

## Week 10 Tutorial

### Goals

1. Produce a draft of your detailed use case model, including use case descriptions, activity diagrams and a refactored use case diagram.

### Assumptions

1. You understand the material from lectures in weeks 4-7.
2. You have a clear understanding of the high-level use case model and a good draft of sections two to five of the requirements specification.

### Activities

Activity 1 (15 min.)	<p>Review the scope of your system based on the information captured in your use case diagrams. Consider the following questions:</p> <ul style="list-style-type: none"> <li>• Upon reflection, is the scope too large or too small?</li> <li>• Does it encompass activities that are other systems' responsibilities?</li> <li>• Does it include everything that users would expect of the system?</li> <li>• Does it represent a complete and coherent system?</li> <li>• Are there use cases that seem unrelated to others in the model? If so, should they be deleted or are there missing use cases?</li> <li>• For each actor, do the use cases cover all ways of using the system that would be expected from the actor's perspective?</li> <li>• Are all actors appropriate users of the system?</li> <li>• Are there any missing actors?</li> </ul>
Activity 2 (15 min.)	<p>Review the priorities given to the use cases.</p> <ul style="list-style-type: none"> <li>• Do all the Must Have use cases reflect key features that <i>must</i> be delivered?</li> <li>• Can a logically staged release plan be created from the current priorities?</li> </ul>
Activity 3 (30 min.)	<p>If you have not already done so, identify at least one significant use case per team member that should be part of the first release. Allocate one of these use cases to each team member.</p> <p>Each team member should write an initial description of the typical scenario for their use case. They should then identify a list of alternative scenarios for the use case and write initial descriptions of at least a few of these.</p> <p>In the last 5 minutes of this activity, swap use case descriptions between team members. Each team member should read another member's description. As you are reading the description, make notes of possible improvements. Give these notes to the use case's author.</p>
5 min.	<b>Break</b>
Activity 4 (30 min.)	<p>Allocate another significant use case to each team member. These should be from the first or second release. Each team member should draw an activity diagram for their use case. The process should be the same as for writing a use case description. Draw the nodes and flows that make up the typical scenario. Then identify alternative scenarios and add their nodes and flows into the diagram.</p> <p>In the last 5 minutes of this activity, swap activity diagrams between team members. Each team member should review another member's diagram. As you are reviewing the activity diagram, make notes of possible improvements. In particular look for missing detail, steps or logical options that are missing from the diagram. Give these notes to the use case's author.</p>

Activity 5 (15 min.)	<p>Consider the information your team has generated through the use case descriptions and activity diagrams. Can you identify how the use case diagrams can be refactored in light of this information?</p> <ul style="list-style-type: none"><li>• Can you see where some use cases seem to share common steps? Should these use cases include a use case that captures these common steps?</li><li>• Can you see where a use case seems to have significant optional steps? Should this use case have an extension point where it is extended by a use case with these steps?</li><li>• Can you see where a group of use cases include a similar process? Should these use cases inherit from a more general use case that provides an abstract description of the process?</li><li>• Do your use case diagrams need to be updated? Do your existing set of use case packages still make sense in light of your refactored use case diagrams?</li></ul>
-------------------------	---

### Required Outcome

You have draft use case descriptions and activity diagrams that describe a large amount of the functionality of the first release of the system. You also have a clear idea how to revise your use case diagrams and how to refine your use case descriptions and activity diagrams.

### Further Work

As a team, revise your use case diagrams to reflect any improvements from activity one, and to add include, extend and generalisation relationships you identified in activity five.

Consider the use cases for which your team has draft descriptions or activity diagrams, identify which of them are involved in a relationship with other use cases. These descriptions or activity diagrams need to be updated to reflect the relationship. The other use cases involved in these relationships need to have either use case descriptions or activity diagrams. Allocate these to team members to complete.

You should aim to have a draft of sections one to seven of your requirements specification to bring to the tutorial next week.