**SOFTWARE REQUIREMENT SPECIFICATION**

**DOCUMENT**

**STUDENT MANAGEMENT**

**Version:** Version 1.0

**ABSTRACT**

This document is intended to be the SRS for developing **STUDENT MANAGEMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | **Student Management** | | |
| **Project Mentor** | Nguyen Quang Bao | | |
| **Contact Information** | Nguyenquangbao575@gmail.com | | |
| **Start Date** | 24/10/2025 | End Date |  |

Table of Contents

[Revision History 4](#_Toc212230899)

[1. Introduction 5](#_Toc212230900)

[1.1. Purpose 5](#_Toc212230901)

[1.2. Intended Audience and Reading Suggestions 5](#_Toc212230902)

[1.3. References 5](#_Toc212230903)

[2. Project Overview 6](#_Toc212230904)

[2.1. Project Description 6](#_Toc212230905)

[2.2. Business Problems 6](#_Toc212230906)

[2.3. Business Need 6](#_Toc212230907)

[2.4. Project Analyst 7](#_Toc212230908)

[2.4.1. Business Function Diagram 7](#_Toc212230909)

[2.4.2. System Context Diagram 7](#_Toc212230910)

[2.4.3. Data Flow Diagram 8](#_Toc212230911)

[2.5. Software Requirement Specification 10](#_Toc212230912)

[2.5.1. High level Fucntional Requirement (FR) 10](#_Toc212230913)

[2.5.2. Stakeholders 11](#_Toc212230914)

[2.5.3. Use case 11](#_Toc212230915)

[2.5.3.1. List of use cases 11](#_Toc212230916)

[2.5.3.2. <<Student Management System>> Use Case Diagram Overview 12](#_Toc212230917)

[2.5.4. List of business rule 12](#_Toc212230918)

[2.5.5. Use Case Specification 13](#_Toc212230919)

[*UC 01: Create a New Place* 13](#_Toc212230920)

[*UC 02: Edit a Place’s Information* 15](#_Toc212230921)

[2.5.6. Software Quality Attributes 18](#_Toc212230922)

[2.5.6.1. Usability 18](#_Toc212230923)

[2.5.6.2. Reliability 18](#_Toc212230924)

[2.5.6.3. Security 18](#_Toc212230925)

[2.5.6.4. Maintainability 19](#_Toc212230926)

[2.5.6.5. Portability 19](#_Toc212230927)

[2.5.6.6. Performance 19](#_Toc212230928)

[2.6. Software Design Description 20](#_Toc212230929)

[2.6.1. Sequence Diagrams 20](#_Toc212230930)

[2.6.1.1. Cập nhật thông tin nhân viên 20](#_Toc212230931)

[2.6.1.2. Quan hệ Gia đình 21](#_Toc212230932)

[2.6.2. Activity Diagrams 22](#_Toc212230933)

[2.6.2.1. Cập nhật thông tin nhân viên 22](#_Toc212230934)

[2.6.3. State Diagram 22](#_Toc212230935)

[2.6.4. Class Diagram 23](#_Toc212230936)

[2.6.5. Allocation Diagram 23](#_Toc212230937)

[Appendix A: Glossary 23](#_Toc212230938)

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Change Item | Description | by | Version |
| 24/10/2025 | Draft | Production, Project description, Business need, BFD, DFD level 0, DFD level 1 | Bao | 1.0 |
| 24/10/2025 | Update | Updated BFD, DFD level 0, 1; Added DFD level 2, High-level Functional Requirements, Stakeholders | Bao | 1.1 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

## Purpose

This document describes the **Student Management System (SMS)**, including all main functions for users, students, and administrators.  
The key features are listed below:

* **As a student (user):** can register for courses, view personal profile, and view the list of available courses.
* **As an administrator (admin):** can perform all CRUD (Create, Read, Update, Delete) operations on classes, users, and courses.

## Intended Audience and Reading Suggestions

|  |  |
| --- | --- |
| Intended Audience | Reading Suggestions |
| Administrator | Review overall system functions and management operations |
| Architect analyst and designer | Overall description and user cases to architect and design the system |
| Quality control | Overall description and user cases to make test plan and write acceptance test |
| Developer | Overall description, focus on Functional + Non-Functional Requirements to understand the process |

## References

|  |  |  |
| --- | --- | --- |
| No | References | Document Information |
| 1 | Nguyen Dang Quang Huy - Duy Tan university | Slide: *Requirements Engineering* |
| 2 | Nguyen Dang Quang Huy – Duy Tan university | SRS Template |

# Project Overview

## Project Description

This project is developed to assist administrators in managing all courses, users and students within Student Management System.

