Tài liệu đặc tả yêu cầu phần mềm cho

Hệ thống quản lý bán thuốc

*Ngày 20 tháng 4 năm 2019*

Version 1

Nhóm 9:

Vũ Duy Mạnh, Nguyễn Công Lương, Phùng Bá Luân, UN LYAN, Paa Inthalone.

**MỤC LỤC**

[Các phiên bản sửa đổi 3](#_Toc6612441)

[1 Giới thiệu 4](#_Toc6612442)

[1.1 Tổng quan 4](#_Toc6612443)

[1.2 Mục đích và mục tiêu 4](#_Toc6612444)

[1.3 Phạm vi 4](#_Toc6612445)

[1.4 Các thuật ngữ 5](#_Toc6612446)

[1.5 Các quy ước 5](#_Toc6612447)

[1.6 Các giả định 5](#_Toc6612448)

[2 Các ràng buộc thiết kế chung 6](#_Toc6612449)

[2.1 Product Environment 6](#_Toc6612450)

[2.2 User Characteristics 6](#_Toc6612451)

[2.3 Mandated Constraints 7](#_Toc6612452)

[2.4 Potential System Evolution 7](#_Toc6612453)

[3 Các yêu cầu phi chức năng 7](#_Toc6612454)

[3.1 Tính khả dụng 7](#_Toc6612455)

[3.2 Vận hành 8](#_Toc6612456)

[3.3 Hiệu năng 8](#_Toc6612457)

[3.4 Tính bảo mật 8](#_Toc6612458)

[3.5 Tính an toàn 8](#_Toc6612459)

[3.6 Tính pháp lý 8](#_Toc6612460)

[3.7 Thuộc tính chất lượng khác 8](#_Toc6612461)

[3.8 Documentation and Training 8](#_Toc6612462)

[3.9 External Interface 8](#_Toc6612463)

[3.9.1 User Interface 9](#_Toc6612464)

[3.9.2 Software Interface 9](#_Toc6612465)

[4 Phân tích hệ thống thông tin của phần mềm 10](#_Toc6612466)

[4.1 Sơ đồ use case tổng quan của hệ thống 10](#_Toc6612467)

[4.2 Sơ đồ use case mức phân rã 11](#_Toc6612468)

[4.2.1 Phân rã use case “Quản lý nhập thuốc” 11](#_Toc6612469)

[4.2.2 Phân rã use case “Quản lý nhân viên” 12](#_Toc6612470)

[4.2.3 Phân rã use case “Quản lý bán thuốc” 13](#_Toc6612471)

[4.2.4 Phân rã use case “Quản lý khách hàng” 14](#_Toc6612472)

[4.2.5 Phân rã use case “Báo cáo thống kê” 15](#_Toc6612473)

[4.3 Sơ đồ tuần tự 15](#_Toc6612474)

[4.4 Sơ đồ hoạt động 15](#_Toc6612475)

[4.5 Sơ đồ lớp 15](#_Toc6612476)

Các phiên bản sửa đổi

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
|  |  |  |  |
|  |  |  |  |

# Giới thiệu

## Tổng quan

Tài liệu này xác định yêu cầu đối với hệ thống quản lý thuốc cho một cửa hàng. Mục đích của tài liệu này là thể hiện các yêu cầu hệ thống theo cách dễ đọc để khách hàng và các bên liên quan có thể hiểu chúng và xác minh chúng cho chính xác nhưng với đủ chi tiết để các nhà phát triển có thể thiết kế và triển khai hệ thống phần mềm.

Tài liệu này không giải quyết các vấn đề của dự án như lịch trình, chi phí dự án, phương pháp phát triển, giai đoạn phát triển, sản phẩm và quy trình thử nghiệm.

Hệ thống bán thuốc là một công cụ dựa trên Web để xuất bản nội dung lên Internet. Nó giúp cho việc quản lý sản phẩm thuốc cũng như giao dịch cho cửa hàng thuốc một cách tiện lợi và dễ dàng hơn.

## Mục đích và mục tiêu

Với bài toán quản lý bán thuốc đã phần nào đáp ứng được công việc của nhà quản lý và việc bán thuốc của cửa hàng trở nên đơn giản hơn, giảm bớt được các chức năng hoạt động thủ công. Thay vào đó là công việc của máy tính thực hiện như: tra cứu, tổng hợp báo cáo, tính toán lãi suất của cửa hàng…đã được hoàn thiện và áp dụng hoàn toàn trong quản lý tự động, khả năng làm việc nhanh chóng mang hiệu quả cao và độ chính xác lớn trong công việc.

