Test Report
InPress – Group 3

Abdul Hadi Muhammad (0951862)
Liu Wenbo (0970709)
Justin Kan (0843763)

Table of Contents

List of Tables/Figures

Change History

The following table shows the change history for this test plan document.

Version	Date	Author	Comments
—	—	AHM, LW, JK	Original content.
0	March 18/2014	AHM, LW, JK	Initial check in

Table 1: Change History

Non-Functional Testing

Performance Testing

Apache JMETER is used to conduct performance and stress testing. JMETER is used instead of Selenium because JMETER has the ability to simulate multiple users log-in's (including log-in time differences) and actions (such as POST / GET).

For our test purposes we focused on the Student Module where it will be heavily used compared to the instructor module. This is because there will be much more students using the Application in a fixed time frame. For example there will be 200 students and One Instructor using the Web Application within the hour versus 20 instructors using it to create their class materials.

We used a more extreme case. The settings are as follow:

Users: 200
Ramp Up Time: 1 second
Loop: Forever

We use **200 users** to simulate how the students in a large class in McMaster. This number is sufficient to cover majority of the large classes. The **Ramp Up Time** is the time between each user that is coming in (or logging in). Finally, the **Loop** is how many times this testing will be executed. To get better samples, we have set the loop to go on forever.

To simulate the usage as close as possible, stress testing is done on a laptop that is using a wireless connection. Majority of the users will be using wireless connection and very few that will be using wired connection. Therefore, wireless connection stress testing is chosen.

Performance Data:

The Min, Max and Average represent the response times of the system itself. Everything is in milliseconds (ms). The error % represents how many times the assertion has failed, or it has been timed out. Finally, the throughput is how many users were completed within the minute.

Instructor Module:

