

System Requirement Specification

(SRS Document)

What is SRS?

- System Requirement Specification (SRS) is a document which describes the software system to be developed in terms of the requirements specified.
- SRS provides a proper basis for the contract between the clients and developers/suppliers
- The main audience of the document is clients who are non-technical in their backgrounds. Therefore, it is very important to present the software solution in simplified form which can be understood by non-technical audience.
- It lays out functional and non-functional requirements of the system & set of use-cases which describes system-user interaction
- This documentation provides a basis to estimate production cost, risk and schedules. SRS also helps to prevent the risk of product failures or rejection of client as the clients can comment on proposed system before starting the development.
- Goals of delivering SRS document are:
 - State the scope of project to clients clearly
 - Facilitate the system reviews at the early stage of the software development process
 - Minimize the risk of entire project failure

Contents of a System Requirement Specification

General content of an SRS document includes:

- Introduction
- Scope of the project
- Functional Requirements
 - Use case Diagrams
- Non-Functional Requirements

1. Introduction

Under this topic you have to explain what you hope to present through this document. In here you have to clearly mention the purpose of the submission of the document. Then you also need to give a brief summary of the document content and a guideline for the audience.

Then provide a small introduction to your addressed problem and to your proposed system.

2. Scope of the project

Under here clearly explain the proposed solution by stating the boundaries and limitations of the system. Here it needs to include:

- Characteristics of the products
- Services provides by the end system
- End result of the project
- Under which conditions the project will be executed
- Constraints of the project
- If you are making some assumptions related to the project
- Acceptance criteria

3. Functional Requirements

Functional requirements define the functions deliver by the system. Functional requirements can be defined as inputs, behaviors and outputs. These are the requirements which can be identified through the use cases.

Functional requirements should include (ofnisystems.com, 2018):

- Descriptions of data to be entered into the system
- Descriptions of operations performed by each screen
- Descriptions of work-flows performed by the system
- Descriptions of system reports or other outputs
- Who can enter the data into the system?
- How the system meets applicable regulatory requirements

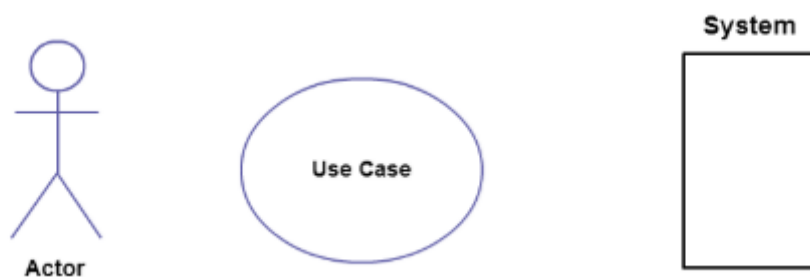
Use-case Diagrams

Use case diagram is a graphical representation of the user-system interaction. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements (whatis.techtarget.com, 2017).

First needs to identify the actors of your system. Actor is any entity which performs a role in the System. For example: if it is a School Management system, students and teachers will be the actors. Actors are represented with the use of drawn skeleton.

Then identify the functions or actions perform by the system. These are known as use cases and represented with the use of oval shape and in the middle of the oval you need to write the name of the use case.

Then draw the system boundary by representing the constraints of the system. System boundary represents with the use of rectangle shape which includes all use cases within it. Normally actors reside outside of the System boundary.



