**System Use Cases**

[AGENCY NAME]

[PROJECT NAME]

[Publish Date]

**Table of Contents**

[Using this Template 1](#_Toc438536409)

[Revisions 2](#_Toc438536410)

[Introduction 3](#_Toc438536411)

[System Use Case Template Outline 4](#_Toc438536412)

[System Use Case Definitions 5](#_Toc438536413)

[Acceptance 11](#_Toc438536414)

# Using this Template

This template contains “suggested language” and assumes that the author of this document will make appropriate additions, deletions, and changes for their specific project needs.

To create a document from this template:

* Replace [bracketed text] on the cover page, in the header, and throughout the document with your project and agency information by filling in the [bracketed text] area in the document text. Filling in the information once, will propagate that field throughout the document.
* Complete the entire template making all necessary adjustments
* Each section contains abbreviated instructions (**Green Font**) and an example using (**Black Font**).
* Delete this “Using This Template” page.
* Update the Table of Contents by clicking on the “References” tab, selecting “Update Table”, then “Update Entire Table” and click “Ok”.
* Save.

To provide any suggested improvements or corrections, please email TBSM.info@tn.gov

# Revisions

| Revision | Description of Change | Author | Effective Date |
| --- | --- | --- | --- |
| v1 | Initial document upload to TBSM intranet site | BSD Team | 09/28/12 |
|  |  |  |  |
|  |  |  |  |

# Introduction

System Use Cases are developed to describe interactions between an actor and an Information Technology (IT) solution where a scenario may be completed in one session.

During high-level design sessions, system use cases can be written in a brief format, eliminating some of the granular detail until the modeling matures to the lowest functional breakdown of a functional requirement. Additional tools and techniques such as user interface prototyping, data flow diagramming, attribute table development, and activity diagramming may be used in conjunction with forming system use cases.

The following system use case template outline may be used to create a common template for individuals assisting with the process so that all sections and formatting are consistent. Different versions of a system use case template may be developed from the following to allow the documentation to become more detailed as system design matures. Use cases are meant to be flexible and should be tailored to an organization’s needs. The following outline is a recommendation only. Some sections such as non-functional requirements could be documented under a separate cover; however, they should be considered when system use cases are developed.

# System Use Case Template Outline

1. System Use Case Name:

Section1, Sub Section 1.1 and 1.2 will appear in all system use cases. For high-level descriptions of a functional area in the initial stage, these may be the only elements included for a “system use case brief”

Level:

Type:

* 1. Business Context
		1. Brief Description
		2. Business Goals and Benefits
		3. Business (Organizational) Area
		4. System Under Design
	2. Stakeholders
		1. Primary Actors
		2. Secondary Actors
		3. External Actors
	3. Triggers

When system use case briefs are complete and next level breakdowns are in process, elements in Section 1 and Section 2 may be included (“system use case outline”)

* 1. Preconditions
	2. Post Conditions
		1. Post Conditions On Success
		2. Guaranteed (or Minimal) Post-Conditions
1. Main Success Scenario
	1. Alternative Scenarios
	2. Extensions
2. Special Requirements

Sections 3 through 11 are used for final granular detail in order to fully vet a system process (“system use case description”). This information is used to communicate with other members of the design team such as an application architect and database administrator to develop system specifications.

* 1. IT Service Level (Non-Functional) Requirements
		1. Usability Requirements
		2. Reliability Requirements
		3. Performance Requirements
		4. Supportability Requirements
		5. Security Requirements
		6. Legal and Regulatory Requirements
	2. Constraints
1. Activity Diagram
2. User Interface
3. Domain (Business Entity) Diagrams
4. Open Issues
5. Information Items
6. Prompts and Messages
7. Business Rules
8. Related Artifacts

The following sections define the elements of a system use case. Some organizations may prefer to include the definitions in their customized template to assist in the training process for those who will develop the system use case package.

# System Use Case Definitions

1. System Use Case Name:

System use case names reflect a goal that an actor (an actor may be a system or an individual in a specific job role) needs to achieve. The name should follow a verb-noun approach with the verb being an action the actor needs to take, and a noun to reflect what is being done or the target of a specific action.

Examples of a business use case name: Record Sales Transaction.

Level:

The levels of a system use case are defined by the level of the IT service being provided. Some texts use terms such as “kite-level” to describe an end-to-end IT service, “sea-level” for an IT service met by an actor in one session, and a “sub-sea-level” to describe an IT service that may be included in an actor’s goal.

