Test Plan Marking Scheme						
Team Number						
Team Name						
Student Name	Safwan Hossain	**macid**	hossam18			
Student Name	Tyler Magarelli	**macid**	magarelt			
Student Name	Eamon Earl	**macid**	earle2			
					Mark	**Out of**
Basics						
Spelling and Grammar – take off 2 for each spelling or grammar mistake after the first two “free” mistakes, to the maximum shown					6	6
Take off 4 marks if the file is not located in the correct spot in the repo or commit not tagged					4	4
Total					**10**	**10**
Style						
Paragraph structure (logical grouping of ideas) Concisely expressed ideas (not wordy) Flow between paragraphs and sections Adequate number of figures and other visuals (could be zero, if this is adequate) “Pointers” in the document to help navigate through Subsections logically organized					7	10
Total					**7**	**10**
Overall Opinion of Content and Originality						
Is the material specific. Is the rational clear and logical. Not simply a list of every test type - plan needs to recognize that there are limited resources of time. Originality - evidence that the students have thought about the issues and shown creativity					5	10
Total					**5**	**10**
Check List						
Title Page, with student names and macids, Table of Contents					0.5	1
List of Figures and Tables (if appropriate), Tables and Figures have captions.					0.5	1
Pages are numbered.					1	1
There is a section for the major revision history.					0.5	1
Tools used for testing are explicitly identified (tools could include unit testing framework, code coverage metrics, static checkers, automated testing, load testing (like JMeter), etc.)					3	5
Identify types of tests (structural (white box), functional (black box), unit), static vs dynamic, manual versus automated.					3	5
Explain how test will be performed.					2	5
Explain plans for automated testing.					2	5
Specific system tests. All tests should be fully described in terms of initial state, input and output. Tests should be numbered and named.					4	5

System tests include tests that should generate exceptions, like trying to access a missing url, or open an nonexistent file, or attempting to solve a singular matrix, or find a solution when no feasible solutions exists.	3	5
Specific system tests explain how the existing implementation will be used in testing, potentially for parallel testing	3	5
Proof of Concept test is explicitly identified.	0	2
Gantt chart breaks the testing into a set of tasks	0	2
Gantt chart includes specific dates and specifically identifies which team members do what	5	5
System test cases are unambiguous.	1.5	2
System test cases build confidence that the domain is covered.	1.5	2
Tests for nonfunctional requirements, like usability and performance, are explicitly identified. Test could include usability testing, stress testing, performance testing, etc. Tests could include varying an input parameter and tracking the input on performance.	6	10
Plans for unit testing - what type of tests? need for stubs or drivers? coverage metrics?	5	8
Total	41.5	**70**
Total Mark	**64**	**100**