Some Info on the Software Requirements Specification Document

09.03.2022

Some general guidelines:
Font: Calibri (Body) or Times New Roman,

Font Size: 12 with 1.5 spacing.

Main sections: Heading 1. Subsections 1.1. Heading2 etc.

Software requirements specification (SRS) document must contain the following:

Front page: with title of your product, Software Requirements Document, IUS logo, Project team members, date at the bottom.

Table of Contents

Abstract: One paragraph abstract should be included. About 300 words.

Document Revision History: It should include document revision history, including a rationale for the creation of a new version and a summary of the changes made in each version. E.g.

Rev1.0 March 22 ,2022 – initial version

Rev 1.1 March 29,2022 – added an additional requirement TODO explain the requirement

Rev 1.2 April 15, 2022 – removed xxx requirement…

Glossary: It should define technical terms used and abbreviations used in the document.

1 Introduction

This should describe the need for the system. It should include main goals, and *briefly* describe system's function and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software. This section should also include general constraints, assumptions and dependencies, as well as intended users of the system. Overview of the remainder of the document also needs to be included in this section.

The introduction section should serve as the user requirements as discussed in class. It should describe the services provided for the user. The non functional system requirements should also be *briefly* described in this section. Details will be included in sections that will follow. This description should use natural language that are understandable to customers.

2 System Features and Use Cases

This chapter should include product features, derived user requirements and corresponding functional requirements. First write down your product's major features with short description and feature priority (Use 4 scale (MoSCoW) style prioritization), including corresponding user requirements, and functional requirements. **Requirement with highest priority (must haves) should be implemented first. NOTE:** a feature represents one or more logically related system capabilities that provide value to a user and are described by a set of functional requirements. Functional requirement is a description of a behavior that a system will exhibit under specific

conditions. User requirement is a goal or a task that specifies classes of users must be able to perform with a system [1].

A note on MoSCoW: First state “Must haves” (minimal requirements, without which the project will fail), followed by “Should haves” (additional desired requirements with high priority, but not essential for a usable product) , “Could haves” (requirements that can be considered if there is time left) and lastly “Would haves” (would be nice to have, wishes for the future).

In this section you also need to include **non functional requirements** (emergent system properties, e.g.: reliability, security, system performance, availability, portability, organizational requirements, etc.) **in more detail (please read Ch4 Req Eng.ppt on functional and non functional requirements, and see example below)**. Interfaces to other systems may also be defined. This section should also include hardware requirements (i.e. the minimal and optimal configurations for the system.) You may also include database requirements, if any, (relationship between the data should not be included in this section but in the design section.)

2.1 System Features

SF1: Manage Login: A short description of the feature, e.g.

Login allows registered users/IUS personnel (IUS staff and IUS students) to access features that a user can access. Once registered, the users shall be able to login to the system in order to buy or borrow the book. Some parts of the system need to be made completely public, e.g.: viewing of all available books in the library to borrow and to buy, search option, etc.

UR1.1. The user shall be able to Register to the library.

> **FR1.1.1 User Registration (Must have):** The user shall be able to register with username as student/staff ID card number or email account username@ius.edu.ba, and an 8-digit password consisting of at least one symbol character.
>
> **FR1.1.2.** A short description of the functional requirement FR1.1.2

….

UR1.2. The user shall be able to Login with valid username and password

> **FR1.2.1 User Login (Must have):** The user shall be able to login by typing in their username and password where the users credentials will be validated. If correct the user logs in. In case of a forgotten password, the user shall be able to retrieve the forgotten password by email.

….

Functional requirements will further need to be written in terms of use cases.

Initially write down the main/primary actors: e.g. Viewer, Guest, Member, Librarian. Also add uses cases related to each primary actor in a table.

Table2.1 Use Case Table

Primary Actor	**Use Cases**
Viewer	1. Search Catalog

	2. View books 3. Register
Guest	4. Login 5. Add to Card 6. Remove from Cart 7. Purchase
Member	8. Request to borrow a book 9. Return book 10. …
Librarian	11. Add book 12. Edit book 13. Remove book 14. Add category 15. Modify category 16. Remove category 17. Add user 18. Modify user 19. Remove user 20. ..

2.2. Use Cases

First start with a use case diagram, see example from ppt slides, chapter 4. Recall that a use case describes a task a user would need to do.

2.2.1 Detailed use cases

Then for each use case, explore the use case in more depth, using the template below. **Note:** the Use Case ID must corresponds to the same number as stated above in System Feature section.

Use case ID and title: UR1.1 Register

Description: The purpose of remote access use case is to describe how can the user register to the system.
Priority: Must have
Pre-condition: Internet connection, browser
Post-condition: An account is created for the user
Basic Flow:

1. The user types the homepage url
2. The user click on 'register new account' link and is taken to account creation page
3. The user enters required information and clicks 'Register' button

What can go wrong

If the username already exists, error message will be displayed, warning the user and asking for new username. If the username does not exist, a confirmation message will appear, informing the user that the account has been created.