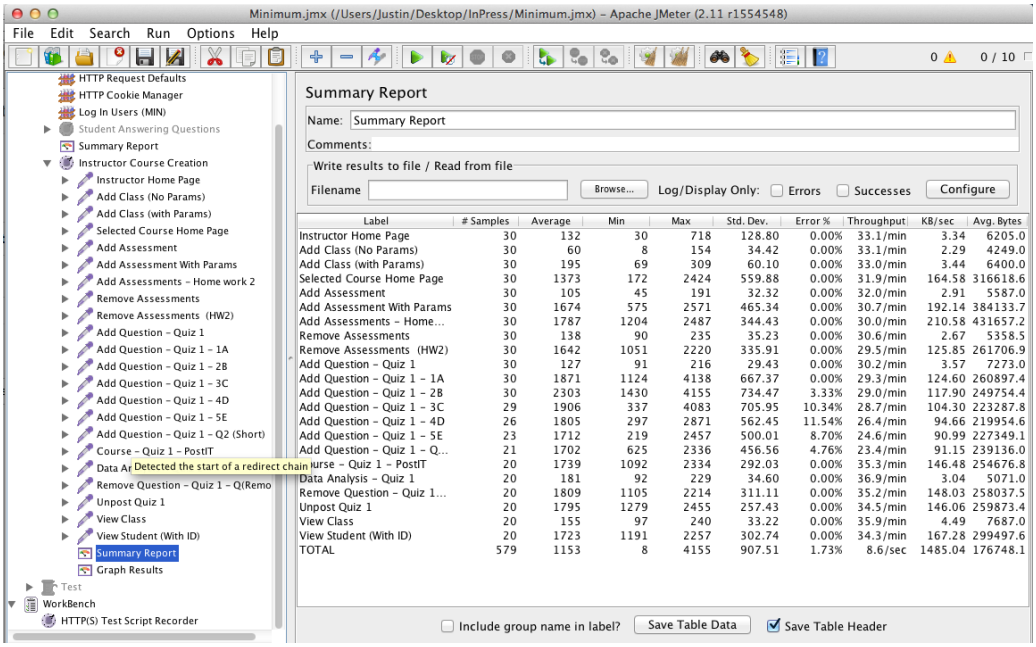

Figure 1: Performance Test: Table of results for Instructors Module

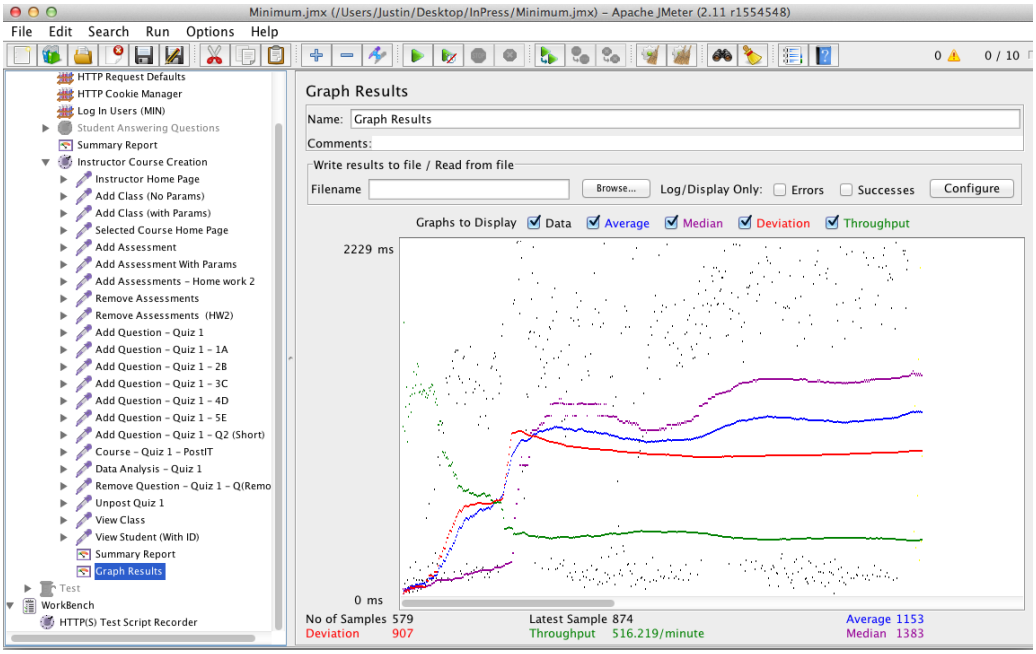

Figure 2: Performance Test: Graph results for Instructors Module

As we can see we have numbers that are bigger than most numbers in minimum and maximum. Since these are in milliseconds it is not very significant. However, this proves that under high usage, the user may experience some minor lags or even dropouts (at a very low rate).

Student Module:

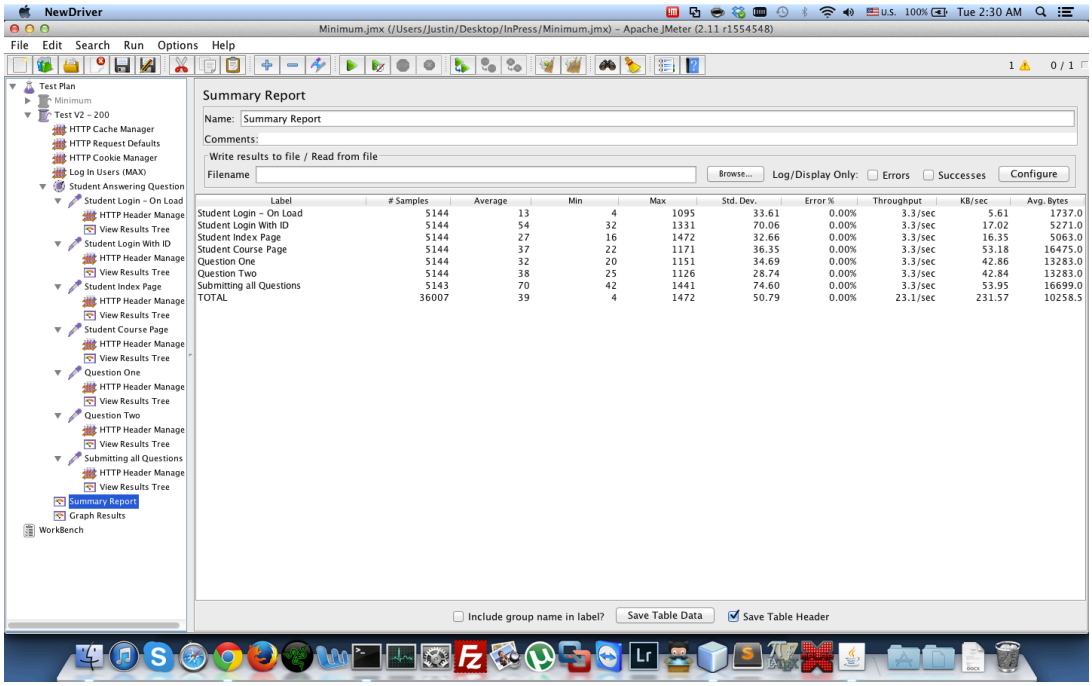

Figure 3: Performance Test: Table of results for Students Module

In comparison the Student Module performs much better than the instructor side. This is very crucial as the main focus is on the students and not the instructor. We can allow some small delay for the Instructor, but the Students should not be experiencing any delays or lag at all as it will impact the lecture progress.

Compatibility Testing

On the host side, inPress is loaded and configured on to a server with Linux (Ubuntu Distribution) installed as the operating system. For local testing, it is also possible to be loaded on Mac OSX. This application can be used virtually anywhere as long as Python, Django and Apache is supported on the host.

Host:

