

For the final project report you will not need to write 3 separate documents: ie. Vision and Scope, User Requirements and SRS document, which we will name Software Requirements Document.

You will submit one final document that will incorporate most important information that needs to be included in all of the 3 above mentioned documents.

The report must include the following:

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

Table of Contents

Table of Contents should be comprised of the following sections:

Document Revision History

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. NOTE: Document Revision History should not be a separate section, but just a bolded, left aligned title left on a single page.

e.g.

Rev1.0 October 10,2020 – initial version

Rev 1.1 October 15,2020 – added an additional feature TODO name the feature.

Rev 1.2 October 15, 2020 – removed User requirement UR1.3 ... name the user requirement

1. Introduction

The introduction should include the purpose of the document, the intended audience (in our case assume that the main audience are the developers of the software product), and structure of the document (for example Section 1 describes the overall description of the document, Section 2 outlines, Section 3...)

2. Overall Description

Overall Description, i.e. overall description/background of a software product that you would like to order as a customer (why are you building the product). Make sure that you include what the problem does, i.e. what is the software intended to solve, and why do you believe it would be important to build it (i.e. what real-world problem would it solve). Introduction should also include

Above should be followed by at least two business objectives, customer needs, vision statement, user classes, operating environment and assumptions and dependencies. Make sure

that each of the business objective, and customer needs and user classes you specify a unique ID , e.g. BO1.

2.1 Business objectives with clear success criteria,

e.g. BO1: Reduce order management costs by 25% by 2022,

BO2:centralize all orders from world-wide customers

....

2.2. Customer needs

e.g. CN1: allow payment to be made electronically using existing or new accounts,

CN2: allow customers to make track purchases for all consumable products using a web browser

....

2.3 Vision statement

Vision template:

- **For** [target customer]
- **Who** [statement of the need or opportunity]
- **The** [product name]
- **Is** [product category]
- **That** [major capabilities, key benefit, compelling reason to buy or use]
- **Unlike** [primary competitive alternative, current system, current business process]
- **Our product** [statement of primary differentiation and advantages of new product]

2.4. User classes

Identify the user classes that you anticipate will use this product, and describe their pertinent characteristics. Also include concise tasks for each user class and identify the favoured user class.

e.g.

UC1: system administrator

- daily patient scheduling, patient billing

UC2: doctor

- schedule summary, scheduling strategy, remote data access

UC3: office manager

- monthly reporting, complaint management

UC4: office assistant

- daily patient scheduling

2.5. Operating environment

Describe the environment in which the software will operate, including the hardware platform; operating systems and versions.

E.g.

The product must work on an isolated PC xxx machine and support xx operating system version xxx, with minimum of xxx GB of RAM. XXX browsers must be supported. OE-3: The system shall permit user access from the corporate intranet; from a VPN Internet connection; and by Windows smartphones and tablets.

2.6. Assumptions and Dependencies: if any.

3. System features and Use cases

This section 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 (either use the 3 scale priority levels discussed in class or 4 scale (MoSCoW)), including corresponding user requirements, and functional requirements. Then include the *big picture* of a feature tree (i.e. without the functional requirements). Thus the feature tree should contain a list of features (e.g. Manage bookmarks, Manage cookies, etc.) of the *named product* (e.g. *web browser*), with corresponding user requirements for each feature. NOTE: each user requirement should start with a verb.

3.1. System Features

FTR1: Manage bookmarks: A short description of the feature.

Priority: Must have.

UR1.1 Add a bookmark

FR1.1.1

FR1.1.2

UR1.2 Edit bookmarks

....

FTR2: Manage cookies

....

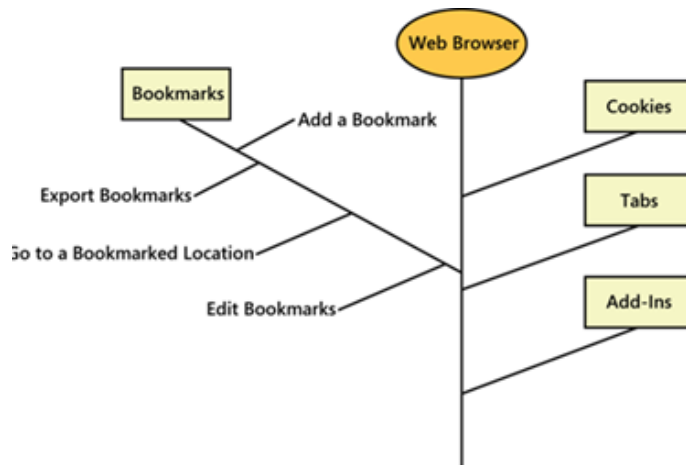


Figure xx Feature tree

The feature tree should be followed by the table (see template below) that clearly shows all of the features, user requirements, and corresponding functional requirements. NOTE: your table should also include the background colors (different versions of gray for example to make the text understandable.)