Use case ID and title: UR1.2 Log-in

The purpose is to describe logging of a user to the system
Priority: Must have
Pre-condition: User has an account
Post-condition: The user successfully logged into the system

Basic Flow:

1. The user types the homepage url
2. The user click selects login, enters username and password, and selects 'login' button.

Alternate Flow": optionally include

What can go wrong

If the incorrect combination username and password is entered, an error message will appear informing the user that username and password do not match and gives the user possibility to login again.

…….

Use case ID and title: UR2.1 Search

The purpose of the use case is to describe how to search/find a specific library item.

Priority: Must have

Pre-condition: User is logged in.

Post-condition: The user is presented with a list of library items that possibly match the entered search criteria.

Basic Flow:

1. The user types the homepage url
2. The user selects the Search link and are taken to search page.
3. The user should be able to search for a library item, using several search options. Those search options are: Item type (book, cd, journal, etc), book author, publisher, price.
4. There should also be a free text search option. A user should be able to select multiple search options in one search.
5. Search results can be viewed in a list. Each element in the list represents a particular library item, that should include library item type, ISBN, date, publisher. There should be max 30 results displayed per page. If the result contains more than 30 items, pagination will be displayed.
6. The list view should include a button when selected should display different filtering options in a filtering menu.
7. After the search results are displayed, user is allowed to either just view the information about the item or to buy it.

What can go wrong

The user may not enter any search criteria and may press Search button. In that case a warning message should be displayed, warning the user to select a search criteria.

…….

NOTE on use case scenarios: in the What can go wrong section, do not forget to include what does your application do to solve the problem, e.g. as in the above example, I wrote "In that case a warning message should be displayed, warning the user to select a search criteria.
"

2.3 Non-Functional Requirements

TODO in this section first provide an introductory text, i.e. what will the section contain and then refer to Table 2.2. Make sure that all the tables and figures used are LABELED, and also referred to in the text.

Table2.2 A list of non-functional requirements

Requirement	Definition	More Details
NFR1.: Hardware interface - I	Minimum hardware requirement for the server. (NOTE: you can simply write down your estimate of the min hardware requirements.)	**Operating System:** … **Processor:** … **Disk Space:** …

		RAM: ...
NFR2.: Hardware interface - II	Minimum hardware requirement for the client.	**Operating System:** … **Processor:** … **Disk Space:** … **RAM:** ...
NFR3.: Software Interface	Software requirements for the system.	Database: … ……
NFR4.: Communication Interface	Use needs connection to the internet to access the system via browser.	Internet connection
NFR5. Performance	…..	…..
…..	….	…….

2.4 System evolution: This should describe the fundamental assumptions on which the system is based and any anticipated changes (such as hardware evolution, changing user needs, etc.)

3 Release Plan

Table3.1 Release Plan

Requirement	Duration	Increment	Priority	Dependencies	Release
FR1	1 days	1	High, must have	-	1
FR2	2 days	1	High, must have	FR1	1
FR3	2 days	3	High, must have		1
FR4	2 days	2	High, must have		1

Timeline (Activity bar chart, or Gantt chart): Create a timeline graph where you clearly set the start and end date of each increment.

E.g. (from Somerville, Project planning chapter [2]. T represents a Task, M represents a milestone (i.e. points where you can assess progress.))