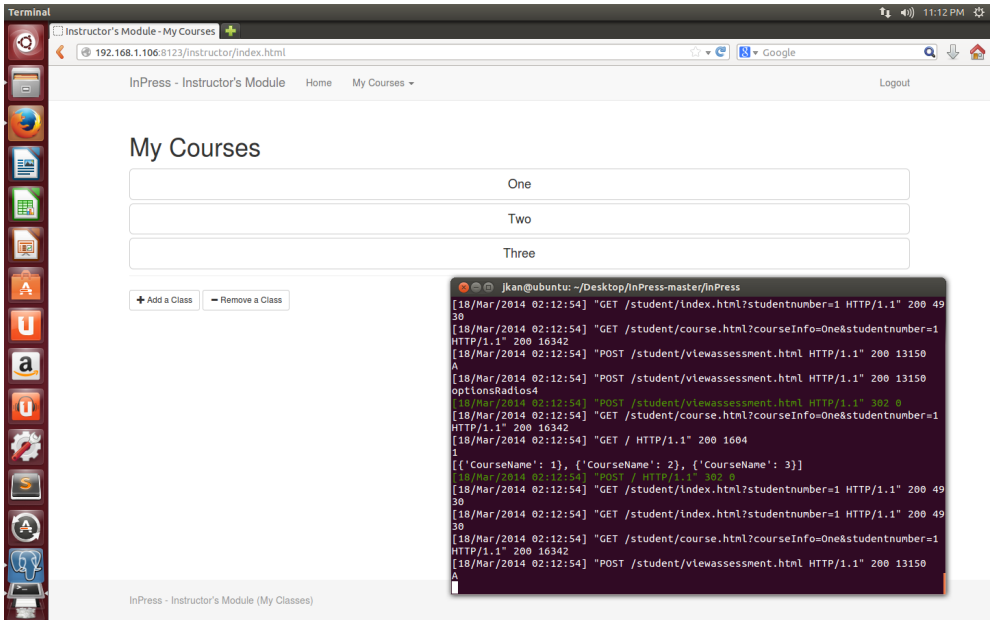

Figure 4: Compatibility with the Host running Ubuntu

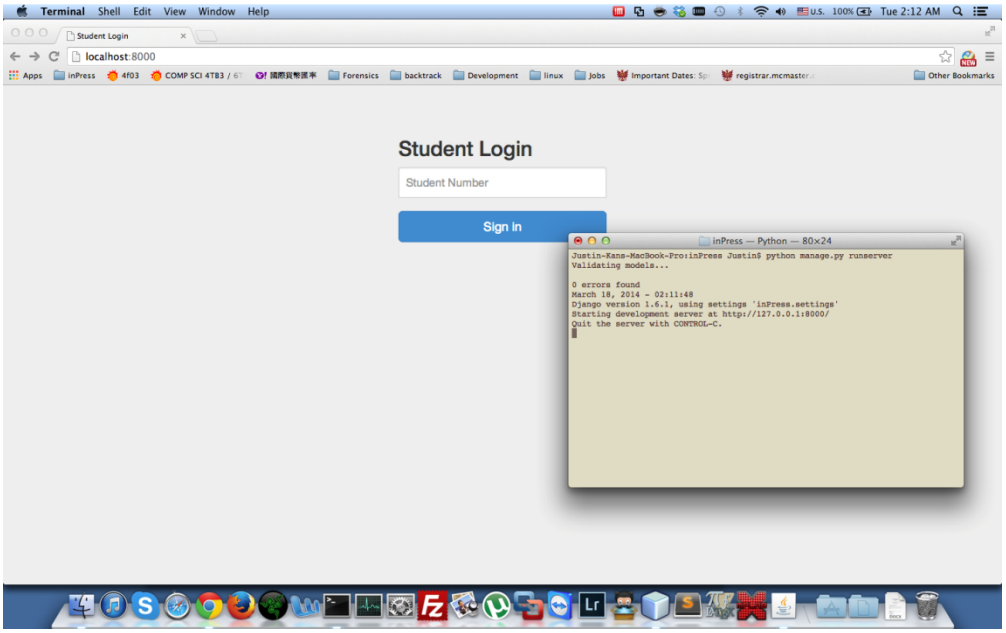

Figure 5: Compatibility with the Host running on OS X

Client:

On the Client Side it is compatible with all major operating systems and browsers. On the mobile platform iOS and Android is supported.

PC:

