Implementing effective services.

USE CASES

Use Cases are traditionally used in the interaction design phase of the development of services to work out the interaction flows. They are a means of roughing out the functionality of a service. Developers are very accustomed to seeing them and will understand them immediately, as will the business people who have over the years had to use them to communicate with programmers.

Use cases are a type of textual requirements specification that capture how a user will interact with a solution to achieve a specific goal. They describe the step by step process a user goes through to complete that goal.

Use cases capture all the possible ways the user and system can interact that result in the user achieving the goal. They also capture all the things that can go wrong along the way that prevent the user from achieving the goal.

Use Cases contain the following elements:

- Name – A clear descriptor that communicates the scope of the use case;
- Brief Description – A brief paragraph describing the scope of the use case;
- Actors – A list of the types of users who can engage in the activities described in the use case;
- Preconditions – Anything the solution can assume to be true when the use case begins;
- Basic Flow – The set of steps the various stakeholders take to accomplish the goal of the use case. A clear description of what the system does in response to each user action is also common.
- Alternate Flows – Capture the less common user/system interactions;
- Exception Flows – The things that can happen that prevent the user from achieving their goal;
- Post Conditions – Anything that must be true when the use case is complete.

The output of a cohesive use-case writing exercise is a series of stories that define the system to be designed. Use Cases also act as input into more detailed and pragmatic design artefacts, such as sketches, wireframes, information architecture diagrams, and service design blueprints.

Give each use case a unique numeric identifier. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labelled use case. Using the template attached to this document, complete the following sections;

- **Use Case Name** - State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun.
- **Created By** - Supply the name of the person who initially documented this use case.
- **Date Created** - Enter the date on which the use case was initially documented.
- **Last Updated By** - Supply the name of the person who performed the most recent update to the use case description.
- **Date Last Updated** - Enter the date on which the use case was most recently updated.
- **Actor** - An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.
- **Description** - Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.
- **Preconditions** - List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition.

- **Postconditions** - Describe the state of the system after the use case execution. Number each postcondition.
- **Priority** - Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.
- **Frequency of Use** - Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.
- **Normal Course of Events** - Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description.
- **Alternative Courses** - Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course"
- **Exceptions** - Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by "EX" to indicate "Exception".
- **Includes** - List any other use cases that are included ("called") by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.
- **Special Requirements** - Identify any additional requirements, such as non-functional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.
- **Assumptions** - List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.
- **Notes and Issues** - List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case Template

Use Case ID:			
Use Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	
Description:	
Preconditions:	
Postconditions:	
Priority:	
Frequency of Use:	
Normal Course of Events:	
Alternative Courses:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	