**Use Case Specification**

**DMT/RM01/TMP**

|  |  |
| --- | --- |
| Project Code |  |
| Project Name |  |

|  |  |  |
| --- | --- | --- |
| Prepared/Modified by | Role | Date of Preparation |
|  |  |  |
| Reviewed by | Role | Date of Review |
|  |  |  |
| Approved by | Role | Date of Approval |
|  |  |  |
| Circulation List |  | Version Number of the template:1.1 |
| Version Number | <version No > |  |

<<Customer>> REVIEW HISTORY

<<Customer comments on the Use case along with the signed off is tracked here>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | <<Version number>> |  |  |  |  |  |
| Date | <<Date of Review>> |  |  |  |  |  |
| Reviewed by | << Reviewer Name>> |  |  |  |  |  |
| Reviewed UI Specification doc | << Whether UI Specification doc is reviewed >> |  |  |  |  |  |
| All Open Queries/issues closed | << Whether all the open queries and issues resolved>> |  |  |  |  |  |
| Agreement on Assumptions | <<Whether all the assumptions have been agreed upon by the customer>> |  |  |  |  |  |
| Sign Off | <<Signature>> |  |  |  |  |  |

Disclaimer:

The scope of the project ‘<<Project Name>>’ is restricted to the contents of this signed off use case.

TABLE OF CONTENTS

[1.Use Case Name: <<Use Case Name>> 4](#_Toc302030199)

[2.Actor(s) 4](#_Toc302030200)

[3.Preconditions 4](#_Toc302030201)

[4.Flow of Events 4](#_Toc302030202)

[4.1 Basic Flow 4](#_Toc302030203)

[4.2 Alternative Flows 5](#_Toc302030204)

[4.2.1 Alternate Flow 1 5](#_Toc302030205)

[4.3 Sub Flows 6](#_Toc302030206)

[4.3.1 Sub Flow 1 6](#_Toc302030207)

[5.Post Conditions 6](#_Toc302030208)

[6.Special Requirements 7](#_Toc302030209)

[7.Extension Points 7](#_Toc302030210)

[<Name of extension point> 7](#_Toc302030211)

[8.Business Rules 7](#_Toc302030212)

[9.Diagrams 8](#_Toc302030213)

[Use Case Diagram 8](#_Toc302030214)

[Activity Diagram 9](#_Toc302030215)

[10.Scenarios 10](#_Toc302030216)

[Success Scenarios 10](#_Toc302030217)

[Failure Scenarios 10](#_Toc302030218)

[11.Issues 10](#_Toc302030219)

[12.UI Specifications 10](#_Toc302030220)

[13.Inter System Dependencies 11](#_Toc302030221)

[14.Integration with an already existing System of the <<Customer>> 11](#_Toc302030222)

[15.Assumptions 11](#_Toc302030223)

REVISION HISTORY OF THE WORK PRODUCT…………………………….. .. 12

# Use Case Name: <<Use Case Name>>

**Use Case ID:**  <<Customer>>.<<System>>.<<Use Case Number>>

**Brief Description:** <<Brief Description of the use case in approx 4-5 lines>>

# Actor(s)

<<List the actors that can interact with this use case.

Note: Actors can be a sub system or other external system>>

# Preconditions

<< List the preconditions that should be in place before executing this use case. These pre conditions include state under which other system or sub system or entities will be in. >>

# Flow of Events

4.1 Basic Flow

<< Basic Flow is the main flow or heart of the Use case.Use case starts when the actor does some action i.e. an actor always initiate a use Case. 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; it defines things like customer information there, to keep the use case from drowning in details. >>

**Name:** << Basic Flow Name>>

Select option.

<< State the option under which this use case is initiated>>

Select action.

<<List down the various selections that the user has after initiating the use case>>

Action Description:

<< State the various actions, which can be done by the user and system in this use case. Actions should be noted in a dialogue form such as ‘User does this..‘ , ‘System does this ……’ . Ideally this should be in bullet or number format

Separate out ultimate actions such as ‘Submit’, ‘Quit’ etc from ‘Action description’ and have separate steps for them as mentioned below.>>

Submit

<< State here the actions, which the system should take on ‘Submit’ request by the Actor.

Include validation in this section such as UI validation, Common Business and Use case specific Business rules validations.

Also include what action system should take if errors or warnings are encountered.>>

Quit.

<<State here the actions, which the system should take on ‘Quit’ request by the Actor.>>

* 1. Alternative Flows

<<More complex alternatives should be described in a separate section, which is referred to in the basic flow of events section. Think of the alternative flow sections like alternative behaviour – each alternative flow represents alternative behaviour (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 behaviour. When an alternative flow ends, the events of the main flow of events are resumed unless otherwise stated.

