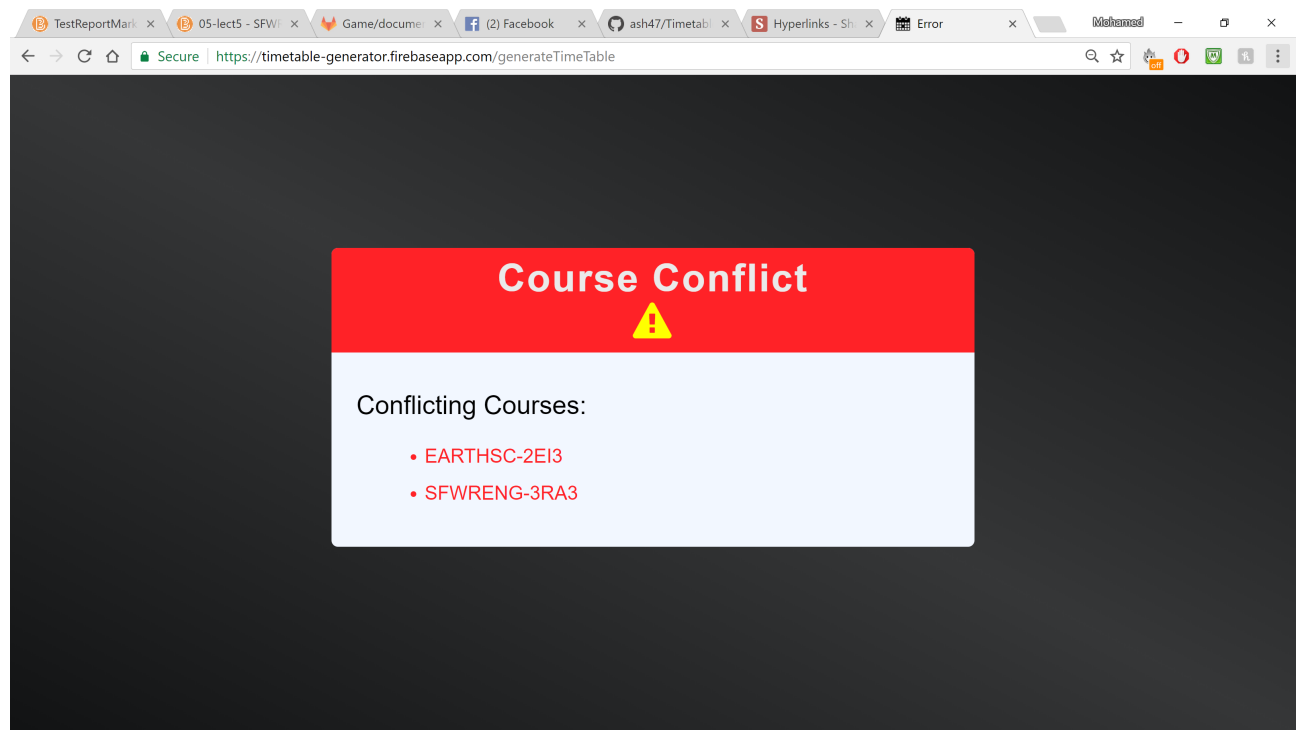


Figure 1: Example Conflict Error

4. Type: Dynamic

Initial State: Homepage

Input: MATLS 2B03, EARTHSC 2EI3

Expected Output: Conflict error page

Actual Output: Conflict error page

### 3.3.2 Test Area 2: Output Timetable

1. Type: Automated

Initial State: Homepage

Input: SFWRENG 3XA3, SFWRENG 3BB4, SFWRENG 3DB3, SFWRENG 3MX3, COMMERCE 1AA3

Expected Output: Valid Schedule in a timetable format.

