**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Practical Medicine Cabinet Application**

|  |  |
| --- | --- |
| **Group 05** | |
| **Group members** | Trần Lê Nhân – SE61301 (Leader)  Nguyễn Khôi Nguyên – SE61025  Trần Ngọc Thuận – SE61329  Phan Quốc Quyền- SE60902 |
| **Supervisor** | Kiều Trọng Khánh |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | PMCA |

-Ho Chi Minh City, ***4th January 2016***-

*This page is intentionally left blank*

***ACKNOWLEDGEMENTS***

We wish to thank various people for their contribution to this project: Our teachers for their advice and participation in the final review, our friends for their valuable technical support.

Special thanks should be given to Mr. Kiều Trọng Khánh, our research supervisor for his professional guidance and the useful, constructive recommendations throughout the course of this project.

# Table of Contents

[Table of Contents 1](#_Toc448743188)

[List of Tables 3](#_Toc448743189)

[List of Figures 3](#_Toc448743190)

[A. Introduction 5](#_Toc448743191)

[1. Project Information 5](#_Toc448743192)

[2. Introduction 5](#_Toc448743193)

[3. Current Situation 5](#_Toc448743194)

[4. Problem Definition 6](#_Toc448743195)

[5. Proposed Solution 7](#_Toc448743196)

[5.1 Feature functions 7](#_Toc448743197)

[5.2 Advantages and disadvantages 8](#_Toc448743198)

[6. Functional Requirements 8](#_Toc448743199)

[7. Role and Responsibility 9](#_Toc448743200)

[B. Software Project Management Plan 10](#_Toc448743201)

[1. Problem Definition 10](#_Toc448743202)

[1.1 Name of this Capstone Project 10](#_Toc448743203)

[1.2 Problem Abstract 10](#_Toc448743204)

[1.3 Project Overview 10](#_Toc448743205)

[2. Project organization 13](#_Toc448743206)

[2.1 Software Process Model 13](#_Toc448743207)

[2.2 Roles and responsibilities 14](#_Toc448743208)

[2.3 Tools and Techniques 15](#_Toc448743209)

[3. Project Management Plan 16](#_Toc448743210)

[3.1 Software development life cycle 16](#_Toc448743211)

[3.2 Phases’ detail 17](#_Toc448743212)

[3.3 All Meeting Minutes 19](#_Toc448743213)

[4. Coding Convention 19](#_Toc448743214)

[C. Software Requirement Specification 21](#_Toc448743215)

[1. User Requirement Specification 21](#_Toc448743216)

[1.1 Guest Requirement 21](#_Toc448743217)

[1.2 User Requirement 21](#_Toc448743218)

[1.3 Staff Requirement 21](#_Toc448743219)

[1.4 Admin Requirement 21](#_Toc448743220)

[1.5 Scheduler Requirement 21](#_Toc448743221)

[2. System Requirement Specification 22](#_Toc448743222)

[2.1 External Interface Requirement 22](#_Toc448743223)

[2.2 System Overview Use Case 22](#_Toc448743224)

[3. Software System Attribute 24](#_Toc448743225)

[3.1 Usability 24](#_Toc448743226)

[3.2 Reliability 24](#_Toc448743227)

[3.3 Availability 24](#_Toc448743228)

[3.4 Security 24](#_Toc448743229)

[3.5 Maintainability 24](#_Toc448743230)

[3.6 Portability 24](#_Toc448743231)

[3.7 Performance 24](#_Toc448743232)

[4. Conceptual Diagram 25](#_Toc448743233)

[D. Software Design Description 26](#_Toc448743234)

[1. Design Overview 26](#_Toc448743235)

[2. System Architectural Design 26](#_Toc448743236)

[2.1 Web Application Architecture Description 26](#_Toc448743237)

[2.2 Mobile Application Architecture Description 27](#_Toc448743238)

[3. Component Diagram 28](#_Toc448743239)

[4. Detailed Description 28](#_Toc448743240)

[4.1 Class Diagram 28](#_Toc448743241)

[4.2 Interaction Diagram 31](#_Toc448743242)

[5. Database Design 37](#_Toc448743243)

[5.1 Entity relationship diagram (ERD) 37](#_Toc448743244)

[6. Algorithms 37](#_Toc448743245)

[6.1 Synchronize data. 37](#_Toc448743246)

[6.2 String Comparison 41](#_Toc448743247)

[6.3 Search medicine by illness mechanism 44](#_Toc448743248)

[6.4 Prevent time-dependent 46](#_Toc448743249)

[6.5 Parsing data from website 48](#_Toc448743250)

[E. Appendix 53](#_Toc448743251)

# List of Tables

[Table 1 : Definitions, Acronyms, and Abbreviations 4](#_Toc448743557)

[Table 2: Problem Definition 7](#_Toc448743558)

[Table 3 : Roles and Responsibilities 9](#_Toc448743559)

[Table 4: Hardware Requirement for Server 13](#_Toc448743560)

[Table 5: Hardware Requirement for Mobile 13](#_Toc448743561)

[Table 6: Software development tools 13](#_Toc448743562)

[Table 7: Roles and Responsibilities Details 15](#_Toc448743563)

[Table 8: Software Development Life Cycle Detail 16](#_Toc448743564)

[Table 9: Phase 1: Requirement Analysis and definition 17](#_Toc448743565)

[Table 10: Phase 2: System and software design 17](#_Toc448743566)

[Table 11: Phase 3: Implementation and unit testing 18](#_Toc448743567)

[Table 12: Integration and system testing 18](#_Toc448743568)

[Table 13: Phase 5: Operation and Maintenance 19](#_Toc448743569)

[Table 14: Conceptual Diagram Data Dictionary 25](#_Toc448743570)

[Table 15: Component dictionary 28](#_Toc448743571)

[Table 16: Class Diagram Description 31](#_Toc448743572)

# List of Figures

[Figure 1: Waterfall model 14](#_Toc448743602)

[Figure 2: System Overview Use Case 23](file:///E:\OneDrive\Capstone\Document\Document%20Final%20Version\PMCA_Group-5_Final%20-%2030.docx#_Toc448743603)

[Figure 3: System Architectural Design 26](#_Toc448743604)

[Figure 4: Mobile Application Architecture 27](file:///E:\OneDrive\Capstone\Document\Document%20Final%20Version\PMCA_Group-5_Final%20-%2030.docx#_Toc448743605)

[Figure 5: Component Diagram 28](file:///E:\OneDrive\Capstone\Document\Document%20Final%20Version\PMCA_Group-5_Final%20-%2030.docx#_Toc448743606)

[Figure 6: Class Diagram 29](file:///E:\OneDrive\Capstone\Document\Document%20Final%20Version\PMCA_Group-5_Final%20-%2030.docx#_Toc448743607)

[Figure 7: Sequence - <User> Add Medicine to Cabinet 32](#_Toc448743608)

[Figure 8: Sequence - <Staff> Approve Medicine 33](#_Toc448743609)

[Figure 9: <Scheduler> Parse Medicine 35](#_Toc448743610)

[Figure 10: Activity - <User> Synchronize data 36](#_Toc448743611)

[Figure 11: Entity relationship diagram (ERD) 37](file:///E:\OneDrive\Capstone\Document\Document%20Final%20Version\PMCA_Group-5_Final%20-%2030.docx#_Toc448743612)

[Figure 12: Synchronize data - Flow chart 40](#_Toc448743613)

[Figure 13: String Comparison- Flow chart Phase 1 42](file:///E:\OneDrive\Capstone\Document\Document%20Final%20Version\PMCA_Group-5_Final%20-%2030.docx#_Toc448743614)

[Figure 14: String Comparison- Flow chart Phase 2 43](#_Toc448743615)

[Figure 15: Search medicine by illness mechanism- Flow chart 45](#_Toc448743616)

[Figure 16: Prevent time-dependent- Flow chart \_ Sync with server time 47](#_Toc448743617)

[Figure 17: Prevent time-dependent- Flow chart \_ Auto update date in offline mode 47](file:///E:\OneDrive\Capstone\Document\Document%20Final%20Version\PMCA_Group-5_Final%20-%2030.docx#_Toc448743618)

[Figure 18: Prevent time-dependent- Flow chart \_ Parsing data from website 49](#_Toc448743619)

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Name** | **Definition** |
| PMCA | Practical Medicine Cabinet Application |
| NDC | National Drug Code |
| Pharmacology | Branch of medicine and biology concerned with the study of drug action. |
| Excipient | Natural or synthetic substance formulated alongside the active ingredient of a medication. |
| Contraindication | A condition or factor that serves as a reason to withhold a certain medical treatment due to the harm that it would cause the patient. |
| Uses (Indication) | A reason to use a certain treatment. |
| Interaction (Drug interaction) | Interactions between medications. |
| Side effect | An effect, whether therapeutic or adverse, that is secondary to the one intended. |
| Prescription | A plan of care written by a physician or other health care professional |

***Table 1 : Definitions, Acronyms, and Abbreviations***

# Introduction

## Project Information

* Project name: **Practical Medicine Cabinet Application**
* Project Code: **PMCA**
* Product Type: **Website & mobile application**
* Start Date: **4th January 2016**
* End Date: **26th April 2016**

## Introduction

Nowadays medicine cabinet has become an essential part in almost families. However, with the ordinary management, it is so hard to manage the expiration date of these medicines as well as the indication, dosage, etc.

Based on our researches and analysis, along with the development and popularity of smart phones, we have developed a system to help users manage their medicine cabinets effectively. In addition, our system supports users to search for medicine information using name or use of the medicine. The database is updated regularly to assure that the search results are accuracy. The system also reminds user to take their medication and manage their prescription.

This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows. We hope the system and our solution will help resolve the problems from the traditional use of medicine cabinet in family.

## Current Situation

Medicine cabinets are always existed in our family to storage medicines that we have bought.

