|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **C:\Users\sonnx0907\Desktop\Untitled.png** | **MINISTRY OF EDUCATION AND RAINING** | | | **FPT UNIVERSITY** | | |
| Capstone Project Document |
| Hospital Finder |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Group 11** | | | **Group Members** | Lê Phước Việt – Team Leader – SE60706  Nguyễn Xuân Sơn – Team Member – SE60727 | | **Supervisor** | Nguyễn Đức Khoan | | **Ext Supervisor** | N/A | | **Capstone Project code** | HospitalF | | |

- Ho Chi Minh City, May / 2014 –

**CAPSTONE PROJECT REGISTER**

Class: Duration time: from 12/05/201 To 03/09/2014

(\*) Profession: <Software Engineer> Specialty: <ES> <IS>

x

(\*) Kinds of person make registers: Lecturer Students

X

1. Register information for supervisor (if have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **E-Mail** | **Title** |
| Supervisor 1 | Nguyễn Đức Khoan |  | khoannd@fpt.edu.vn | Mr. |

2. Register information for students (if have)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Full name** | **Student code** | **Phone** | **E-mail** | **Role in Group** |
| Student 1 | Lê Phước Việt | SE60706 | 0949051851 | vietlpse60706@fpt.edu.vn | Team leader |
| Student 2 | Nguyễn Xuân Sơn | SE60727 | 01214184300 | sonnxse60727@fpt.edu.vn | Team member |
| Student 3 | Dương Thị Hoàng Anh | 60434 | 0908616730 | anhdth60434@fpt.edu.vn | Team member |

3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

English: Hospital Finder.

Vietnamese: Tìm kiếm bệnh viện

Abbreviation:

- HospitalF

- Building the web application which compatible with both computer and mobile devices supports user to find the appropriate hospitals for their conditions. The system can find the appropriate hospitals and shows the result list and on map. In this map, user can interact with the system to see hospital image, name, address, short description, and the rating. The system can show full description, more images and users’ reviews. User also can write a review, rating and share information of the hospital if they have a social account.

(\*) 3.2. Main proposal content (including result and product)

1. Theory and practice (document):

* Student should apply the software development process and the UML
* Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
* 3 tiers should be applied
* Server side technique:
  + Database design, OOA, OOD, OOP, MVC, Java, PHP or .Net technology, …
* Client side technique
  + HTML5, CSS, JavaScript, JQuery, Ajax, Android, …
* Communication technique
  + Exchange information and transfer data in effective in networks
* Research
  + Google Map API, Social Network Integration, …

1. Program:

* Main functions
  + Admin
    - Manage the systems, manage account, configure systems …
  + Staff
    - Update and create the new users, update hospital information ...
    - Make the statistic
  + Users
    - Make query/request, login using social account, rating, write the reviews

1. Other products:

N/A

4. Other comment (propose all relative thing if have)

N/A

|  |  |
| --- | --- |
| **Supervisor**  *(Sign and full name)* | HCM city, date 12/05/2014  **On behalf of Registers**  *(Sign and full name)* |

|  |
| --- |
|  |

**Table of Contents**

[**Table of Acronyms and Abbreviations** 10](#_Toc396111604)

[**A.** **Introduction** 11](#_Toc396111605)

[**1.** **Project information** 11](#_Toc396111606)

[**2.** **Introduction** 11](#_Toc396111607)

[**3.** **Current Problem** 11](#_Toc396111608)

[**4.** **Proposed solution** 11](#_Toc396111609)

[**5.** **Functional Requirement** 12](#_Toc396111610)

[**5.1.** **Search Hospital** 12](#_Toc396111611)

[**5.2.** **Hospital Management** 12](#_Toc396111612)

[**5.3.** **User Management** 12](#_Toc396111613)

[**5.4.** **Meta data management** 12](#_Toc396111614)

[**5.5.** **Statistics** 12](#_Toc396111615)

[**5.6.** **Mobile Devices** 12](#_Toc396111616)

[**6.** **Role and responsibility** 12](#_Toc396111617)

[**B.** **Project Management Plan (PMP)** 13](#_Toc396111618)

[**1.** **Problem Definition** 13](#_Toc396111619)

[**1.1.** **Name of this Capstone Project** 13](#_Toc396111620)

[**1.2.** **Problem Abstract** 13](#_Toc396111621)

[**1.3.** **Project Overview** 13](#_Toc396111622)

[**1.3.1.** **The Current System** 13](#_Toc396111623)

[**1.3.2.** **The Proposed System** 13](#_Toc396111624)

[**1.3.2.1.** **Computers** 13](#_Toc396111625)

[**1.3.2.2.** **Mobile Devices** 14](#_Toc396111626)

[**1.3.3.** **Boundaries of the System** 14](#_Toc396111627)

[**1.3.4.** **Development Environment** 15](#_Toc396111628)

[**1.3.4.1.** **Hardware requirements** 15](#_Toc396111629)

[**1.3.4.2.** **Software requirements** 15](#_Toc396111630)

[**2.** **Project Organization** 15](#_Toc396111631)

[**2.1.** **System Process Model** 15](#_Toc396111632)

[**2.2.** **Roles and Responsibilities** 16](#_Toc396111633)

[**2.3.** **Tools and Techniques** 17](#_Toc396111634)

[**2.3.1.** **Back-end** 17](#_Toc396111635)

[**2.3.2.** **Front-end** 17](#_Toc396111636)

[**3.** **Project Management Plan** 18](#_Toc396111637)

[**3.1.** **Overview** 18](#_Toc396111638)

[**3.2.** **Detail** 19](#_Toc396111639)

[**3.2.1.** **Iteration 1** 19](#_Toc396111640)

[**3.2.2.** **Iteration 2** 19](#_Toc396111641)

[**3.2.3.** **Iteration 3** 19](#_Toc396111642)

[**3.2.4.** **Iteration 4** 20](#_Toc396111643)

[**4.** **Convention Rules** 20](#_Toc396111644)

[**C.** **System Requirements Specification (SRS)** 20](#_Toc396111645)

[**1.** **User Requirement Specification** 20](#_Toc396111646)

[**1.1.** **Guest requirement** 21](#_Toc396111647)

[**1.2.** **Normal User requirement** 21](#_Toc396111648)

[**1.3.** **Hospital User requirement** 21](#_Toc396111649)

[**1.4.** **Administrator requirement** 21](#_Toc396111650)

[**2.** **System Requirements Specification** 22](#_Toc396111651)

[**2.1.** **External Interface Requirements** 22](#_Toc396111652)

[**2.1.1** **User Interfaces** 22](#_Toc396111653)

[**2.1.2** **Software Interfaces** 22](#_Toc396111654)

[**2.1.3** **Communication Protocol** 22](#_Toc396111655)

[**2.2.** **Function Requirements** 22](#_Toc396111656)

[**2.2.1.** **System Overview** 22](#_Toc396111657)

[**2.2.2.** **Use case List** 23](#_Toc396111658)

[**2.2.2.1.** **<GU> Overview Use Case** 23](#_Toc396111659)

[**2.2.2.2.** **<NU> Overview Use Case** 24](#_Toc396111660)

[**2.2.2.3.** **<HU> Overview Use Case** 25](#_Toc396111661)

[**2.2.2.4.** **<AD> Overview Use Case** 25](#_Toc396111662)

[**2.3.** **Non-Functional Requirements** 26](#_Toc396111663)

[**2.3.1.** **Reliability** 26](#_Toc396111664)

[**2.3.2.** **Availability** 26](#_Toc396111665)

[**2.3.3.** **Security** 26](#_Toc396111666)

[**2.3.4.** **Maintainability** 26](#_Toc396111667)

[**2.3.5.** **Portability** 27](#_Toc396111668)

[**2.3.6.** **Performance** 27](#_Toc396111669)

[**3.** **Entity Relationship Diagram** 28](#_Toc396111670)

[**D.** **System Design Description (SDD)** 29](#_Toc396111671)

[**1.** **System Architecture Design** 29](#_Toc396111672)

[**2.** **Component Diagram** 29](#_Toc396111673)

[**3.** **Detailed Description of Components** 29](#_Toc396111674)

[**4.1.** **Class Diagram** 30](#_Toc396111675)

[**4.** **Sequence Diagram** 31](#_Toc396111676)

[**4.1.** **Normal Search Hospital** 31](#_Toc396111677)

[**4.2.** **Advanced Search Hospital** 31](#_Toc396111678)

[**4.3.** **Location Search Hospital** 31](#_Toc396111679)

[**4.4.** **Import Hospital Using Excel** 31](#_Toc396111680)

[**4.5.** **Save valid Excel records** 32](#_Toc396111681)

