eCornell

Defining Scope
Checklist

Instructions:
You need to identify the operations your system must perform in a variety of use case situations.

These steps help you create a context diagram which will show the interactions of your system with other elements. With the help of your context diagram, you then will be able to create a list of use cases your system can perform, and a list of the elements and stakeholders that will interact with your system. Check off each step as you complete it.

Steps	Is Complete When	✓
Before You Begin: Remove any name around your system. Instead refer to it as "The System." At this first stage of design, not naming the solution is helpful to prevent drilling down to specific solution too quickly.	Your system has no name.	
Step 1: Begin to make a context diagram, by inserting "The System" to the center of your context diagram. Begin to add elements to your context diagram. These should be listed outside "The System" box.	You have added the elements to your diagram and followed the guidelines using the same size boxes, same font type, and added no color to the diagram.	
Step 2: Define your stakeholders.	You have a list of all the stakeholders associated with your project. In many cases, you may also want to declare a few or even just one primary stakeholder as your client, those whom you must satisfy to in order to get the project completion approval or get paid.	

Steps	Is Complete When	✔
Step 3: If needed, add to your context diagram based on the stakeholders that will interact with your system.	You have "The System" box in the system boundary box and all other stakeholder boxes listed around and outside of the system boundary box.	
Step 4: Determine what else your system interacts with, focusing on the items that would make your system have to do something different or something special. Consider unwanted or inappropriate interactions with your system.	You've added boxes of other things that the system may interact with that are outside of what you as the designer have direct control over.	
Step 5: Determine how the outer box items will use or interact with your system and what your system must do with the outer box items.	You have connected each outer box with a line to "The System" box and at least one end of each line is labelled with an interaction. You have created your first pass at a context diagram.	
Step 6: Take a break. You've put a lot of information into your context diagram and as a result it may have become cluttered. Now's a great time to take a break before you try to review it as a whole or reorganize it.	You've been able to clear your mind for a bit of the work you've done so far and you're ready to review the hard work you've done.	
Step 7: Iterate on your context diagram by breaking up, combining, and/or removing outer box items.	You have reviewed your context diagram for both completeness and conciseness. You have rearranged the boxes, lines, and words in the diagram so that they are easy to read.	

Steps	Is Complete When	✔
Step 8: In a separate document, develop a list of use cases. This is done by looking over your context diagram and then creating a list of the situations your system will be used in.	You have created your first case of use cases.	
Step 9: Add in use cases from other sources (e.g. a use case internal to your system or a direct request from a stakeholder).	You have a list of use cases that covers all of the major functionalities your system will have to perform and scenarios it will have to handle.	
Step 10: Refine your use cases list by considering cases that might be included in existing use cases or those that might extend existing use cases.	You have refined the set of use cases to fill in the gaps in your previous iterations.	
Step 11 Run through this checklist: 1. Can you clearly state the initial and ending conditions of your use cases? 2. Could you describe what occurs during the use case as a series of step-by-step functions that your system must perform? Could you do so without it feeling *too long* or running across several use cases? 3. Does each use case capture various functions that might otherwise have been missed if this use case wasn't considered?	You have modified and improved your use case list to a reasonable level.	