To manage it, user usually use one of these ways:

**Manually manage:**

After bought medicines, users put them into their medicine cabinet and then just find them when they need.

However medicine’s expired date is one of important thing that we usually forget. Expired medicines must be throw away and replace by the new ones. Besides, when buying medicines, users usually do not care enough for the instructions, indications or bought a small part of the whole medicine packet without the instruction, indications. Moreover, they just put medicines in the medicine cabinet and do not administrate its.

**Using websites, apps to help:**

In Viet Nam, there are some apps, websites help user to find the medicine‘s indication, dosage by medicine’s name or National Drug Code (NDC) like:

Websites:

* <http://vinapharm.com.vn/index.php>
* <http://thuoc.net.vn/>, [www.thuoc.vn](http://www.thuoc.vn/)
* [http://www.thuocbietduoc.com.vn](http://www.thuocbietduoc.com.vn/)

Apps:

* Ứng dụng Bệnh Và Thuốc - [http://www.bhmedia.vn/…/ung…/193-ung-dung-benh-va-thuoc.html](http://www.bhmedia.vn/ung-dung-mobiles/kho-ung-dung/ung-dung-doi-song/193-ung-dung-benh-va-thuoc.html)
* Tủ Thuốc - [https://www.microsoft.com/…/store/apps/tu-thuoc/9nblggh3z8q1](https://www.microsoft.com/en-us/store/apps/tu-thuoc/9nblggh3z8q1)
* Thuoc tay, tu thuoc - [https://play.google.com/store/apps/details…](https://play.google.com/store/apps/details?id=com.pharmacy.knowledge)
* Từ điển thuốc - [https://play.google.com/store/apps/details…](https://play.google.com/store/apps/details?id=com.sau.dictionary.drug)

But none of these websites, apps help user to manage their medicine cabinet.

Over the world we can found some apps can help users manage medicine cabinet like:

* DrugHub- [https://itunes.apple.com/…/drughub-medicine-ca…/id421028414…](https://itunes.apple.com/ca/app/drughub-medicine-cabinet-in/id421028414?mt=8)

However, it does not suitable for Vietnamese users and also does not suggest medicine for user by theirs treatment or illness.

Besides that, if the medicine which users input not exists in database, apps or websites not let users to find it from other reputation websites about medicine.

## Problem Definition

With the current situation, we can see some disadvantages that exists with manually manage or using website, apps to help manage:

|  |  |  |  |
| --- | --- | --- | --- |
| Problem | Description | Manually manage | Using websites, apps to help |
| Lack of administration medicine | Users usually buy medicines and put it to the medicine cabinet and do not care any more about them. | They do not know what and how many medicines they have, number of each medicine and the expired date. | Available apps, websites not support users manage their medicines. |
| Manage medicine’s expired date | Users are put medicines in the medicine cabinet and forget to check their expired date so they usually let them out of date. | They do not have medicine when they need or event use the expired medicine that can be harmful for their health. | Available apps, websites not have the function to manage medicine’s expired date and notify user when medicine expired. |
| Not suggest medicine for user by their treatment or illness: | When user have some illness, they want to get some suggest for medicine to help them. | Only based on user’s experiences to get suggest medicine.  Or go to doctor to get help. | Available apps, websites only can search medicine by name or code but cannot have suggestion medicine with treatment or illness |
| Search medicine that not exist in database | Medicines that user input are to new or not exist in the current apps, websites database. | User only know what they have in their cabinet. | Available apps, websites can only search medicine in their database; cannot search from other source and add to apps, website database. |

***Table 2: Problem Definition***

## Proposed Solution

From the above analysis, we recommend solutions combines and supplements the existing shortcomings of the current system. Our proposed solution is to develop the Practical Medicine Cabinet Application named “PMCA system” to support users manage the medicine cabinet such as the expired date of these drugs as well as the indication, dosage, etc. Besides, the application also supports users to look up drug information and helpful hints for drug users.

### Feature functions

Mobile application: developing functions on mobile supports user manage their medicine cabinet and prescription. Mobile function also supports user search for medicine information and remind them to take medication:

* Manage medicine cabinet: users input name of medicine or scan the barcode on the medicine package with the expired date, the system will match the user’s medicine with the system’s medicine dictionary. Thereby supporting the management of their medicine cabinet and storage information of each medicine without the worry of loss
* Notify functions: A background scheduler will notify users when their medicines are expiring and remove the expired ones.
* Search medicine: allow the users search for medicine information by name or the illness. Provides suggest mechanism even when user type a miss-spelled medicine name, search result is sorted by the most convenience for user.
* Suggest post: every day system will suggest healthy post for user, the post which related with user’s medicine cabinet will be shown first.
* Manage user’s prescription: allows user to input their prescription and active reminder for each prescription. When activated, the reminder will notify user to take medicine along with the information about dosage.
* Offline mode: application allow users to manage their cabinet in offline mode. When running in this mode, the application still ensure the correctness of other functions related to time such as tracking user’s cabinet.
* Synchronize data: the mechanism for data synchronization helps the data added in the offline mode does not lose or resolve conflict data incurred by the interaction of user to their cabinet from web app.

Web application: design for staff, admin to manage the system and also allow user to interact with their cabinet.

* Helps staff manage the medicine dictionary, posts, keywords, barcodes, parser configuration, configure scheduler.
* Background processes:
  + Run periodically to parse data from other websites to database.
  + Update the frequency of medicine and the n-gram index.
  + Tracking user’s cabinet, notify user when their medicine is expiring.

### Advantages and disadvantages

* Advantages:
* Easy to use: system and user communicate via web or smart phone app which makes imported drugs (via scan barcode), notify user, etc. easier.
* The medicine database completed, accurate and updated regularly: medicine information is taken from reputable sites, can be update regularly by staff of system scheduler.
* Solve the needs to learn in detail how to use the medicine as well as its information of users: system provides search medicines feature which helps users have a detailed information.
* Disadvantages:
* Require users to interact and fully update in order to help system works correctly and efficiently.

## Functional Requirements

Function requirements of the system are listed as below:

* User component:
* Manage medicine cabinet: input medicines and their expired date, check quantity and expired date of medicines in the medicine cabinet.
* Search medicine: users can search and map the indication, dosage and administration.
* Manage user’s prescription: user can create prescription and create a reminder for it to take medication.
* View post have related to medicine had in the cabinet.
* Staff component:
* Update medicine database: The staff choose the source sites, then system will parse and update data via these sites.
* Configure the system: configure the system’s settings to update data from other sites and schedule the automatic update.
* System component:
* Parse: The system will get data source from reputable sites.
* Notify: The system will show the notifications when the medicines expired date are coming.
* Suggests the medicine: The application suggests the medicine with the indication, dosage, etc. when the users search the treatment or illness by the application.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | khanhkt@fpt.edu.vn |
| 2 | Trần Lê Nhân | Developer | Leader | nhantlse61301@fpt.edu.vn |
| 3 | Trần Ngọc Thuận | Developer | Member | thuantnse61329@fpt.edu.vn |
| 4 | Nguyễn Khôi Nguyên | Developer | Member | nguyennkse61025@fpt.edu.vn |
| 5 | Phan Quốc Quyền | Developer | Member | quyenpqse60902@fpt.edu.vn |

***Table 3 : Roles and Responsibilities***

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name:** Practical Medicine Cabinet Application
* **Vietnamese name:** Tủ thuốc tiện dụng
* **Abbreviation:** PMCA

### Problem Abstract

Today in every household medicine cabinet is becoming more popular as well as smartphones. Hence, we provide a mobile application that give users a helpful method to manage medicine in their cabinet, as well as those medicine’s expired date. The application also aids users by reminding them about incoming expired date in case they forget. In addition, we develop a web site as an alternative using way for the application in some situations. Moreover, the application can suggest medicine based on the illness and remind them to take medication based on their prescription. Manage prescription when they have demand.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* + **Lack of knowledge about medicine:** Our project need the team member to know more about medicine knowledge which none of us have it.
  + **Parsing mechanism problem:** Our project can help user get information of medicine that not exists in our database by parsing from other website but with other website it have difference structure so we may cannot get correct information.
  + **Cooperation of user:** Our software only work efficiently based on user’s cooperation in input medicine in their medicine cabinet. Through a way when medicine expired and refill when medicine run out if need when software notify its.
  + **New with develop mobile platform:** Our team member are all new with develop mobile application so this part may cost most of time of project.
  + **Lack of UI, UX (user experience) designer:** Our team member all study Information System and no one have study UI, UX design so that some UI may misunderstand or hard to use with normal user.

#### The Proposed System

* + By the study of the behavior of users, we have designed so that users can interact with the system a simple way, while reminding them to update their status of the medicine cabinet so the system can provide best support through guidance and tutorials.
  + To solve the problem of medicine hint feature, in any suggestions, we warn the user that these hints are approximate, the medication and treatment must follow the guidance of doctors.

##### Web Site

Web Site is an essential part of the system. The user’s site provides some functions like the mobile application, while the staff’s site performs gathering data task.

* For users:
* Manage medicines in the medicine cabinet.
* Search medicine for detail information like indication, dosage and administration.
* View post that system suggests have related to medicine in their cabinet
* Manage prescription in the medicine cabinet.
* Remind user to take their medication.
* For staffs:
* Configure settings to update data (news and medicines) from other sites and schedule the automatic update.
* Update data manually.
* Manage post.
* Manage keyword.
* Manage medicine dictionary.
* Parse data manual.
* Approve data.
* For Scheduler:
  + - * Auto parse data
      * Track user’s medicines
* Notify user
* Admin
  + - * Manage account

##### Mobile Application

Mobile application is the mainly part used by users. It has the following core functions:

* Manage medicine and their expired date.
* Scan barcode with camera for quick adding new medicine.
* Notify about expiring medicine.
* Suggests the medicine when users search the treatment or illness by the application.
* Search medicine for detail information like indication, dosage and administration.
* Remind user to take medication based on their prescription.
* Suggest post have related to medicine in their cabinet.