It also aims to optimize the course registration process and allow students to easily view their personal profiles and the list of available courses.

In summary, the system provides:

* Efficient management of courses, students, and users for administrators.
* Simplified course registration and personal information management for students.

## Business Problems

Currently, students find it difficult to register for courses, view their personal profiles, and access the list of available courses.

At the same time, administrators face challenges in managing a large number of courses, users, and students manually, which leads to inefficiency and errors.

Therefore, there is a need for a centralized system that can:

* Automate course registration and student management.
* Reduce administrative workload.
* Provide real-time access to information for both students and administrators.

## Business Need

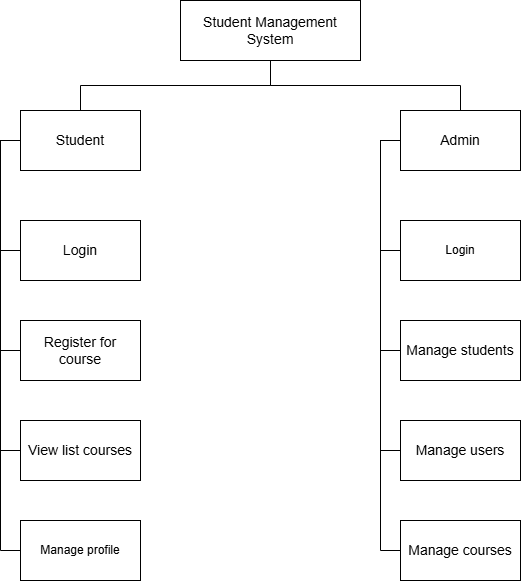
The Student Management System is needed to improve the efficiency and accuracy of managing academic information.  
Currently, most of the operations, such as course registration, user management, and student information updates, are performed manually.  
This manual process is time-consuming, prone to human errors, and lacks real-time data access.

By developing this system:

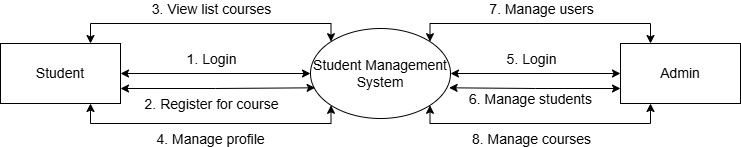
* Administrators can easily manage courses, users, and students through an integrated digital platform.
* Students can register for courses, view their profiles, and access course information conveniently.
* The system ensures data consistency, reduces workload, and improvesoverall productivity for educational institutions.

## Project Analyst

## Business Function Diagram



## System Context Diagram

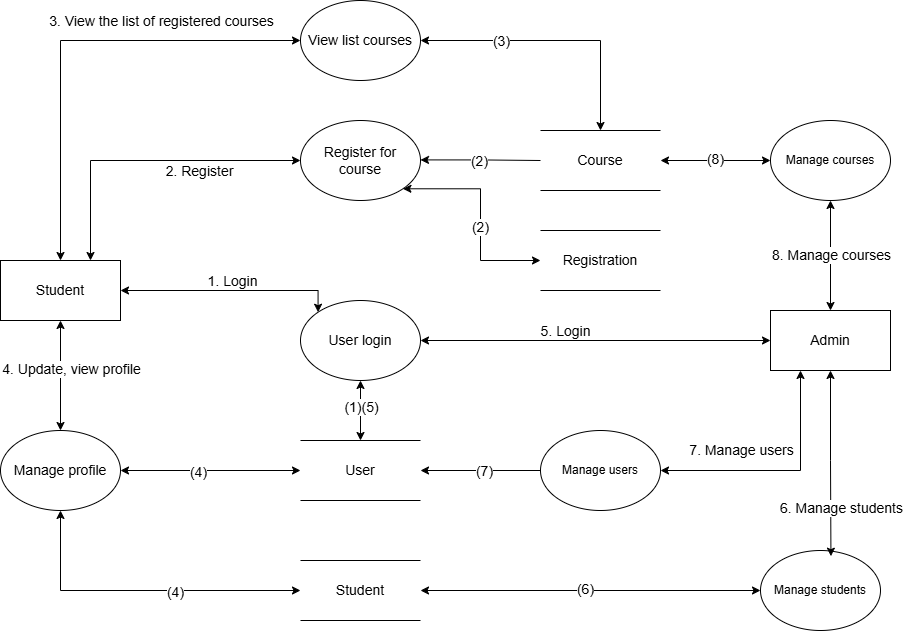


The “Student Management” system with key features that meet business requirements includes:

