<Template instructions: Text in this style should be removed. Text in Normal style should be replaced. After using the template for some time you might remove some of the sections if they are unused, or adapt it more closely to suit the general style in use within your company. Copyright in this template remains with CRaG Systems>

<The change log should be updated for every significant change made to the document. Use the format n.m for the version number>

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Description** | **Changed By** | **Date** |
| 1.0 | First draft | Fred Bloggs | 1/1/00 |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| 1. **Brief Description:** | XXX *<2 to 3 sentences describing the use case>* |
| 1. **Preconditions:** | XXX *<the stable state(s) that the system must be in for the use case to start>* |
| 1. **Business Trigger:** | XXX *< the business event that causes the first interaction>* |
| 1. **Basic Flow:** | |
| 1. xxxx xxx xxxxx *<a line in the use case normally describing an interaction across the system boundary (‘black box’ view), but possibly describing gross internal functionality (‘white box’ view). Keep the level of abstraction as high as possible while still providing sufficient information to describe what is done. Avoid referring to the technology of the interface if possible and describe what information or command is passed. Write complete sentences including source (noun), action (verb), optional parameters (nouns), possibly with actual values and target (noun). Avoid concatenating sentences with commas and ‘ands’>* 2. xxxx xxx xxxxx *<further lines as above. Try to keep them in pairs or 1 in to n out. Don’t include things that happen outside the system but don’t cross the system boundary>* 3. xxxx xxx xxxxx *<if there is a large time gap between interactions, consider separating the temporally cohesive sections into separate use cases >* 4. xxxx xxx xxxxx *<continue until the outcome described by the name of the use case has been fulfilled. The last step should be one going out, probably back to the primary actor indicating that the use case is complete>* | |
| 1. **Post Condition:** | XXX *<the state of the system at the end of the basic flow, or things guaranteed to be true at the end of a successful use case>* |
| 1. **Alternate Flow: XXXX** *<name of the alternate flow starting with an active verb>* | |
| 1. When in XXX *<insert the number of the line in which the condition occurs>* 2. xxxx xxxx xxxx, then: *<define the condition under which the alternate flow is executed e.g. ‘the item number entered is found to be invalid’>* 3. xxx xx xxxxx *<write the lines just as you would in the basic course, describing what happens largely as interactions across the boundary adding further lines as necessary>* 4. xxx xx xxxxx *<the last line must describe what happens next: the use case terminates; the use case restarts where it left off; the use case jumps back and restarts at an earlier step; the use case jumps forward and restarts at later step>* 5. **Post Conditions:** XXXX *<you might wish to describe post-conditions for an alternate flow where they differ from those of the basic flow>* | |
| 1. **Alternate Flow: XXXX** *<add the names of further alternate flows as you think of them. Add the detail of the alternate flow after the basic flow has been detailed. It is possible to have extensions on the extensions. Write them exactly the same way as other extensions>* | |
| 1. **Sub-Flow: XXX** *<where there is procedure that is common to more than one flow in the use case, or where there are different flows following a case statement or selection statement, create sub-flows and ‘call’ them from the using flow. Name the sub-flows starting with an active verb>* | |
| 1. xxxx xxxx xxxx *<write the lines of the sub-flow just as you would in the basic course, describing what happens largely as interactions across the boundary. Add further lines as necessary>* | |
| 1. **Sub-Flow: XXX** *<add further sub-flows as necessary>* | |
| 1. **Business Rules:** | |
| 1. xxxx *<include here a description of detailed internal algorithms or procedures that are not part of the externally visible behaviour, but are vital to the functional definition of the system >* 2. xxxx *<or specify data structures or constraints or technology requirements that are specific to the use case and have no place in the procedural part of the use case >* | |
| 1. **Activity Diagram:** | |
| 1. *<if there is complex iteration and selection, include an activity diagram, or a reference/hyperlink to one here. Activity diagrams should not duplicate or replace the text of the flows but augment it where prose is difficult to use to describe complex conditionality >* | |
| 1. **Prototype Screen:** | |
| 1. *<if you have a prototype screen for this use case, include it or a reference/hyperlink to it here. If might be at proof of concept or detailed level depending on the importance of the screen design to the user. Make sure the text of the use case is consistent with the prototype >* | |