#### Boundaries of the system

The system can:

* Allow the users input the name of medicine with the expired date that are put in their medicine cabinet, or they can scan the barcode on packet
* Search and map the indication, dosage and administration, ...
* Notify the users the medicines that are expiring.
* Allow users to search the treatment or illness on application, the application suggests the medicine in their cabinet with the indication, dosage...
* Parse reputable websites and update the database automatically
* System suggests post have related to medicine in the cabinet.
* System reminds user to take medication based on their prescription.

Besides, the system cannot:

* Replace doctors in the treatment of users: all suggestions are approximate, users should consider carefully before applying these suggestions and take full responsibility when problems occur.

#### Future plans

Current systems support only a tool for users to use their medicine cabinet more effective as well as suggestions only drugs for common diseases. With further research and development, the system can apply these advanced features:

* Upgrade suggests medicine functionality, instead of the simple suggestion (ordinary 1 medicine for 1 disease), the system will return the advanced suggestions (combination multiple medicine) that treatment brings better results.
* The system will store the medication history of user, there by the system will show the useful article that related to their problem daily, enabling users to improve their health.

#### Development Environment

##### Hardware requirements

* **For server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable(4 Mbps) | Cable(8 Mbps) |
| Operating System | Window Server 2008 | Window Server 2012 |
| Computer Processor | Intel® Core i3 1.4GHz | Intel® Core i5 2.50 GHz |

***Table 4: Hardware Requirement for Server***

* **For Mobile**

|  |  |  |
| --- | --- | --- |
| **Windows** | **Minimum** | **Recommended** |
| Internet Connection | Wi-Fi or 3G (4 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.2.2 (Jelly Bean) | Android 5.0 (Lollipop) |
| Computer Processor | Cortex-A15 Quad-core 1.6 GHz | Cortex-A15 Quad-core 1.9 GHz |
| Computer Memory | 1GB RAM | 1GB or more |

***Table 5: Hardware Requirement for Mobile***

##### Software development tools:

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Window 7, Window 8 | Operating system and platform for development |
| Environment | .NET Framework 4.5 | Specification for developing web application |
| Modeling tool | Microsoft Visio 2013 | Used to implement website and web service |
| IDE | Microsoft Visual Studio 2013 | Used for implement website and Android Mobile App. |
| Design Model tool | StartUML v5.0 | Used for creating modal and diagrams. |
| DBMS | SQL server 2008 R2 | Used to create & manage the database for system |
| Document storage | Google Drive | Used for storing document |
| Store and manage source code | TFS Server | Used to store all source code |

***Table 6: Software development tools***

## Project organization

### Software Process Model

Project is developed under waterfall model.



***Figure 1: Waterfall model***

***Reference:*** *Page 30, chapter 2, Software process model, SOFTWARE ENGINEERING 9th Edition,by Ian Sommerville.*

This project is developed under waterfall model. We apply customized waterfall model to capable with current situation in our team. We choose this model because the following reasons:

* This is a project with clear requirement.
* Based on researches and clarify standard regimen of diseases are stable, clear, fixed and well understood by all team members.
* We can mix requirement Analysis, Design, Coding and Testing in order to fit with our team’s working process.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Kiều Trọng Khánh | Superviser | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| 2 | Trần Lê Nhân | Team leader, BA, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Trần Ngọc Thuận | Team member,  BA, DEV,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 4 | Nguyễn Khôi Nguyên | Team member,  BA, DEV,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 5 | Phan Quốc Quyền | Team member,  BA, DEV,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |

***Table 7: Roles and Responsibilities Details***

### Tools and Techniques

|  |  |  |
| --- | --- | --- |
|  | **Tools** | **Techniques** |
| **Front-end** | Visual studio | - HTML5  - CSS  - JavaScript |
| **Back-end** | Visual studio | - Website: ASP.NET MVC 4  - Website: Entity Framework 6  - Web Service: ASP.Net API 2. |
| **Web server** | Microsoft IIS Version 8.5 | N/A |
| **Mobile application** | Android studio 2.0 | N/A |
| **Database management system** | - MS SQL Server 2008 R2 Express.  - SQL lite. | N/A |

## Project Management Plan

### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Description | Deliverables | Resource needed | Dependencies and Constrains | Risks |
| Requirements Definition | - Collect and analyze requirements | - SRS | 7 man-days | - N/A | - Missing  requirement  - Unclear scope of project  - Lack of member share of understand |
| System and Software Design | - Design structure of the system | - Software architecture | 14 man-days | - Depend on requirement after analyzed and system architecture design. | - Lack of experience.  - Not fulfil requirement |
| Implementation and Unit Testing | - Implement the system | - Runnable system | 60 man-days | - Depend on all previous phrases: requirement and system design |  |
| Integration and System Testing | - The systematic discovery and debugging of defects | - Test case | 14 man-days | - Depend on all previous phrases: requirement ,system design and implementation |  |
| Operation and Maintenance | - Deploy, maintain and expand the system | - The installation, migration, support, and maintenance of complete systems | 20 man-days | Depend on all previous phrases: requirement ,system design, implementation and testing |  |

***Table 8: Software Development Life Cycle Detail***

### Phases’ detail

#### Phase 1: Requirements Definition.

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1.Collect requirements | Find which systems currently provide similar service, their strength and weakness. | NhanTL, NguyenNK, ThuanTN, QuyenPQ |
| 2.Identify and clarify main functions | Define which main function system should provide. | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |
| 3.Requirements specification | Defining the requirements in detail. | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |
| 4.Requirements validation | Checking the validity of the requirements | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |

***Table 9: Phase 1: Requirement Analysis and definition***

#### Phase 2: System and Software Design

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Architectural design | Identify the overall structure of the system, the principal, their relationships and how they are distributed. | NhanTL, NguyenNK |
| 2. Interface design | Define the interfaces between system components. | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |
| 3. Component design | Design how each system component will operate. | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |
| 4. Database design | Design the system data structures and how these are to be represented in a database. | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |

***Table 10: Phase 2: System and software design***

#### Phase 3: Implementation and Unit Testing

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Implement web and mobile front-end | Create GUI for web and mobile application | NhanTL, NguyenNK |
| 2. Implement parser | Implement parser to gather data from other sites base on Staff configuration. | NguyenNK |
| 3. Implement Schedule and system notification | * For Staff to schedule automatic update database * System notify user when medicine will expired | NhanTL |
| 4. Implement matching | Matching data when user input medicine or scan with barcode. | ThuanTN |
| 5. Implement fulltext search | Apply full-text search when user search medicine or looking for suggest medicine for disease. | QuyenPQ |
| 6. Unit test | Developer write and run to ensure that code meets its design and behaves as intended. | NhanTL, NguyenNK, ThuanTN, QuyenPQ |

***Table 11: Phase 3: Implementation and unit testing***

#### Phase 4: Integration and System Testing

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Component testing | Individual components are tested independently. | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |
| 2. System testing | Testing of the system as a whole. Testing of emergent properties is particularly important | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |
| 3. Acceptance testing | Testing with customer data to check that the system meets the customer’s needs. | NhanTL, NguyenNK,  ThuanTN, QuyenPQ |

***Table 12: Integration and system testing***

#### Phase 5: Operation and Maintenance

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Maintain and expanding system | Maintain current system  Update new features | N/A |

***Table 13: Phase 5: Operation and Maintenance***

### All Meeting Minutes

Refer to “Meeting Minutes” folder:

<https://drive.google.com/open?id=0B01CoMZGnpUnT3k4a3R0OXBfNlE>

## Coding Convention

C#: Using to develop website and web service.

Summary:

* Naming Convention:
  + For variable’s name, use camel case. Eg: minValue, maxValue…
  + For function name, class name, use Pascal case. Eg: AddIncome, AddExpense… Reasonable Spending Suggestion 10
* Layout Convention:
  + Write only one statement/declaration per line.
  + Indent continuation one tab stop (four spaces).
  + Add at least one blank line between method definitions and property definitions.
  + Use parentheses to make clauses in an expression apparent.
* Commenting Convention:
  + - Place the comment on a separate line, not at the end of a line of code.
    - Begin comment text with an uppercase letter.
    - End comment text with a period.
    - Insert one space between the comment delimiter (//) and the comment text.
    - Do not create formatted blocks of asterisks around comments.
* Language guidelines: Using C# coding convention from <http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>

Android: Using to develop mobile application.

* Naming Convention:
  + - Follows camelcase syntax for naming the class, interface, method and variable.
    - If name is combined with two words, second word will start with uppercase letter always e.g. actionPerformed(), firstName, ActionEvent, ActionListener etc.
  + Follow Field Naming Conventions
    - Non-public, non-static field names start with m.
    - Static field names start with s.
    - Other fields start with a lower case letter.
    - Public static final fields (constants) are ALL\_CAPS\_WITH\_UNDERSCORES.
* Language guidelines: Using Android Code Convention from <http://source.android.com/source/code-style.html>

# Software Requirement Specification

## User Requirement Specification

### Guest Requirement

Guest is a person who does not have access the system. Guest can use some functions in the system. To use all functions, guest must login. These are some functions guest can use:

* Login
* Register

### User Requirement

User is guest who already login to the system by his/her account and uses service of system. The customer can use some following functions:

* Search Medicine
* Get Suggested Medicines
* Manage Medicine Cabinet
* View post
* Create prescription and remind for it

### Staff Requirement

Staff is the person who works directly with system to manage medicine, system and parser. Staff can use some following functions:

* Configure system
* Parse data manually
* Manage medicine dictionary
* Matching data

### Admin Requirement

Admin is a person with permission to manage some aspect of the system. Admin can use some following functions:

* Manage account

### Scheduler Requirement

Scheduler is a part of PMCA allows system to do some functions automatically. These automatic functions are:

* Enable the parser
* Notify the users

## System Requirement Specification

### External Interface Requirement

#### User Interface

* General requirement for graphics user interface is the GUI should be simple, clear, intuitive, and reminiscent.
* The user interface uses Vietnamese language.
* The user interface uses consistent palette of colors between the text and the background.
* The user interface displays best on 1024x768-screen size.

