<Logo of the Company/Group might be here. Use Cairo University logo>

<Company/Group name>

Find My Drug

Software Requirements Specification (SRS)
Version: 1.0

G5_SE02_v1.0

Revision History

SEC/ BN	Prepared/ Modified by	E-mail	Version	Date	Approved by	Descriptions/ Remarks
2/6	Kerolos Ishac	kero_es7ak@hotmail.com				
2/11	Mohamed Abd Elhady Ali	mohamed_abdelhady96@hotmail.com				
	Ahmed Samir Ibrahim	asamir97@hotmail.com				
	Mohamed Ali	muhamedali166@gmail.com				

Distribution list

Name	E-mail	Notes
Eng/Lydia Wahid	ENG.LYDIAWAHID@HOTMAIL.COM	
Dr/ Ahmed Kassab		

Table of Contents

Software Requirements Specification

1. Introduction

<The introduction of the SRS should provide an overview of the entire SRS. It should include the purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS.>

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for 'Find My Drug' website.

It will illustrate a complete declaration for the development of the website's system, the system's constraints under which it must operate and interactions with any external applications.

This document is primarily intended as a reference for developing the first version of 'Find My Drug' website, for the development team and it will be proposed to the customer for its approval.

1.2 Scope

'Find My Drug' is a service provider website that provides a platform for searching for drugs in the nearby pharmacies.

The website aims to reduce patients' suffering from the drug unavailability problem. It facilitates finding drugs by reducing the time you would take to manually search for drugs in pharmacies.

1.3 Definitions, Acronyms and Abbreviations

<This subsection should provide the definitions of all terms, acronyms, and abbreviations required to interpret properly the SRS. This information may be provided by reference to the project Glossary.>

DBMS	Database Management system
US	User Story
UC	Use case
NFR	Non Functional Requirement
DB	Database
LE	Egyptian Pound
FB	Facebook
PHMCS	Pharmacies
PHMC	Pharmacy

1.4 References

[1] Laravel PHP Framework version 5.4 documentation.

[2] jQuery mobile UI Framework version 1.4.5 documentation

1.5 Overview

The remainder of this document includes six chapters,the second one provides information about other similar services to 'Find My Drug', the third one briefly discusses the cost estimate of implementing the system and deploying it.

The fourth chapter goes over the feasibility of this project from different aspects, like if this system can be engineered using current technology and within budget and time constraints.

The fifth chapter provides the requirements specification in detail and a description of the different system interfaces, the sixth one goes over a schedule of all iterations of the project.

2. Market Survey

<Perform a search over the Internet for similar projects. If you know someone in the market who has a similar project, go and interview him/her. Now is a good time to give up being shy! For each similar project, give a separate section as follows>

2.1 Pharmacychecker

2.1.1 Project Description

Pharmacychecker is a website which provides a tool to search for drugs , getting a list of pharmacies having this drug , comparing different pharmacies prices and contact with a certain pharmacy to order your drug .

2.1.2 Functional Specifications

<This should provide a structured listing (1, 1.1, 1.2, etc.) of the functional specification in this project. It is better to group the specification into categories. This will help you to classify specifications in the summary section>

2.1.2.1 Drugs

2.1.2.1.1 Searching for a drug

The user shall be able to search for a specific drug by its name and get a list of all pharmacies –which have a partnership with pharmacychecker- having that drug.

2.1.2.1.2 Comparing prices

The user shall be able to compare certain drug prices in different pharmacies.

2.1.2.1.3 Local and Global pricing

The user shall be able to get the local and global price of a drug .

2.1.2.1.4 Drug details

The user shall be able to read a certain drug details if he searched for that drug

2.1.2.1.5 Ordering drug

The user shall be able to order drug from a specific pharmacy

2.1.2.2 Pharmacies

2.1.2.2.1 Rating a pharmacy

The user shall be able to rate a pharmacy after successful shipping.

2.1.2.2.2 Reviewing a pharmacy

The user shall be able to write a review for a specific pharmacy.

2.1.2.2.3 Viewing all pharmacies

The user shall be able to get a list of all pharmacies and links to their profiles.

2.1.2.2.4 Reporting a pharmacy

The user shall be able to report a specific pharmacy.

2.1.2.2.5 Pharmacies meet critical safety standards

The user shall be able to get a list of pharmacies which meet the critical safety standards.

2.1.2.3 Navigation through the site

2.1.2.3.1 Blog for news

The system shall provide a blog for latest drugs' articles.