Examples of kite-level goals in a sales environment are: Process Customer Orders and Ship Customer Orders. Sea-level goals are: Record Sales Transaction, Create Warehouse Ticket, Fulfill Customer Order, Process Customer Payment, and Create Shipping Label. Sub-sea-level goals in the sales transaction are below the user’s level such as Create Bank Transaction, Compute Sales and Use Tax, Distribute Order to Warehouse, etc.

Type:

The type of use case describes how it is used in the system under discussion; the types are Base Use Case, Extending, Generalized, or Included. Use case types are developed when a system use case model is developed to graphically display how actors are going to achieve their goals.

A base use case refers to the goal the actor needs to achieve such as Record Sales Transaction.

An extending use case adds or “extends” functionality to a base use case but cannot stand alone as a specific goal. In the sales scenario, an extending use case example is Process Over Credit Limit.

A generalized use case expresses a parent-child relationship in a given scenario. In an online sales transaction the base (or parent) use case might be Order Products and contain two generalized or child use cases such as Purchase Downloads and Purchase Warehouse Products.

An included use case refers to a function that may be re-used in the system such as Search Catalog. The user in a sales transaction may need to search the catalog while performing the sale; however, the component can also be used by the Inventory Control Manager to look up the status of a product. An included use case is used by two or more base use cases in the system under discussion.

* 1. Business Context
		1. Brief Description

Provide a brief description (one paragraph) about the IT service.

.

* + 1. Business Goals and Benefits

Provide a rationale for the existence of the business process as well as the goals the actor needs to achieve when the outcome is successful.

* + 1. Business (Organizational) Area

Name the area being described; for example, Sales (high-level).

* + 1. System Under Design

Identify the IT system or components referenced by this use case.

* 1. Stakeholders
		1. Primary Actors

The primary actors are entities (person, system or event) that interact with the use case.

Actors are not specific individuals but can be a specific job role, such as Sales Representative. Actors may also be timed events such as “end-of-month”.

* + 1. Secondary Actors

The secondary actors are those who do not act directly with the use case but have a vested interest in a successful outcome.

Examples of secondary actors in the sales transaction would be the accounting director and the purchasing department.

* + 1. External Actors

External actors include any system, authority, or entity where the actor is outside the process but are taken into consideration because decisions made in this process may have a downstream effect in another process, or may be used to help increase the overall perspective of the system design.

Examples in the sales transaction would be the sales and use tax authority, the marketing department and the warehousing and distribution department.

* 1. Triggers

Name all of the events that cause the use case to begin, including temporal events.

Examples are: “Sales representative receives a customer call”, “End-of-Month”.

* 1. Preconditions

List all of the conditions that must be true (excluding trigger information) before the use case can begin.

In a sales transaction, “inventory figures are up to date”, “the customer’s account has been approved” are conditions that exist before this process begins.

* 1. Post Conditions

List all of the conditions that must be true upon completion of the use case. In the following two bullets, the post conditions will be divided into a successful versus non-successful completion.

* + 1. Post Conditions On Success

List all of the conditions that will be true when the use case completes successfully.

In the sales example: “The customer’s order is recorded” and “The payment is received” are outcomes of a successful completion.

* + 1. Guaranteed (or Minimal) Post-Conditions

List the conditions that will be true when the use case does not complete normally.

“The payment is not accepted”, “Inventory levels are not adjusted”, and “The order is not recorded” are examples of conditions that must be true for a sales transaction that does not complete normally.

1. Main Success Scenario

The main success scenario of a use case is to describe the most common path to successfully achieve the goal of the use case. Describe the scenario in easy to understand language so that all stakeholders comprehend the nature of the process and the way to achieve the system goals. While eliciting a success scenario, stakeholders may mention additional ways to achieving success or, more commonly, bring up paths that describe when things go wrong and how to resolve issues that arise. The additional paths (or flows) will be described in the following two sections. The steps in a main success scenario should be numbered in order to tie a step in the main success to alternative scenarios and extensions.

Continuing the sales transaction:

1. Sales representative searches for an item in inventory.
2. The system provides a list of items that meet the search criteria.
3. Sales representative chooses an item for purchase indicating the quantity.
4. Customer indicates their order is complete.
5. Service representative submits the order.
6. The system checks the customer’s account status and finds a regular customer.
7. The system sums the order and displays the final total.
8. Sales representative accepts account information and submits the payment.
9. The system verifies the account and processes the payment.
	1. Alternative Scenarios

Alternative scenarios describe situations in the main success scenario where a failure causes the flow to take a different branch. Every possible failure needs to be considered to fully realize the scope and complexity of a scenario. When outlining (referencing) the main success scenario, it is helpful to use an outline format in order to detail the complexity. For example, if the main success scenario has ten steps and the failure point happens at step 5, then the alternate may be labeled 5.1, with the next possible failure for step 5 being 5.2, etc. The steps to resolve failure 5.1 may be outlined 5.1.1, 5.1.2, etc. until the alternate path is completed with either “the use case ends” or “return to MSS (main success scenario) step 6” (or wherever the failure point returns to the successful path).