#### Hardware Interface

* Android Smartphone:
  + RAM: 1GB
  + CPU: 1.2GHz

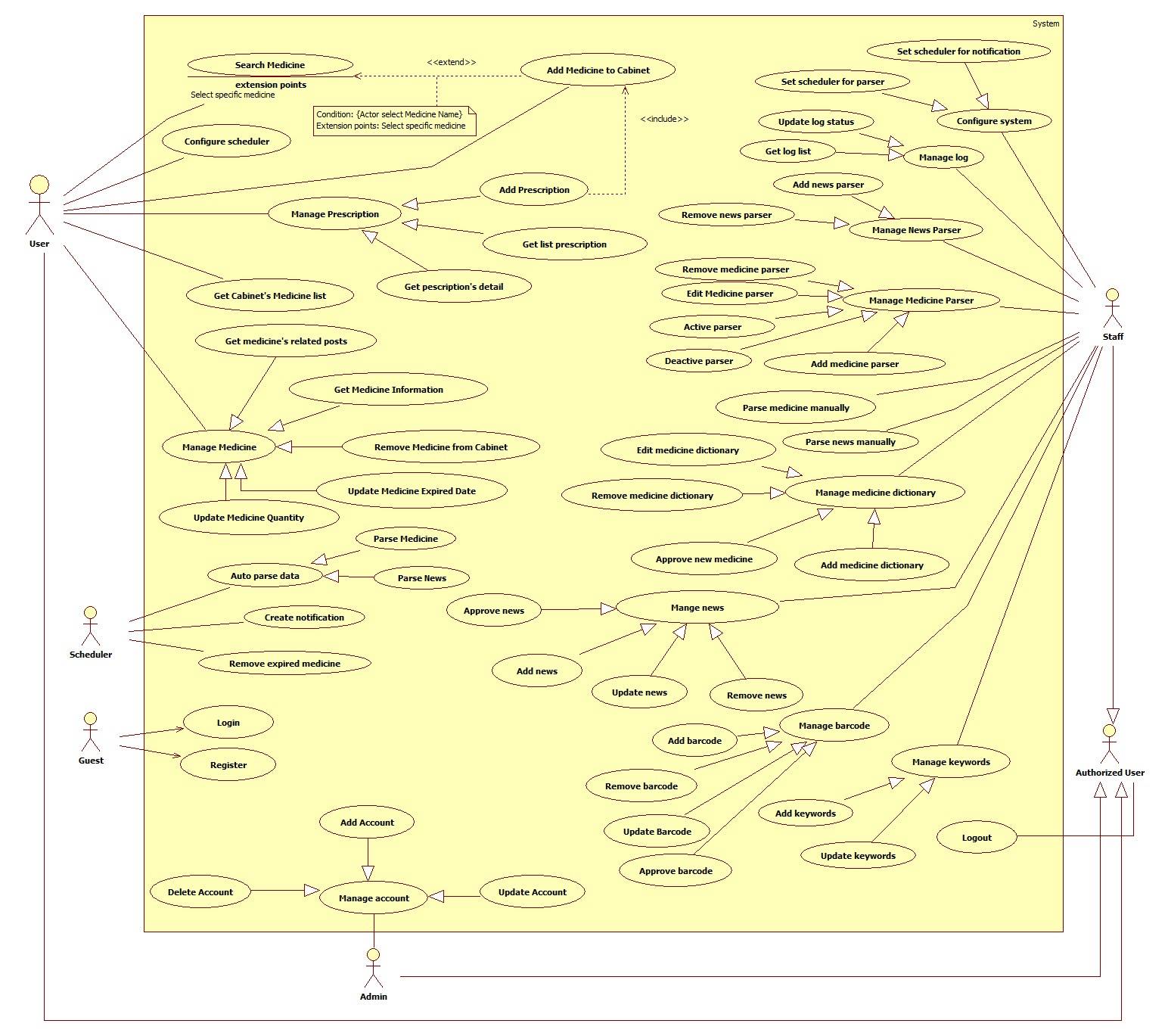
#### Software Interface

* Web application: work with Firefox (v30 or above), Chromes (v14 or above), Internet Explorer (v10 or above) browse.
* Mobile application: Android operating system (v 4.0 or above).

#### Communication Protocol

* Use HTTP protocol 1.1 for communication between the web browser and the web server.

### System Overview Use Case



**Figure 2: System Overview Use Case**

## Software System Attribute

### Usability

#### Graphic User Interface

* + - All the texts, labels and notifications must be written in Vietnamese.

#### Usability

* + - Admin, staff should need less than 2 days of training to be productive with the system.
    - The system is easy to use for user without training.

### Reliability

* + - Application will help user can manage medicine record and prescription.
    - Application will notify user about expired day of medicine.
    - All medicine in system must be approved by staff.
    - All website that is parsed must have high precision about information of medicine, high reliability.

### Availability

* + - Server have back-up method to make sure that if it have problem while running then all necessary data must be protected and restore as soon as.
    - Mobile application have local storage to store data in case of no Internet connection

### Security

* Privacy: Each role of user has a specific permission to interact with system. System always checks the authorization and authentication before doing anything.

### Maintainability

* The system is divided into separated modules.

### Portability

* The system can be deployed into many type of servers those have IIS8.
* Admin, staff, customers and guests can use application on Google Chrome version 42 or above.

### Performance

* Requests from web application are responded in less than 5 seconds at 4 Mbps bandwidth speed and 1GHz Processing Speed of CPU.

## Conceptual Diagram



|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| **Entity Name** | **Description** |
| User | Abstract entity describes a user in system |
| Notification | Contain notification information |
| Medicine cabinet | Contain medicine cabinet information. |
| Medicine info | Contain medicine information |
| News | Contain news information |
| News Parser | Contain Medicine News configure. |
| Medicine Parser | Contain Medicine Parser configure. |
| Excipient | Contain Excipient information. |
| Pharmacology | Contain Pharmacology information. |
| Keywords | Contain Keywords information. |
| Staff | Role Staff of system. |
| Admin | Role Admin of system. |
| Barcode | Contain Medicine’s barcode information. |
| Prescription | Contain prescription information. |
| MedicineCabinetInfo | It is sub class that is created by medicine cabinet and medicine info. It contain information of medicine in cabinet. |

***Table 14: Conceptual Diagram Data Dictionary***

# Software Design Description

## Design Overview

* This document describes the technical and user interface design of PMCA System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
  + Section 2: gives an overall description of the system architecture design.
  + Section 3: gives component diagrams that describe the connection and integration of the system.
  + Section 4: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
  + Section 5: describe a fully attributed Entity Relationship Diagram.

## System Architectural Design

D:\Downloads Software\HinhA3\Architec.png

***Figure 3: System Architectural Design***

### Web Application Architecture Description

In Web Application, the system is developed under MVC architecture style. We choose this architecture for Web application because of following advantages:

* Web application contains Web services (Public API for mobile app) with MVC architecture, we can separate business code with Controller and View. So we can use the business code in web service without repeat the code.
* We can organize the code better for maintainability, extensibility, reusability so we can expand the scope for more function.
* MVC architecture make it easier to split the big project into small modules and make it easier to assign each module for members in our team.

### Mobile Application Architecture Description

The application is developed as an Android native application. In general, the application architecture conforms to Android architecture.

C:\Users\Admin\Desktop\android.png

***Figure 4: Mobile Application Architecture***

**Reference**: Android Developer Guider – Application Fundamentals

<http://developer.android.com/guide/components/fundamentals.html>

In our Android application, the application is developed under MVC architecture style. We choose this architecture for Android application because of following advantages:

* + - We can organize the code better for maintainability, extensibility and reusability.