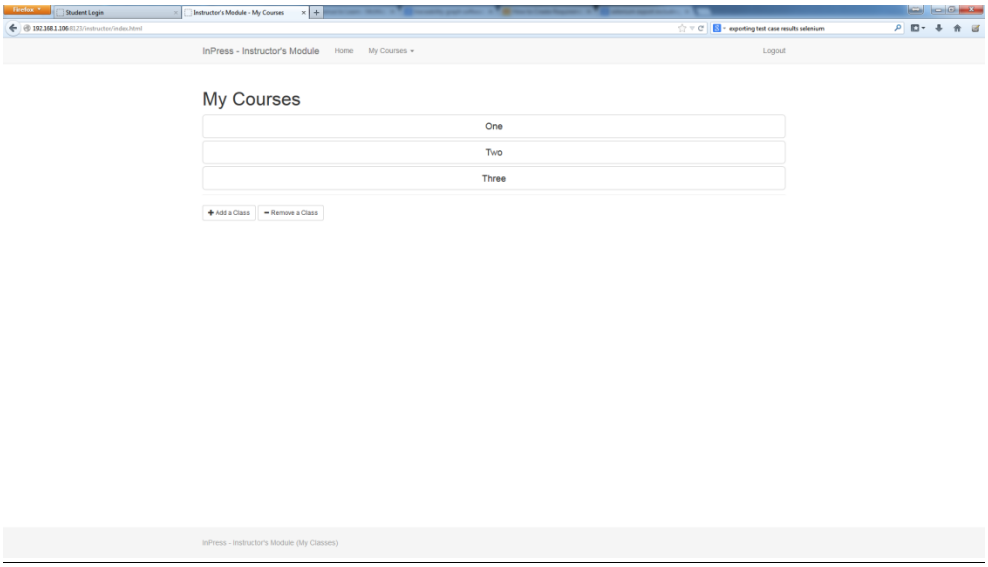

Figure 6: Compatibility: Windows 7 & Mozilla Firefox

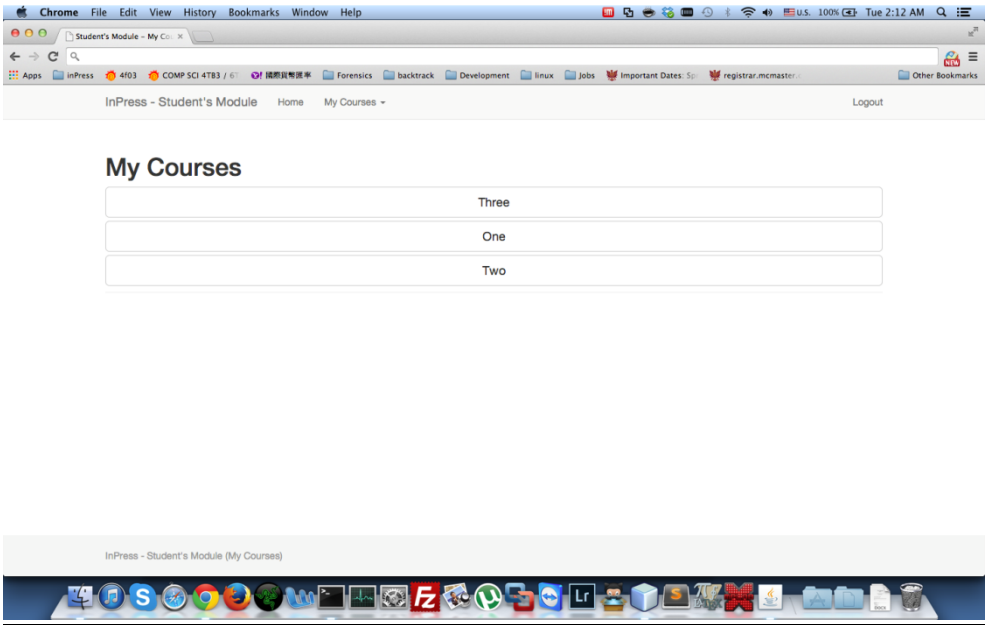

Figure 7: Compatibility: OS X & Google Chrome

Mobile Devices:

