<Name of the project>
Software Requirement Specification

Project Code: <Code of the project>

Document Code: <Code of the document >– v<x.x>

<Location, issued date of the Document>

RECORD OF CHANGE

*A - Added M - Modified D - Deleted

Effective Date	Changed Items	A* M, D	Change Description	New Version

SIGNATURE PAGE

ORIGINATOR:	<Name>____________________	<Date>_____________________
	<Position>	
REVIEWERS:	<Name>____________________	<Date>_____________________
	<Position>	
	<Name, if it's needed>________	<Date>_____________________
	<Position>	
APPROVAL:	<Name>____________________	<Date>_____________________
	<Position>	

TABLE OF CONTENTS

1 INTRODUCTION

[The introduction of the ***Software Requirements Specification (SRS)*** *provides an overview of the entire* ***SRS****. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the* ***SRS****.]*

[Note: The ***SRS*** *document captures the complete software requirements for the system, or a portion of the system. Following is a typical* ***SRS*** *outline for a project using only traditional, natural-language style requirements—with* ***no use-case modeling.*** *It captures all requirements in a single document, with applicable sections inserted from the Supplementary Specifications (which would no longer be needed). For a template of an* ***SRS*** *using use-case modeling, which consists of a package containing Use Cases of the use-case model and applicable Supplementary Specifications and other supporting information.]*

[Many different arrangements of an ***SRS*** *are possible. Refer to [IEEE830-1998] for further elaboration of these explanations, as well as other options for* ***SRS*** *organization.]*

1.1 Purpose

[Specify the purpose of this SRS. The SRS fully describes the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]

1.2 Scope

[A brief description of the software application that the SRS applies to, the feature or other subsystem grouping, what Use-Case model(s) it is associated with, and anything else that is affected or influenced by this document.]

1.3 Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to the project's Glossary.]

1.4 References

[This subsection provides a complete list of all documents referenced elsewhere in the SRS. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

1.5 Overview

[This subsection describes what the rest of the SRS contains and explains how the document is organized.]

2 OVERALL DESCRIPTION

[This section of the SRS describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

- *product perspective*
- *product functions*
- *user characteristics*
- *constraints*
- *assumptions and dependencies*
- *requirements subsets]*

2.1 Product Perspectives

<Giới thiệu tổng quan về sản phẩm /Hệ thống. Vai trò, trách nhiệm của các thành phần - components trong sản phẩm

Thường trong phần này sẽ có sơ đồ thể hiện tổng quan , phạm vi của sản phẩm/hệ thống và kèm theo mô tả giao diện giữa các thành phần, nhiệm vụ các thành phần cần phải thực hiện>

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

<Dưới đây là ví dụ- Below is an example>

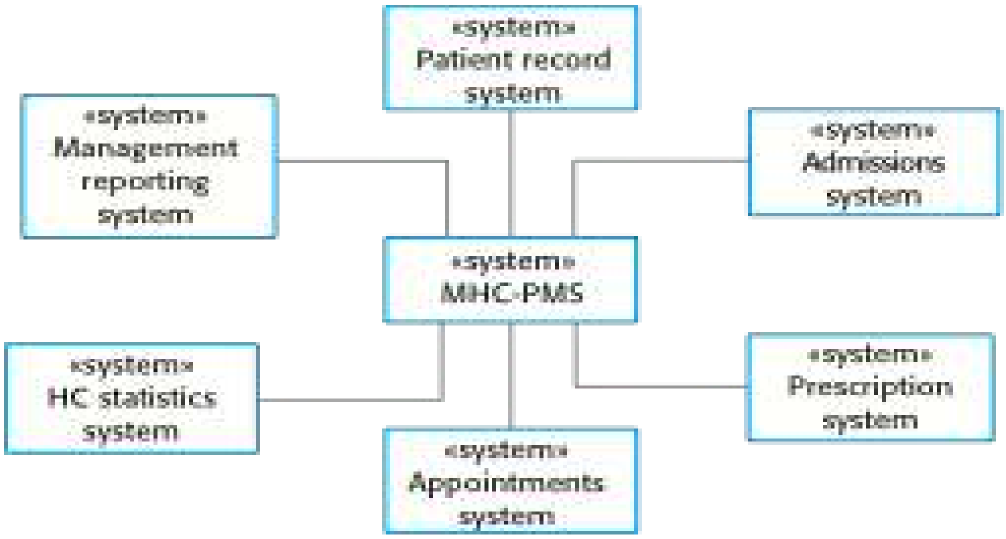

«system» Patient record system
«system» Management reporting system
«system» Admissions system
«system» MHC-PMS
«system» HC statistics system
«system» Prescription system
«system» Appointments system

Figure 2-1: System Overview of MHC-PMS

The system has six main components - Management reporting system, HC statistics system... ...

- Management reporting system: this component will receive the summary information about patient records and then produce monthly prescribed medicine report... and then send it to related stakeholders...
- HC Statistics System: ....

2.2 Product Functions

<Khái quát các chức năng của phần mềm cần phải cung cấp>

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

<Dưới đây là ví dụ-Below is an example>

The software system will provide the main functions:

- Authenticate and authorize users : the user only use software functions when he/she has been authenticated and authorized.
- Add New Staff Record:
- Update Staff Record:
- ...

2.3 User Requirements

<List of all requirements comes from end users>

2.4 User Characteristics

<Khái quát các loại người dùng và đặc điểm của các loại người dùng>

<Dưới đây là ví dụ>

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy. Below is an example>

The system will have the following categories of the users:

- User: ..
- Manager:

2.5 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.6 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

2.7 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

3 SPECIFIC REQUIREMENTS

[This section of the SRS contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the Use Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.]

3.1 Functionality

[This section describes the functional requirements of the system for those requirements that are expressed in the natural language style. For many applications, this may constitute the bulk of the SRS package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods may also be appropriate; for example, organization by user or organization by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modeling tools, and the like, are employed to capture the functionality, this section of the document would refer to the availability of that data, indicating the location and name of the tool used to capture the data.]

3.1.1 < Function 1/Use-case 1>

[The requirement or use-case description.]

3.1.1 Screen Definition

3.1.2 Use-case specification

3.1.2 Login <Vi du- sample Use Case>

3.1.3 Screen Design

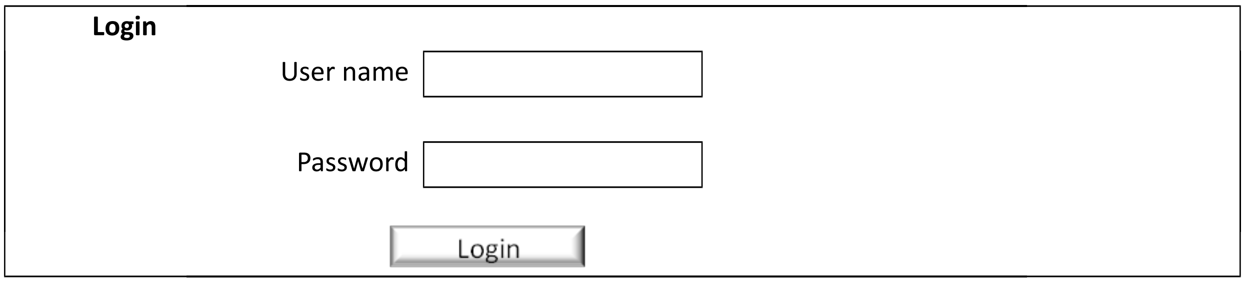
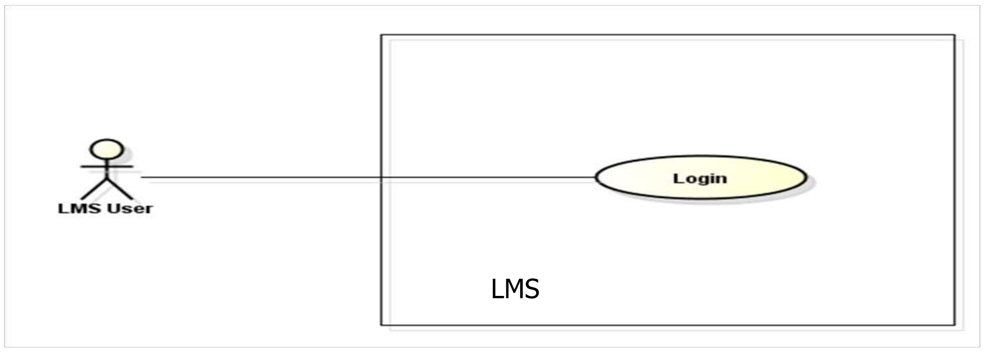