## Component Diagram

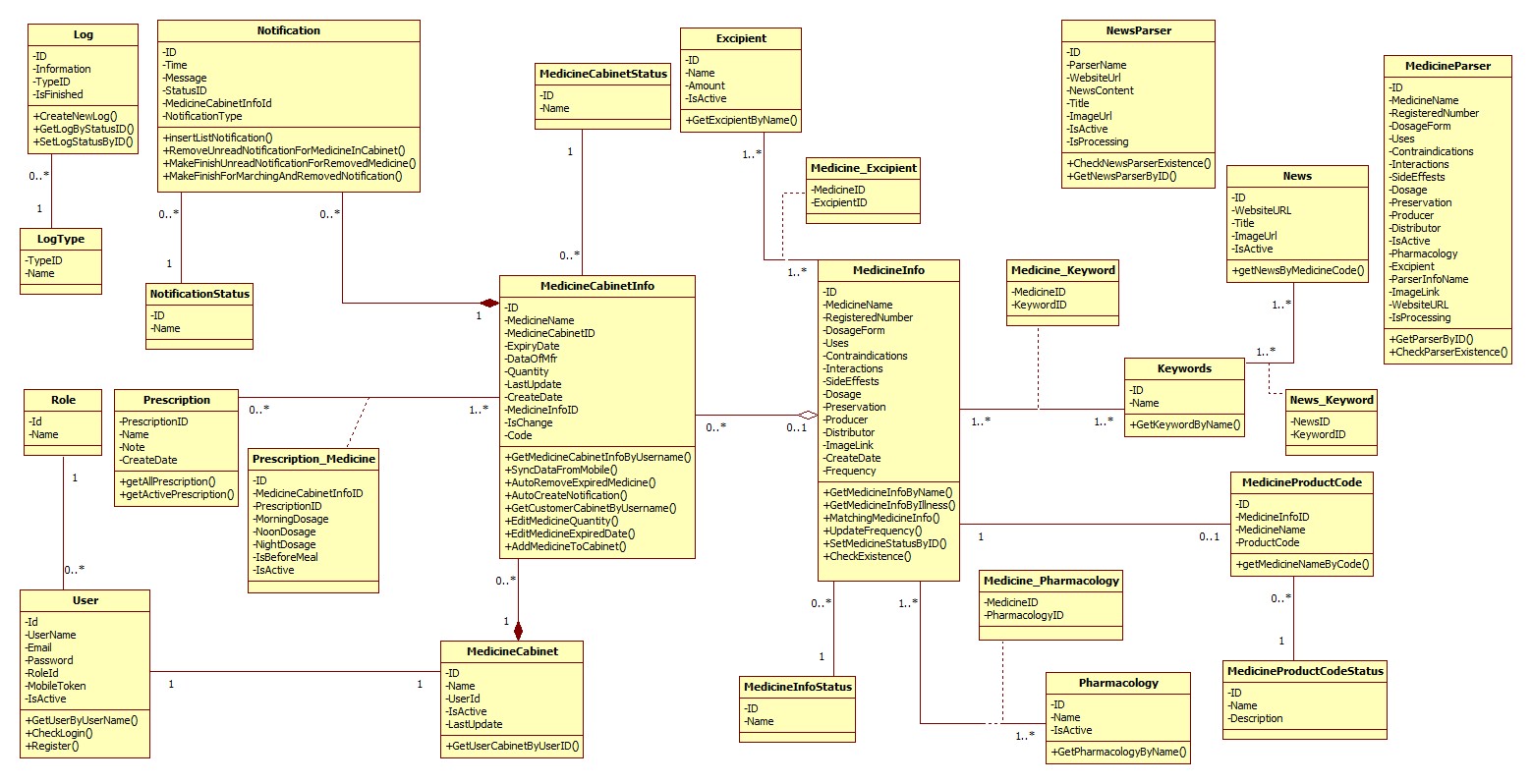
***Figure 5: Component Diagram***

|  |  |
| --- | --- |
| Component dictionary: describe component | |
| Component Name | **Description** |
| Web Application | Web application package: View, Controller |
| Web Services | Provide API for mobile applications to interact with the system. |
| Parser Component | Component to handle parser in the system. |
| Automation Component | Component to handle notification and scheduler in the system. |
| User Component | Component to handle user activities in the system. |
| Staff Component | Component to handle user activities in the system. |
| Admin Component | Component to handle admin activities in the system. |
| Business Service | Common object to handle domain business operations for each components |
| Google cloud Services | Handle push notification to mobile. |

***Table 15: Component dictionary***

## Detailed Description

### Class Diagram



**Figure 6: Class Diagram**

|  |  |  |
| --- | --- | --- |
| Class dictionary: Describe Class | | |
| Class Name | **Mapping column with Conceptual diagram** | **Description** |
| Log | Log | Contain the log information |
| Log Type | N/A | Not exist in conceptual diagram. But needed in class diagram to contain the log type information |
| Role | Role | Contain the role information |
| User | User | Contain the user information |
| Notification | Notification | Contain the notification information |
| Notification status | N/A | Not exist in conceptual diagram. But needed in class diagram to contain the status of notification. |
| Medicine Cabinet Info | Medicine Cabinet Info | Contain the information of cabinet |
| Medicine Cabinet | Medicine Cabinet | Contain the medicine cabinet information |
| Excipient | Excipient | Contain the excipient information |
| Medicine Info | Medicine Info | Contain the medicine information |
| Medicine Status | N/A | Not exist in conceptual diagram. But needed in class diagram to contain the status of medicine |
| Medicine\_Excipient | N/A | Not exist in conceptual diagram. But needed in class diagram to contain the excipient of medicine |
| Medicine\_Pharmacology | N/A | Not exist in conceptual diagram. But needed in class diagram to contain the pharmacology of medicine |
| Pharmacology | Pharmacology | Contain the pharmacology information |
| Medicine parser | Medicine parser | Contain the medicine configure information. |
| News parser | News parser | Contain the news configure information |
| MedicineProductCode | Barcode | Contain the barcode of medicine information |
| MedicineProductCodestatus | N/A | Not exist in conceptual diagram. But needed in class diagram to contain the status of barcode. |
| Keywords | Keywords | Contain the keyword information |
| Medicine\_Keyword | N/A | Contain the keyword of medicine information |
| News | News | Contain the news information |
| News\_Keyword | N/A | Contain the keyword of news information |
| Prescription | Prescription | Contain the prescription information |
| Prescription\_Medicine | N/A | Not exist in conceptual diagram. But needed in class diagram to contain the medicine on prescription. |

**Table 16: Class Diagram Description**

### Interaction Diagram

#### Website

##### User

###### <User> Add Medicine to Cabinet

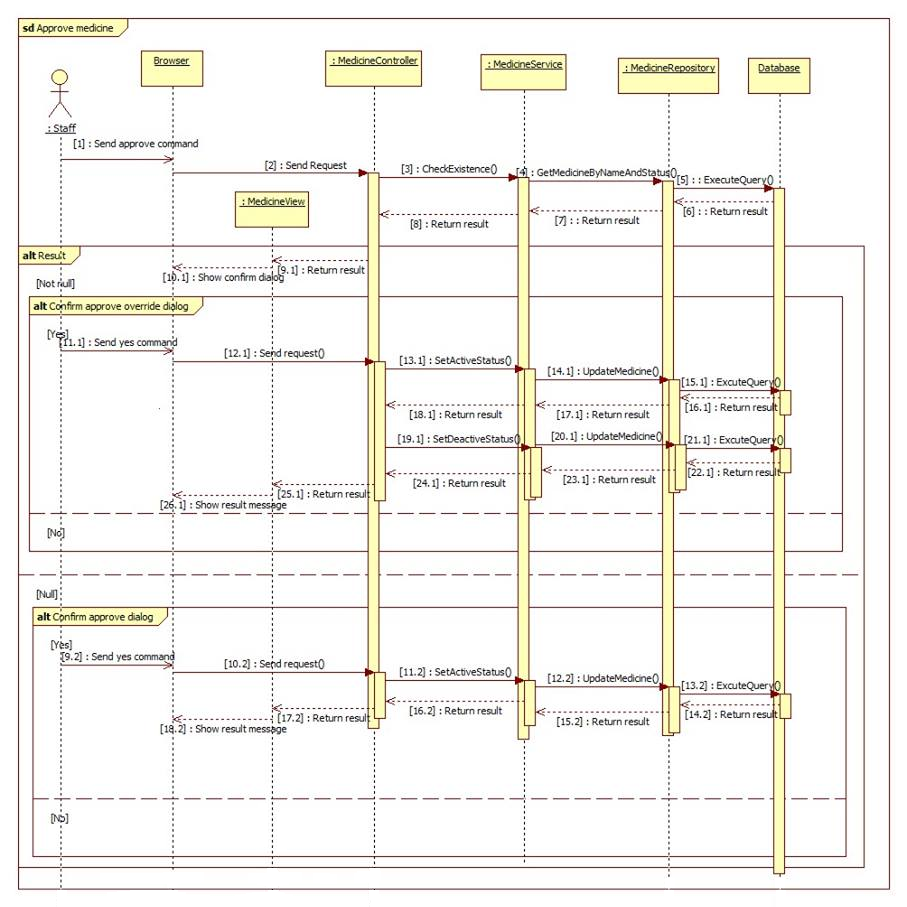
Summary: This diagram show process of how user add medicine to their cabinet in system

**Figure 7: Sequence - <User>** **Add Medicine to Cabinet**

##### Staff

###### <Staff> Approve Medicine

Summary: This diagram show process of how Staff Approve medicine.

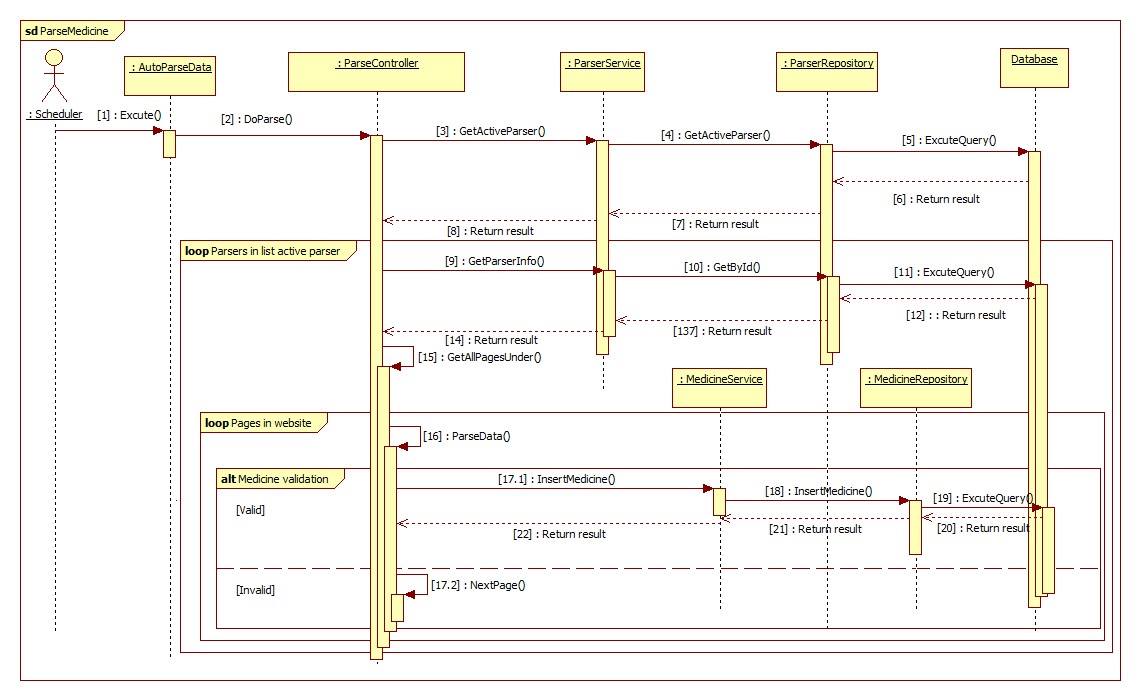


**Figure 8: Sequence - <Staff>** **Approve Medicine**

##### Scheduler

###### < Scheduler > Parse Medicine

Summary: This diagram show process of how Scheduler parse medicine information to database.



