

<System Name>
Use Case Specification: <Use-Case Name>

Version <1.0>

LIS453LE – Fall 2013	Version: <01>
Use Case Specification: <Use-Case Name> <Use Case Name>	Date: <2013- XX-XX>
<document identifier>	

Revision History

Date	Version	Description	Author
2013-XX-XX	V 01	First Draft	Your name goes here

LIS453LE – Fall 2013	Version: <01>
Use Case Specification: <Use-Case Name> <Use Case Name>	Date: <2013- XX-XX>
<document identifier>	

Use Case Specification: <Use-Case Name>

[The following template is provided for a Use-Case development, which contains the textual properties of the use cases.]

1. Use Case Name

1.1 Brief Description

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

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 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.]

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. 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 flows are more complex, use a separate section to describe it. For example, an **Alternative Flow** subsection explains how to describe more complex alternatives.]*

]

2.2 Alternative Flows

2.2.1 < First Alternative Flow >

*[More complex alternatives should be described in a separate section, which is 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, many times because of 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 Sub-flow >

[Alternative flows may, in turn, be broken down 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 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,

LIS453LE – Fall 2013	Version: <01>
Use Case Specification: <Use-Case Name> <Use Case Name>	Date: <2013- XX-XX>
<document identifier>	

concise, and understandable way.]

3. Special Requirements

[A special requirement is typically a non-functional 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 < First Special Requirement >

4. Pre-Conditions

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

4.1 < Pre-condition One >

5. Post-Conditions

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

5.1 < Post-condition One >

6. Extension Points

[Extension points of the use case.]

6.1 <Name of Extension Point>

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

7. Key Scenarios

7.1 <First Key Scenarios Name>

[Give a brief description of the path through the flows above that the scenario takes. If you have illustrated this scenario with an activity diagram, then refer readers to the activity diagram.]