2.1.2.3.2 Slideshow

The system shall provide a slideshow section to share the website goals and aims with the users .

2.1.2.3.3 In the news

The system shall provide an "In the news " section which highlights what media outlets are saying about pharmacychecker.

2.1.2.3.4 Patients Assistance Program

The system shall provide an alternatives to users who cannot afford expensive drugs .

2.1.3 Non-Functional Specifications

<This should provide a structured listing (1, 1.1, 1.2, etc.) of the non-functional specification in this project. It is better to group the specification into categories. This will help you to classify specifications in the summary section>

2.1.3.1 Security

The system shall be secure since credit cards are stored in the database there must be good security to protect them.

2.1.3.2 Usability

The system shall be flexible regarding searching tool since not every patient know exactly the name of the drug he wants.

2.1.3.3 Legality

The system shall be legal and in accordance to law.

2.1.3.4 Impartiality

The system shall be impartial and not partial to some pharmacies due to some business issues.

2.1.3.5 Efficiency

The system shall be efficient specially regarding drugs shipping , drugs shall be shipped within a week.

2.1.4 Limitations

<You should look for limitations present in this project and their reasons>

2.1.4.1 Limited delivery

The website only deliver drugs to usa citizens

2.1.4.2 Pharmacies

1. Pharmacies which wants to register to a partnership must have a website .
2. Only few pharmacies are available in the website .

2.1.4.3 Drugs

1. The website doesn’t handle well different names for same drug.

2.1.4.4 Efficiency

2. Searching tool isn’t pretty efficient at all .
3. The website database isn’t up-to-date , since there is no synchronizing between the website database and pharmacies database.

2.2 Egyptian Medical Index

2.2.1 Overview

Egyptian Medical Index is a mobile app which simply helps you to search for a specific drug , to get its current price , searching for a specific pharmacy to get its location or even searching for a hospital to get its information

2.2.2 Functional Specifications

<This should provide a structured listing (1, 1.1, 1.2, etc.) of the functional specification in this project. It is better to group the specification into categories. This will help you to classify specifications in the summary section>

2.2.2.1 Drugs

2.2.2.1.1 Searching for a drug

The user shall be able to search for a specific drug and get in return its information , price and even its chemical structure.

2.2.2.2 Pharmacies

2.2.2.2.1 Searching for a pharmacy

The user shall be able to search for pharmacies and get in return the pharmacy location and information about the pharmacy.

2.2.2.3 Hospitals

2.2.2.3.1 Searching for a hospital

The user shall be able to search for a hospital and get in return its information including its location .

2.2.2.4 Doctors

2.2.2.4.1 Search for a doctor by name

The user shall be able to search for a specific doctor by his name.

2.2.2.4.2 Search for a doctor by speciality

The user shall be able to search for doctors by their specialities.

2.2.2.4.3 Search for a doctor by city name

The user shall be able to search for doctors in a specific city .

2.2.3 Non-Functional Specifications

<This should provide a structured listing (1, 1.1, 1.2, etc.) of the non-functional specification in this project. It is better to group the specification into categories. This will help you to classify specifications in the summary section>

2.2.3.1 Up-to-date

The system shall be up-to-date with new updates in drugs prices

2.2.3.2 Efficiency

The system shall be very fast regarding returning search results.

2.2.3.3 Accuracy

The system shall be accurate regarding the information it gives to users about drugs/pharmacies.

2.2.3.4 Fast to learn

The user shall learn everything about the application service within 4 clicks (the application only has 4 tabs each tab represents searching tool for doctors or drugs or pharmacies or hospitals).

2.2.4 Limitations

<You should look for limitations present in this project and their reasons>

2.2.4.1 Database

The application database is relatively small since many drugs are not in the application database as well as many pharmacies and hospitals.

2.2.4.2 Lagging

The application suffer from lagging and freezing

2.2.4.3 No drug-pharmacy relation

The user cannot find out the pharmacy having the drug he searched for.

2.3 Summary

<This should summarize the previous section. You should classify the specifications

above into categories and specify the essential and non-essential specifications>

2.3.1 Functional Specifications

<This should provide a structured listing of the functional specification in similar projects grouped into categories. Write two sections for essential and non-essential specifications>

2.3.1.1 Essential Specifications

2.3.1.1.1 Drugs

2.3.1.1.1.1 ***Searching for a drug***

The user shall be able to search for a specific drug by its name and get a list of all pharmacies having this drug.

2.3.1.1.1.2 ***Drug details***

The user shall be able to read a certain drug details if he searched for that drug.

2.3.1.1.2 Pharmacies

2.3.1.1.2.1 ***Searching for a pharmacy***

The user shall be able to search for a specific pharmacy and get its location.

2.3.1.1.2.2 ***Viewing all pharmacies***

The user shall be able to view all pharmacies according to some filters he chose.

2.3.1.2 Non-Essential Specifications

2.3.1.2.1 Reporting

The user shall be able to report a specific pharmacy.

2.3.1.2.2 Rating

The user shall be able to rate a specific pharmacy.

2.3.1.2.3 Reviewing

The user shall be able to review a specific pharmacy.

2.3.1.2.4 Pharmacies' profile

The user shall be able to view pharmacies profiles.

2.3.2 Non-Functional Specifications

< This should provide a structured listing of the non-functional specification in similar projects grouped into categories. Write two sections for essential and non-essential specifications>

2.3.2.1 Essential Specifications

2.3.2.1.1 Efficiency

The system shall be efficient regarding searching tools and other services that the system provides.

2.3.2.1.2 Up-to-date

The system database shall be up-to-date with new drugs added or updated as well as pharmacies

2.3.2.1.3 Accuracy

The system shall be accurate about the information it provides for drugs and pharmacies.

2.3.2.2 Non-Essential Specifications

2.3.2.2.1 Fast to learn

The system shall be easy to learn all of its services within few clicks and navigations.

2.3.2.2.2 Lagging

The system shall not suffer from lagging.

2.3.3 Limitations

<You should summarize the limitations present in these projects and analyze their reasons. Discuss how your project may overcome these limitations>

1. All the previous services lack the big database and link of drug and pharmacies having that drug in an accurate way.
2. The previous services don’t serve all countries .
3. Ordering a drug is limited to specific countries.
4. Only few pharmacies are available in the previous services

3. Cost Estimate

<Give a rough estimate for the cost required for both implementing the system (cost for development, paying the engineers, programmers, testers, etc) as well as deployment costs (database licenses, servers, networks, hardware devices, maintenance) in a tabular form as follows>

Item	Estimated Cost	
	US Dollars	**L.E.**
1. **Development Cost**		
1.1. Developer Cost/hour	30	480
1.2. Total Developers Cost/hour	140	1920
Total	170	2400
2. Deployment Cost		
1.1 Domain Cost/year	52	832
1.2 Hosting Cost/year	60	960
Total	112	1783
Total Estimated Cost		

Table 2: Cost Estimate

<Provide different tables for different alternative solutions. At this phase we only require a broad estimate, not too accurate since you have not specified every requirements, yet search over the web for reasonable cost estimates>

4. Feasibility Study

5.3.1. Economic & Technical Feasibility Study :

This project is going to be developed at a sufficient cost so that it will be acceptable by all clients that need such type of applications.

Concerning the technical aspect, it is possible to develop the system using current technology and the available tools : PHP, Laravel Framework , MySQL , Java script , HTML ,CSS .

5.3.2. Organisational Feasibility study :

Our software application provides a solution for ordering a drug.it saves a lot of effort for both pharamacies and customers . so in this way we are reducing costs and time wasted

wasted and also we are promoting flexibility efficiency and personalization.

5.3.3. Integrability Feasibility study :

We are developing a cross-platform software application.we are making a web and mobile application. also our website will be responsive so that it can be user-friendly and easily viewed on so many

devices (mobile phones ,tablets ,etc…).

5. Requirements Specifications

5.1 User Requirements (User Stories)

<The services provided for the user should be described in this section. This section should be understandable by system users who do not have detailed technical knowledge. Each requirement should take a unique identifier as follows. Remember that you will be using these identifiers later on in the design, implementation and even testing phases>

<This should give a brief description for each functional requirement along with the importance priority for each according to the user>

Customer Stories:

5.1.1 :

ID	TITLE	DESCRIPTION	PRIORITY
CS01	Generic Searching	A customer can search for drug by typing in the search bar	High

5.1.2 :

ID	TITLE	DESCRIPTION	PRIORITY
CS02	Categorical Searching	A customer can search for specific drug in categories	Medium

5.1.3 :

ID	TITLE	DESCRIPTION	PRIORITY
CS03	Sending Note	A customer can send a note to the pharmacist if he couldn’t find the required drug	Medium

5.1.4 :