Note: Alternate flow should resume back to Basic Flow or Use case Ends. Always define the return or exit step>>

Alternate Flow 1

<< Alternate Flow should be divided in the below mentioned 4 sections:

**Entry Point:** State from where is this alternate flow called

**Condition:** Condition under when this alternate flow will be executed

**Action Description:** State the various actions, which can be done by the user and system in this use case. Action should be noted in a dialogue form such as ‘User does this……‘ , ‘System does this ……’ . Ideally this should be in bullet or number format

**Exit Point:** State where should the Alternate flow resume back. >>

Sub Flows

<<Functionality, which can be commonly used within a use case, is separated out in a section, which can be referred in the basic flow and alternate flow, such common functionality is described in a Sub flow. Think of the sub flow as a function, which can be called more than once in a use case. Sub flow has no existence outside this use case similar to Alternate flow. Sub flow always resumes back to the same point from where it was called. >>

Sub Flow 1

<< Sub Flow should be divided in the below mentioned 2 sections:

Entry Point: State from where do we enter in this Sub flow

Action Description: State the various actions, which can be done by the user and system in this use case. Action should be noted in a dialogue form such as ‘User does this……‘ , ‘System does this ……’ . Ideally this should be in bullet or number format >>

# Post Conditions

<< Post conditions are the STATE where the system, sub-system and /or entities will be after Basic and /or Sub flow and/or Alternate flows are executed.

State the Post conditions for Basic flow + each and every Sub flow and Alternate flow.>>

| Flow Name | Post Condition |
| --- | --- |
| << Alternate/Basic/ Sub Flow Name>> | << Post condition of the system, sub system and/or entities>> |
|  |  |
|  |  |
|  |  |

# 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 system common requirements such as operating systems and environments, compatibility requirements, and design constraints should be captured in Supplementary Specification.>>

# Extension Points

<<Mention the Extension points of the use case.>>

<Name of extension point>

<<Use extension points to specify the point of an extended use case where an extending use case's behaviour should be inserted>>

# Business Rules

<<Identify any Business Rules applicable to this Use Case. Any generic business rule should be captured in a separate Common Business rules document or in the supplementary specification]

| Business Rule Name | Business Rule Description | System action (if BR fails) |
| --- | --- | --- |
| << Business Rule Name example BR01>> | << Description of Business Rule>> | <<Action to be taken by the system on the failure of Business Rule>> |

# Diagrams

Use Case Diagram

<< Gives the relationship between Actors and Use cases [i.e. Main Use case, Include and Extends called by Main use case>>



Activity Diagram

<< Activity Diagram gives the high level interaction between the user, system and sub systems. Ideally only one activity diagram should be made per use case. >>



# Scenarios

[Identify the scenarios using Basic Flow, Sub flow and Alternate flows]

Success Scenarios

[List different success scenario.]

* <<Name of success Scenario>>
* <<List the flows involved i.e. basic and/or alternate and/or Sub flows involved in this success scenario>>

Failure Scenarios

[List different failure scenario]

<< Failure scenarios should include exceptions, validation of Use case and Common Business Rules, UI Validation and other failure conditions of the use case>>

# Issues

<< List any potential problems or known dependencies that are likely to cause this use case to fail (technical failure, staff absence, etc).

Note that this section should not have Queries related to this use case here, they should be tracked in a separate excel. If you wish you could link to that excel?

>>

1. UI Specifications

<< Provide a link to the UI specification document of the Use case. Please don’t embed the document here>>

1. Inter System Dependencies

<<Mention the related functionality within the application that is impacted because of this use case. E.g variable or value settings in this use-case which will have a direct impact on the functionality of another use-case. Or vice-versa.>>

**Module:** <<Specify the Module, which will get impacted because of this use case>>

Use case name: <<Use case Name>>

**Impact**: <<Mention the impact on the above mentioned Use case because of this use case>>

1. Integration with an already existing System of the <<Customer>>

<< This is especially applicable if the project at hand is an enhancement to an existing system.>>

**Module:** <<Specify the Module, which will get impacted because of this use case>>

**Entity:** <<List down the entities, which can be impacted because of this use case>>

**Information: <<**Mention the Impact in brief. >>

1. Assumptions

<< List down all the assumptions considered by this use case>>

**REVISION HISTORY OF THE WORK PRODUCT**

**<to be maintained by projects>**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Date | Version # | Section Changed | Details of changes made | Approved By |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |