

Sample Use Case Template

Use Case Name *[Includes a unique reference number, the business name and the version extension]*
[For example #0001 – Maintain Party Details – V1.15].

1. Functional Requirements

1.01 Description	<p><i>Represents a summary of the overall scope and content of the use case.</i></p> <p><i>Typically one or two sentences that describe the goal of the use case.</i></p> <p><i>It does not detail the flow of events but captures the context and purpose of the use case. The reader should not have to look at the other sections of the use case to understand the supported behavior.</i></p>
1.02 Subject Area	<p><i>The category that best describes where this use case fits. For example ,it could be a department within the organization, a business function, or it could map to a named package of related use cases.</i></p>
1.03 Generalises/ specialises on	<p><i>If this use case is abstract and serves as a generalisation, list the specialisations here. If a specialization of an abstract use case, indicate the parent use case here.</i></p>
1.04 Primary Actor(s) [secondary actor(s)]	<p><i>The name of the external actor that triggers the use case for the purpose of achieving a goal.</i></p> <p><i>More than one actor may initiate a use case. In this case, a single abstract actor should be defined from which all initiating real actors will inherit the associated use case(s) behavior.</i></p> <p><i>Secondary actors may be included. These are external entities called upon by the use case to provide a service.</i></p>
1.05 Triggers	<p><i>The event or events that initiate the use case (usually triggered by the actor).</i></p> <p><i>The trigger can be a business event (such as the receipt of a document) or it can be a temporal event (such as an anniversary or month end). There may be multiple triggers for a use case (list them all!).</i></p>
1.06 Includes	<p><i>An included use case represents the common behaviour that is applicable to this, and other use cases. For ease of management, it is externalised in its own document.</i></p>
1.07 Pre-conditions	<p><i>Pre-conditions state the constraints that must be met before the use case can be executed. For example:</i></p> <ul style="list-style-type: none"> <i>• the system may have to be in a certain state before the initiation of the use case is enabled.</i> <i>• the use case may require a previous sequence of successful use cases before it can be invoked.</i>
1.08 Flow of Events	<p>1.08.1 Main Flow:</p> <ol style="list-style-type: none"> 1. 2. 3. ... <p><i>The use case should describe what the actor does and what the system does in response. It should be phrased in the form of a dialog between the actor and the system.</i></p> <p><i>The use case should describe what happens inside the system, but not how or why. If information is exchanged, be specific about what is passed back and forth. (Alternatively introduce an entry in the Glossary of Terms to keep the complexity of the use case manageable but define the data associated with it there!!).</i></p> <p><i>A picture is sometimes worth a thousand words (though there is no substitute for clean, clear prose). If an Activity Diagram is useful to present a complex decision process, by all means use it. Similarly, for state-dependent behavior, a state-transition diagram often clarifies the behavior of a system better than pages upon pages of text.</i></p>

Sample Use Case Template

	<p>1.08.2 Alternate Flows: [A1] [A2] [A3]</p> <p><i>Alternate Flows identify the conditions in the normal flow of events that cause the execution of an alternate path from the main flow. For example, there might be an alternate flow based on a variation of the input data.</i></p> <p><i>The alternate flow is identified in the main body by stating the conditions causing the invocation and a cross-reference a set of steps (documented in this section).</i></p> <p><i>To control Alternate Flow explosion for a given use case, complex alternate flows can be turned into subordinate use cases in their own right. We should consider specialisations of a use case (i.e inheritance relationship) or an extension to a base use case (i.e. "extends" stereotype on a relationship between the use cases).</i></p>
	<p>1.08.3 Exception Flows: [E1] [E2]</p> <p><i>Exception Flows identify the behaviour in "error" situations when the normal flow of events (or one of the alternative paths) does not perform as expected. For example, a party is not found on a party search.</i></p> <p><i>The exception flow is identified in the main body with a cross-reference to the set of steps (documented in this section).</i></p>
1.09 Extends	<p><i>A use case identified in the Extends section represents a reference to a use case that is being extended by this use case. The extended use case (the one referenced here) is valid as a complete use case in its own right. It does not know about this use case that extends it.</i></p> <p><i>This use case (the extending use case) contains the specific behaviour that can be "plugged" into the extended use case (with an indication where it fits).</i></p>
1.10 Post-conditions	<p><i>Alternative ways that the use case may end (either successful or unsuccessful in nature). Sometimes listed with the conditions under which the specific outcome is achieved.</i></p> <p><i>Note: The post-conditions can be useful in identifying events that get published such that other interested parties (objects/components) may subscribe to them to do further processing.</i></p>
1.11 Input Summary	<i>Summary level listing of the data expected as input to this Use Case.</i>
1.12 Output Summary	<i>Summary level listing of the data produced by this Use Case.</i>

2. Non-Functional Requirements

Non-Functional Requirement Number	Description

The Non-Functional Requirements section is likely to be specific to a particular implementation (scenario in our case). It may be that this section is externalised from the Business Use Case that is used across the organisation as a whole (not just a single implementation).

Sample Use Case Template

3. Business Rules

Business Rules Number	Description

This Business Rules section focuses on identifying the business rules such that they can be reused across multiple use cases but maintained in a single place.

This externalisation of the business rules will allow the organisation to not only manage the rules at analysis time but, manage the rules later in the development lifecycle too.

4. Business Requirements

Requirement Reference	Business Requirements Number	Description
Rnn		

This Business Requirements section cross-references the use case against requirements stated in another document (such as a 'traditional' Statement of Requirements, for example).

This section may be replaced when other tools (e.g. RequisitePro) have been evaluated (assumes that is appropriate for the organisation).

5. Outstanding Issues

Issues Number	Description

The Outstanding Issues section is used to record unanswered questions with respect to the scope of this use case.

6. Version History

Version Number	Description of Changes	Date

The Version History section is concerned with detailing the whole history of the use case. The latest version entry should correspond with the current version of the document.

7. Metrics

Complexity	<i>For example 1 = easy peasy through to 5 = very complex</i>
Time	<i>The purpose of this section is to quantify the resource expended on realizing the use case for input to post-implementation review; also provides a basis for future estimates and costings.</i>
Money	
Effort	

Other headings to consider include: Priority; Related materials e.g. protocol descriptions, job procedures etc.

Sample Use Case Template

Notes: