

Team project

This report will be an update to your proposal with some additions. The document should have the following parts.

1. The name of the project, the team number, and the list of team members.
2. A project description – This may be a rewritten or expanded version of your original submission.
3. Use cases – This will be a bit more formalized version of the “user interaction script” of the previous submission. There should be one or two use cases along with a description of the presented cases and an indication of whether to anticipate building more.
4. Some sketches of display screens (These need not be in great detail. You can use text or pictures. These can be modifications of the images in the first submission.)
5. A list of essential and enhancement items (One list for the things that you think will be essential and another list for the things that build on the essential elements. You need not complete all of these. These items can be modifications of the first submission.)
6. The domain model – A presentation of the essential items in the domain the logic of the items and their interaction. In addition to a descriptive paragraph there should be a listing and description of the items based on noun/verb separation, CRC cards, or UML diagrams. The domain model should not contain references to the GUI in terms of what is displayed to the user or what the user inputs. You can think of this as a list of the things in the domain and what they do. This is not yet to a software design level. If you want to add a more detailed design with an indication of fields, methods, and visibilities that would be good but is not required at this point. It will be required for the next report.
7. A time budget – Update your time budget. This update should include at least the amount of time currently expended and the amount of time remaining in the budget.

This artifact will continue to guide your work. You will add more detail in future reports. Note that each report will build on the previous report. The reports should give you almost all the information you will need for the final documentation of your project.

A template for a use case

Name	A clear verb/noun or actor/verb/noun descriptor that communicates the scope of the use case.
Brief Description	A brief paragraph of text describing the scope of the use case.
Precondition	Anything the solution can assume to be true when the use case begins
Basic Flow	The set of steps the actors take to accomplish the goal of the use case. A clear description of what the system does in response to each user action.
Alternate Flows	Capture the less common user/system interactions, such as being on a new computer and answering a security question. This is optional
Exception Flows	The things that can happen that prevent the user from achieving their goal, such as providing an incorrect username and password. This is optional
Post Conditions	Anything that must be true when the use case is complete

Note that some of these items are optional. The text should be in ordinary language as far as possible.

A template for a domain model

For each thing (entity) in the domain

Name	Common name for the client, designer, and implementer to use. Remember this is at the conceptual level. What do we call this thing?
Description	Express the intent of the thing in ordinary language. What is this thing?
Responsibility(or interface)	Express what the thing is supposed to do or the requests to which it will reply. What does this thing do?

Note that this could be based on CRC cards or UML diagrams. For our current purposes it is enough of a start to describe things in ordinary language. If you would rather use CRC or UML feel free. You should note that the intent of any of these is the same at this point. The intent is to describe the domain. This template is a simple and I hope straightforward account of what should be in the domain model. It can be refined as you go along.

A 'test' of your efforts

Use the use case and the domain diagram in a walkthrough. In the walk through don't worry about the mechanism of the user interaction. This will be coming. The walk through is to make sure that there is something in the domain that can be used in a response to the user action. Don't dive into a lot of detail here. It is good enough at this point to know that things A, B, and C would be used by the system in responding to the user action.