[**4.6.** **Create Hospital** 32](#_Toc396111682)

[**4.7.** **View Hospital Detail** 33](#_Toc396111683)

[**4.8.** **Display list of hospitals** 33](#_Toc396111684)

[**4.9.** **Update Hospital Information** 34](#_Toc396111685)

[**4.10.** **Comment** 34](#_Toc396111686)

[**4.11.** **View Statistics** 34](#_Toc396111687)

[**5.** **Algorithms** 36](#_Toc396111688)

[**5.1.** **Geographic Information Retrieval** 36](#_Toc396111689)

[**5.1.1.** **Definition** 36](#_Toc396111690)

[**5.1.2.** **Examples** 36](#_Toc396111691)

[**5.2.** **Haversine Formula** 37](#_Toc396111692)

[**5.2.1.** **Definition** 37](#_Toc396111693)

[**5.2.2.** **Example** 37](#_Toc396111694)

1. **Introduction**
2. **Project information**

* Project name: Hospital Finder
* Project code: HospitalF
* Product type: Responsive design website
* Start date: May 12, 2014
* End date: September 03, 2014

1. **Introduction**

* The needs of health services for people is raising every day. A list of hospitals or clinics center in a specific region is not so difficult to find, but finding suitable health services for a specific illness or disease is a headache problem. A nearer and faster curing service, a better chance for a patient to improve his/her condition. The more health services people required, the more specific information people want to know about their chosen centers. Understanding that needs, we offer a brand new website system for hospital service in Vietnam, named Hospital Finder.

1. **Current Problem**

* A system for finding appropriate health services base on patients’ conditions is not yet published.
* Vietnam doesn’t have an official published list of hospitals and clinics that offer health services in the country, and many non-official websites provide different versions of the list so the information is not consistent.
* A better list is supported by Google Map by using “Hospital” or “Health” keywords, but some locations and names are not relevant to health services because of attached advertising tag keywords.

1. **Proposed solution**

* Hospital Finder is a website application that’s built for finding appropriate hospitals with users’ conditions.
* The result will be displayed on a visual map, users can interact with it to show full description, images and another users’ reviews about any health centers. Users can also write comments, create rating or feedback to administrators if they see the provided information is not suitable or incorrect.
* The system is compatible with both computers and mobile devices.

1. **Functional Requirement**
2. **Search Hospital**

* Users can search for hospitals or clinics by using those features below:
* Patients’ conditions.
* Centers in a specific area in Vietnam.
* Centers around current location or at a specific location address.
* Centers that located at a specific distance reference to the current location of user.
* Search results and the shortest way to a specific center will be displayed on an interactive visual map.

1. **Hospital Management**

\_ Administrators can import hospitals to the system using a pre-defined Excel template or create them manually.

\_ Hospital members can manage, add or update the information of their hospitals.

1. **User Management**

\_ Administrators have the right to create new hospital members. Keep track of current using members and have the right to active or de-active their accounts.

1. **Meta data management**

* Administrators have the right to manage data of specialities, diseases, supporting services or facilities of the system.
* Those data will be used as basement data for the whole system.

1. **Statistics**

\_ System supports generating statistics about those features below:

* Search query that provides the highest counting result.
* Pecentage of each type of hospital in the system.
* Percentage of hospital that being managed by hospital user accounts.

1. **Mobile Devices**

\_ Hospital Finder system is compatible with both computers and mobile devices.

1. **Role and responsibility**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Full Name** | **Roles** | **Position** | **Contact Detail** |
| 1 | Nguyễn Đức Khoan | Project Manager | Instructor | khoannd@fpt.edu.vn |
| 2 | Lê Phước Việt | Designer, developer, tester | Leader | vietlpse60706@fpt.edu.vn |
| 3 | Nguyễn Xuân Sơn | Designer, developer, tester | Member | sonnxse60727@fpt.edu.vn |

1. **Project Management Plan (PMP)**
2. **Problem Definition**
3. **Name of this Capstone Project**

* Hospital Finder

1. **Problem Abstract**

* The needs of health services for people is raising every day. A list of hospitals or clinics center in a specific region is not so difficult to find, but finding suitable health services for a specific illness or disease is a headache problem. A nearer and faster curing service, a better chance for a patient to improve his/her condition. The more health services people required, the more specific information people want to know about their chosen centers. Understanding that needs, we offer a brand new website system for hospital service in Vietnam, named Hospital Finder.

1. **Project Overview**
2. **The Current System**

* A system for finding appropriate health services base on patients’ conditions is not yet published.
* Vietnam doesn’t have an official published list of hospitals and clinics that offer health services in the country, and many non-official websites provide different versions of the list so the information is not consistent.
* A better list is supported by Google Map by using “Hospital” or “Health” keywords, but some locations and names are not relevant to health services because of attached advertising tag keywords.

1. **The Proposed System**
   * 1. **Computers**

**Guest:**

* Search hospitals and clinics base on:
  + - Patients’ conditions.
    - Centers in a specific area in Vietnam.
    - Centers around current location or at a specific location address.
    - Centers that locate at a specific distance reference to current location.
* They can also view information of a specific hospital or clinic.
* Feedback about information of the system.

**Normal Users:**

* Have all priorities that supported for guests except Login.
* Create ratings and comments on a specific hospital or clinic.

**Hospital users:**

* Manage information about their hospitals or clinics.
* View users’ feedback about their hospitals or clinics.
* Manage doctor of their hospitals or clinics.
* View statistics.

**Administrators:**

* Create hospital users’ accounts.
* Manage normal and hospital users.
* View users’ feedback about the system.
* Manage and import list of hospitals or clinics.
* View statistics.
  + 1. **Mobile Devices**
* Only support functions for guests.

1. **Boundaries of the System**

* Hospital Finder can be used on both computers and mobile devices.
* Services for Administrators and Hospital users are not supported on mobile version.
* English and another foreign languages are not supported for searching.
* The list of hospitals and clinics in a specific area is not automatically updated.
* Auto detect tag keywords and search hospital base on alias name are not support in this version.
* The system is only an intermediate service between users and hospitals. It is not provide any service or method to manage business of a hospital or clinic online.

1. **Development Environment**
   * 1. **Hardware requirements**

For servers:

|  |  |  |
| --- | --- | --- |
| **Infrastructure** | **Minimum Requirements** | **Recommended** |
| Operating system | Window 7 | Window 8 |
| Processor | Intel® Core 2 Duo | Intel® Core i5-460M |
| Memory | 1 GB RAM | 2GB or higher |
| Internet connection | Cable or wireless connection (1.5Mb) | Cable or wireless connection (5Mb) |

For mobiles devices:

|  |  |  |
| --- | --- | --- |
| **Infrastructure** | **Minimum Requirements** | **Recommended** |
| Operating system | Android 4.0 | Android 4.4 |
| Memory | 512 GB RAM | 1GB RAM or higher |
| Internet connection | Wireless connection (1.5Mb) | Wireless connection (5Mb) |

* + 1. **Software requirements**

\_ Operating system: Window 7 Ultimate

\_ Smart phone operating system: Android 4.3

\_ IDE for server and web application: Microsoft Visual studio 2012 Ultimate

\_ Database: Microsoft SQL 2008 Express SP 2

\_ Google code and Tortoise SVN

\_ Browser: Google Chrome Version 34.0.1847.137

1. **Project Organization**
2. **System Process Model**

****

*For more information, please refer to the website:*

*http://blogs.globalteckz.com/characteristics-of-agile-methodology-in-software-development/*

1. **Roles and Responsibilities**

|  |  |  |
| --- | --- | --- |
| **Full name** | **Roles** | **Responsibilities** |
| Nguyễn Đức Khoan | Project Manager | * Give user requirements * Control developing process * Support |
| Lê Phước Việt | Team Leader | * Manage members and tasks * Designer * Developer * Tester |
| Nguyễn Xuân Sơn | Team Member | * Designer * Developer * Tester |

1. **Tools and Techniques**
2. **Back-end**

\_ HTML 5

\_ CSS 3

\_ Javascript

\_ Jquery

\_ Ajax

1. **Front-end**

\_ ASP.NET MVC 4

\_ Linq

\_ Android toolkit

\_ Microsoft IIS 7

\_ Microsoft SQL 2008 Express SP 2

1. **Project Management Plan**
   1. **Overview**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteration** | **Description** | **Output** | **Time** | **Note** |
| **1** | * Study technology that will be used in the project. * Identify business and system requirements. | * Business of the Hospital Finder system. | 20 days | * Might have misunderstanding between user and project team. |
| **2** | * Finding suitable searching algorithm. * Prepare meta-data. | * Searching algorithms. * Data of hospitals, specialities, diseases, supporting services and facilities. | 20 days | * Need to combine and optimize search technology and algorithm. * Lack of medical published data. |
| **3** | * Implement search methods. * Implement method to display hospital detail. * Implement methods to manage doctors in a specific hospital or clinic. | * User can search hospitals base on 3 search options: Normal search, advanced search and location search. * View hospital details. * Methods to add, update and display doctors’ information. | 30 days | * Search methods return unexpected results. * Data in results page and hospital detail page might be confused. |
| **4** | * Implement methods to manage hospitals * Implement methods to manage meta-data and users. * Implement statistics methods. | * Method to add, update hospitals and display a list of them. * Method to manage specialities, diseases, supporting services and facilities. * Method to add, update users. * Method to view statistics. | 20 days | * Excel template need to be defined. * Lack of data of specialities, diseases, supporting services and facilities. |

* 1. **Detail**

1. **Iteration 1**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Resource** |
| **Identify business of the system** | Decide which services will be provided to the users. | VietLP, SonNX, |
| **Report 1** | Create introduction report HospitalF system. | SonNX |
| **Project Management**  **Plan.** | Create plan for the whole project. | VietLP |
| **Website Prototype.** | Design website prototype | VietLP |
| **Design ERD diagram.** | Design entity relationship diagram. | SonNX |
| **Study Facebook service and Google APIs** | Study how to integrate Facebook and Google service to the project. | SonNX |

1. **Iteration 2**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Resource** |
| **Find search algorithms** | Find and study search algorithms. | VietLP, SonNX |
| **Geographic information retrieval algorithm** | Study and implement geographic information retrieval algorithm | VietLP, SonNX |
| **Report 2** | Create project management report for HospitalF system. | SonNX |
| **Prepare meta-data** | Find data of hospitals, specialities, diseases, supporting services and facilities. | SonNX |
| **Report 3** | Create use case and SRS document. | VietLP, SonNX |

1. **Iteration 3**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Resource** |
| **Normal search** | Implement normal search mode. | SonNX |
| **Advanced search** | Implement advanced search mode. | SonNX |
| **Location search** | Implement location search mode. | VietLP |
| **Hospital detail** | Implement hospital detail page. | VietLP |
| **Testing** | Test methods to search hospital and manage doctors of a specific hospital or clinic. | VietLP, SonNX |
| **Report 4** | Create UML diagrams and SDD document. | VietLP, SonNX |

1. **Iteration 4**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Resource** |
| **Manage hospital methods** | Implement methods to add, update, and view list of hospitals. | SonNX |
| **Manage meta-data** | Implement methods to add, update, view list of specialities, diseases, supporting services and facilities. | SonNX |
| **Manage users** | Implement methods to add, update, and view list of users | VietLP |
| **Feedback** | Implement methods to create, and view list of feedbacks. | VietLP |
| **Statistics** | Implement statistics methods. | VietLP |
| **Testing** | Test methods that will be implemented in iteration 4. | VietLP, SonNX |
| **Report 5** | Create test case and SIT document | VietLP, SonNX |
| **Report 6** | Create user manual report. | SonNX |

1. **Convention Rules**

* Using C# coding convention at page:  
  http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx

1. **System Requirements Specification (SRS)**
2. **User Requirement Specification**
   1. **Guest requirement**

Guests are the people who can use the system without any authentications process. They are supported with the following functions below:

\_ Search hospitals and clinics base on:

+ Patients’ conditions.

+ Centers in a specific area in Vietnam.

+ Centers around current location or at a specific location address.

+ Centers that locate at a specific distance reference to current location.

\_ View information of hospitals or clinics, and create feedback about the system.

* Feedback about information of the system.
  1. **Normal User requirement**

Normal Users are the people who use the system with their specific authenticated Facebook accounts. They have the right to use all the functions that supported for Guests (except Login), and the following functions below:

\_ Create rating.

\_ Comment or reply comments.

* 1. **Hospital User requirement**

Hospital Users are the people who have special accounts to manage information of a specific hospitals or clinic. They are supported with the following functions below:

\_ Change password.

\_ Manage information about their hospitals or clinics.

\_ View users’ feedback about their hospitals or clinics.

\_ Add doctors.

\_ View statistics.

* 1. **Administrator requirement**

Administrators are the people who have the highest priorities in the system. They are supported with the following functions below:

* Create hospital users’ accounts.
* Manage normal and hospital users.
* View users’ feedback.
* Manage hospitals.
* Manage meta-data (Specialities, diseases, supporting services and facilities).
* View statistics.

1. **System Requirements Specification**
2. **External Interface Requirements**
3. **User Interfaces**

* User interface of the website must be clear and easy to use.
* Mobile interface must be compatible with multiple sizes of smart phone and tablet screens.
* Error messages must be highlight, easy to see and understand, and do not contain disturb or abused words or grammars.

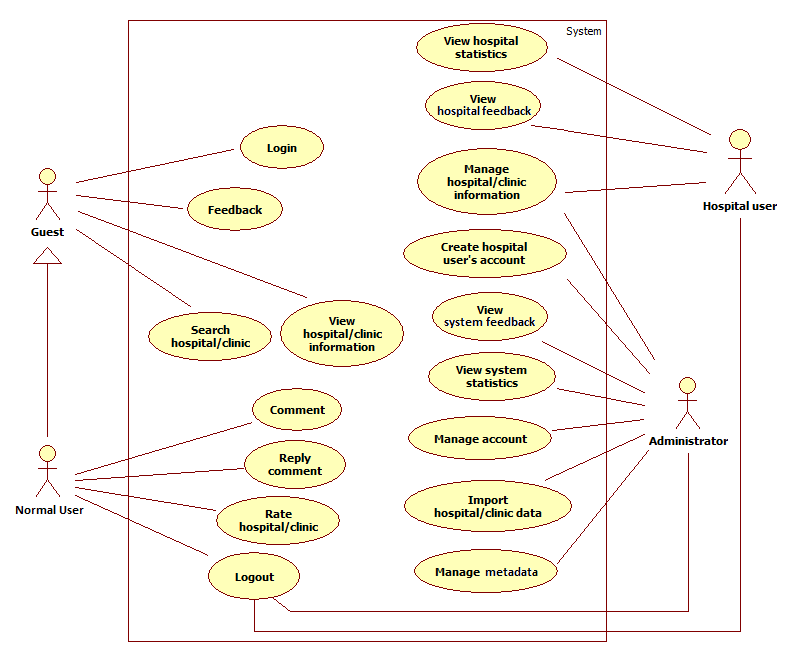
1. **Software Interfaces**

* Web browser: Google Chrome and Fire Fox.
* Support Java Script and HTML5.

1. **Communication Protocol**

* HTTP Protocol will be used to communicate between web browsers, Android devices and web server.

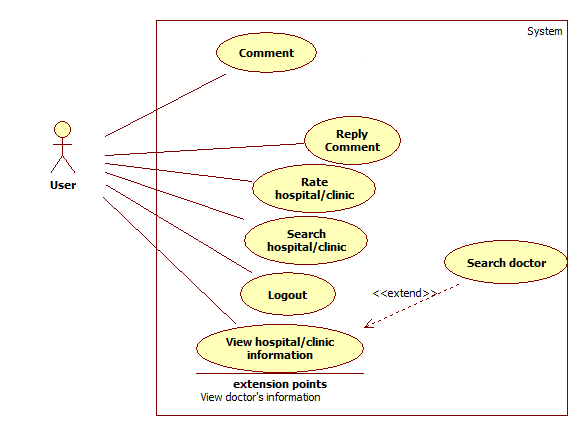
1. **Function Requirements**
2. **System Overview**



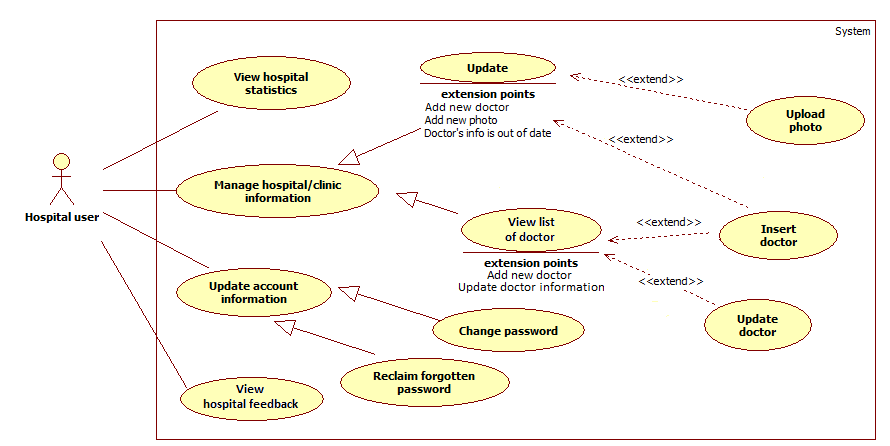
1. **Use case List**
   * + 1. **<GU> Overview Use Case**