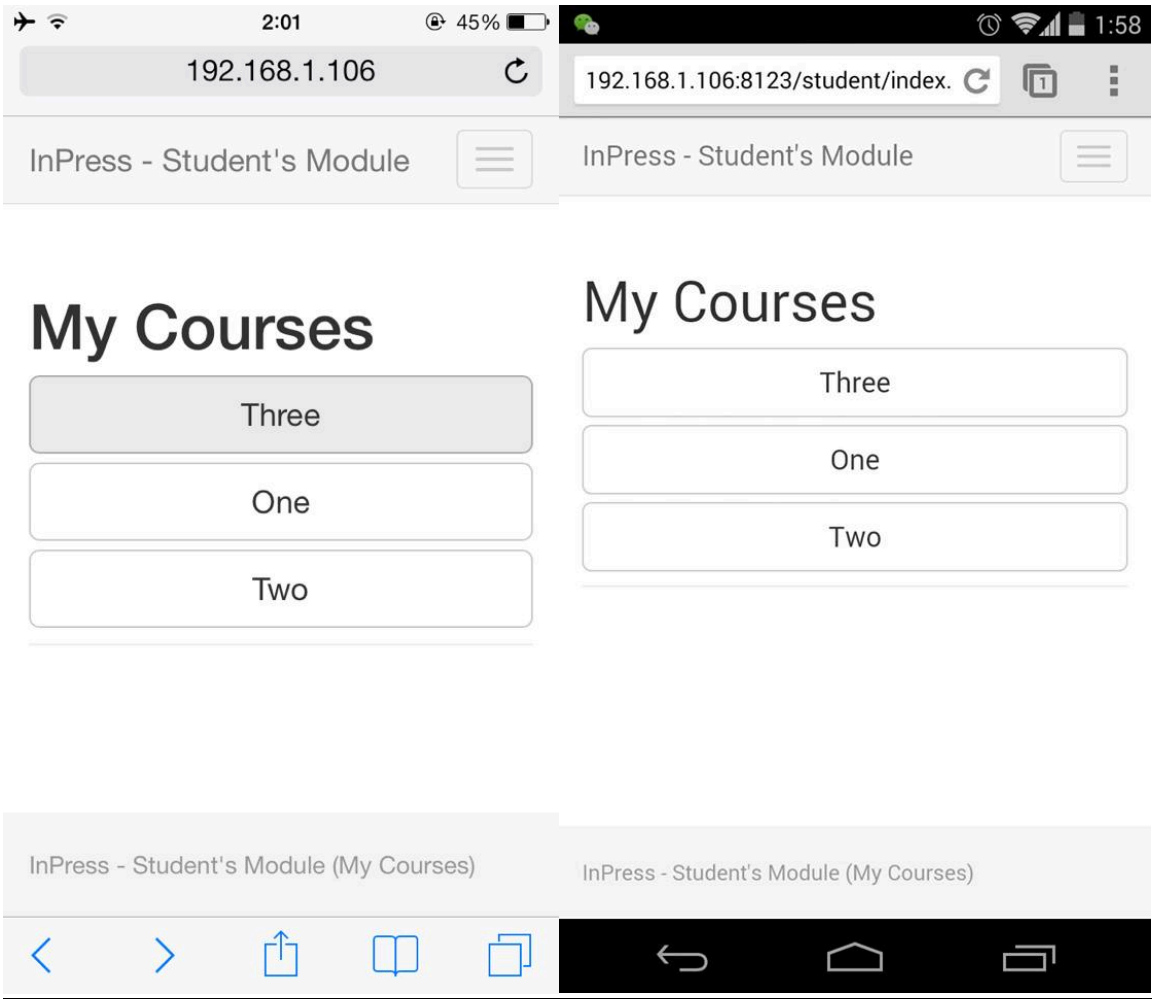

Figure 8: Mobile Devices Compatibility: iOS (left) & Android (right)

Usability/Robustness Testing

In the upcoming week, there are plans to test InPress in a live classroom. A discussion of the usability of the product will be discussed with the class after the students and instructors have worked with the product.

After testing our product with the system test cases outlined below, we also feel confident that InPress is extremely robust, and can handle various expected and unexpected scenarios. Further tests of robustness will follow as we proceed with enhanced testing in a live class.

System Testing:

Automated Testing

Automated testing is done through Selenium. Below are the automated testing using Selenium for the both the Students and the Instructors Module. Green (both light and dark) means that the test has passed. Upon travelling to each new page it will validate to see if it has gone to the right page (Asserts of certain texts in that page). The table below proves that automation using selenium was done, and was successful. Each dark green row represents a new page.

Student		
selectWindow	null	
open	/	
type	name=studentnumber	1
clickAndWait	//button[@onclick='submit']	
assertTitle	Student's Module - My Courses	
clickAndWait	xpath=(//a[contains(text(),'One')])[2]	
assertTitle	Student's Module - Course List	
clickAndWait	xpath=(//a[contains(text(),'Start!')])[2]	
assertTitle	Student's Module - My Courses	
type	id=answer	A
assertText	css=label.col-sm-2.control-label	exact:Question:
assertText	css=h1	A1
click	link=Next Question	
assertText	css=label.col-sm-2.control-label	exact:Question:
assertText	css=h1	A1
click	id=optionsRadios2	
click	link=Submit	
assertText	css=th	Assessment(s) for today
clickAndWait	link=Go to all the exisiting test(s) and quizz(es).	
assertText	css=h1	One
click	//div[@id='wrap']/div[2]/div/div[2]/div/h4/a/div/div	
click	css=div.col-md-10	
clickAndWait	//button[@type='submit']	
assertText	css=th	Assessment(s) for today
clickAndWait	link=Home	
assertTitle	Student's Module - My Courses	
clickAndWait	link=Logout	
assertTitle	Student Login	

Student

Instructor		
selectWindow	null	
open	/instructor/index.html	
assertTitle	Instructor's Module - My Courses	
clickAndWait	link=Add a Class	
assertTitle	Instructor's Module - Add a Course	
assertText	css=label	Course Name
type	id=ClassName	Test01
type	id=ClassCode	TT 101
type	id=classList	C:\Users\jkvc48\Desktop\st200.txt
clickAndWait	css=button.btn.btn-default	
assertTitle	Instructor's Module - My Courses	
clickAndWait	xpath=(//a[contains(text(),'Test01')])[2]	
clickAndWait	link=Add an Assessment	
assertTitle	Instructor's Module - Add an Assessment	
assertTitle	Instructor's Module - Add an Assessment	
type	id=AssessmentName	Assessment 1
clickAndWait	css=button.btn.btn-default	
assertText	css=label	Assessment 1
clickAndWait	link=Student Enrollment	
assertTitle	Instructor's Module - Manage Course Enrollment	
type	id=newStudent	1
click	id=addoption	
type	id=newStudent	2
click	id=addoption	
addSelection	id=selectMultipleStudents	label=2
click	id=btnRemoveMultipleOptions	
clickAndWait	css=button.btn.btn-default	
click	css=div.col-md-10	
assertText	id=isPosted0	Assessment 1 is not posted
clickAndWait	//a[@onclick="Posting('Assessment 1', '0');"]	
clickAndWait	link=Add a Question	
assertTitle	Instructor's Module - Add a Question	
type	id=question	Question One