Actual Output: Color-coded schedule with correct courses and timings

2. Type: Automated

```

Conflict Test 1
'SFWRENG-3RA3', 'EARTHSC-2EI3'
✓ boolean value should be true, which means there are conflicts

Conflict Test 2
RELIGST-2TA3, SFWRENG 3BB4
✓ boolean value should be true, which means there are conflicts

Conflict Test 3
'BIOLOGY-2F03', 'SFWRENG-3MX3'
✓ boolean value should be true, which means there are conflicts

Conflict Test 4
'MATLS-2B03', 'EARTHSC-2EI3'
✓ boolean value should be true, which means there are conflicts

```

Figure 2: Conflict Test Results

Initial State: Homepage

Input: ENGINEER-1C03, MATH-1ZC3, MATH-1ZB3, ECON-1BB3, PHYSICS-1E03, MATLS-1M03

Expected Output: Valid Schedule in a timetable format.

Actual Output: Color-coded schedule with correct courses and timings

3. Type: Manual

Initial State: Homepage

Input: RELIGST-2TA3, RELIGST-2QQ3, POLSCI-2I03, POLSCI-4CA3

Expected Output: Valid Schedule in a timetable format.

4. Type: Manual

Initial State: Homepage

Input: EARTHSC-2EI3, EARTHSC-2C03, GEOG-2UI3, GEOG-2GI3, BIOLOGY-2F03

Expected Output: Valid Schedule in a timetable format.



Time	Monday	Tuesday	Wednesday	Thursday	Friday
8:00					
8:30					
9:00					
9:30	SFWRENG-3BB4 C01 HSC 1A6 Emil Sekerinski	SFWRENG-3XA3 C01 T13 125 Asghar Bokhari		SFWRENG-3BB4 C01 HSC 1A6 Emil Sekerinski	
10:00	SFWRENG-3RA3 C01 TSH B128 Ryszard Janicki	SFWRENG-3BB4 C01 HSC 1A6 Emil Sekerinski	SFWRENG-3RA3 C01 TSH B128 Ryszard Janicki	SFWRENG-3RA3 C01 TSH B128 Ryszard Janicki	
11:00			SFWRENG-3MX3 T03 ITB 139 Martin George Von Mohrenschildt		
11:30					
12:00					
12:30					
13:00					
13:30	SFWRENG-3MX3 C01 TSH B128 Martin George Von Mohrenschildt	SFWRENG-3RA3 T01 BSB 137 Ryszard Janicki	SFWRENG-3MX3 C01 TSH B128 Martin George Von Mohrenschildt	SFWRENG-3MX3 C01 TSH B128 Martin George Von Mohrenschildt	
14:00					
14:30	SFWRENG-3BB4 T03 T13 107 Emil Sekerinski			SFWRENG-3XA3 L01 ITB 236 Asghar Bokhari	
15:00					
15:30					
16:00					
16:30	SFWRENG-3XA3 L01 ITB 236 Asghar Bokhari				
17:00					
17:30					
18:00					
18:30					
19:00					
19:30					
20:00					
20:30					
21:00					
21:30					
22:00					

Figure 3: Example Output

Actual Output: Color-coded schedule with correct courses and timings

### 3.3.3 Tests for Non-functional Requirements

#### Test Area 3: Speed Performance

1. Type: Manual  
Initial State: Homepage  
Input: SFWRENG 3XA3, SFWRENG 3BB4, SFWRENG 3DB3, SFWRENG 3MX3, COMMERCE 1AA3  
Expected Output: Output should be displayed within 3 seconds  
Actual Output: Schedule was displayed in less than 1 second
2. Type: Manual  
Initial State: Homepage  
Input: ENGINEER-1C03, MATH-1ZC3, MATH-1ZB3, ECON-1BB3, PHYSICS-1E03, MATLS-1M03  
Expected Output: Output should be displayed within 3 seconds

```

Output Test 1
'PHYSICS-1E03', 'ENGINEER-1P03', 'PHYSICS-1D03', 'CHEM-1E03', 'MATH-1ZA3', 'MATH-1ZC3', 'MATH-1ZB3', 'MATLS-1M03', 'ENGINEER-1C03', 'ENGINEER-1D04', 'ECON-1BB3', 'ECON-1B03'
✓ boolean value should be false, which means there arent any conflicts

Output Test 2
SEWRENG-3XA3, SEWRENG-3BB4, SEWRENG-3DB3, SEWRENG-3MX3, COMMERCE-1AA3
✓ boolean value should be false, which means there arent any conflicts

Output Test 3
'RELIGST-2TA3', 'RELIGST-2QQ3', 'POLSCI-2I03', 'POLSCI-4CA3'
✓ boolean value should be false, which means there arent any conflicts

Output Test 4
'EARTHSC-2EI3', 'EARTHSC-2C03', 'GEOG-2UI3', 'GEOG-2GI3', 'BIOLOGY-2F03'
✓ boolean value should be false, which means there arent any conflicts

Output Test 4
'EARTHSC-2EI3', 'EARTHSC-2C03', 'GEOG-2UI3', 'GEOG-2GI3', 'BIOLOGY-2F03'
✓ boolean value should be false, which means there arent any conflicts

