Online Flowerisit ordering System

Use-Case Specification: Online Ordering Form

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Brief Description 2

2. Basic Flow of Events 2

3. Alternative Flows 2

3.1 <Area of Functionality> 2

3.1.1 < A1 First Alternative Flow > 2

3.1.2 < A2 Second Alternative Flow > 2

3.2 <Another Area of Functionality> 2

3.2.1 < AN Another Alternative Flow > 2

4. Subflows 2

4.1 <S1 First Subflow > 2

4.2 < S2 Second Subflow > 2

5. Key Scenarios 2

6. Preconditions 2

6.1 < Precondition One > 2

7. Postconditions 2

7.1 < Postcondition One > 2

8. Extension Points 2

8.1 <Name of Extension Point> 2

9. Special Requirements 2

9.1 < First Special Requirement > 2

10. Additional Information 2

Use-Case Specification: <Use-Case Name>

[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.** **Brief Description**

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

# **2.** **Basic Flow of Events**

[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 if it is not defined. It is better to say the actor enters the customer’s name and address. A Glossary of Terms (or a more formal Domain Model) is essential to keep the complexity of the use case manageable¾you 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 flow of events. If it only takes a few sentences to describe what happens when there is an alternative, do it directly within the flow. 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.

Complex flow of events should be further structured into sub-flows. In doing this, the main goal should be improving the readability of the text. Subflows can be invoked many times from many places. Remember that the use case can perform subflows in optional sequences or in loops or even several at the same time..

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 flow charts, activity diagrams 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.]

# **3.** **Alternative Flows**

[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 behavior¾ 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.

Start each alternative flow with an initial line clearly stating where the alternative flow can occur and the conditions under which it is performed.

End each alternative flow with a line that clearly states where the events of the main flow of events are resumed. This must be explicitly stated.

Using alternative flows improves the readability of the use case. 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.]

## 3.1 <Area of Functionality>

[Often there are multiple alternative flows related to a single area of functionality (for example specialist withdrawal facilities, card handling or receipt handling for the Withdraw Cash use case of an Automated Teller Machine). It improves readability if these conceptually related sets of flows are grouped into their own clearly named sub-section. ]

### 3.1.1 < A1 First Alternative Flow >

[Describe the alternative flow, just like any other flow of events.]

#### **3.1.1.1** **< An Alternative Subflow >**

[Alternative flows may, in turn, be divided into subsections if it improves clarity. Only place subflows here is they are only applicable to a single alternative flow.]

### 3.1.2 < A2 Second Alternative Flow >

[There may be, and most likely will be, a number of alternative flows in each area of functionality. Keep each alternative flow separate to improve clarity.]

## 3.2 <Another Area of Functionality>

[There may be, and most likely will be, a number of areas of functionality giving rise to sets of alternative flows. Keep each set of alternative flow separate to improve clarity.]

### 3.2.1 < AN Another Alternative Flow >

# **4.** **Subflows**

## 4.1 <S1 First Subflow >

A subflow should be a segment of behavior within the use case that has a clear purpose, and is "atomic" in the sense that you do either all or none of the actions described. You may need to have several levels of sub-flows, but if you can you should avoid this as it makes the text more complex and harder to understand.

## 4.2 < S2 Second Subflow >

[There may be, and most likely will be, a number of subflows in a use case. Keep each sub flow separate to improve clarity. Using sub 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.]

# **5.** **Key Scenarios**

[List the most important scenarios of the use case. Simply provide a short name and accompanying description to uniquely identify each key scenario. There will potentially be many scenarios possible with this use-case specification: it is important to focus on the most important or frequently discussed scenario’s that are either exemplars of this use case or are of concern or specific importance to the actor stakeholders.]

# **6.** **Preconditions**

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

## 6.1 < Precondition One >

# **7.** **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.]

## 7.1 < Postcondition One >

# **8.** **Extension Points**

[Extension points of the use case.]

## 8.1 <Name of Extension Point>

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

# **9.** **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 requirements¾such as operating systems and environments, compatibility requirements, and design constraints¾should be captured in this section.]

## 9.1 < First Special Requirement >

# **10.** **Additional Information**

[Include, or provide references to, any additional information required to clarify the use case. This could include overview diagrams, examples or any thing else you fancy.]