Instructor		
type	id=MC1	Answer 1
type	id=MC2	Answer 2
type	id=MC3	Answer 3
type	id=MC4	Answer 4
click	name=MCRadio	
clickAndWait	css=button.btn.btn-default	
assertText	css=label	Assessment 1
clickAndWait	link=Add a Question	
assertTitle	Instructor's Module - Add a Question	
type	id=question	Question Two
click	link=Short Answer	
type	id=SAText	Answer to Q2
clickAndWait	css=button.btn.btn-default	
assertTitle	Instructor's Module - My Course	
assertText	css=div.col-md-10	Assessment 1
clickAndWait	link=Data Analysis	
assertText	css=h1	Data Analysis for Assessment 1
click	link=Individual Result	
assertText	//div[@id='accordion']/div[2]/div	Individual Result
click	link=Summary	
assertText	css=h4.panel-title	Summary
click	link=My Courses	
clickAndWait	link=Test01	
assertText	css=h1	Test01
clickAndWait	css=span.glyphicon.glyphicon-remove	
assertText	id=isPosted0	Assessment 1 is posted
clickAndWait	//a[@onclick="Posting('Assessment 1', '0');"]	
assertText	id=isPosted0	Assessment 1 is not posted
clickAndWait	link=Remove an Assessment	
assertText	css=h1	Remove an assessment
clickAndWait	link=Assessment 1	
clickAndWait	link=Home	
assertTitle	Instructor's Module - My Courses	
clickAndWait	link=Remove a Class	
assertText	css=h1	Remove a Course
clickAndWait	xpath=(//a[contains(text(),'Test01')])[2]	

Instructor		
assertTitle	Instructor's Module - My Courses	
clickAndWait	link=Logout	
assertTitle	Instructor Login	

Table 1: Automated Test Results

Manual Testing

Instructor Module

Test cases	Scenario	Expected Result	Result
T1 - Login Functionality	**a)** Input correct instructor's username and password on instructor's login homepage and click login **b)** Input incorrect instructor's username and password on instructor's login homepage and click login	Only the correct instructor credentials will login.	**a)** Redirected to instructor's individual home page **b)** Redirected back to Instructor's login page **PASSED**
T2 - Add a Class (Navigation to Webpage)	1) Log in as an instructor 2) Click the "Add a Class" button	The "Add a Class" button will direct user to "Add a Course page"	Instructor is directed to the "Add a Course" webpage **PASSED**
T3 - Add a Class (Class Creation)	1) Follow **T2** 2) Create a class with the following information: i) Name: Programming ii) Code: CS 101 iii) Class List: a txt file with the following student numbers - 0951871, 0987127, 0913411	Class can be added with all the information recorded correctly.	Instructor is directed back to his homepage with a new course named "Programming" listed. **PASSED**
T4 - Add a Class with no class list should fail to create	1) Follow **T2** 2) Create a class with the following information: i) Name: Programming ii) Code: CS 101 iii) Class List: Leave blank	If there is no class list, instructor cannot create a class, and be warned with an error message.	A 504 Django Error appears. **FAILED**
T5 - Add a Class with no class name should fail to create	1) Follow **T2** 2) Create a class with the following information: i) Name: ii) Code: CS 101 iii) Class List: a txt file with the following student numbers - 0951871, 0987127, 0913411	If there is no class name, instructor cannot create a class, and will be warned with an error message.	Instructor is directed back to his homepage with a new course named "" listed. **FAILED**

T6 - Remove a Class	1) Follow **T3** to create a new course 2) Navigate to the Instructor's homepage 3) Click on "Remove a Class" 4) Click to remove the course you added in step 1	Class will be removed.	Instructor is directed back to his homepage with the new course removed. **PASSED**
T7 - Add an Assessment (Navigation to Webpage)	1) Follow **T3** to create a new course 2) Click on the added course on the Instructor's homepage 3) Click the "Add an Assessment" button	The "Add an Assessment" button will direct users to "Add an Assessment page"	Instructor is directed to the "Add an Assessment webpage. **PASSED**
T8 - Add an Assessment (Assessment Creation)	1) Follow **T7** 2) Create an Assessment with following information: i) Name: Test #1 ii) Effective Date: Today	Assessment can be added with all the information recorded correctly.	Instructor is directed to the Course Homepage with the new assessment - "Test #1" added **PASSED**
T9 - Add an Assessment with no name should fail to create	1) Follow **T7** 2) Create an Assessment with following information: i) Name: ii) Effective Date: Today	If there is no assessment name, instructor cannot create a class, and will be warned with an error message.	Instructor is directed to the Course Homepage with the new assessment - "" added **FAILED**
T10 - Add an Assessment (Future Effective Date)	1) Follow **T7** 2) Create an Assessment with following information: i) Name: Test #1 ii) Effective Date: Future	Assessment will be added with the status "offline".	Instructor is directed to the Course Homepage with the new assessment - Test #1 added with status of "Offline" **PASSED**
T11 - Add an Assessment (Past Effective Date)	1) Follow **T7** 2) Create an Assessment with following information: i) Name: Test #1 ii) Effective Date: Past	Assessment will be added with the status "Post Date Passed".	Instructor is directed to the Course Homepage with the new assessment - Test #1 added with status of "Post Date Passed" **PASSED**
T12 - Posting an Assessment with Date = Today	1) Follow **T8** 2) Expand the assessment you created, and click Post	Assessment will change status to "Online".	Webpage refreshes, and the assessment's status becomes Online **PASSED**
T13 - Posting an Assessment with Date = Past	1) Follow **T11** 2) Expand the assessment you created, and click Post	Assessment will change status to "Post Date Passed"	Webpage refreshes, and the assessment's status becomes "Post Date Passed"