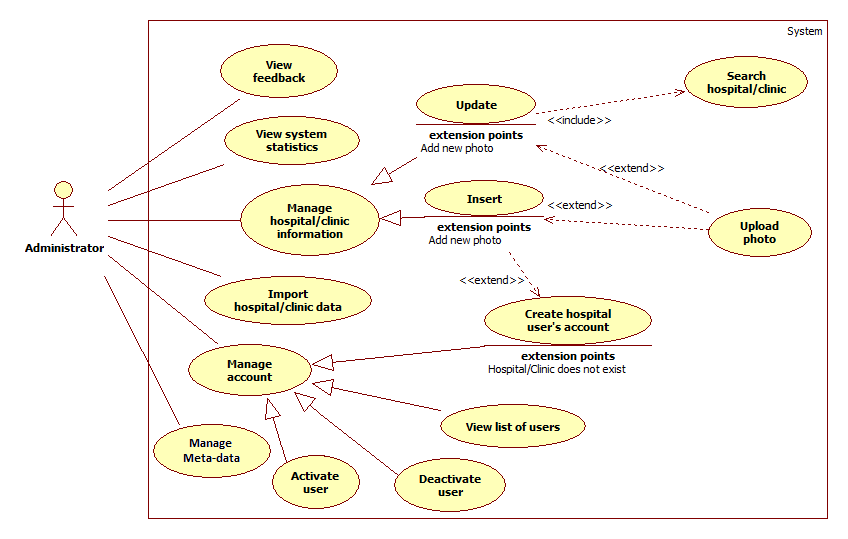
* + - 1. **<NU> Overview Use Case**



* + - 1. **<HU> Overview Use Case**



* + - 1. **<AD> Overview Use Case**



1. **Non-Functional Requirements**
2. **Reliability**

* Locations and information of hospitals and clinics must be precise as it be in reality.

1. **Availability**

* Services of the system must be provided 24/24.

1. **Security**

* Information of users must be kept secretly.
* Each type of user has a specific priorities in the system.
* Always check for authorization before executing any business.

1. **Maintainability**

* Code for Controller, Model, and View must be divided clearly.
* Complicated query must be written in Structure Query Language (SQL).
* All constant values and error messages