ID	TITLE	DESCRIPTION	PRIORITY
CS04	Requesting Drug	A customer can click on the required drug and get the nearest pharmacy which has the drug	High

5.1.5 *:*

ID	TITLE	DESCRIPTION	PRIORITY
CS05	View Pharmacy	A customer can view information of the pharmacy which has the required drug	High

5.1.6 *:*

ID	TITLE	DESCRIPTION	PRIORITY
CS06	Change Pharmacy	A customer can get the next nearest pharmacy which sells the drug	High

Pharmacist Stories:

5.1.7 *:*

ID	TITLE	DESCRIPTION	PRIORITY
PS01	Editing info	A pharmacist can edit the pharmacy's information like phone number and name etc ..	Medium

5.1.8

ID	TITLE	DESCRIPTION	PRIORITY
PS02	Accepting Drug	A pharmacist can accept drug request	High

5.1.9

ID	TITLE	DESCRIPTION	PRIORITY
PS03	Reporting Users	A pharmacist can report a user	High

5.1.10

ID	TITLE	DESCRIPTION	PRIORITY
PS04	Decline Drug Request	A pharmacist can decline a drug request	High

5.1.11

ID	TITLE	DESCRIPTION	PRIORITY
PS05	Requests History	A pharmacist can view his request history	Low

5.1.12

ID	TITLE	DESCRIPTION	PRIORITY
PS06	Login	A pharmacist can view Login	High

5.1.13

ID	TITLE	DESCRIPTION	PRIORITY
PS07	Logout	A pharmacist can logout	Low

5.1.14

ID	TITLE	DESCRIPTION	PRIORITY
PS08	Pending Requests	A pharmacist can view pending requests	High

Visitor (Pharmacy Applicant) Stories:

5.1.15

ID	TITLE	DESCRIPTION	PRIORITY
VS01	Apply For Pharmacy	A visitor can apply for having a pharmacist account by filling the form	High

5.1.16

ID	TITLE	DESCRIPTION	PRIORITY
VS02	Edit Application	A visitor can edit his previous form after clicking submit	Medium

Admin Stories:

5.1.17 :

ID	TITLE	DESCRIPTION	PRIORITY
AS01	Accept/Reject Pharmacy Application	An Admin can accept or reject pharmacy Application	High

5.1.18 :

ID	TITLE	DESCRIPTION	PRIORITY
AS02	View Applications	An Admin can View Current Applications	Medium

5.1.19 :

ID	TITLE	DESCRIPTION	PRIORITY
AS03	Edit Application	An Admin can edit the application information before accepting it	Medium

5.1.20 :

ID	TITLE	DESCRIPTION	PRIORITY
AS04	View Reports	An Admin can view reports on customer	High

5.1.21 :

ID	TITLE	DESCRIPTION	PRIORITY
AS05	Block User	An Admin can block a customer	Medium

5.1.22 :

ID	TITLE	DESCRIPTION	PRIORITY
AS07	Login	An Admin can login	High

5.1.23

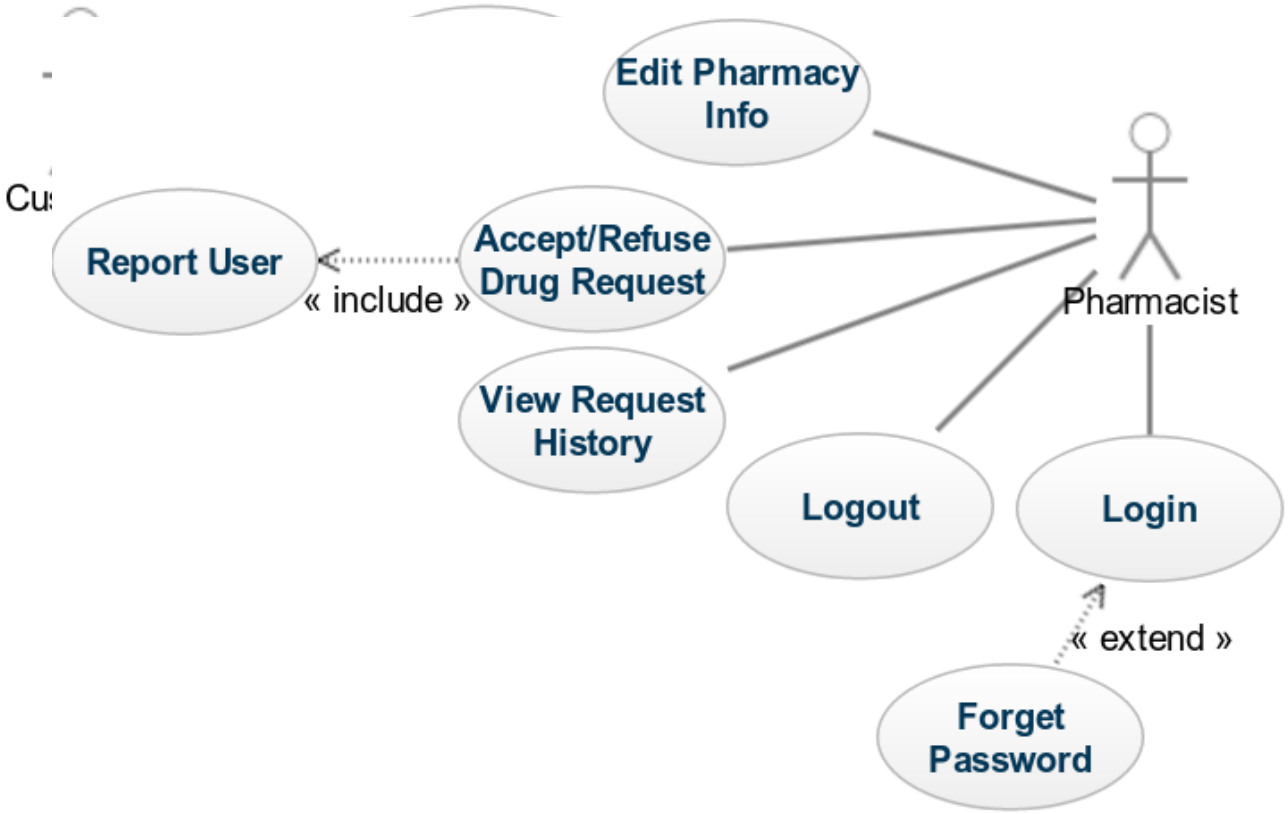
ID	TITLE	DESCRIPTION	PRIORITY
AS06	Logout	An Admin can logout	High

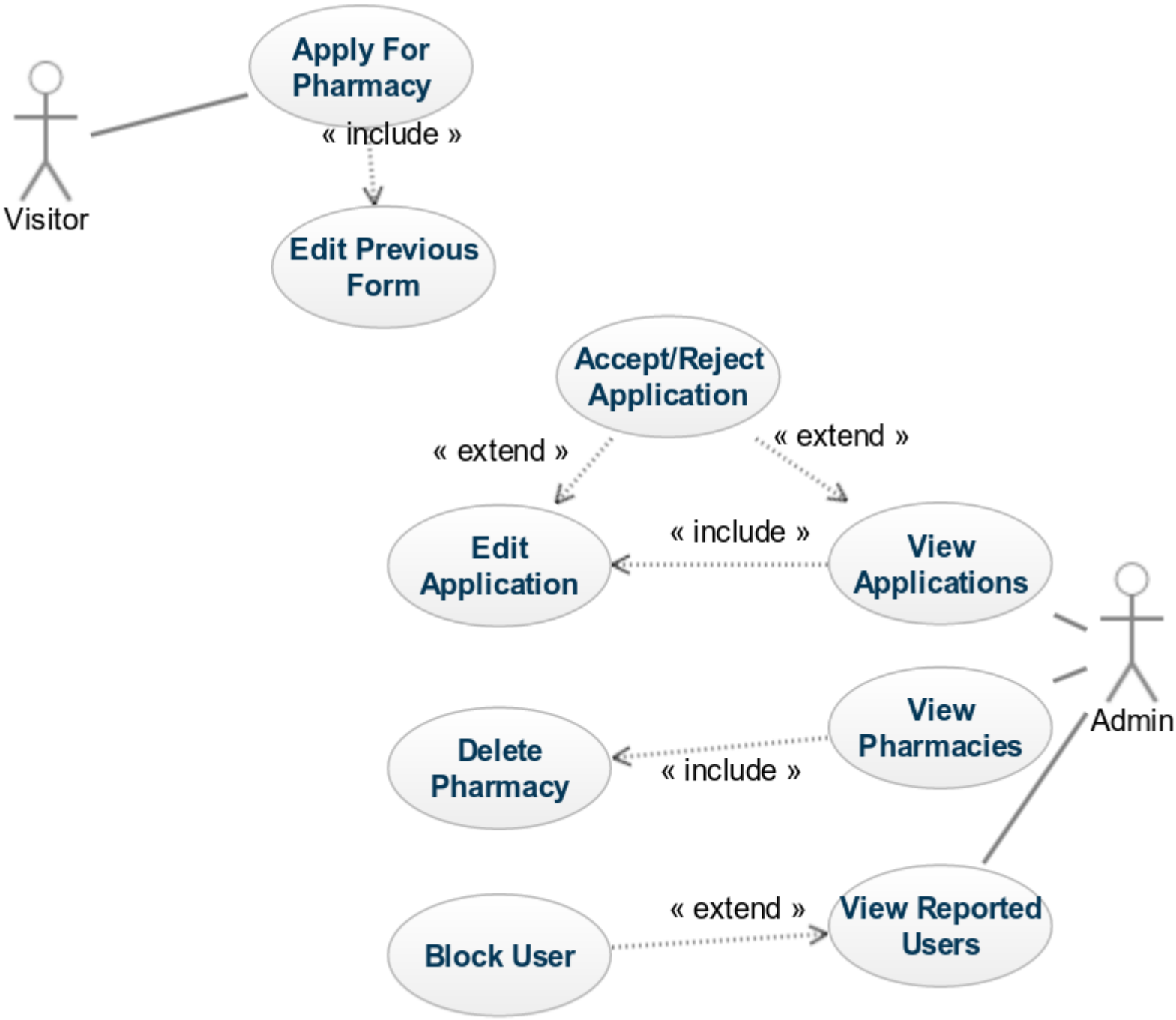
5.1.24 <Requirement Description and priority>