Có 4 mục tiêu chính của “Hệ thống quản lý bán thuốc”, đó là:

1. **Nhanh chóng và hiệu quả:** hầu hết các công việc đều diễn ra trên máy tính với các thao tác đơn giản. Mọi yêu cầu của khách hàng đều sẽ thực hiện một cách nhanh chóng, không mất quá nhiều thời gian chờ đợi như trước đây.
2. **Chính xác và đầy đủ:** Các yêu cầu của khách hàng được đáp ứng một cách đầy đủ và chính xác, không có sự sai lệch như khi áp dụng việc quản lý bằng giấy tờ một cách thủ công như trước kia. Độ chính xác gần như tuyệt đối.
3. **Quản lý dễ dàng:** Người làm công tác quản lý có thể tra cứu tất cả các thông tin trong kho dữ liệu một cách nhanh chóng và kết quả trả về hoàn toàn chính xác và hết sức nhanh chóng.
4. **Giảm tải:** Là hệ quả của việc giải quyết các vấn đề trên, nhân viên bán hàng sẽ không phải chịu nhiều công việc như trước nữa. Làm cho năng suất công việc cải thiện đáng kể cũng như tăng hiệu quả cao phục vụ.

## Phạm vi

Hệ thống thực hiện việc xây dựng một phần mềm quản lý cơ sở dữ liệu ở mức một cửa hàng nhỏ, với số lượng thuốc không quá lớn. Mục tiêu chính của phần mềm là giải quyết sự chậm trễ và sai sót trong việc phục vụ yêu cầu của khách hàng hiện nay như nhập thuốc, xuất thuốc, lập và đưa ra các báo cáo. Ngoài ra phần mềm còn đem lại sự tiện lợi và chính xác trong việc thanh toán cũng như nhập xuất các loại thuốc

Trong thời gian tới, nhóm chúng em sẽ nghiên cứu và phát triển thêm các chức năng cũng như tăng quy mô của phần mềm.

## Các thuật ngữ

This section defines potentially unfamiliar or ambiguous words, acronyms and abbreviations. If terms such as “shall”, “should” and “may” are used to indicate importance the meaning of these terms should be defined here.

***Example:***

**Use case** – describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

**Scenario** – one path through a use case

**Actor** – user or other software system that receives value from a use case.

**Role** – category of users that share similar characteristics.

**Product** – what is being described here; the software system specified in this document.

**Project** – activities that will lead to the production of the product described here. Project issues are described in a separate project plan.

**Shall** – adverb used to indicate importance; indicates the requirement is mandatory. “Must” and “will” are synonyms for “shall”.

**Should** – adverb used to indicate importance; indicates the requirement is desired but not mandatory.

**May** – adverb used to indicate an option. For example, “The system may be taken offline for up to one hour every evening for maintenance.” Not used to express a requirement, but rather to specifically allow an option.

**Controls** – the individual elements of a user interface such as buttons and check boxes.

## Các quy ước

This section describes presentation conventions use in the document.

***Example:***

Portions of this document that are incomplete will be marked with TBD. Each TBD item will have an owner and estimated date for resolving the issue.

## Các giả định

In this section list any assumptions on which the requirements, as they are described here, depend. Assumptions are conditions, usually outside the control of the performing organization, that are taken for granted.

Be careful to only document assumptions that are outside the control of the performing organization. For example, the following is not a valid assumption for the requirements document: “We assume that all requirements will be documented.” You might be *assuming* this but there is no point in documenting it as an assumption here because it is something that is primary within the scope and control of the performing organization. The purpose of this section is to document assumptions that are outside the control of the performing organization—conditions that should be noted because someone will be taking responsibility for ensuring they hold.

A distinction can also be made between assumptions that pertain to the requirements and those that pertain to the project as a whole. The software project management plan is the most logical place to document assumptions that pertain to the project as a whole.

***Example:***

It is assumed that the client has an ODBC compliant database installed and this database will be accessible from the machine where the system will run.

It is assumed that the web hosting ISP will allow server-side scripts to access the file system.

# Các ràng buộc thiết kế chung

## Product Environment

Most software systems don’t exist in isolation. They interact with other systems and are part of a larger system or environment. This section describes the boundaries between the proposed system and its environment. The product context may include other hardware/software systems or a general business context. It may be helpful to specify the product environment with a block diagram that shows the major interfaces between the system under development and its environment.

If the system will use an existing database management system describe the interface here.

Note the user interface, which characterizes an interface between the system and its environment, is described below.

***Example:***

The innovative publishing system will be a component of the existing online course management system used in the computer science department at UMKC. The online course management system has built-in support for authentication. The innovative publishing system will use the existing course management system to identify and authenticate readers. The online course management system also includes a relational database which will be used through the JDBC/ODBC interface.

## User Characteristics

It’s important for developers to have a complete and accurate image of the end users. Even when the requirements of the user interface are described in detail the developer will still make many tiny design decisions during design and implementation. Knowing the general characteristics of the end users will help the developer make better decisions.