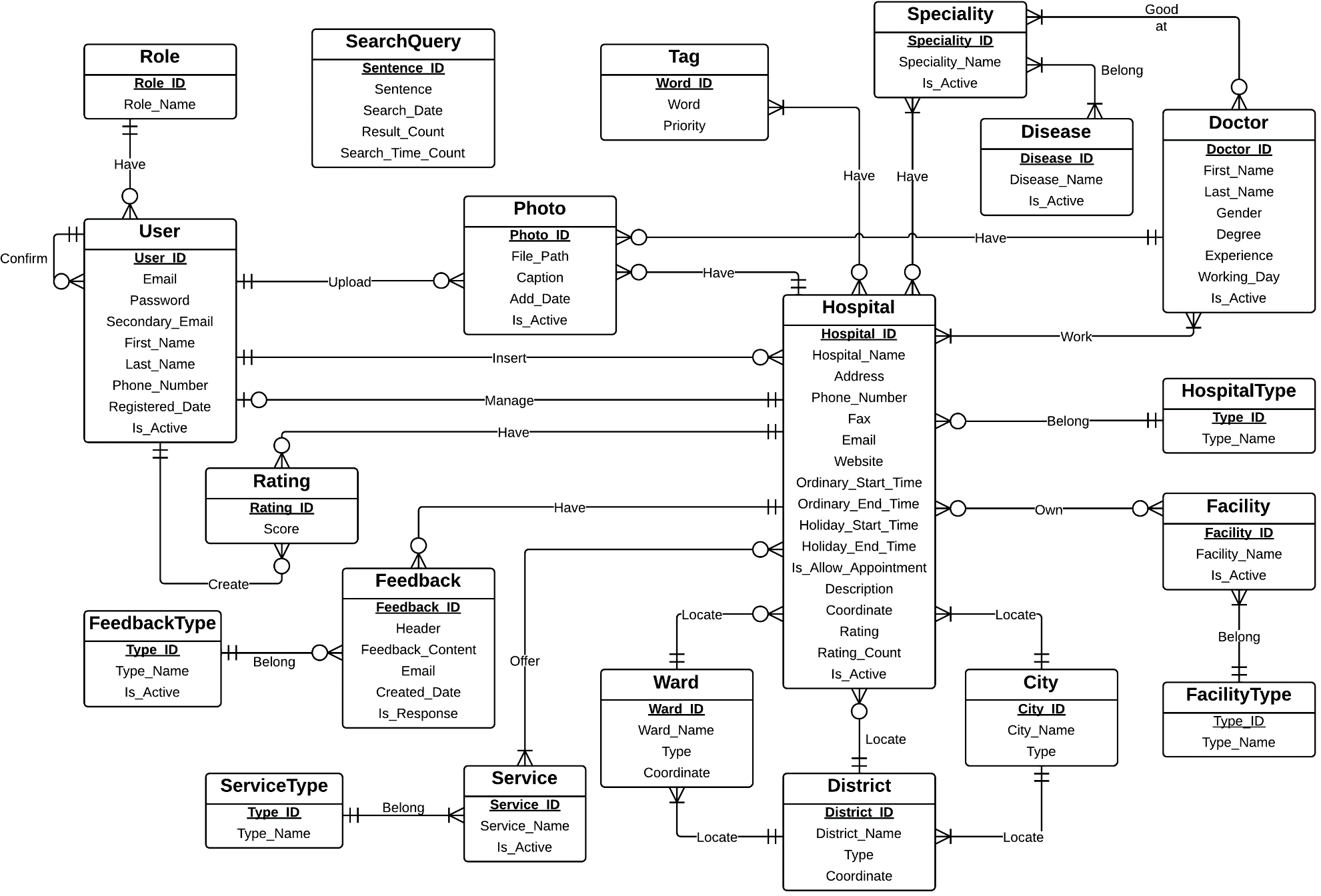
1. **Portability**

N/A

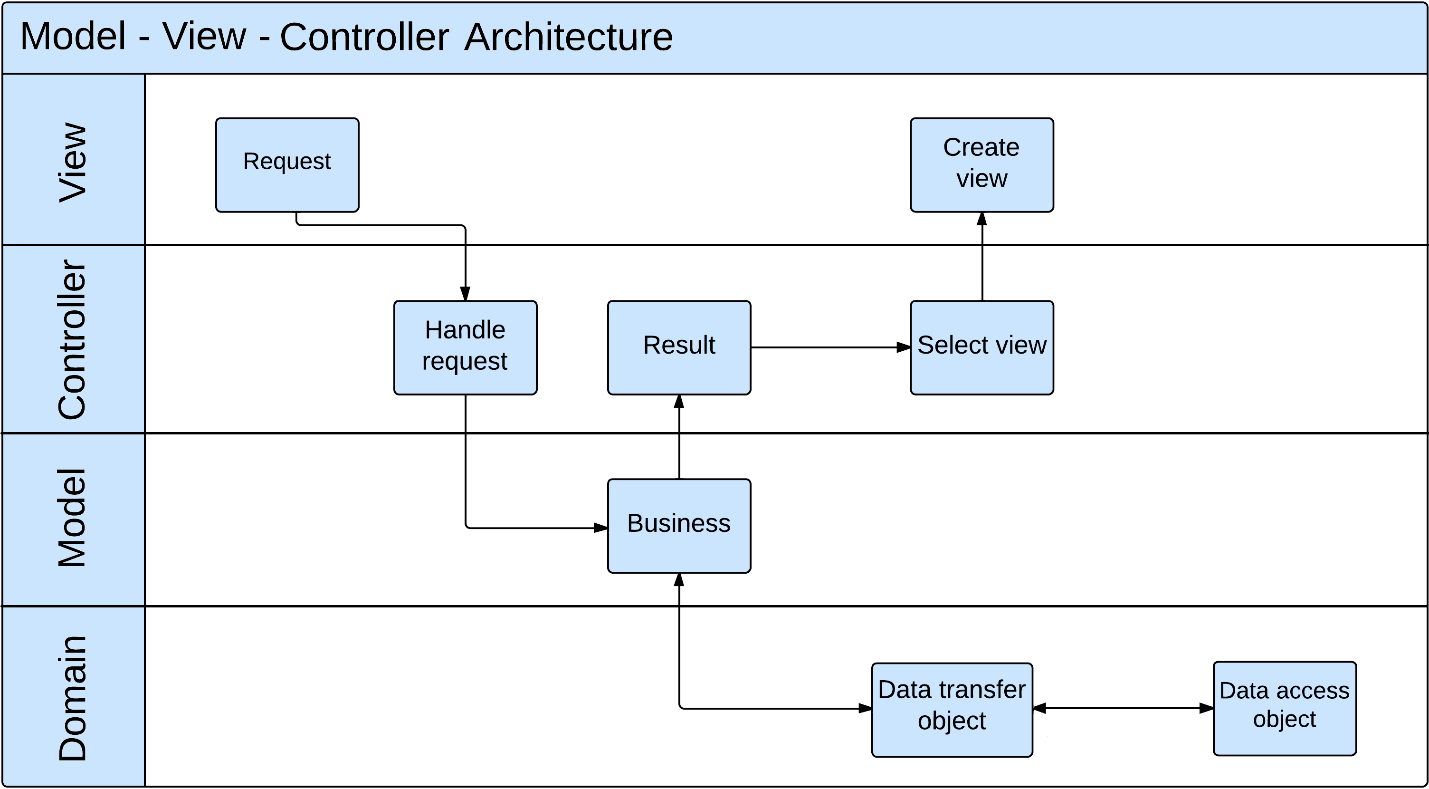
1. **Performance**

* Search process must be done in less than 5 seconds with network and hardware conditions are equal or better than development environments.
* Words and sentences suggestion must be showed in less than 3 seconds right after users input search query.