5.2 System Requirements (Use Cases)

<The services provided for the user should be described in this section from the system view not a user view. This section would be understandable by the development team. Each requirement should take a unique identifier as follows and map to the user story. Remember that you will be using these identifiers later on in the design, implementation and even testing phases.>

<Show first the use case diagram>


Edit Pharmacy Info
Report User
« include »
Accept/Refuse Drug Request
Pharmacist
View Request History
Logout
Login
« extend »
Forget Password


Apply For Pharmacy
« include »
Visitor
Edit Previous Form
Accept/Reject Application
« extend »
« extend »
Edit Application
« include »
View Applications
Admin
View Pharmacies
Delete Pharmacy
« include »
« extend »
View Reported Users
Block User

5.2.1

Name: Pharmacy Login

Identifier UC01

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user must not be logged in first

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the user opens the website and chooses to login as a pharmacy
2. Use case ends when the user enters a valid username and password then he'll be directed to the pharmacy's homepage.

Alternate Course A: Description of the alternate course
Condition: user enters non valid username or password
A.6 inform the user that there is an error and redirect him to the previous page and show "forgot your password?" button.

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. user can view his pharmacy's homepage

Map to: PS06

5.2.2

Name: Change own Info

Identifier UC02

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user is logged in

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the pharmacist views the edit my info page
2. pharmacist changes the required info
3. Use case ends when the user enters the password to validate the editing and saves

Alternate Course A: Description of the alternate course
Condition: some fields are left empty or containing invalid info
A.7 user gets an indicative message showing the field which he entered wrong
A.8 user is redirected to the previous page

Post conditions (*List the state(s) the system can be in when this use case ends*)

2. info is edited and user is redirected to the home page

Map to: PS01

5.2.3

Name: View Drug Requests

Identifier UC03

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user must be logged in as a pharmacy

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the user receives a drug request or the user opens the requests tab
2. Use case ends when the user accepts or declines the request

Alternate Course A: Description of the alternate course
Condition: pharmacist finds offensive note or excessive requests by the same customer
A.9 pharmacists reports the customer

Post conditions (*List the state(s) the system can be in when this use case ends*)

3. the pharmacist returns to requests page

Map to: PS08

5.2.4

Name: View Requests History

Identifier UC04

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user is logged in as a pharmacy

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the user clicks on view requests history
2. Use case ends when the user changes to another page

Alternate Course A: Description of the alternate course
Condition: there is no previous requests
A.10 an indicative message is shown

Post conditions (*List the state(s) the system can be in when this use case ends*)

4. nothing changes user stays on the same page until he clicks on home page

Map to: PS05

5.2.5

Name: Pharmacy Logout

Identifier UC05

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user is logged in as pharmacy

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when user clicks on logout

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. user is redirected to login page