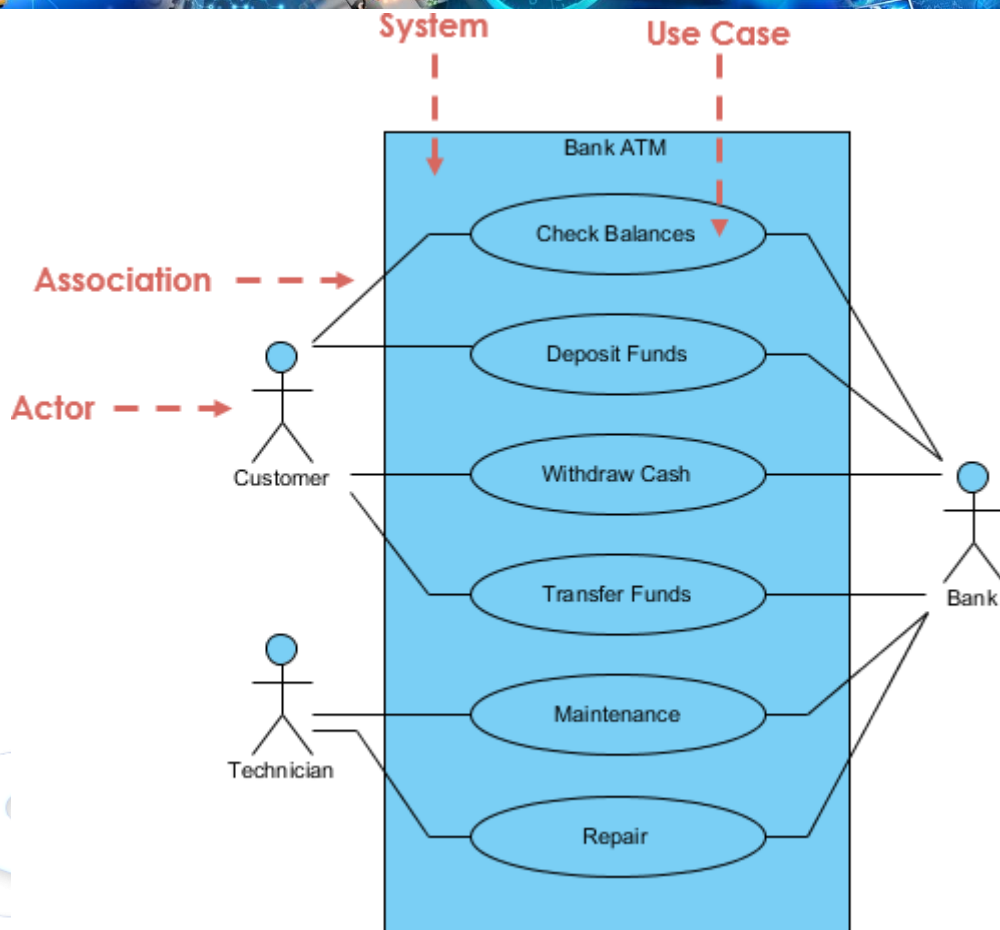


Figure 1: Sample Use case for ATM System (Source: Visual Paradigm Community Circle, 2018)

Draw the use-case diagram to entire to show the user-system interactions and internal system interactions separately. Do not forget to draw the system boundary to separate the users and system.

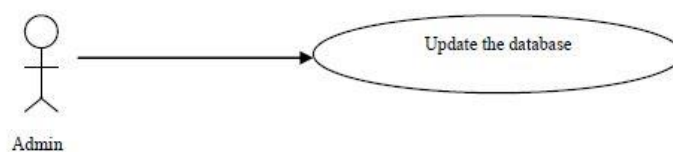
Then describe each use-case (Use case Description) separately using a proper format. Sample format is given below.

Use case 01 (number)	
Individual Use case Diagram	
Use Case Name	
Actor	
Overview (Description about the use case)	
Pre-Conditions	
Description (Steps of the activity)	1. 2.

	3.
Alternative flows (only if any)	Description: 1. 2.
Additional Descriptions (only if any)	
Post condition	

Example Use case Description taken from actual project is given bellow.

1.3.11 Use Case 11



Use Case Name: Update the database

Actor: Admin

Overview: Updates the database

Pre-conditions:

1. Administrator must log into the system

Description:

1. Administrator updates the database
2. End of the Use Case

Additional Descriptions:

1. Administrator updates the database only if there is any new data regarding the crops

Figure 2: Sample Use Case Description (AgroFriends,2014)

4. Non-functional Requirements

Non-functional requirements help define the constraints of the system. While functional requirements define what the system needs to do, the non-functional requirements specify the criteria which can be used to measure and judge the operations of the system.



Non-functional requirements of a system cover the following area:

- Security
- Reliability
- Performance
- Maintainability
- Scalability
- Usability
- etc

Some examples for Non-functional requirements:

Usability Requirements: (Source: AgroFriends,2014)

1. All the users may not be from a computer-based environment. Therefore, the proposed system interfaces are simple to understand and they are graphically aided.
2. All the system users should be able to login to the system by entering their user name and their preferred password.

Performance Requirements: (Source: AgroFriends,2014)

1. System must load quickly and should take maximum 5 seconds to perform functions
2. Database query processing time shall not be exceeded 50 milliseconds