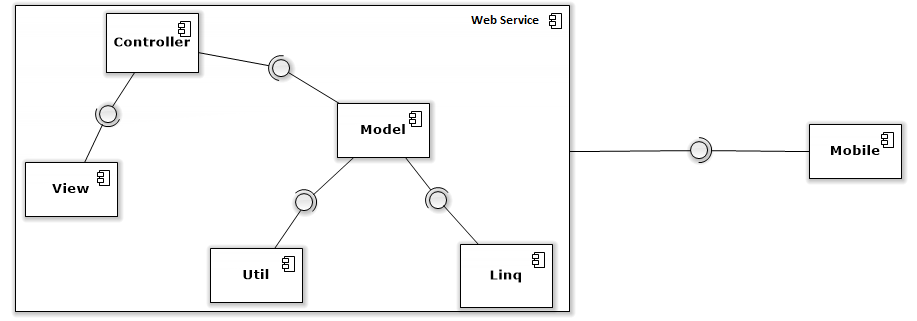
1. **Entity Relationship Diagram**

****

1. **System Design Description (SDD)**
2. **System Architecture Design**

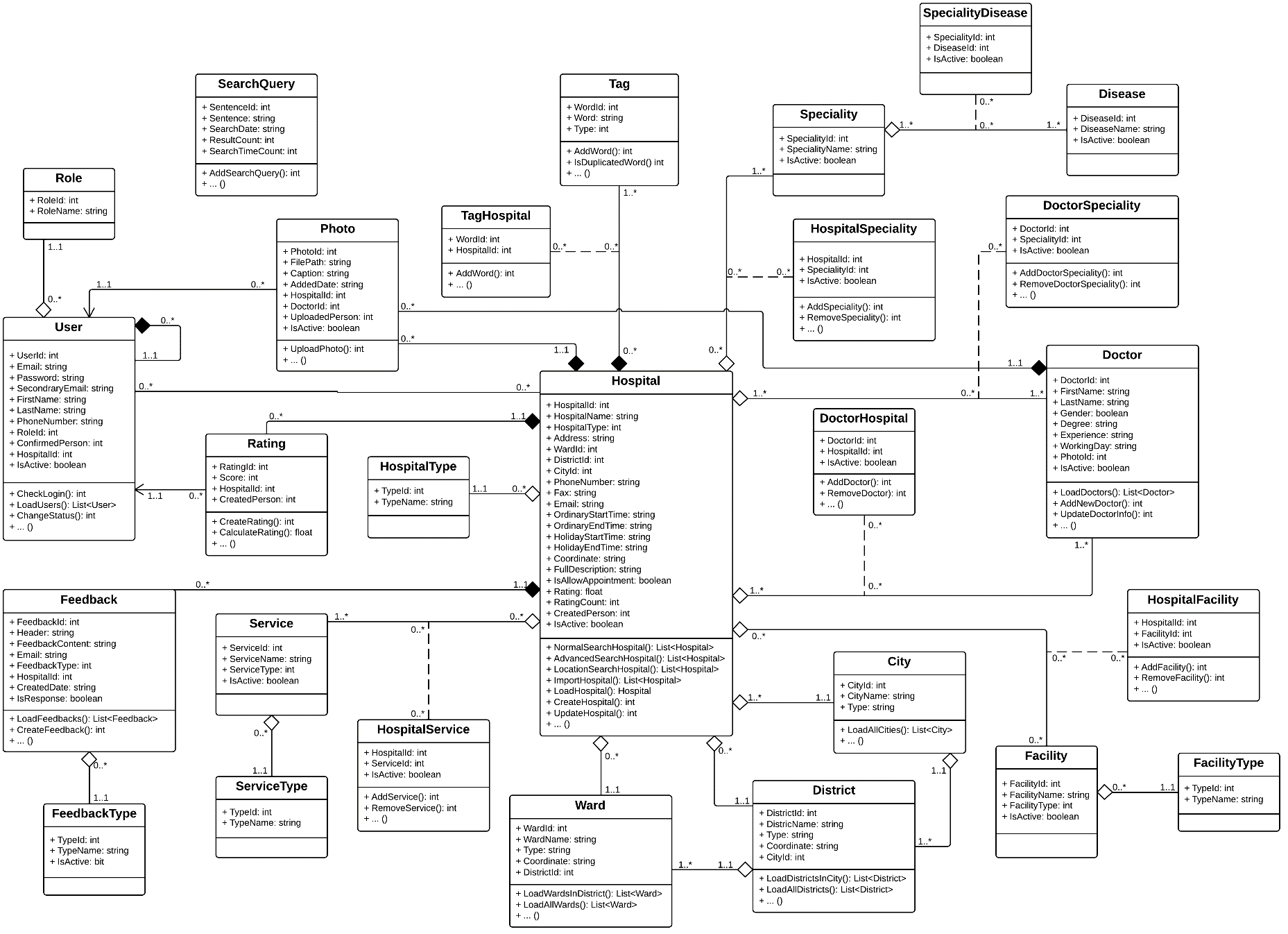


1. **Component Diagram**

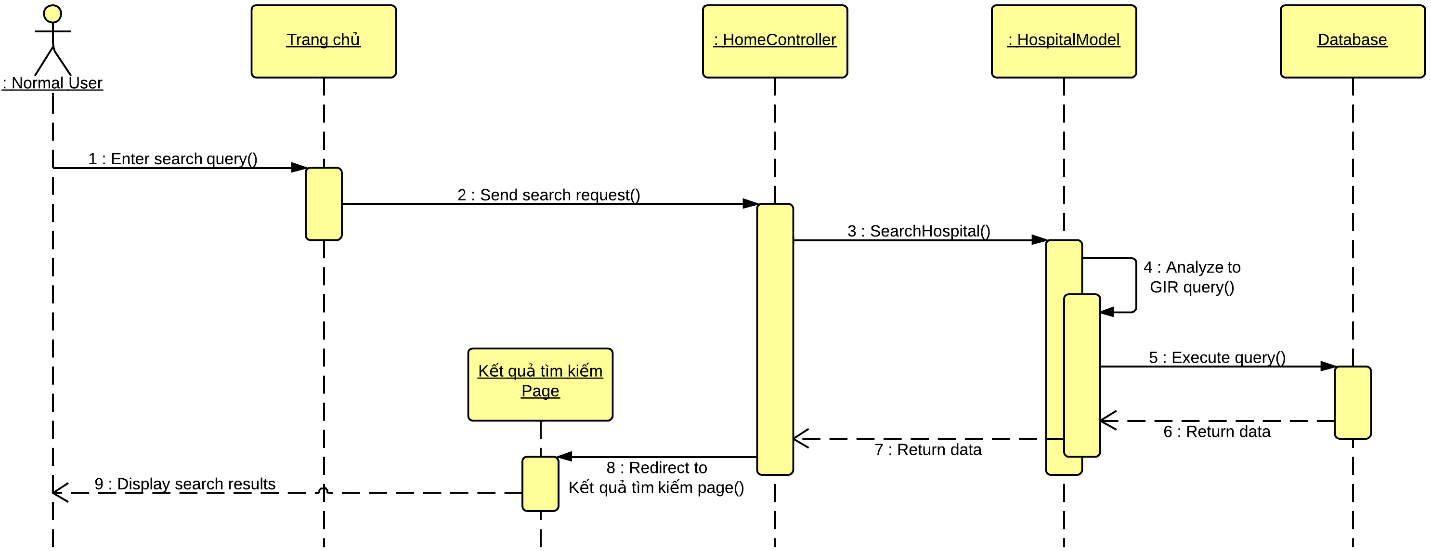


1. **Detailed Description of Components**

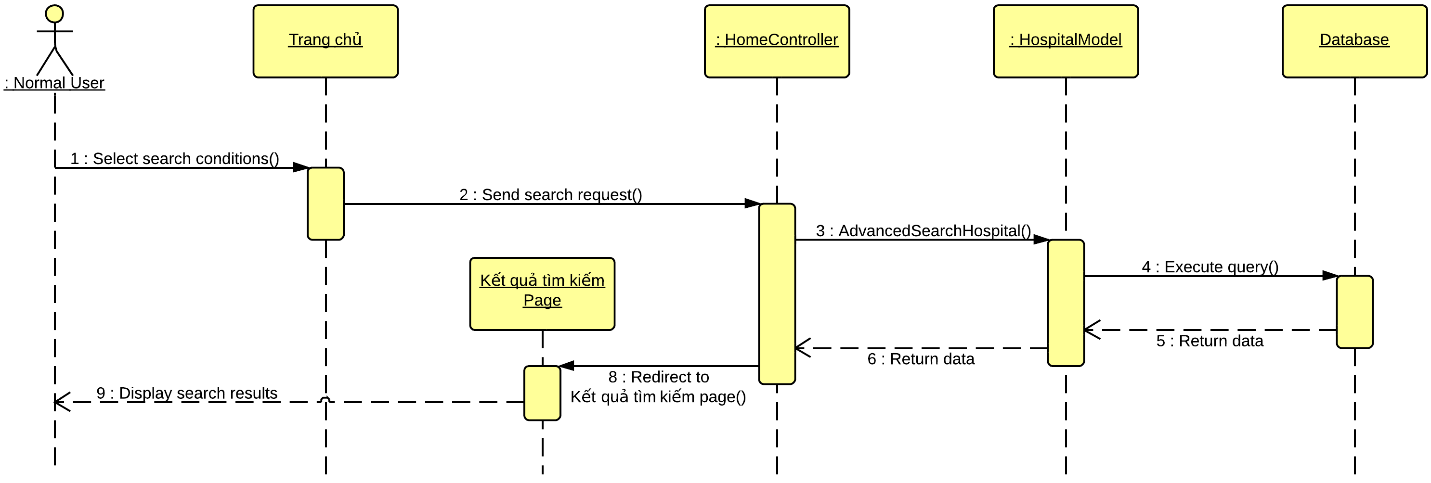
1. **Class Diagram**



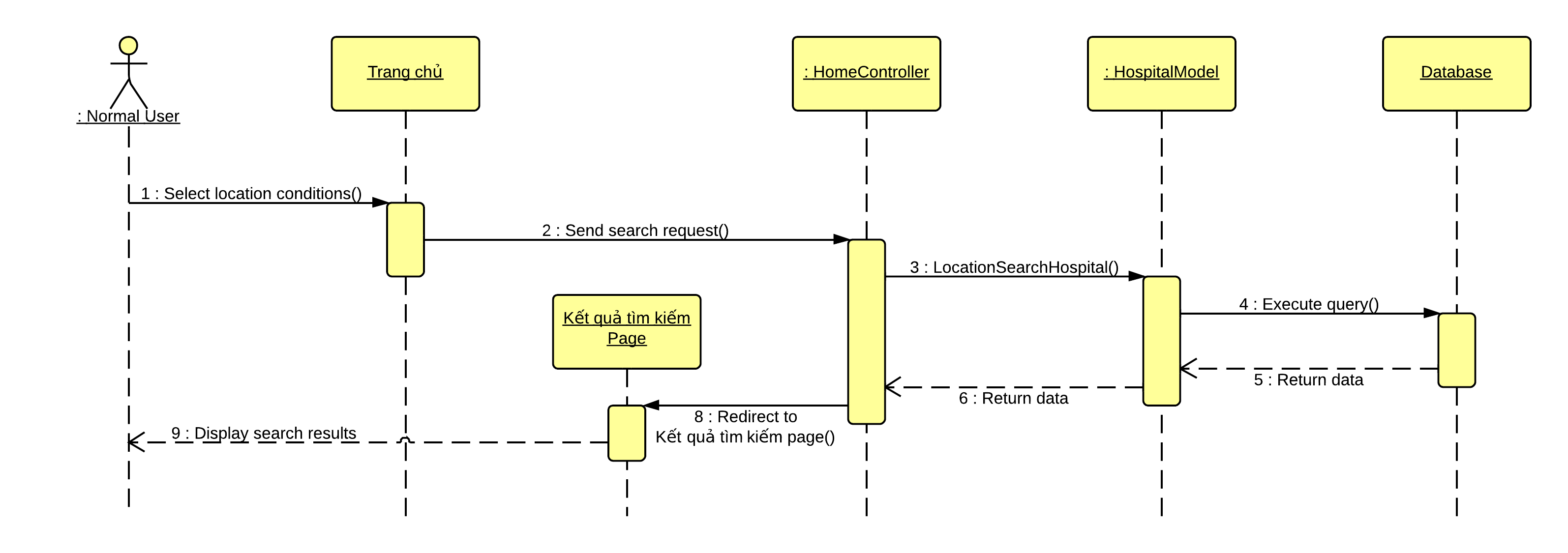
1. **Sequence Diagram**
   1. **Normal Search Hospital**



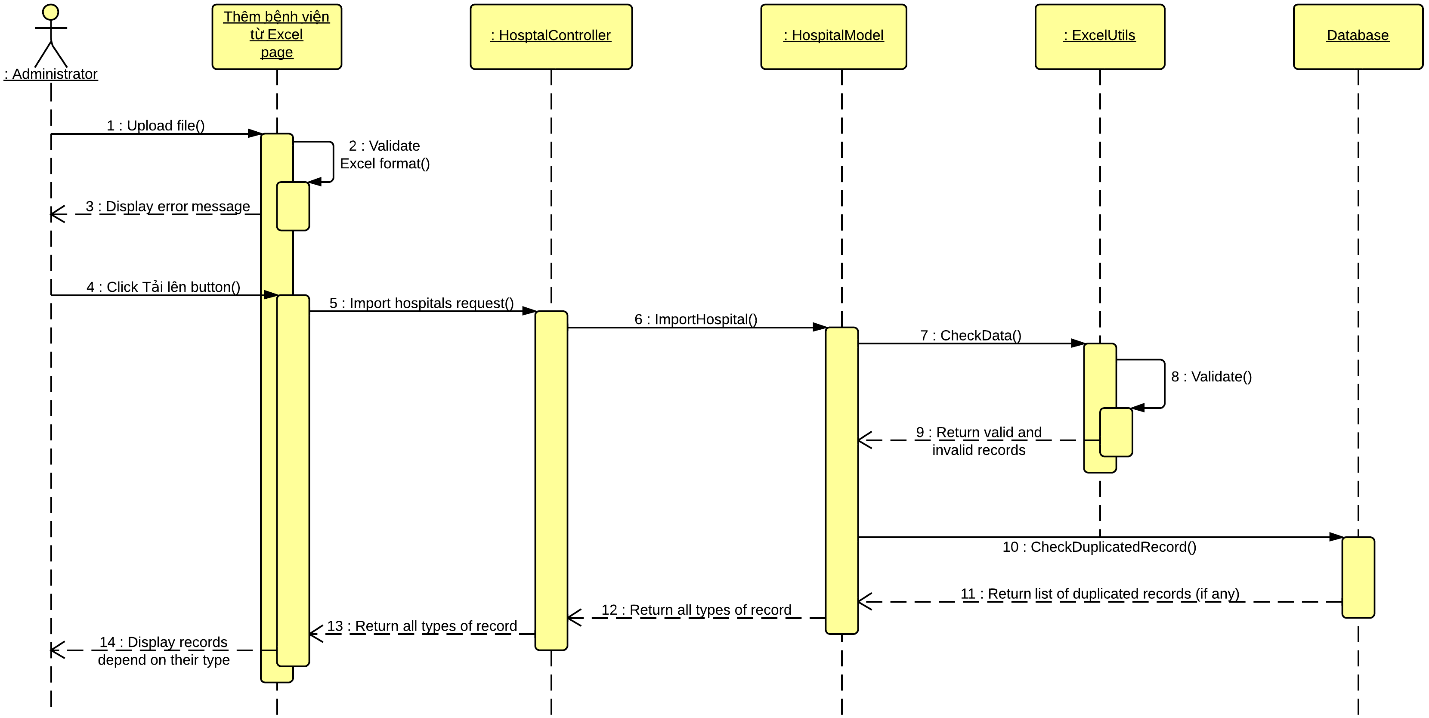
* 1. **Advanced Search Hospital**



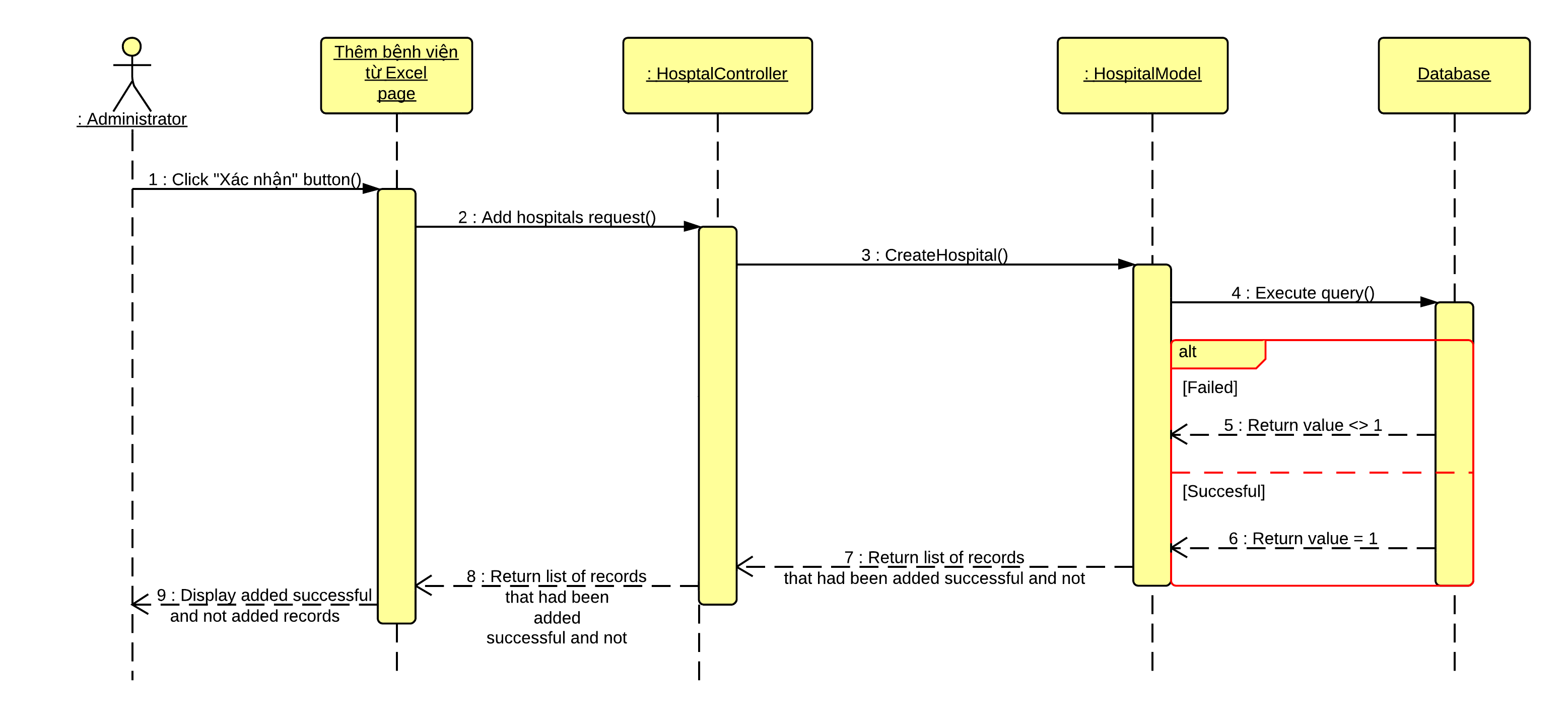
* 1. **Location Search Hospital**



* 1. **Import Hospital Using Excel**



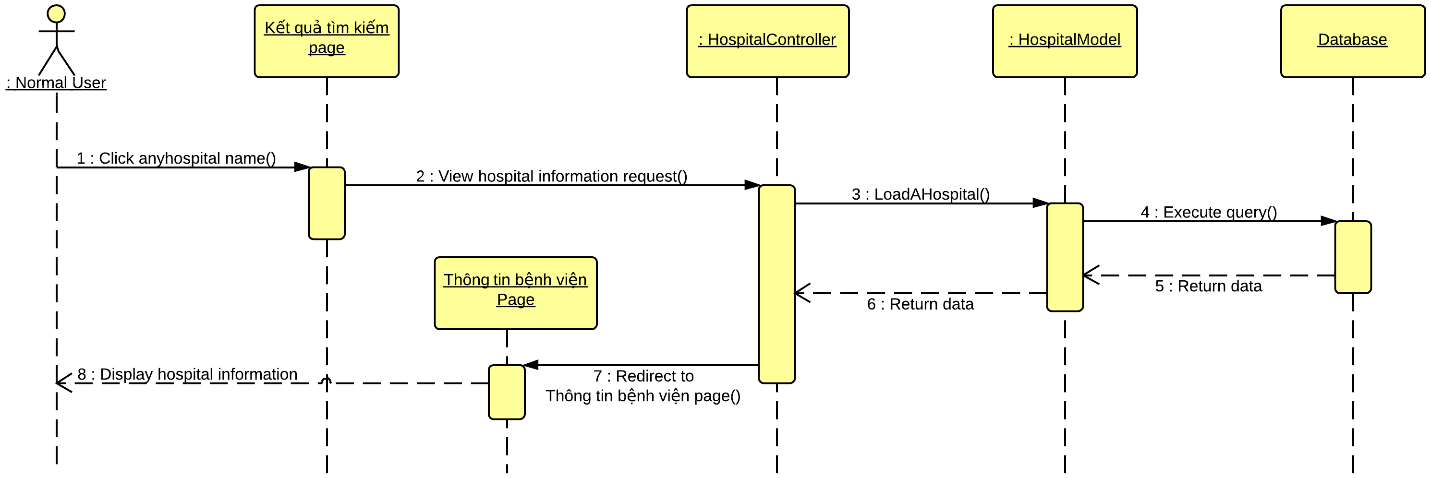
* 1. **Save valid Excel records**



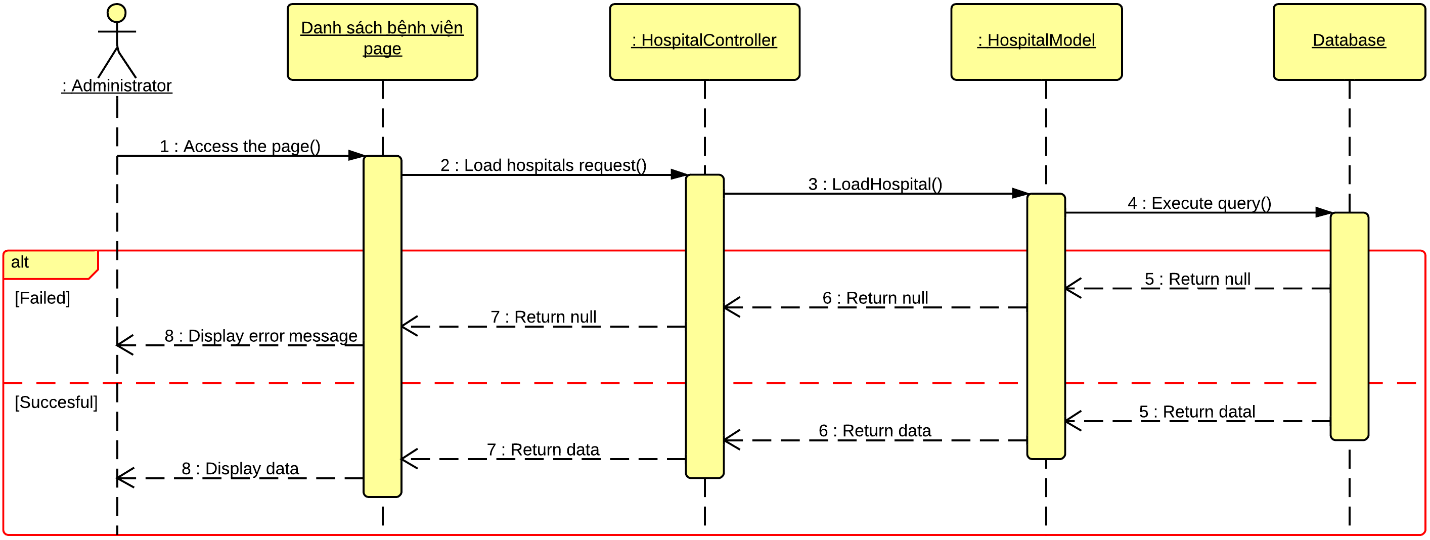
* 1. **Create Hospital**



* 1. **View Hospital Detail**



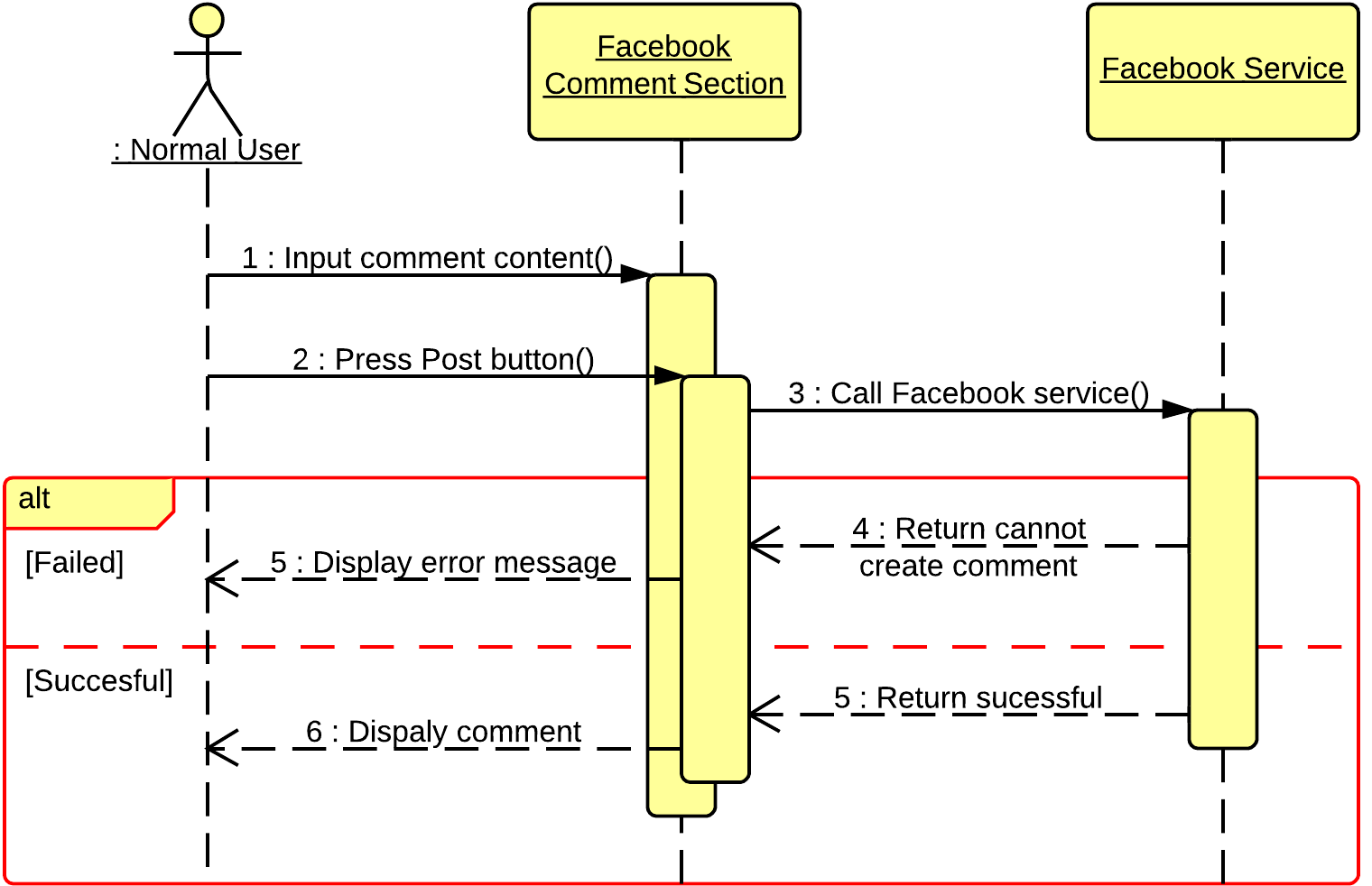
* 1. **Display list of hospitals**



* 1. **Update Hospital Information**



* 1. **Comment**



* 1. **View Statistics**



1. **Algorithms**
2. **Geographic Information Retrieval**
   1. **Definition**

Gegographic Information Retrieval (GIR) is a technique of analyzing and breaking down input query into pre-defined well-known information.

GIR is applied in many systems that support searching specific locations base on unknown or undefined input search query from users.

More specifically, in Hospital Finder system, GIR is used to break down input search query into 3 stages: **{What}, {Relation}, {Where}**. With:

* **{What}**: Contains what people want to search for, or what information people want to know through the search engine.
* **{Relation}**: Contains mostly preposition.
* **{Where}**: Contains information of specific address or locations. For example:
* In English: In, at, on, of, between, near, located in, ….
* In Vietnamese: Ở, tại, thì, của, tọa lạc, nằm trên, ….
  1. **Examples**

For more details, the following examples will break down sentences into 3 stages of {What}, {Relation} and {Where} using GIR algorithms:

***Example 1***: **Bệnh viện răng hàm mặt ở thành phố Hồ Chí Minh.**

* After running GIR algorithm, the sentence above will be analyzed into 3 stages with:
* **{What}**: Bệnh viện răng hàm mặt
* **{Relation}**: ở
* **{Where}**: thành phố Hồ Chí Minh

***Example 2***: **Bệnh viện răng hàm mặt thành phố Hồ Chí Minh.**

* After running GIR algorithm, the sentence above will be analyzed into 3 stages with:
* **{What}**: Bệnh viện răng hàm mặt
* **{Relation}**: N/A
* **{Where}**: thành phố Hồ Chí Minh

***Example 3***: **Bệnh viện răng hàm mặt.**

* After running GIR algorithm, the sentence above will be analyzed into 3 stages with:
* **{What}**: Bệnh viện răng hàm mặt
* **{Raltion}**: N/A
* **{Where}**: N/A

1. **Haversine Formula**
2. **Definition**

The haversine formula is an equation important in [navigation](http://en.wikipedia.org/wiki/Navigation), giving [great-circle distances](http://en.wikipedia.org/wiki/Great-circle_distance) between two points on a [sphere](http://en.wikipedia.org/wiki/Sphere) from their [longitudes](http://en.wikipedia.org/wiki/Longitude) and [latitudes](http://en.wikipedia.org/wiki/Latitude). It is a special case of a more general formula in [spherical trigonometry](http://en.wikipedia.org/wiki/Spherical_trigonometry), the law of haversines, relating the sides and angles of spherical triangles.

* **C** stands for Great Circle Distance, with **C** equal:
* **R** is the radius of the earth (6371km)
* **d** is the distance between the two points (Along a great circle of the sphere, see spherical distance), with **d** equal:
* : latitude of point 1 and latitude of point 2
* : longitude of point 1 and longitude of point 2

*For more detail, please refer to page:* [*http://en.wikipedia.org/wiki/Haversine\_formula*](http://en.wikipedia.org/wiki/Haversine_formula)

1. **Example**

* Give two points A(53,156; -1,8444), B(52,2047; 0,1406)
* The following code is executed using javascript.

*/\*\* Converts numeric degrees to radians \*/*

*if (typeof(Number.prototype.toRad) === "undefined") {*

*Number.prototype.toRad = function() {*

*return this \* Math.PI / 180;*

*}*

*}*

*var R = 6371; // km*

*var dLat = (lat2-lat1).toRad();*

*var dLon = (lon2-lon1).toRad();*

*var h = Math.sin(dLat/2) \* Math.sin(dLat/2) +*

*Math.cos(lat1.toRad())\* Math.cos(lat2.toRad()) \**

*Math.sin(dLon/2) \* Math.sin(dLon/2);*

*var c = 2 \* Math.atan2(Math.sqrt(h), Math.sqrt(1-h));*

*var d = R \* c;*

* Result: c = **105,8km**