Example:

5.1 The system checks the customer’s account status and the customer has reached their credit limit.

5.1.1 The system displays an over credit limit to the Sales Representative.

 5.1.2 Sales representative places the order on hold.

 5.1.3 The system updates the order to a hold status.

 5.1.4 The use case ends.

2.2 Extensions

Extensions are documented for alternate paths the main success scenario may take when it is not a failure point.

An example follows:

6.1 The system retrieves the customer’s account status and finds it is a preferred customer.

6.1.1 The system applies a ten percent discount.

6.1.2 Return to MSS 7.

1. Special Requirements

The special requirements section identifies all non-functional requirements required by this particular use case. The list of examples below should not be considered as exhaustive, and if the requirement does not apply it can be skipped. Global or system wide non-functional requirements are documented under a separate cover.

Example:

* 1. IT Service Level (Non-Functional) Requirements
		1. Usability Requirements

Usability requirements describe the user friendliness goals for a specific user.

An example of a usability requirement is that the user must be able to complete a transaction successfully 90% of the time without guidance or specific instructions.

* + 1. Reliability Requirements

Reliability requirements describe measurements on accuracy, precision, availability, redundancy, and error handling.

An example of an accuracy requirement is that the function must detect 90% of system faults.

* + 1. Performance Requirements

Performance requirements are measurements for turn-around time, stress requirements, etc.

An example is that this use case must be able to support 2,000 concurrent users.

* + 1. Supportability Requirements

Supportability requirements include scalability, configuration ability and compatibility with other systems, etc.

An example of scalability is that the system must be able to increase the number of concurrent users without disruption to the system.

* + 1. Security Requirements

Security requirements describe issues around access to data, legal or homeland security requirements, privacy restrictions, etc.

An example of a security requirement is who in the organization (role) can update information.

* + 1. Legal and Regulatory Requirements

Legal and regulatory requirements may include information on sales taxes, or regulation requirements.

An example of a legal requirement is that all sales going across state lines must have sales tax applied.

* 1. Constraints

List any technological or architectural constraints on this use case.

An example of a technical constraint is that the solution must use the existing database.

1. Activity Diagram

Activity Diagrams are a type of User Modeling Language (UML) notation that expresses workflows for activities that are highly complex or difficult to articulate. The symbols for Activity Diagrams allow for the ability to have parallel processes, as well as, merging or forking activities in a way not expressed by traditional flowcharts. This section of the use case is optional when the use case does not need this type of granular notation. If UML is not available to the agency, then traditional flowcharting techniques may be used to express complexity.

1. User Interface

Use this section to describe the user interface needed for this use case. The description might include a prototype of the user interface but is not meant to be the final design of the product.

1. Domain (Business Entity) Diagrams

Include the class diagram depicting the business class and the relationships it has with other entities, as well as the multiplicities of all of the objects that need to be included in this use case.

1. Open Issues

List any assumptions, notes or problems that need to be addressed regarding this business process. When requirements are complete, this section should be blank.

1. Information Items

Include links to business rules or other artifacts that have relevance to this use case.

1. Prompts and Messages

Messages that need to be used in this process flow should be included in this section, or create a message catalogue in order to keep all messages to the user uniform.

An example of a message is “The customer is over their credit limit, please refer to accounting”.

1. Business Rules

Business rules can be referenced by a policy number, hyperlink to a business rules document or listed in the use case when a repository does not currently exist.

Business rules include items such as, “The system must check the status of the customer’s credit limit before creating a sales order.”

1. Related Artifacts

Include references to complex computations, decision tables, etc. that may be needed to articulate the system use case.

# Acceptance

(This section should be modified for best application to specific projects. Include all project team members that should have some level of authority regarding document review and approval.)

Approved by:

 Date:

<Approvers Name>

[PROJECT NAME] Executive Sponsor

 Date:

<Approvers Name>

[PROJECT NAME] Business Sponsor

 Date:

<Approvers Name>

[PROJECT NAME] Project Director/Manager

 Date:

<Approvers Name>

[PROJECT NAME] Stakeholder