If the specific users of the system are known list them here. More likely there will be user roles or categories of uses. For each group of users list their responsibilities, characterize their knowledge of the domain and describe their characteristics including technical sophistication, background and education.

Prioritize users if necessary. This is especially important when user characteristics conflict. For example, if the system must accommodate experience users as well as novices it is important to know which should be given priority in case it’s not possible to accommodate both groups.

## Mandated Constraints

Ideally requirements will be specified in terms of functionality needed and developers will have free rein to design and implement a solution. In practice there are constraints on the eventual design and implementation.

Constraints may be mandated technologies. For example, the client may specify that a specific database management system, programming language, and/or operating system be used.

Constraints limit design and implementation options.

Constraints might be absolute, desirable or optional. If constraints aren’t absolute the motivation for the constraint should also be given.

## Potential System Evolution

The resulting software system should be maintainable and extensible. Knowing the types of anticipated changes aids significantly in establishing an architecture that will accommodate the types of expected changes. This section suggests ways the system is likely to be extended or modified in the future.

# Các yêu cầu phi chức năng

## Tính khả dụng

It’s hard to image a software system that doesn’t have usability as one of its highest nonfunctional quality requirements. It’s not enough to just say that the system should be usable though. Usability requirements must be stated in a quantifiable and testable way.

One method of specifying usability requirements is to specify efficiency, effectiveness and satisfaction goals for specific scenarios of use (section 4) carried out by representative users (section 2.2). A simpler alternative is to design a survey to measure user satisfaction and get consensus on who will take the survey and what will be considered an acceptable aggregate score.

## Vận hành

This section describes the general characteristics of the physical environment for the product.

***Example:***

The users’ environment is noisy so the system shouldn’t depend on the user hearing audible output.

## Hiệu năng

Thời gian khởi động hệ thống không quá 3 giây. Với 30 người dùng đồng thời, không có thao tác nào mất hơn 5 giây và 95% thao tấc sẽ mất ít hơn 2 giây.

## Tính bảo mật

Hệ thống sẽ không bị kẻ gian truy cập bất hợp pháp. Có các cơ chế phòng ngừa và ngăn chặn, dù kẻ gian có tài nguyên mạnh như thế nào cũng không thể can thiệp được vào hệ thống.

## Tính an toàn

Đảm bảo thuốc được bán cho khách hàng là chính xác, không có trường hợp bán nhầm thuốc hoặc bán thuốc hết hạn sử dụng cho khách hàng.

## Tính pháp lý

Some security and safety requirements may also be legal requirements. For example, federal law protects confidentiality of medical records.

***Example:***

Student social security numbers will not be visible to other students.

## Thuộc tính chất lượng khác

There are specific sections above for non-functional quality attributes such as security, performance, etc. In this section describe any other non-functional quality attributes such as portability, availability, etc.

## Documentation and Training

An important part of the total system is the documentation and training that is provided with the system. This section should describe the types and quantity of documentation and training that will be provided with the product.

## External Interface

External interfaces may be user interfaces or software interfaces.

### User Interface

The requirements document shouldn’t contain design or implementation details. The *logical* user interface should be described here. The description here shouldn’t unnecessarily constrain design and implementation options.

The general personality of the interface should be described here. For example, should the interface convey a conservative, professional, authoritative or fun attitude? What is the look and feel? Style? User characteristics were given above in section 2.2 but it may be helpful to characterize the average user here as adult, teenager or child.

User interfaces may contain a mixture of media types. It may be helpful to describe the desired/permissible user interface in terms of media elements.

Should the interface be intuitive or will training be provided. If training is required what types of training will be provided (online help, separate user manual, formal classroom training).

Ease-of-use requirements should be stated in a way that can be verified. For example, “the product should be easy to use” isn’t verifiable because it’s impossible to define “easy” in a measurable way. Must better is “75% of users will be able to use 80% of the features within 20 seconds without prior training”.

### Software Interface

The software interfaces may be locally addressable API’s or remote interfaces using technologies such as web services.

***Example:***

The operations defined by use case 4-7 below will also be available programmatically via XML over HTTP. The exact protocol is described the WSDL document at xyz.

# Phân tích hệ thống thông tin của phần mềm

## Sơ đồ use case tổng quan của hệ thống

A picture containing text, map

Description automatically generated

## Sơ đồ use case mức phân rã

### Phân rã use case “Quản lý nhập thuốc”

A close up of text on a white background

Description automatically generated

### Phân rã use case “Quản lý nhân viên”

A picture containing text

Description automatically generated

### Phân rã use case “Quản lý bán thuốc”

A picture containing text, map

Description automatically generated

### Phân rã use case “Quản lý khách hàng”

A picture containing text, map

Description automatically generated

### Phân rã use case “Báo cáo thống kê”

A picture containing text, map

Description automatically generated

## Sơ đồ tuần tự

## Sơ đồ hoạt động

## Sơ đồ lớp