

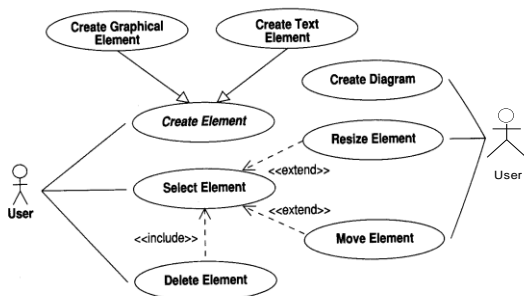
Deliverables

Week 6 – Report: The Problem and the Early Solution (Total: 20%)

1. Use Case Requirements Specification and Planning (10%)

- A **use case diagram** or **diagrams** representing the main sets of sequences of user-system interaction.
- A corresponding set of **written use case descriptions**.
- A **Gantt chart** indicating the main development strands and deliverables over the project lifetime.

e.g. From Chapter 4 of the Module Notes

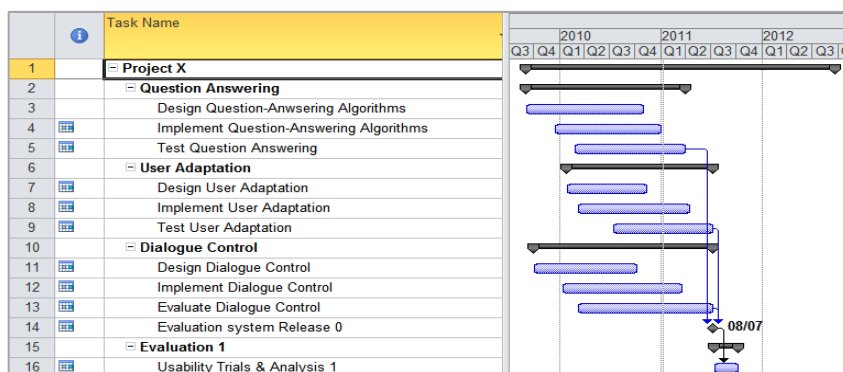


Use case diagram

Flow of Events for the <i>Select Element</i> use-case	
Objective	<i>To select an element in the workspace</i>
Precondition	<i>There is an active diagram containing at least 1 element</i>
Main Flow	<ol style="list-style-type: none"> 1. The user selects the selection tool (if necessary) 2. The user moves the cursor over an element 3. The user presses the mouse button 4. The element becomes selected and the control points are displayed 5. The user releases the mouse button
Alternative Flows	<p>At 3, there may not be an element. In this case no element is selected</p> <p>At 3, the element may already be selected. In this case, it remains selected</p>
Post-condition	<i>The element is selected and its control points are displayed</i>

Use case description

e.g. From Chapter 3 of the Module Notes



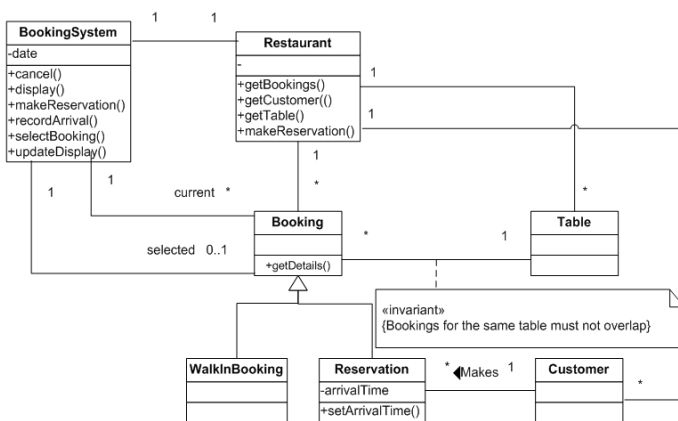
Gantt Chart

(Week 6 Deliverable continued overleaf)

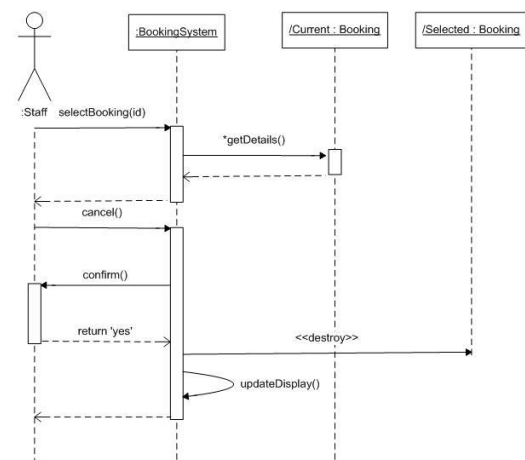
2. System Analysis (10%)