			PASSED
T14 - Posting an Assessment with Date = Future	1) Follow **T10** 2) Expand the assessment you created, and click Post	Assessment will change status to "Offline until <date>"	Webpage refreshes, and the assessment's status becomes "Offline until <date>" **PASSED**
T15 - Add a Question (Navigation to Webpage)	1) Follow **T8** 2) Expand the assessment you created 3) Click the "Add a Question" button	Add question button will direct user to "Add a Question" page.	Instructor is directed to the "Add a Question" webpage **PASSED**
T16 - Add a Question (Multiple Choice)	1) Follow **T15** 2) Create a Multiple Choice question with the following information: i) Question: Which is a programming language? ii) Answer - English, Java, Python, C++ 3) Select English as a bullet point	Instructor will successfully add a question with type multiple choice.	Instructor is directed to the Course homepage with the new question added to the assessment **PASSED**
T17 - Add a Question (Multiple Choice with Blank Question)	1) Follow **T15** 2) Create a Multiple Choice question with the following information: i) Question: ii) Answer - English, Java, Python, C++ 3) Select English as a bullet point	Instructor cannot add a question with no question text.	Instructor is directed to the Course homepage with the new question added to the assessment. The question field is blank. **FAILED**
T18 - Add a Question (Multiple Choice with no correct answer)	1) Follow **T15** 2) Create a Multiple Choice question with the following information: i) Question: ii) Answer - English, Java, Python, C++	Instructor cannot add a multiple-choice question with no correct choice.	A 504 Django Error appears. **FAILED**
T19 - Add a Question (Short Answer)	1) Follow **T15** 2) Create a Short Answer question with the following information: i) Question: Is Python a computer language? ii) Answer - Yes	Instructor can add a short answer question by using correct inputs.	Instructor is directed to the Course homepage with the new question added to the assessment **PASSED**
T20 - Remove Questions	1) Follow **T16** and **T19** to create a Multiple Choice and Short Answer questions 2) Remove both questions by clicking the X beside it	Instructor can remove the question by click on the X button.	Instructor is directed to the Course homepage with the deleted questions removed from the assessment **PASSED**

T21 - Student Enrollment (Add and Remove Students)	1) Follow **T3** 2) Click on the course created 3) Click on the "Student Enrollment" button 4) Remove 0987127 from the student list, and add 094191. 5) Ensure that 0987127 is not able to access the course, and 094191 is by logging in with those student numbers	Instructor can add or remove student from student list.	Instructor is able to add and remove the student successfully. Students that are not listed in the class list are unable to see the course, and students listed in the class list are. **PASSED**
T22 - Data Analysis	1) Follow **T28** 2) Navigate to the Instructor's Module and navigate to the corresponding course 3) Click "Data Analysis"	Data analysis will show the number of student that participate and their choices.	Instructor is able to see who completed the assessment, what they chose, and the corresponding graphs for it. **PASSED**

Table 2: Test Results – Instructor Module

Student Module

Test cases	Scenario	Expected Result	Result
T23 - Login Functionality	**a)** 1**)** Follow **T3** 2) Logout, and access the student module 3) Login with the following student number - 0951871 **b)** Access the student module, and login with an unregistered number with no course attached to it	Only the correct student credentials will login.	**a)** Redirected to student's individual home page with at least one course listed **b)** Redirected back to student's login page **PASSED**
T24 - Viewing Assessments (with Post Date today)	1) Follow **T12** 2) Logout, and Login with a student number registered in the class created in 1 3) Click on corresponding course	The assessment which is posted and active date is today will show up in student's course page.	Student is able to see the assessment on the course homepage **PASSED**
T25 - Viewing Assessments (with Post Date past)	1) Follow **T13** 2) Logout, and Login with a student number registered in the class created in 1 3) Click on corresponding course	Student will only can see the past assessments in the assessment history	Student is unable to see the assessment on the course page, but is able to see it in course history **PASSED**
T26 - Viewing	1) Follow **T14**	Student will not be able to	Student is unable to see the

Assessments (with Post Date future)	2) Logout, and Login with a student number registered in the class created in 1 3) Click on corresponding course	see the future active assessments from anywhere.	assessment on the course page, and is unable able to see it in course history **PASSED**
T27 - Answering Questions (Multiple Choice/Short Answer)	1) Follow **T16, T17** 2) Logout, and Login with a student number registered in the class created in 1 3) Click on corresponding course 4) Click on the assessment available, and answer the questions.	Student can start today's posted assessment and answer questions.	Student is able to answer to the questions properly, and no errors are outputted. **PASSED**
T28 - Student is able to see correct answers	1) Follow **T16, T17** 2) Logout, and Login with a student number registered in the class created in 1 3) Click on corresponding course 4) Click on the assessment available, and answer the questions. 5) Click on Submit, and click See my Answers	Student will be able to check the correct answer in the course assessment history page.	Student is able to see the correct answers. **PASSED**

Table 3: Test Results – Student Module

Field Test

We conducted two field tests towards the end of our project. A Chemical Engineer Professor (Kevin Dunn) allowed us to use InPress in his Process Control Class.

The first test started out normal when everybody was logging in. However, in the midst of everyone logging in the server started to response very slowly, to a point that no one could get in. While we were monitoring the servers we realized that the swapping and the cpu usage was at max. We decided to reboot the host but it was so irresponsive to a point that we concluded it has crashed.

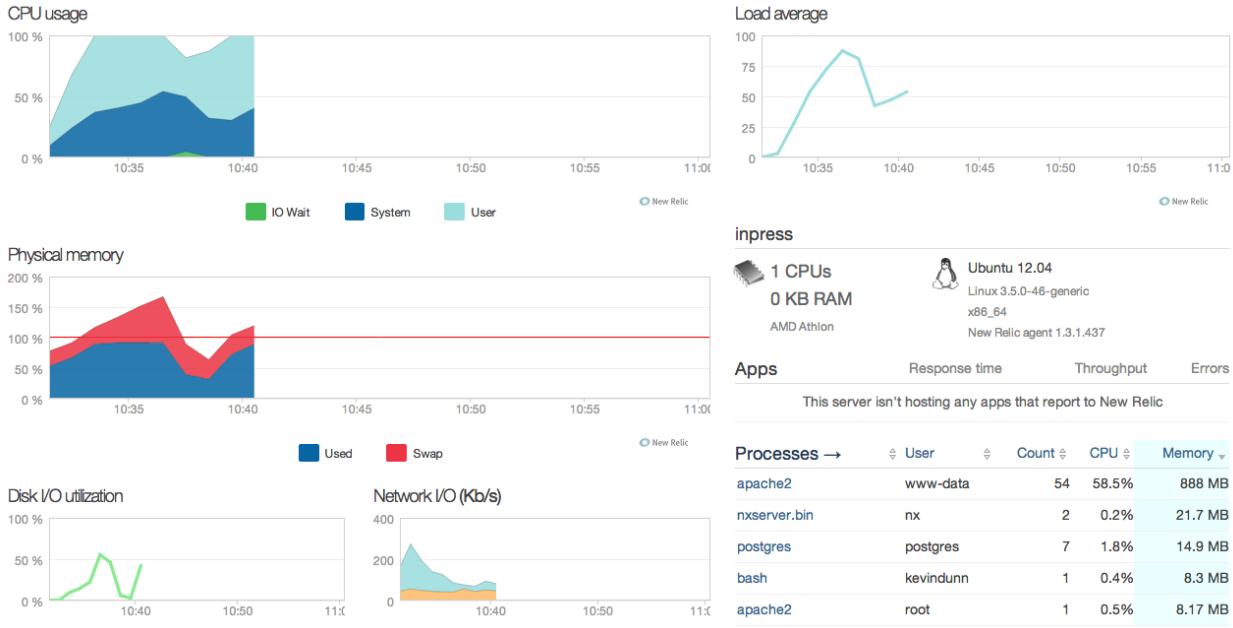

Figure: Screenshot of server before crashing in the First Field Test

We investigated as to why the server could not handle it by stress testing the server again to see if it was InPress or the server itself. Turns out that it was the server, we realized that was our apache servers will have spare servers / processes for each user. So for worse case, each user will have 5 spare processes. If we have 60 students in the class this would mean 300 processes and another 60 for PostgresSQL. This would come to a total of 360 processes, and 240 extra processes. Each lingering processes will take up some memory, and eventually it will take all the memory from the server hence extremely slow response time. Our original stress testing conducted on another server to see if InPress can handle all the users. We should have conducted our test on the field server instead of our own server.

After the investigation we reloaded it with new configurations (no spare servers were allowed), we did another stress test. Below are before and after results:

	Login Time (ms)	Full Run (ms)
Before	61980	N/A
After	43321	10161

From the results we can see that we could only get the Login Time samples to complete and could not do a full test on other pages as well. Just doing the login page was enough to nearly crash the server itself (having to restart apache each time just to kill lingering processes). With the configuration we were able to get a reduced time for login and have a Full Run at an average of 10

seconds. With this we were able to conduct our second field test successfully (with some heart attack moments where there was a couple of second delays).

Summary of Changes

Out of 28 test cases spanning InPress, 4 test cases failed. The nature of the test cases that failed was due to unexpected input from the user, and were deemed medium in severity. The failure of these test cases yielded more reinforced checking of parameters provided by the user. Prior to being submitted to the backend, the frontend of InPress now ensures that the input received from the user is in correct form, and outputs an error on the webpage if it is not. All test cases were executed again after these additional checking in the code, and all of them subsequently passed after this code change. The failure of these test cases provided a more robust product.

Updated with Field Test report and results, with new apache configuration we were able to reduce the time drastically and have close to none spare lingering apache processes.

Traceability Graph

Requirements	**Test Case**	**Module**
R1: Create an Instructor Account	T1	Instructor Module
R2: Login as an Instructor	T1	Instructor Module
R3: Login as a Student	T23	Student Module
R4: Create a course	T2 - T5	Instructor Module
R5: Delete a Course	T6	Instructor Module
R6: Add an Assessment	T7 - T11	Instructor Module
R7: Remove an Assessment	T20	Instructor Module
R8: Add questions to an assessment R14: LaTex in Questions/Solutions	T15-T19	Instructor Module
R9: Posting/UnPosting Assessments R10: Effective Date for Assessment	T10 – T14 T24 – T26	Instructor Module Student Module
R11: Data Analysis	T22	Instructor Module
R12: Data Analysis for Students (Answer Key) R15: Answer Questions	T27 – T28	Student Module
R13: Add and Remove Students	T21	Instructor Module

Table 4: Traceability Graph