Team 15  
Deliverables

short line

**Overview:**

1. Project Plan Document
2. Project Process Document
3. Design Document
4. Code base
5. Testing Schedule Document
6. Group Report
7. Peer Assessment Report
8. Video
9. Project Plan: (Spec)

Showing the tasks that you identified for your team, how much effort you planned for each task, when it was scheduled to start and complete, and which team member(s) were responsible for it. This will likely take the form of a PERT style analysis or a Gantt chart. Bear in mind that the execution of the plan may well end up quite different from the initial plan. This is quite normal as sometimes things turn out differently to what you expect. If the variations are small, you don’t need to keep the plan up to date. If the execution is widely different from the plan, you should consider updating the plan and presenting both the initial and revised plans in your submission, together with commentary to explain why the plan fell apart.

Summary:

* ~~List of tasks identified~~
* ~~Effort Planned for each task identified~~
* ~~Schedule (PERT or Gantt)~~
* ~~Team member assigned for each task~~
* ~~Updating Plan, if executions were off predictions~~
* ~~Submit revised and initial plans~~
* Commentary on why revision was needed

1. Project Process Document: (Spec)

This project will be developed over a series of incremental developments (“sprints” in the Agile terminology). Your process document should record what you did for each incremental phase. For each phase this will include, but is not limited to, the target objectives, user requirements or user stories you set for yourself, any functional, non-functional or domain requirements that you identified that needed to be addressed, any other reasonable assumptions that you made as part of any requirements analysis process, any key design and implementation decisions that you took, and a review of how successful you were. You can blend elements of the Waterfall and Agile development process models as you see fit.

Summary:

* ~~Sprint Document~~
* ~~Sprint Document including record of action for each phase~~

Each Phase should have:

* ~~Target Objective~~
* ~~User Requirements~~
* ~~Functional, non- functional or domain Requirements~~
* ~~Key design (if applicable)~~
* ~~Implementation decisions~~
* ~~Review on the level of success~~

1. Design Document: (Spec)

Documenting both high level and low level designs of your software systems. Such a document should include, although is not limited to, class diagrams, sequence diagrams, activity diagrams, state transition diagrams, use case analyses and other UML style diagrams. The design does not need to be exhaustive. You only need to do as much design as your team deems necessary to deliver each of the incremental development phases.

Summary:

* Drawing of Low level designs
* Drawing (Physical or Digital) high level designs
* UML style Diagrams
* Class diagrams
* Sequence Diagrams
* Acitivity Diagrams
* State transition diagrams
* Design for each incremental phase (if phase requires one)

1. Code Base: (Spec)

Including all non-code files or other data necessary to allow your submission to be compiled and executed by the assessors. The code should be properly documented (e.g. using Javadocs if you are working in Java), and be commented to an appropriate standard. As well as the completed codebase, you should retain a snapshot of your codebase at the end of each incremental development phase as set out in your process document.

Summary:

* Code files and Non-Code files
* Check if files are easily compiled and executed by others
* Code be documented and commented
* Snapshot the codebase which was related to the phase

1. Testing Schedule: (Spec)

Showing how you tested the software at both unit test and system test levels. It is acceptable for bugs and other issues to remain in the software as long as they are clearly identified, documented and that you have provided appropriate commentary to explain why they exist and what you would do about them.

Summary:

* Showing how we tested the software (Unit testing and System Test levels)
* Identify and Document bugs (can remain in the software)
* Bugs found are to be explained and what you would do about them

1. Group Report: (Spec)

Documenting how the project went for you, with a critical analysis of the performance of your team, and describing what worked well for your team and what did not. The report must also contain records of any group meetings that you had, together with a summary of actions agreed and a weekly log recording your progress.

Summary:

* Document how project went for you (Individually) and critical analysis of the performance of your team (What went well and Even better If (WWW and EBI))
* Group meeting documents included

1. Peer Assessment: (Spec)

A document where the team agrees the distribution of a pool of marks to reflect the relative contributions of each team member to the group project as a whole. Your individual mark for this Software Engineering module will be a composite of your team score and an individual score determined by your peer assessment. If any individual does not make a contribution to the group effort, the assessors reserve the right to award an individual score of 0 for that team member for the module as a whole.

Summary:

* Each person being assessed and marked based on their contributions as a whole

1. Video: (Spec)

Providing evidence of the final working state of your program. The purpose of this video is to assist the markers in evaluating the working state of your final delivered program, and to enable you to showcase any particular features that you are particularly pleased with.

Summary:

* Evidence of final working state of program
* Include a voiceover to narrate any features