```

Figure 4: Output Test Results

Actual Output: Schedule was displayed in less than 1 second

### 3.3.4 Test Area 4: UI

1. Type: Manual  
 Initial State: Homepage  
 Input: Any McMaster Course  
 Expected Output: Add course to list of courses  
 Actual Output: Course is added to list
  
2. Type: Manual  
 Initial State: Homepage with list of courses chosen  
 Input: Pressing the "Remove" button  
 Input: Any McMaster Course  
 Expected Output: Add course to list of courses  
 Actual Output: Course is removed from list
  
3. Type: Manual  
 Initial State: Homepage with list of courses chosen  
 Input: Pressing the "Generate" button  
 Expected Output: Timetable  
 Actual Output: Timetable

**3.3.5 Test Area 5: Maintainability and Robustness**

1. Type: Static  
Initial State: Not running  
Input: Code walk-through and inspection  
Expected Output: N/A  
Actual output: The result of this was a cleanup/commenting of the code
2. Type: Dynamic  
Initial State: Homepage  
Input: ENGINEER-1C03, MATH-1ZC3, MATH-1ZB3, ECON-1BB3, PHYSICS-1E03, MATLS-1M03, ECON-1B03, MATH-1ZA3, ENGINEER-1D04, CHEM-1E03  
Expected Output: Valid Schedule in a timetable format  
Actual output: The result was a valid, working schedule with all necessary items

**3.3.6 Test Area 6: Browser Compatibility**

1. Type: Static  
Initial State: Not running  
Input: Visit website using different browsers  
Expected Output: N/A  
Actual Output: Tested on Chrome, Edge, and Firefox, and responded well to each

## 4 Traceability

Requirement	TA1	TA2	TA3	TA4	TA5
2.7.1	X				
2.7.2		X	X	X	
3.2				X	
3.3			X		
3.5					X

Table 2: Traceability Matrix to Requirements

Module	Label
Input Parameters Module	M1
Scheduler Module	M2
Output Module	M3
Course Module	M4

Table 3: Module Labels

Requirement	M1	M2	M3	M4
2.7.1	X			X
2.7.2		X	X	X
3.2	X		X	X
3.3		X	X	
3.5	X	X	X	X

Table 4: Traceability Matrix to Modules

## 5 Code Coverage Metrics

In total, we estimate that all the tests conducted covered approximately 75% of the lines of code contained in the project.

**Code Coverage:** Measures the degree to which the source code of a program has been tested. This includes Equivalence testing, Boundary testing, and Control-flow testing.

### 5.1 Equivalence Testing

In the case of the Timetable Generator, the inputs, or courses, are divided into classes such as term 1 courses, term 2 courses, and full semester courses. This is because the generator handles each of these categories separately, so there is a greater chance for bugs and failures. Other classes could be relative to course codes. For example valid course codes, and invalid course codes.

### 5.2 Boundary Testing

Some boundary cases would be using two unit courses as a lower bound/edge, and using a 6 or higher unit course as an upper bound/edge.

### 5.3 Control-flow Testing

Obviously complete control-flow coverage is impossible, but in order to maximize coverage, test cases were used that exercised as many statements, edges, conditions, and paths as possible. It is difficult to fully describe how this was done, seeing as many tests were conducted in addition to the ones listed in this document. Test cases from various McMaster programs' schedule's were used, various different conflict tests were conducted, different amounts of courses were used, and many other tests were used to extend the coverage.