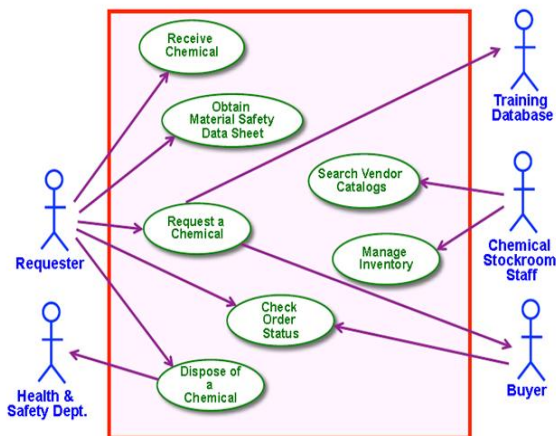
Feature ID	FTR1	Feature Name	Manage bookmarks
Priority	Must have		
User Requirement ID	UR1.1	User Requirement name	Edit bookmarks
Functional Requirement ID	FR1.1.1	F1.1.1 Description goes here...	
Functional Requirement ID	FR1.1.2	F1.1.2 Description goes here...	
...			
User Requirement ID	UR1.2	User Requirement name	Export bookmarks
Functional Requirement ID	FR1.2.1	F1.2.1 Description goes here...	
Functional Requirement ID	FR1.2.2	F1.2.3 Description goes here...	
Feature ID	FTR2	Feature Name	Manage tabs
Priority	Must have		
User Requirement ID	UR2.1	User Requirement name	Add Tabs
Functional Requirement ID	FR2.1.1	F2.1.1 Description goes here...	
Functional Requirement ID	FR2.1.2	F 2.1.2 Description goes here...	

Table 1.1 Product Features, user requirements and corresponding functional requirements.

Based on the Table 1.1. create a user case diagram that will include all the actors and use cases. Each use case should correspond to your identified user requirement.

3.2. Use case diagram

E.g. of a use case diagram:



3.2.1. 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 the User Requirement ID in Table 1.1.

User story		
Description		
Use case ID		
Priority		
Trigger		
Preconditions		
Postconditions		
	User/Actor	System response
Normal Flow		
Alternate flow		
Exceptions/What can go wrong		
Business rules		
Assumptions		
Notes and Issues	Notes and Issues	Resolution
	-	-

NOTE: in the above What can go wrong section make sure that you describe how the product should respond to anticipated error conditions and to invalid inputs and actions. For example for user login, if the user enters the wrong user name and/or password the system should inform the user that incorrect user name and/or password has been entered and prompt the user to re-enter the correct username with matching password.

4. Non Functional requirements

This section should include non functional product requirements:

4.1 Quality attributes (such as performance, availability, usability, reliability, security, etc.) which need to be specific, quantifiable and verifiable, see chapter 14 for examples.

E.g.

e.g.

PER-1. Authorization of an ATM withdrawal request shall take no more than 2.0 seconds.
PER-2. The anti-lock braking system speed sensors shall report wheel speeds every 2 milliseconds with a variation not to exceed 0.1 millisecond

4.1.1 Performance

CO-1: The system shall use the current corporate standard Oracle database engine.

CO-2: All HTML code shall conform to the HTML 5.0 standard.

4.1.2 Reliability

REL-1. No more than 5 experimental runs out of 1,000 can be lost because of software failures.

REL-2. The mean time between failures of the card reader component shall be at least 90 days.

.....

4.1.3. Usability

....

4.1.4.....

4.2 External interfaces (if any, for example if your system needs to communicate/receive input from other devices), design and implementation constraints

4.3. Internationalization and localization. Internationalization and localization requirements ensure that the product will be suitable for use in nations, cultures, and geographic locations other than those in which it was created. Such requirements might address differences in currency; formatting of dates, numbers, addresses, and telephone numbers; language, including national spelling conventions within the same language (such as American versus British English), symbols used, and character sets; given name and family name order; time zones; international regulations and laws; cultural and political issues; paper

sizes used; weights and measures; electrical voltages and plug shapes; and many others. Internationalization and localization requirements could well be reusable across projects.

5. User interface prototype

Include dialog maps for major/most critical user requirements of your system.

e.g. from chapter 17:

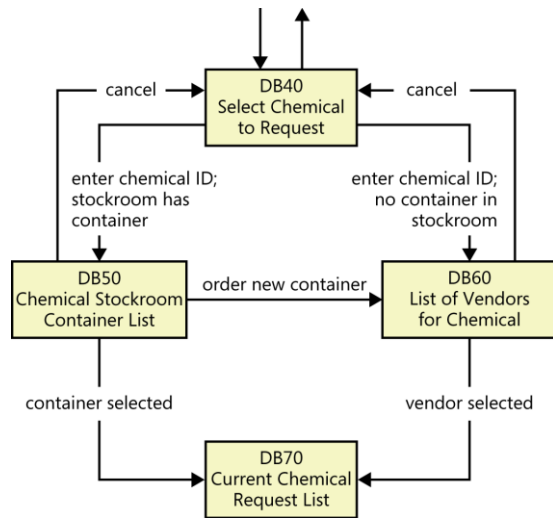


FIGURE 17-6 Portion of the dialog map for the “Request a Chemical” use case.

From the dialog map(s) provide either low or high fidelity prototype most critical user interface display that are included in your dialog map(s). Make sure that you name each UI display.

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)

Appendix A: Glossary