Map to: PS07

5.2.6

Name: Generic Search

Identifier UC06

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user isn't blocked

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

2. Use case begins when user starts typing in the search bar
3.
4. Use case ends when the user hits enter and get the search results

Alternate Course A: Description of the alternate course
Condition: user doesn't find the drug he searched for
A.6 user is offered to send a note or try Categorical searching

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. user gets a list of drugs matching what he typed

Map to: CS01

5.2.7

Name: Categorical searching

Identifier UC07

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user isn't blocked

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the user choses to browse drugs by categories
2. Use case ends when the user finds his drug

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. user finds his drug

Map to: CS02

5.2.8

Name: Note Sending

Identifier UC08

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user isn't blocked

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when user choses to send note to the pharmacies
2. user then choses what kind of note he wants to send (image/text)
3. Use case ends when user sends the note

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. user is redirected to the waiting for reply from pharmacies page

Map to: CS03

5.2.9

Name: Requesting Drug

Identifier UC09

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user is not blocked
2. a drug is already chosen or a note has been selected

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when user chooses a drug by any method
2. Use case ends when the user gets the nearest pharmacy

Alternate Course A: Description of the alternate course
Condition: no pharmacy replies
A.7 user gets an indicative message

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. database is updated
2. a request is sent to the nearest pharmacies

Map to: CS04

5.2.10

Name: View Pharmacy

Identifier UC10

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user isn't blocked
2. user requested a drug
3. a pharmacy accepted the request

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the user is waiting for the request and a pharmacy accepts it
2. Use case ends when the user gets information about the pharmacy

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. database is updates with all the requests data and the requests is removed
2. user can view another pharmacy which accepted the request

Map to: CS05,CS06

5.2.11

Name: Apply For Pharmacy

Identifier UC11

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. user must not be logged in first

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the user opens the website and chooses to Register as a pharmacy
2. Use case ends when the user clicks submit

Alternate Course A: Description of the alternate course
Condition: The user doesn't enter the main information about the pharmacy and the location
A.6 user gets an indicative message showing the field which he entered wrong
A.7 user is redirected to the previous page

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. the user's data and approve button is shown

Map to: VS01

5.2.12

Name: Edit Application

Identifier UC12

Preconditions (*List the state(s) the system can be in before this use case starts*)

1. the user apply for a pharmacy

Basic Course (*Describe the "normal" processing path, aka, the Happy Path*)

1. Use case begins when the user is submitting the information
2. Use case ends when the user clicks submit

Alternate Course A: Description of the alternate course
Condition: Indicate what happened
The user doesn't enter the main information about the pharamcy and the location
A.6 user gets an indicative message showing the field which he entered wrong
A.7 user is redirected to the previous page

Post conditions (*List the state(s) the system can be in when this use case ends*)

1. pharmacy's homepage is viewed

Map to:VS02

5.2.13

Name: Accept or reject pharmacies' applications

Identifier: UC13

Preconditions

1. Admin logged in as admin to have full access to accept or reject pharmacies' applications.
2. Admin navigated to control panel

Basic Course

3. Admin navigated to control panel.
4. Admin click on view pharmacies' applications button.
5. Admin accept or reject each pharmacy request.

Alternate Course A:
Condition: Pharmacy's form needs to be edited.
Steps:

3. Admin click on edit button on a certain form
4. Admin is navigated to edit panel
5. Admin edit the pharmacy's application form
6. Admin click on save

Post conditions

2. Database updated
3. Admin navigate to home page to see some statistics

Map to: AS01,AS02,AS03

5.2.14

Name: Block users

Identifier :UC14

Preconditions

2. Admin logged in and redirected to home page

3. Admin navigate to control panel

Basic Course

1. Admin navigate to control panel.
2. Admin click on view reports button.
3. Admin block a specific user

Alternate Course A

Condition: Admin decides not to block reported user

Steps:

1. Admin decides that reports on that user are not many enough.
2. Admin continue evaluating other reports

Post conditions

1. Admin navigate to other tabs in the control panel.
2. Admin read more users' reports' statistics.
3. Admin navigate to home page to see more about the system statistics.

Map to: AS04, AS05

5.2.15

Name: Logging in

Identifier: UC15

Preconditions

1. Admin navigated to the application

Basic Course

1. Admin navigated to the application.
2. Admin entered his/her username and password
3. Admin becomes logged in .

Alternate Course A:
Condition: Invalid username or password
Steps:

4. Admin is asked to enter username or password again.
5. Admin is asked to click on forget password if he didn't remember his/her password

Alternate Course B:
Condition: User is already logged in
Steps:

3. Admin is offered a logout button to logout if he/she wants.
4. Admin is given a full access from the system to edit any of DB entries.