Figure 3-1: Screen Design of Login

Table 3-1: Screen Definition

#	Field Name	Type	Mandatory	Max Length	Description
1	User Name	Text	Yes	20	
2	Password	Text	Yes	20	Display "*" instead of clear character
3	Login	Button			Go to home page
4	Close	Button			

3.1.4 Use-Case Specification



Use-Case Description/Scenarios

Use-case ID	UC001	**Author**	

Use-case Name		Login
Actor		User
Description		<Mô tả khái quát chức năng của hệ thống, và tình huống bắt đầu (initial desc) của UC Description of use case, what use will do >
Precondition		<Mô tả điều kiện cần để UC có thể được thực hiện It is necessary condition for use-case to be operated >
Trigger condition		<Mô tả điều kiện đủ để UC được thực hiện - Sufficient condition for use to be operated >
Post-condition		<Điều kiện UC cân thỏa mãn sau khi thực hiên. The additional condition that use-case must meet after it have been operated>
Main flows (normal flows)		
Step	**Actor**	**Action Description**
1	User	Type URL of FAP into location field of Internet Browser and enter
2	Software	Display Login screen with the following fields: - User name - Password - Login button - Close button
3	User	Enter the user name & password into Login screen then click on "Login" button
4	Software	Validate the entered user name & password, then display Home page

Exception/(What went wrong)

#	Description
1	When the user type a wrong username or a wrong password, the system shall display an error message on the Login screen.

3.1.3 <Function 3/Use-case 3>

3.1.5 Screen Definition

3.1.6 Use-case Specification

3.1.4 <Function 3/Use-case 3>

3.2 Usability

[This section includes all those requirements that affect usability. For example,

specify the required training time for a normal users and a power user to become productive at particular operations

specify measurable task times for typical tasks or base the new system's usability requirements on other systems that the users know and like

specify requirement to conform to common usability standards, such as IBM's CUA standards Microsoft's GUI standards]

3.1.5 <Usability Requirement One>

[The requirement description goes here.]

3.3 Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

Availability—specify the percentage of time available (xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.

Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?

Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system's output.

Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code *(bugs/KLOC) or bugs per function-point(bugs/function-point).*

Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a "critical" bug; for example, complete loss of data or a complete inability to use certain parts of the system's functionality.]

3.1.6 <Reliability Requirement One>

[The requirement description.]

3.4 Performance

[The system's performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

Response time for a transaction (average, maximum)

Throughput, for example, transactions per second

Capacity, for example, the number of customers or transactions the system can accommodate

Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

Resource utilization, such as memory, disk, communications, and so forth.

3.1.7 <Performance Requirement One>

[The requirement description goes here.]

Interfaces

3.5 Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.]

3.1.8 <Supportability Requirement One>

[The requirement description goes here.]

3.6 Design Constraints

[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

3.1.9 <Design Constraint One>

[The requirement description goes here.]

3.7 On-line User Documentation and Help System Requirements

[Describes the requirements, if any, for o-line user documentation, help systems, help about notices, and so forth.]

3.8 Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

3.9 Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

3.1.10 User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

3.1.11 Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

3.1.12 Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS but with which this software application must interact.]

3.1.13 Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so forth.]

3.10 Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

3.11 Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notices, wordmark, trademark, or logo compliance issues for the software.]

3.12 Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

4 SUPPORTING INFORMATION

[The supporting information makes the SRS easier to use. It includes:

- *Table of contents*
- *Index*
- *Appendices*

These may include use-case storyboards or user-interface prototypes. When appendices are included, the SRS should explicitly state whether or not the appendices are to be considered part of the requirements.]