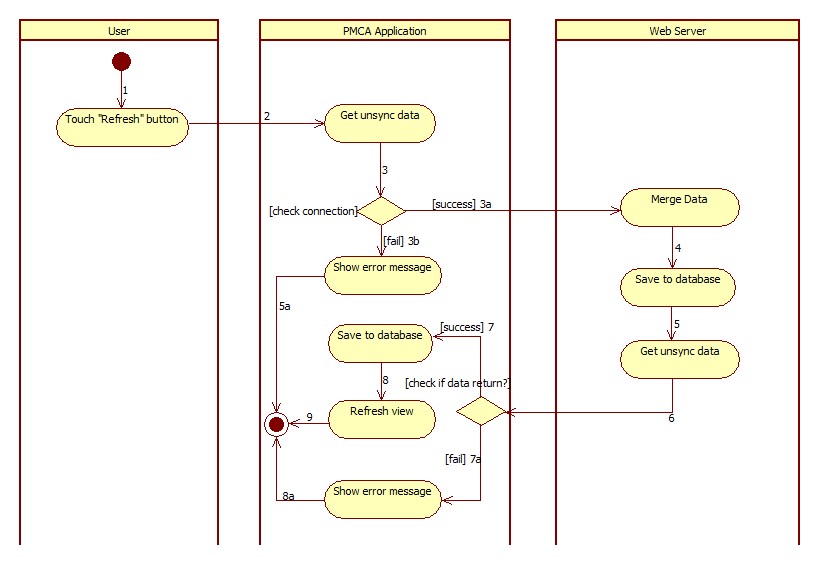
**Figure 9: <Scheduler> Parse Medicine**

#### Mobile Application

##### User

###### <User> Synchronize data

Summary: This diagram show how system synchronize data from mobile application to server and vice versa.



**Figure 10: Activity - <User> Synchronize data**

## Database Design

### Entity relationship diagram (ERD)

**Figure 11: Entity relationship diagram (ERD)**

## Algorithms

### Synchronize data.

#### Definition

* This algorithm proposes method to synchronize data from mobile to server and vice versa.

#### Define Problem

* + - Each user can interact with their medical cabinet via mobile app or website, the mobile app can run in offline mode and the system can also update user medicine’s information. Hence, the problem is how to synchronize data between mobile app and the web server without lost data.

#### Solution

* + - To solve this problem, we create a merging method:
    - Each medicine has a flag variable named “IsChange” that turn on whenever the medicine changes, turn off when the merging method finish.
    - Each medicine has a variable named “LastUpdate” that records the medicine’s last changing date.
* Step 1: [Application] Get all medicine has IsChange = true as a request data to server.
  + - * Step 2: [Server]
* Input: List of changed medicines from mobile.
* For each medicine from input data, the merging method run with the following steps:
* Step 2.1: Check if the medicine exists in the server cabinet

False: Add a new medicine to the cabinet (Step 2.2a)

True: Merge the changed medicines (Step 2.2b).

* Step 2.2a: Add the medicine to the server cabinet

Check if the system contains the medicine’s detail:

True: insert to server database with IsChange = true

False: insert to server database with IsChange = false.

* Step 2.2b: Merge the changed medicine by its last updating time:
  + Input: the mobile’s medicine and the corresponding one in server.

Compare the last changing date of the 2 medicines:

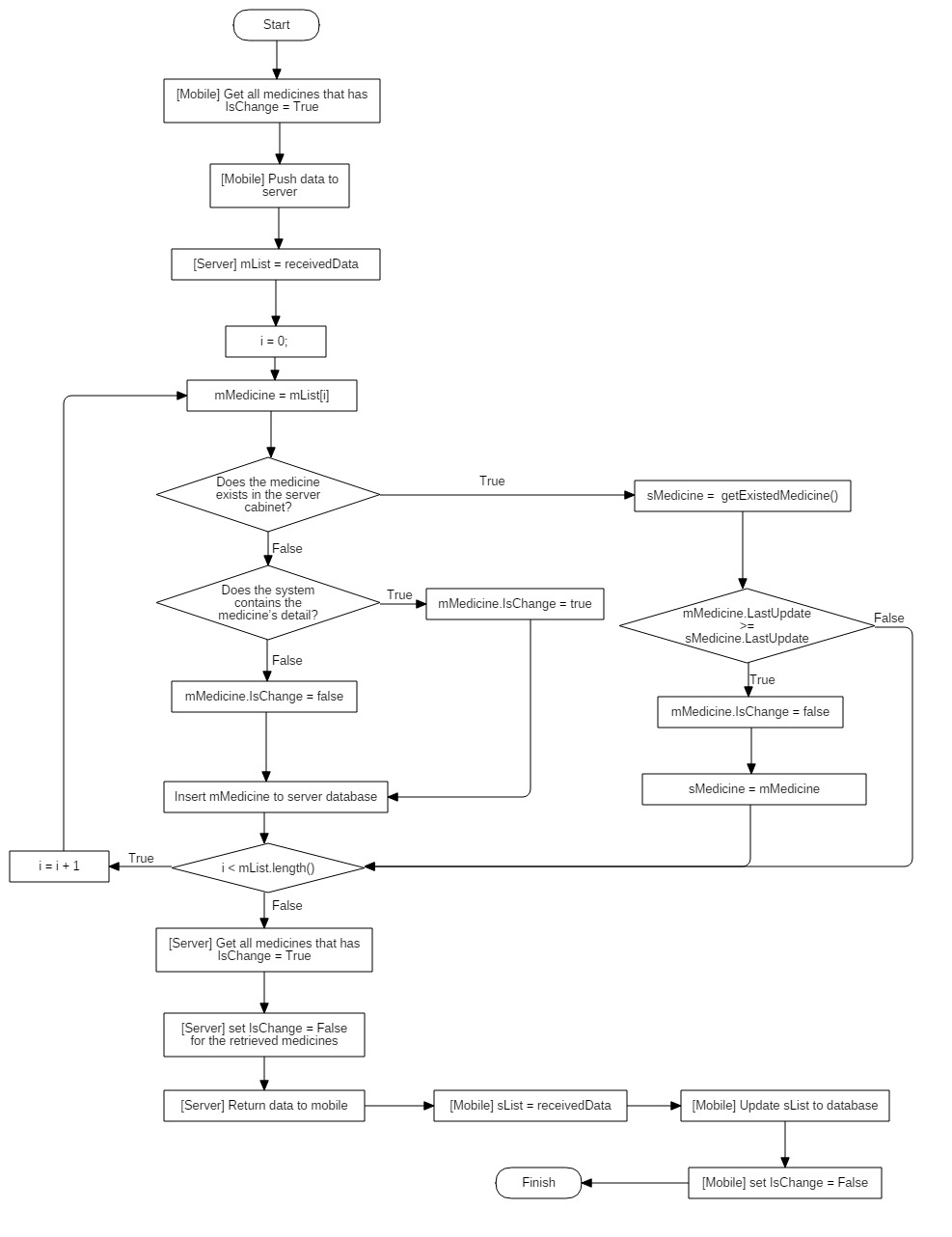
* + If the last updating time of the mobile’s medicine is later than or equal to the server’s one: override the mobile’s medicine to server with IsChange = false.
  + Else: Do nothing
    - * Step 3: [Server] Get all the medicines that have IsChange = true as response data and return to application, then update their status “IsChange” = false to the server database.
    - Output: List of changed medicines from server.
      * Step 4: [Application]

Check if the medicine exists in the mobile’s cabinet

* False: Add to cabinet.
* True: Override to the corresponding one in the mobile’s cabinet.

Turn off all the mobile’s medicines “IsChange” flag.

#### Flow Chart



**Figure 12: Synchronize data - Flow chart**

#### Complexity

The complexity of algorithms is.

### String Comparison

#### Define Problem

* + - Because most of the medicine names are written in English so it’s difficult for Vietnamese to write the exact name of medicine. They often misspelled such as “Panadon” instead of “Panadol” and so on.

#### Solution

* We use n-gram algorithm with n = 2 (bigram) to compare the similarity between input keyword and medicine name. the bigram formula is:

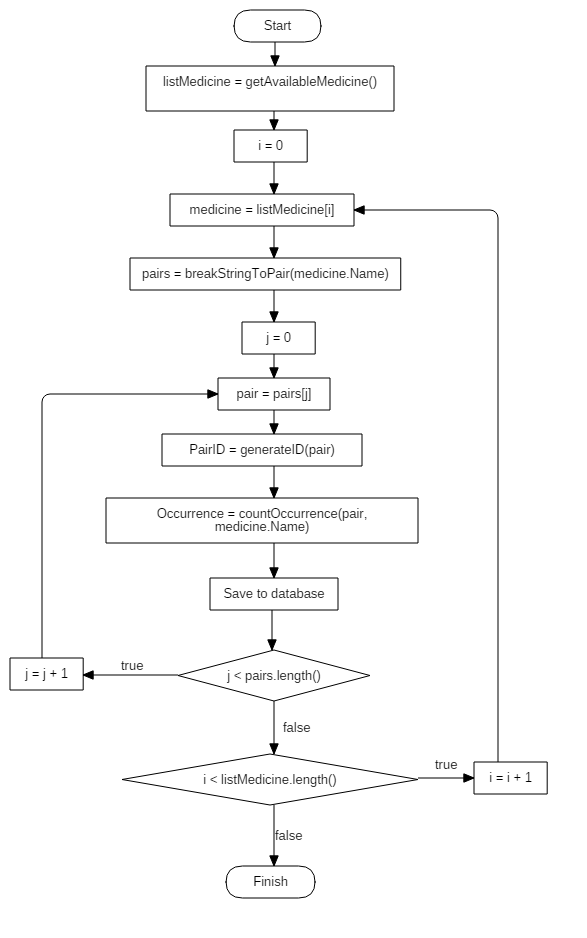
Only medicine name with similarity above 50% will be suggest to user.