* For User (Student) role:
* Login to the system.
* Register for course
* View list courses
* View personal profile
* Update personal profile
* For Admin role:
  + Manage students (fully CRUD functional)
  + Manage users (fully CRUD functional)
  + Manage courses (fully CRUD functional)

## Data Flow Diagram

**DFD level 1**



**DFD level 2**

Student Role

A diagram of a register

AI-generated content may be incorrect.

Admin Role

A diagram of a flowchart

AI-generated content may be incorrect.

## Software Requirement Specification

## High level Fucntional Requirement (FR)

|  |  |  |
| --- | --- | --- |
| FR1 | Title | User Authentication |
| Stakeholder | Student / Admin |
| Description | The system allows users to register new accounts, log in, and log out securely. Registered users can access system features according to their assigned roles |
| FR2 | **Title** | **Enroll Course** |
| Stakeholder | Student |
| Description | The system allows students to enroll in available courses based on their interests or requirements |
| FR3 | **Title** | **View Course List** |
| Stakeholder | Student |
| Description | The system allows students to view the list of available and enrolled courses |
| FR4 | **Title** | **Manage Personal Profile** |
| Stakeholder | Student |
| Description | The system allows students to view and update their personal information |
| FR5 | **Title** | **Manage Students** |
|  | Stakeholder | Admin |
|  | Description | The system allows admins to add, update, delete, and view student records |
| FR6 | **Title** | **Manage Users** |
|  | Stakeholder | Admin |
|  | Description | The system allows admins to manage user accounts, including activating, deactivating, or changing permissions |
| FR7 | **Title** | **Manage Courses** |
|  | Stakeholder | Admin |
|  | Description | The system allows admins to create, modify, delete, and view course details |

## Stakeholders

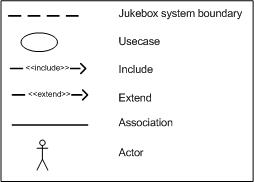
|  |  |
| --- | --- |
| Stakeholder | Description |
| Students | Users who use the Student Management System to register for courses, view enrolled courses, and manage their personal profiles |
| Admins | Users who are responsible for managing courses, students, and user accounts within the system |

## Use case

## List of use cases

|  |  |  |
| --- | --- | --- |
| Use case ID | Use case name | Functional Req. |
| UC.01 | Create new place | FR1.1 |
| UC.02 | Edit a Place’s Information | FR1.2 |
| UC.03 | Login | FR1.3 |

## <<Student Management System>> Use Case Diagram Overview



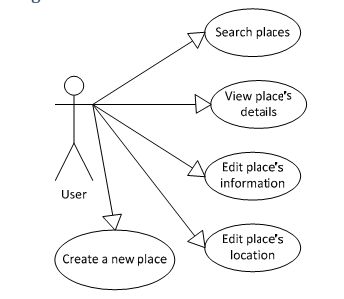
## List of business rule

|  |  |  |
| --- | --- | --- |
| Code | Business rules | Use case |
| BR1 | Rank of a user will be se based on the user’s  contribution point (CP) | UC.02 |
| BR2 | Contribution Point (CP) is calculated as follow:  ‐ Starter point for every user is 10 point  ‐ Create a place: + 20 point  ‐ Your place get deleted: ‐ 25 point | UC.02 |
| BR3 | For user has more than 1000 point (senior user):  ‐ Can edit a place directly  ‐ Can upload place’s image directly  ‐ Can approve suggested edit  ‐ Can approve suggested images  ‐ Set a revision as current | UC. 02 |
| BR4 | For user has less than 1000 point (junior user):  ‐ an revision of a place created by these users  need approvals from senior users or admins to be  displayed | UC.02 |