- a) An initial **class relationship model** (analysis model) representing the most important concepts in the application domain. These concepts may also be regarded as candidate classes for the subsequent design and implementation. They will be derived from the Use Case Requirements Specification. Include brief written commentary on the analysis model. Credit will be given for evidence of an exploratory prototype implementation (see comment on Minutes below).
- b) **Use case realisations** in the form of sequence diagrams that show the main sets of sequences of interaction identified in the Use Case Requirements Specification. Include brief written commentary on the use case realisations. Credit will be given for evidence of an exploratory prototype implementation (see comment on Minutes below).

e.g. From Chapter 5 of the Module Notes



Class relationship model (analysis model)



Use case realisation (sequence model)

Keep agreed team minutes each week and include these as an Appendix to your Week 6 report. A Weekly Team Minutes template is available. Interim deliverables referenced in the minutes (e.g.: a working requirements analysis; a partial software design) may be included in a printed Supplement to the minutes. Groups should submit to SVN any prototype code that they have developed and tested in the course of compiling their report. SVN should also be used as a repository for evolving project documentation. The latest version of prototype code should be described at a very high level (screen dumps, test outputs, etc.) in a further Appendix to the Week 6 report.

3. Peer assessment for the Problem and Early Solution

Along with their Week 6 report, teams must submit their Peer Assessment 1 form, in which team members assess and agree the contribution that their fellow team members have made to the Use Case Requirements Specification and System Analysis exercises. The Peer Assessment 1 scores are used to calculate each individual student's mark from the raw mark that the assessor gives to this deliverable. Teams should meet in good time in order to conduct the Peer Assessment 1 exercise.

(PTO for Week 11 Deliverables.)

Week 11 – The System, the Process and the Final Report (Total 40%)

1. Final Report: Design Documentation (5%)

- a) A documented **interface design**: the look and feel of the system represented by screen dumps/screen mock-ups and brief descriptive text
- b) More detailed **class relationship models** and **sequence diagrams** (see Week 6 above), encompassing the user interface (the 'presentation layer') and also including updated and enhanced documentation for the core application (the 'application layer'). Include brief written commentary.

2. Final Report: Implementation-Related Documentation (5%)

- a) A printed and completed **Test Plan** based on the original Use Case Requirements Specification. The Test Plan should describe tests for all key elements of system functionality; it should indicate that the tests have been applied and that the relevant system functionality has passed. The printed Test Plan should normally document a combination of black box and white box testing, and may include details of unit tests of more complex features (see also Adherence to Process below).

Keep agreed team minutes each week and include these as an Appendix to your Final Report.

3. The Working System (20%)

The working system is to be developed in Java. Swing components will normally be used for the graphical user interface, though use of more specialised Java-compatible GUI toolkits is permissible, provided that any third-party components are clearly identified in the project report. A disk or memory key with the full working system must be submitted with the Final Report. The system will be demonstrated in Week 11 – schedule to be announced.

- a) A system that delivers *basic* functionality similar to that described between the broken lines (.....) in the *Group Product Description* will attract up to **10%**. As a rule of thumb, a system whose functionality is significantly less rich than the basic functionality described might be awarded 3%; a good basic system with slightly less complexity than was described might receive 6%; and a system whose basic behaviour is closely comparable in complexity to the functionality described would be worth 10%. The system will be at least as usable as the sample board game implementation provided – and probably more so: marks for basic functionality may be adjusted in the light of usability.
- b) 'Value-added' features (features that were not covered by the description of basic functionality) will attract up to 10 further percentage points - **10%** of the module mark. Value-added features might include: good playability, even with large numbers of players; additional squares and facilities; a novel university-related theme; an interesting reinterpretation or extension of board game features for the computer; suitable celebrations when a player wins. Again, by way of guidance, 'not many' extra features might be awarded an additional 3%; 'a good selection' of extra features might receive '6%'; while 'lots' of extra features would be worth 10%.

4. Adherence to Process (10%)

Adherence to process includes the following:-

- Regularly updated, version-managed code and documents, typically evidenced by a populated SVN repository and printouts of key revision histories (change logs).
- Good code coverage (i.e. test coverage) through widespread, methodical use of unit testing, typically evidenced by screen dumps of system-generated coverage statistics and outputs of sample test runs.
- Well-organised team minutes, appropriately dated and signed. Where the team minutes make reference to interim deliverables, these may be included in a printed Supplement to the minutes (see Week 6 also).
- Submitted, implemented code (flash memory or CD-ROM).

Printed material representing evidence of adherence to process should be included as a further Appendix to your Final Report.

5. Peer assessment for the system and the final report

In addition, in Week 11 teams must submit their Peer Assessment 2 form, in which team members assess and agree the contribution that their fellow team members have made to the creation of the system and the final report. The Peer Assessment 2 scores are used to calculate each individual student's mark from the raw mark that the assessor gives to this deliverable. Again, teams should meet in good time in order to conduct the Peer Assessment 2 exercise.