* To improve suggest performance, we implement n-gram in sql server. This process consists of 2 phases as follow:

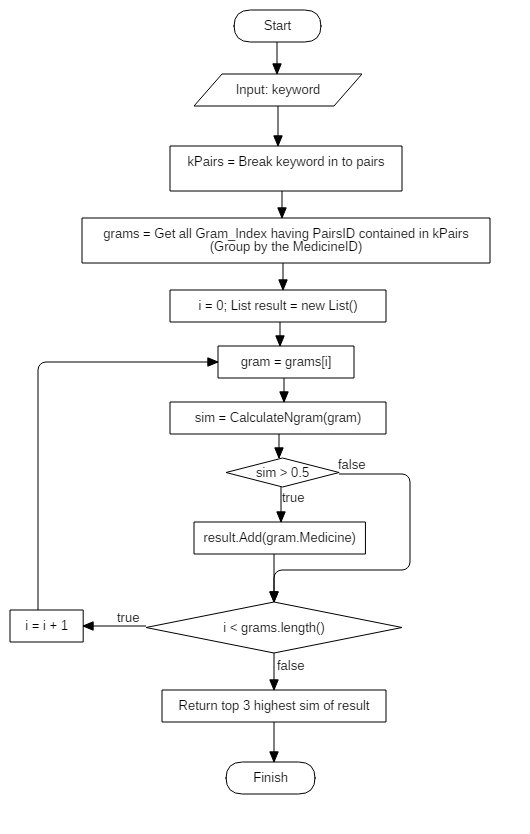
##### Phase 1: Prepare in database:

Every day, a scheduler is enable to update ngram index for medicine with following step:

* + - * Get all available medicine in system which has IsNgramUpdated = false.
      * For each medicine, break the medicine name into pairs of 2 characters and calculate the occurrence of them.
      * Retrieved data will be stored in table “Ngram\_Index” with the following properties:
        + MedicineID : the ID of the calculated medicine.
        + PairID: the unique integer number represent each pair which calculated according to the formula:
* d is the string containing 50 valid characters.
* pos(c1, d): the position of character c1 in d.
* pos(c2, d): the position of character c2 in d.
* Ex:
  + - * + Occurrence: the number of occurrences of this pair in the medicine name.



**Figure 13: String Comparison- Flow chart Phase 1**

* + Phase 2: When user input keyword
    - * Break keyword into pairs of 2 character then calculate the ID of each pair.
      * Search in “Gram\_Index” table to get all records having “PairID” contained in keyword’s pairIDs. Group the search result by the MedicineID.
      * For each group, calculate the n-gram value of the medicine compared with the keyword:
      * The medicine with sim (name, keyword) > 0.5 will be suggest for user.

**Figure 14: String Comparison- Flow chart Phase 2**

### Search medicine by illness mechanism

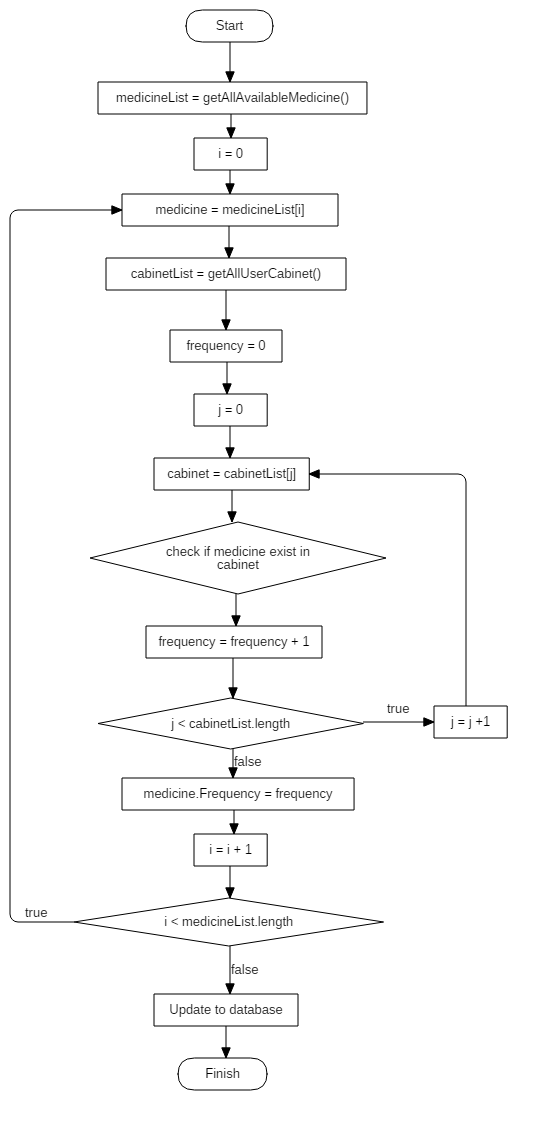
#### Define problem

* + - With most of Vietnamese user, they usually call the common medicines by the illness that its cure instead of its name such as “thuốc ho”, “thuốc cảm” and slowly they became the name of the medicine. This habit make its very difficult for users to check the medicine’s detail information.

#### Solution

* + - Our system provide search function that allow user to input the illness then response the list of medicines that was sorted by the most convenient order for users, as follow:
    - We use full text index provided by SQL Server 2008 to improve search performance.
    - Returned results sorted based on 3 criteria: relevance to keywords, priority medicines available in the user's cabinet and finally the popularity of such medicines.
    - The popularity of each medicine was updated every day, is calculated by counting the number of occurrences of the medicine on each user's medicine cabinet.

#### Flow chart



**Figure 15: Search medicine by illness mechanism- Flow chart**

#### Complexity: O(n2)

### Prevent time-dependent

#### Define problem

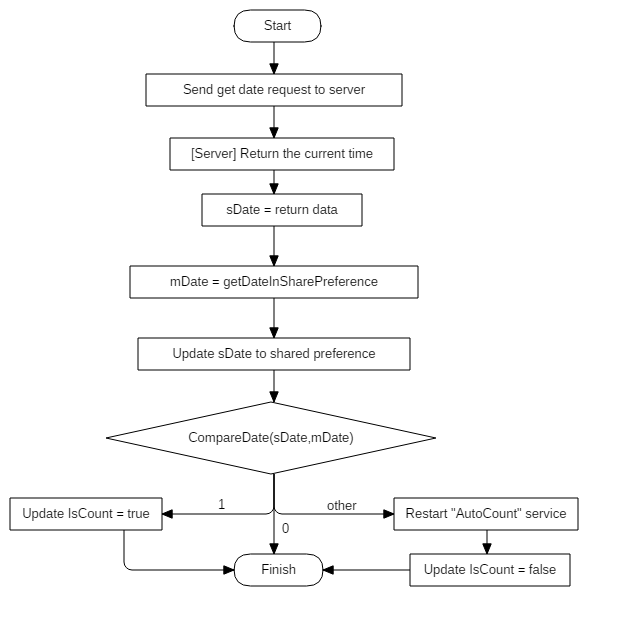
* + - One of the important functions of the application is tracking the status of user’s medicine based on the current day. This function requires working well in offline mode without being dependent on the device’s time.

#### Solution

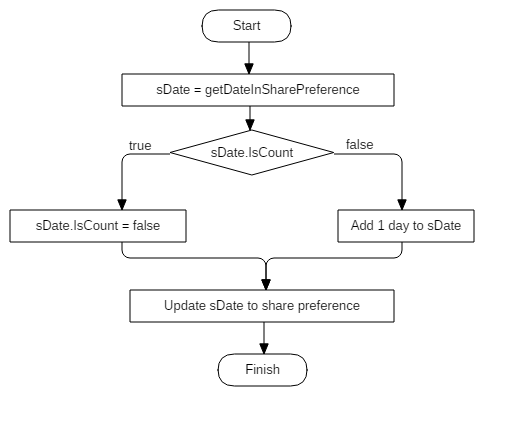
Each application have 2 variables in share preference:

* ServerDate: used to store the server's date, initiate when user login the first time via app.
* IsCount: flag variable, default value = false.
  + - Create a service named “AutoCount” that automatic update the “ServerDate” every 24 hours when the application running in offline mode:
      * IsCount = false: add 1 day to “ServerDate”.
      * IsCount = true: set IsCount = false.
    - Create a method to synchronize the server’s date with the “ServerDate” in application when running in online mode:
      * Compare the returned server date (sDate) with the one in application(mDate):
        1. sDate – mDate = 1: update sDate to “ServerDate”, set IsCount = true.
        2. sDate = mDate: do nothing.
        3. Else: update sDate to “ServerDate” then restart the “AutoCount” service with IsCount = false.
    - All functions related to the current date comparison will use “ServerDate” instead of device’s date.