Post conditions

5. Admin navigates to home page
6. Admin navigates to control panel

Map to: AS07,AS06

5.2.16 <Use Case Description>

5.3 Non-functional Requirements

< These are the constraints imposed by the user or understandable from the user stories and their business. Each requirement should have a unique identifier that can be used as a reference in other documents.>

5.3.4. Fast response time

ID:*NFR1*
Description: *The System Should be able to respond quickly during any request such as Searching for drug or even navigating through the application.*
Addressing:*CS01,CS02,CS03,CS04*

5.3.5. Security

ID:*NFR2*
Description: *The System shall be highly secure to prevent private pharmacies' information to be stolen.*
Addressing:*CS05,PS01*

5.3.6. Services

ID:*NFR3*
Description: *The system should be dependable errors regarding drug*

requests should be minimized as possible.

Addressing:*CS04,PS02,PS04*

5.3.7. Availability

ID:*NFR4*
Description : *The system should be available at all times (patients' health may vary and drug availability matters).*
Addressing:*CS01,CS02,CS03,CS04,CS05,PS02,PS04*

5.3.8. Up-to-date

ID:*NFR5*
Description: *The system should be up-to-date regarding pharmacies' and drug information.*

Addressing:*AS01,AS06*

5.3.9. Fast loading

ID:*NFR6*
Description: *The system should load fast without lag .*
Addressing: *CS01,CS02,CS03,CS04,CS05,CS06*

5.3.10. Cross platform

ID:*NFR7*
Description: *The system should operate on different platforms without errors.*
Addressing:*CS01,CS02,CS03,CS04,CS05,CS06*

6. High level plan

<Give a high level plan of the schedule of all iterations you expect the project to undertake in a tabular form as follows.>

Iteration 1:

User Stories	Estimated Time
CS02	2H
CS04	4H
CS05	1H
PS02,PS03	4H
PS06,PS07	1H
PS08	5H
Total Time	17H

Iteration 2:

User Stories	Estimated Time
CS06	4H
PS01	1H
PS03	4H
PS05	1H
AS01,AS02,AS03	2H
AS04	1H
AS06,AS07	1H
Total Time	14H

Iteration 3:

User Stories	Estimated Time
1.CS01	4H
2.CS03	4H
VS01,VS02	4H
AS05	2H

Total Time	14H

Table 1: High Level Plan

<Please be reasonable in your time estimation. It does not have to be exact, we are here to learn, and now is a good time to learn some time management. *For the software engineering course make each iteration runs for three weeks only. In this course, you will be asked to deliver three iterations only.*>

7. Supporting Information

[The supporting information makes the SRS easier to use. It includes: a) Table of contents, b) Index, c) 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.]

Appendix A

Traceability Matrix

	PS							CS						VS		AS						
ID	01	02	04	05	06	07	08	01	02	03	04	05	06	01	02	01	02	03	04	05	06	07
UC01					✓																	
UC02	✓																					
UC03							✓															
UC04				✓																		
UC05						✓																
UC06								✓														
UC07									✓													
UC08										✓												
UC09											✓											
UC10												✓	✓									
UC11														✓								
UC12															✓							
UC13																✓	✓	✓				
UC14																			✓	✓		
UC15																					✓	✓
NF1								✓	✓	✓	✓											

NF2	✱											✱										
NF3		✱	✱									✱										
NF4		✱	✱					✱	✱	✱	✱	✱										
NF5																✱					✱	
NF6								✱	✱	✱	✱	✱	✱									
NF7								✱	✱	✱	✱	✱	✱									

Appendix B
Dependability Matrix

	PS								CS						VS		AS						
ID	01	02	03	04	05	06	07	08	01	02	03	04	05	06	01	02	01	02	03	04	05	06	07
PS01						✱																	
PS02						✱																	
PS03						✱																	
PS04						✱																	
PS05	✱	✱	✱	✱	✱		✱	✱															
PS06						✱																	
PS07						✱																	
PS08						✱																	
CS01											✱	✱	✱	✱									
CS02																							
CS03										✱													
CS04										✱													
CS05										✱													
CS06										✱													
VS0																✱							

1																					
VS02																					
AS01																					
AS02																					
AS03																					
AS04																					
AS05																					
AS06																					
AS07