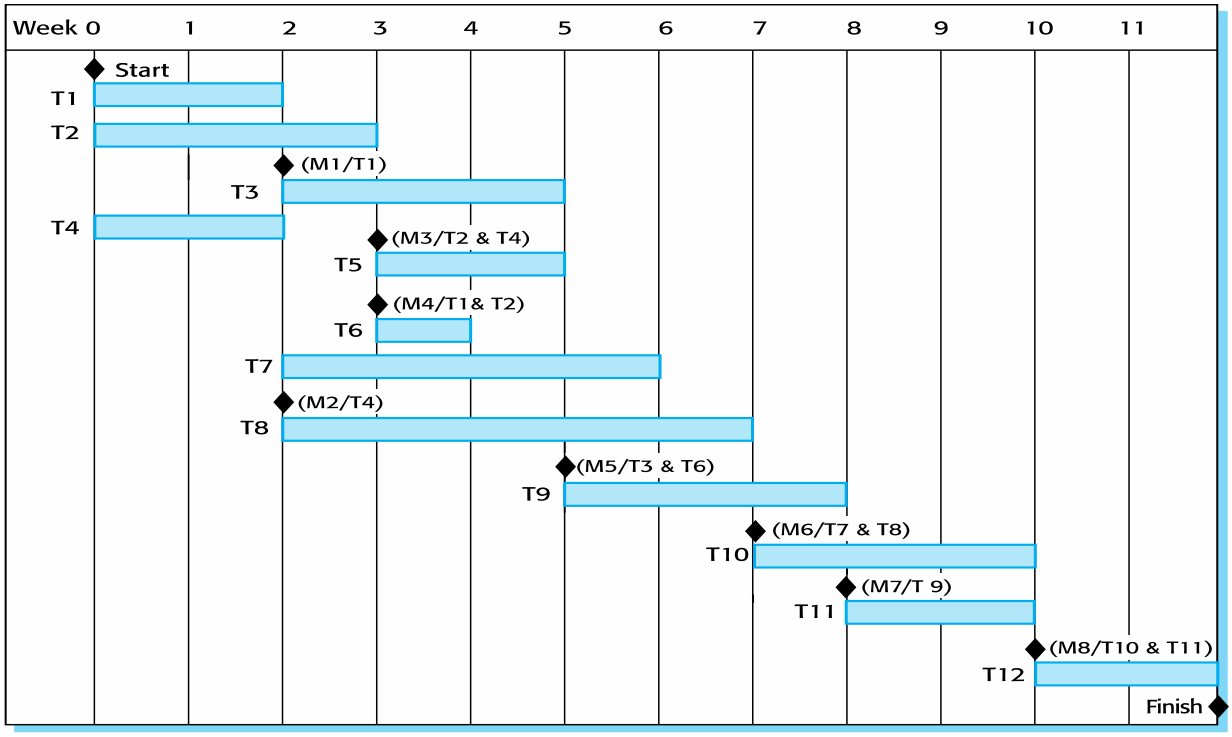

Figure 3.1 Timeline

An example of Staff allocation chart [2]:

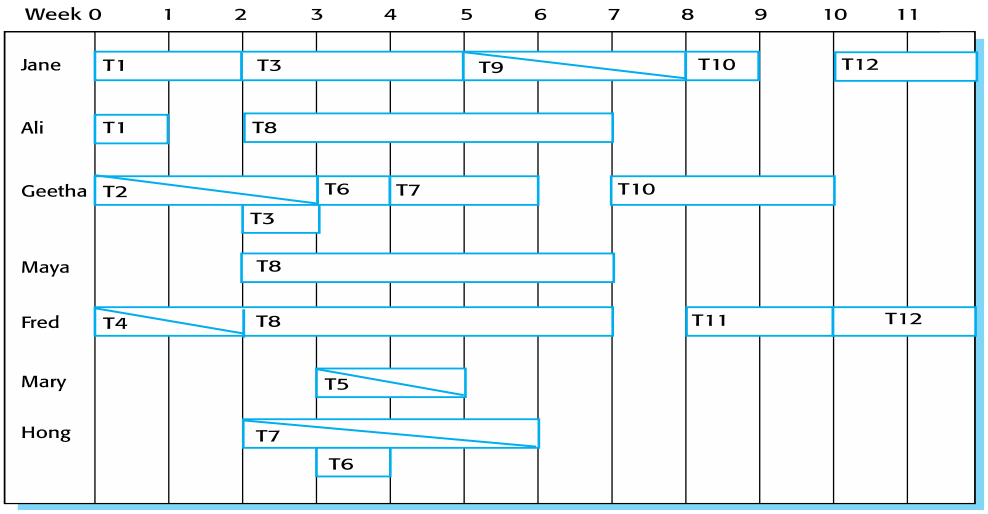

Figure 3.2 Staff allocation chart

References: Include any references used when writing this document. The references should be numbered, and each of the included references must also be present at the appropriate location in the text (the corresponding reference number)

Contribution Table

The document should include in detail the organization of each group, brief task description designated to each group member, completed/uncompleted tasks. You may also include info on the meetings held (date and duration), absence/presence of each team member, issues related to the project, etc**. (Please see the Project Contribution Table to be handed in below. I will not mark the work without the Project Contribution Table, if possible signed by each team member.) Note that the contribution table should be added as part of this document, and not a separate document.**

Each team member must be involved in all stages of software engineering process (requirements document, design, implementation and testing) **Take care to divide the work fairly and have equal task distribution among the team members, both in documentation and coding).**

Project contribution table

Project Name:

Team members:

MILESTONE (Circle one of the following:): Software Requirements Document/ Software Design Document/Coding/Presentation preparation

Date:

Team member name	**Task assigned (**a short description which should not be ambiguous!**)**	**Status of the tasks** (either completed/uncompleted. If uncompleted by the designated team member, write the member or the team who completed the task instead.)

REFERENCES

[1] Software Requirements, K. Wiegers and J. Beatty, Microsoft Press, 2013.

[2] Ian Sommerville, SOftare Engieering, 10th edition, 2015