#### Flow chart

* Sync with server time:

**Figure 16: Prevent time-dependent- Flow chart \_ Sync with server time**

* Auto update date in offline mode

**Figure 17: Prevent time-dependent- Flow chart \_ Auto update date in offline mode**

### Parsing data from website

#### Define Problem

Our system need a dictionary for users to search medicine by name or by illness symptom. In addition we also suggest posts which are related to the medicines that users have in their cabinet. If staff add medicine or post one by one manually, it will take so much time and effort.

#### Requirement

The algorithm get medicines and posts information from a website and automatically insert data into system database.

#### Solution

To solve this problem, we have a parser which follow these steps to parse data from a specific website:

* Parser has a crawler which will visit all internal pages of the website and push their URL to a queue (each URL in the queue is not overlap).
* For each URL in the queue, parser will send request and fetch data from the response based on the configured XPATHs.
* Data after parsing will be validate. If it is valid, it will be inserted to database.
* Staff just only check the data has been parsed and approve it.

#### Flowchart

C:\Users\Dell\Downloads\Parse data.png

**Figure 18: Prevent time-dependent- Flow chart \_ Parsing data from website**

# Task sheet

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Product Deliverables** | **Task** | **Nhân TL** | **Nguyên NK** | **Thuận TN** | **Quyền PQ** | **Size** |
| 1 | Report1 - Introduction | Project Information | **O** |  |  |  | 1 |
| Introduction |  | **O** |  |  | 1 |
| Current Situation |  |  | **O** |  | 1 |
| Problem Definition |  |  |  | **O** | 1 |
| Proposed Solution | **O** |  |  |  | 1 |
| Functional Requirement |  | **O** |  |  | 1 |
| Review document | **O** |  |  |  | 2 |
| 2 | Report2- Software Project Management Plan | Problem Definition |  |  | **O** |  | 1 |
| Project organization |  |  |  | **O** | 1 |
| Project management plan | **O** |  |  |  | 1 |
| Coding Convention |  | **O** |  |  | 1 |
| User Requirement Specification |  |  | **O** |  | 1 |
| Review document | **O** |  |  |  | 2 |
| 3 | Report 3- Software Requirement Specification | User Requirement Specification |  |  |  |  | 1 |
| **System Requirement Specification** |  |  |  |  | 1 |
| Guest requirement |  |  |  | **O** | 1 |
| User requirement |  | **O** |  |  | 1 |
| Staff requirement |  |  | **O** |  | 1 |
| Admin requirement |  |  |  | **O** | 1 |
| Scheduler requirement | **O** |  |  |  | 1 |
| **System Requirement Specification** |  |  |  |  | 1 |
| Externalize Interface Requirement | **O** |  |  |  | 1 |
| System Overview Use case | **O** |  |  |  | 1 |
| **List of Use Case** |  |  |  |  | 1 |
| <Guest> Login |  |  |  | **O** | 1 |
| <Guest> Register |  |  |  | **O** | 1 |
| <Authorized> Log out |  |  |  | **O** | 1 |
| <Admin> Add Account |  |  |  | **O** | 1 |
| <Admin> Delete Account |  |  |  | **O** | 1 |
| <Admin> Update Account |  |  |  | **O** | 1 |
| <User> Search Medicine |  |  |  | **O** | 1 |
| <User> Get Medicine Information |  |  | **O** |  | 1 |
| <User> Add Medicine to Cabinet |  |  | **O** |  | 1 |
| <User> Remove Medicine from Cabinet |  |  | **O** |  | 1 |
| <User> Get Cabinet's Medicine list |  |  | **O** |  | 1 |
| <User> Update Medicine |  |  | **O** |  | 1 |
| <User> Configure scheduler | **O** |  |  |  | 1 |
| <User> Get prescription's detail | **O** |  |  |  | 1 |
| <User> Get list prescription | **O** |  |  |  | 1 |
| <User> Get medicine's related posts | **O** |  |  |  | 1 |
| <Staff> Set scheduler for notification |  | **O** |  |  | 1 |
| <Staff> Add new parser |  | **O** |  |  | 1 |
| <Staff> Edit parser |  | **O** |  |  | 1 |
| <Staff> Remove parser |  | **O** |  |  | 1 |
| <Staff> Activate parser |  | **O** |  |  | 1 |
| <Staff> Deactivate parser |  | **O** |  |  | 1 |
| <Staff> Parse medicine manually |  |  |  | **O** | 1 |
| <Staff> Add medicine manually |  |  |  | **O** | 1 |
| <Staff> Add barcode |  |  |  | **O** | 1 |
| <Staff> Remove barcode |  |  |  | **O** | 1 |
| <Staff> Update barcode |  |  |  | **O** | 1 |
| <Staff> Approve barcode |  |  |  | **O** | 1 |
| <Scheduler> Parse medicine |  |  | **O** |  | 1 |
| <Scheduler> Notify user |  |  | **O** |  | 1 |
| <Scheduler> Create notification |  |  | **O** |  | 1 |
| <Scheduler> Remove expired medicine |  |  | **O** |  | 1 |
| Software System Attribute |  | **O** |  |  | 1 |
| Conceptual Diagram |  | **O** |  |  | 2 |
| System Architectural Design | **O** |  |  |  | 2 |
| 4 | Report 4- Software Design Description | Component Diagram |  |  | **O** |  | 2 |
| **Detailed Description** |  |  |  |  | 1 |
| Class Diagram |  |  |  | **O** | 2 |
| Class Diagram Explanation |  |  |  | **O** | 2 |
| **Sequence Diagram** |  |  |  |  | 1 |
| **<Website>** |  |  |  |  | 1 |
| <User> Search Medicine |  |  |  | **O** | 1 |
| <User> Get Cabinet's Medicine list |  |  | **O** |  | 1 |
| <User> Add Medicine to Cabinet |  |  | **O** |  | 1 |
| <User> Get Medicine Information |  |  | **O** |  | 1 |
| <User> Remove Medicine from Cabinet |  |  | **O** |  | 1 |
| <User> Update Medicine Expired Date |  |  | **O** |  | 1 |
| <User> Update Medicine Quantity |  |  | **O** |  | 1 |
| <Staff> Parse medicine manually |  | **O** |  |  | 1 |
| <Staff> Parse News manually |  | **O** |  |  | 1 |
| <Scheduler> Create notification |  |  |  | **O** | 1 |
| <Scheduler> Parse medicine |  |  |  | **O** | 1 |
| <Scheduler> Parse medicine |  |  |  | **O** | 1 |
| **<Mobile>** |  |  |  |  | 1 |
| <User> Login | **O** |  |  |  | 1 |
| <User> Add New Medicine | **O** |  |  |  | 1 |
| <User> Add Prescription | **O** |  |  |  | 1 |
| <User> Update Alarm status | **O** |  |  |  | 1 |
| <User> Medication reminder | **O** |  |  |  | 1 |
| <User> Scan barcode | **O** |  |  |  | 1 |
| <User> Synchronize data | **O** |  |  |  | 1 |
| **Interface** |  |  |  |  | 1 |
| User Interface Design | **O** |  | **O** |  | 1 |
| **Database Design** |  | **O** |  |  | 1 |
| **Algorithms** |  |  |  |  | 1 |
| Synchronize data | **O** |  |  |  | 1 |
| String Comparison | **O** |  |  |  | 1 |
| Search medicine by illness mechanism | **O** |  |  |  | 1 |
| Prevent time-dependent | **O** |  |  |  | 1 |
| Parsing data from website |  | **O** |  |  | 1 |
| **System Implementation & Test** |  |  |  |  |  |
| 5 | Report 5 - Software Implementation and Test Document | Introduction |  |  | **O** |  | 1 |
| Database Relationship Diagram |  |  |  | **O** | 1 |
| Performance Measures |  |  |  | **O** | 1 |
| Test Plan |  |  |  | **O** | 1 |
| **System Testing Test Case** |  |  |  |  |  |
| Communication Diagram |  |  |  | **O** | 1 |
| Test cases |  |  |  | **O** | 1 |
| Review document | **O** |  |  |  | 1 |
| 6 | Report 6 - Software User's Manual | Installation Guide |  |  | **O** |  | 1 |
| **User’s Guide** |  |  |  |  | 1 |
| Web app |  |  | **O** |  | 1 |
| Mobile app |  |  | **O** |  | 1 |
| Review document | **O** |  |  |  | 1 |
|  |  |  |  |  | 1 |
| 7 | **Architecture** | Design project structure | **O** |  |  |  | 1 |
| Apply framework & design pattern | **O** |  |  |  | 1 |
| 8 | **Implementation** | **Algorithms** |  |  |  |  | 1 |
| Synchronize data | **O** |  |  |  | 2 |
| String Comparison | **O** |  |  |  | 2 |
| Search medicine by illness mechanism |  |  |  | **O** | 2 |
| Prevent time-dependent | **O** |  |  |  | 2 |
| Parsing data from website |  | **O** |  |  | 3 |
| **UI** |  |  |  |  | 1 |
| Web app | **O** |  | **O** |  | 2 |
| Mobile app | **O** |  |  |  | 2 |
| **Web app** |  |  |  |  |  |
| Staff |  | **O** |  |  | 2 |
| Admin |  |  | **O** | **O** | 1 |
| User |  |  | **O** | **O** | 2 |
| Scheduler | **O** |  |  |  | 1 |
| Web services | **O** |  |  |  | 2 |
| Mobile app | **O** |  |  |  | 2 |

# Appendix

1. SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.
2. C# Coding Conventions (C# Programming Guide) for Visual Studio 2013 by Microsoft

<https://msdn.microsoft.com/en-us/library/ff926074(v=vs.120).aspx>

1. Android Code Style for Contributors

<http://source.android.com/source/code-style.html>

1. Android Developer Guider – Application Fundamentals

<http://developer.android.com/guide/components/fundamentals.html>

1. UML for drawing diagram (version 1)

<http://staruml.sourceforge.net/v1/>