
In aeternum

In aeternum
Use-Case Specification: Register

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

In aeternum	Version: <1.0>
Use-Case Specification: Register	Date: 29.10.2016
<document identifier>	

Revision History

Date	Version	Description	Author
28.10.2016	1.0	First Attempt	Louisa

In aeternum	Version: <1.0>
Use-Case Specification: Register	Date: 29.10.2016
<document identifier>	

Table of Contents

1.	Use-Case Name	4
1.1	Brief Description	4
2.	Flow of Events	4
2.1	Basic Flow	4
2.2	Alternative Flows	4
2.2.1	< First Alternative Flow >	4
2.2.2	< Second Alternative Flow >	5
3.	Special Requirements	5
3.1	< First Special Requirement >	5
4.	Preconditions	5
4.1	< Precondition One >	5
5.	Postconditions	5
5.1	< Postcondition One >	5
6.	Extension Points	5
6.1	<Name of Extension Point>	5

In aeternum	Version: <1.0>
Use-Case Specification: Register	Date: 29.10.2016
<document identifier>	

Use-Case Specification: Register

[The following template is provided for a Use-Case Specification, which contains the textual properties of the use case. This document is used with a requirements management tool, such as Rational RequisitePro, for specifying and marking the requirements within the use-case properties.]

The use-case diagrams can be developed in a visual modeling tool, such as Rational Rose. A use-case report, with all properties, may be generated with Rational SoDA. For more information, see the tool mentors in the Rational Unified Process.]

1. Use-Case Name

1.1 Brief Description

[The description briefly conveys the role and purpose of the use case. A single paragraph will suffice for this description.]

This use case is one of the often used ones, because it create a new user account.

1.2 Function Points

Register	Count	Simple	Average	Complex	FTR/RET	DET	Points	Function p
Number of User Input (EI)	5	1	0	0	0	0	5	15
Number of User Outputs (EO)	2	1	0	0	0	0	1	8
Number of User Inquiries (EQ)	7	1	0	0	0	0	4	21
Number of Files (ILF)	2	1	0	0	1	1	1	14
Number of External Interfaces (EIF)	0	1	0	0	0	0	0	0

2. Flow of Events

2.1 Basic Flow

[This use case starts when the actor does something. An actor always initiates use cases. The use case describes what the actor does and what the system does in response. It is phrased in the form of a dialog between the actor and the system.]

The use case describes what happens inside the system, but not how or why. If information is exchanged, be specific about what is passed back and forth. For example, it is not very illuminating to say that the actor enters customer information. It is better to say the actor enters the customer's name and address. A Glossary of Terms is often useful to keep the complexity of the use case manageable—you may want to define things like customer information there to keep the use case from drowning in details.]

*Simple alternatives may be presented within the text of the use case. If it only takes a few sentences to describe what happens when there is an alternative, do it directly within the **Flow of Events** section. If the alternative flow is more complex, use a separate section to describe it. For example, an **Alternative Flow** subsection explains how to describe more complex alternatives.]*

A picture is sometimes worth a thousand words, though there is no substitute for clean, clear prose. If it improves clarity, feel free to paste graphical depictions of user interfaces, process flows or other figures into the use case. If a flow chart 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. Use the right presentation medium for your problem, but be wary of using terminology, notations or figures that your audience may not understand. Remember that your purpose is to clarify, not obscure.]

In aeternum	Version: <1.0>
Use-Case Specification: Register	Date: 29.10.2016
<document identifier>	

Here you can find our UML diagramm (tbd).

Mockup

Feature File

2.2 Alternative Flows

2.2.1 < First Alternative Flow >

*[More complex alternatives are described in a separate section, referred to in the **Basic Flow** subsection of **Flow of Events** section. Think of the **Alternative Flow** subsections like alternative behavior—each alternative flow represents alternative behavior usually due to exceptions that occur in the main flow. They may be as long as necessary to describe the events associated with the alternative behavior. When an alternative flow ends, the events of the main flow of events are resumed unless otherwise stated.]*

2.2.1.1 < An Alternative Subflow >

[Alternative flows may, in turn, be divided into subsections if it improves clarity.]

2.2.2 < Second Alternative Flow >

[There may be, and most likely will be, a number of alternative flows in a use case. Keep each alternative flow separate to improve clarity. Using alternative flows improves the readability of the use case, as well as preventing use cases from being decomposed into hierarchies of use cases. Keep in mind that use cases are just textual descriptions, and their main purpose is to document the behavior of a system in a clear, concise, and understandable way.]

3. Special Requirements

[A special requirement is typically a nonfunctional requirement that is specific to a use case, but is not easily or naturally specified in the text of the use case's event flow. Examples of special requirements include legal and regulatory requirements, application standards, and quality attributes of the system to be built including usability, reliability, performance or supportability requirements. Additionally, other requirements—such as operating systems and environments, compatibility requirements, and design constraints—should be captured in this section.]

3.1 Usability

Because our target group are older people, we will make the register very user-friendly. There should be a “Help”-Button and also the description for the input fields should be easy to undersand for everyone.

4. Preconditions

[A precondition of a use case is the state of the system that must be present prior to a use case being performed.]

4.1 Entering Register-Form

A visitor of the website can enter the register form by click a button (label “Start” and a “+” icon” on the first page he see after calling our page.

5. Postconditions

[A postcondition of a use case is a list of possible states the system can be in immediately after a use case has finished.]

5.1 Save account

The database must save all information(Name, Birthday, Password) for the new account.

In aeternum	Version: <1.0>
Use-Case Specification: Register	Date: 29.10.2016
<document identifier>	

5.2 Go on

After saving the account information the user will be redirected to a new page, where he can insert more information.

6. Extension Points

[Extension points of the use case.]

(tbd)

6.1 <Name of Extension Point>

[Definition of the location of the extension point in the flow of events.]