## Use Case Specification

### *UC 01: Create a New Place*

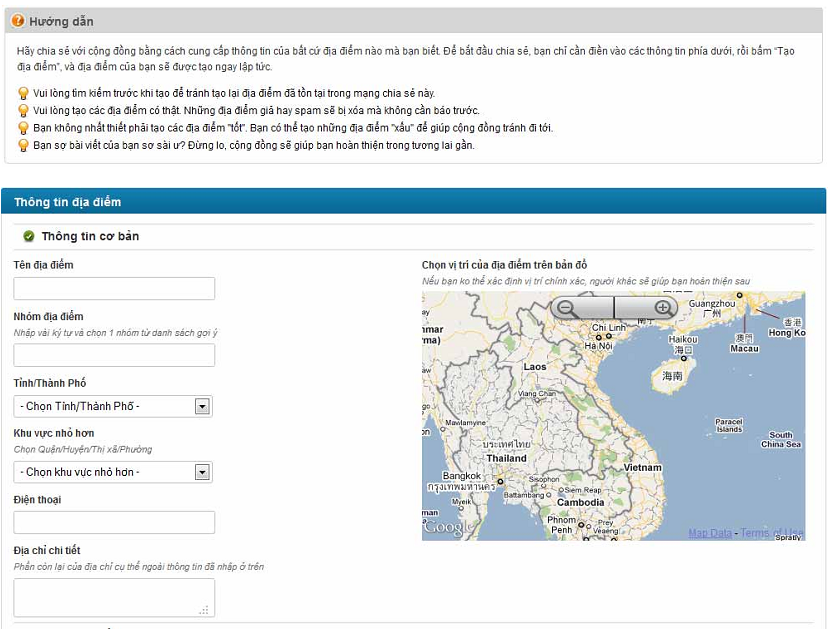
1. Use Case Diagram



1. Use Case Specìication

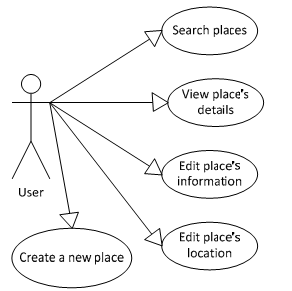
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | **UC1.01** | | | | |
| Use case name | **Create new place** | | | | |
| Create by | Hung Nguyen | | Last updated by | | Hung Nguyen |
| Date created | June 20, 2010 | | Date last updated | | July 01, 2010 |
| Actor | Engineer | | | | |
| Description | This use case allows the user to create a place in this system when the user knows an interesting place and wants to share it to everyone | | | | |
| Trigger | Click to “**T­ạo địa điểm**” in navigator at home page | | | | |
| Pre-condition | User logged into the system successfully. | | | | |
| Post-condition | System redirects user to the place detail page of the just‐created place. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | Click to “**T­ạo địa điểm**” in navigator at home page | | Redirect to Create Place page | |
| 2 | Fills all mandatory data fields | |  | |
| 3 | Click to “**T­ạo địa điểm**” button. | | Redirect to new place’s details page. | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1 | Click to “**T­ạo địa điểm**” in navigator at home page | | Redirect to Create Place page | |
| 2 | Fills all mandatory data fields | |  | |
| 3 | Click to “**Bỏ Qua**” button. | | Redirect to new place’s details page. | |
| Exceptions | **Step** | **Actor Action** | | **System Response** | |
| 1 | Click to “**T­ạo địa điểm**” in navigator at home page | | Redirect to Create Place page | |
| 2 | User fills some data fields. Some mandatory data fields are skipped. | |  | |
| 3 | Click to “**T­ạo địa điểm**” button. | | Shows all required messages for all missing mandatory data fields. | |
| Priority | High | | | | |
| Business rule | N/A | | | | |
| Description: | When user have a place to share, they only need click to “T­o địa điểm” menu on the navigator and get redirect to Create Place page to create a new place. In order to create a new place, user need enter place name, category, city/province, district/town/ward and detail address. Place phone number and description are  optional.   * Place name: required; must be less than 200 characters * Category: required; there is a suggest list displayed below as user typing place’s category in the text box. User must choose a category from the list. An unknown or new or invalid category won’t be accepted and a new place won’t be created * City/ province/ district/ town/ ward: user need choose from combo box list * Phone number: this number will be used to contact to place’s owner/manager * Detail Address: this is detail address of new place. It contains all parts left of place address except (city/ province/ district/ town/ ward) * Description: user can write review,comment, introduction, recommendation for new place * Place location: user can zoom in and out on the map and click to a location on the map to define place’s location   When user click to “**Tạo địa điểm**” button, the system will open Place Detail page of newly created place. By click to “B­ qua” button, the user will go back to previous page. | | | | |

1. Prototype



### *UC 02: Edit a Place’s Information*

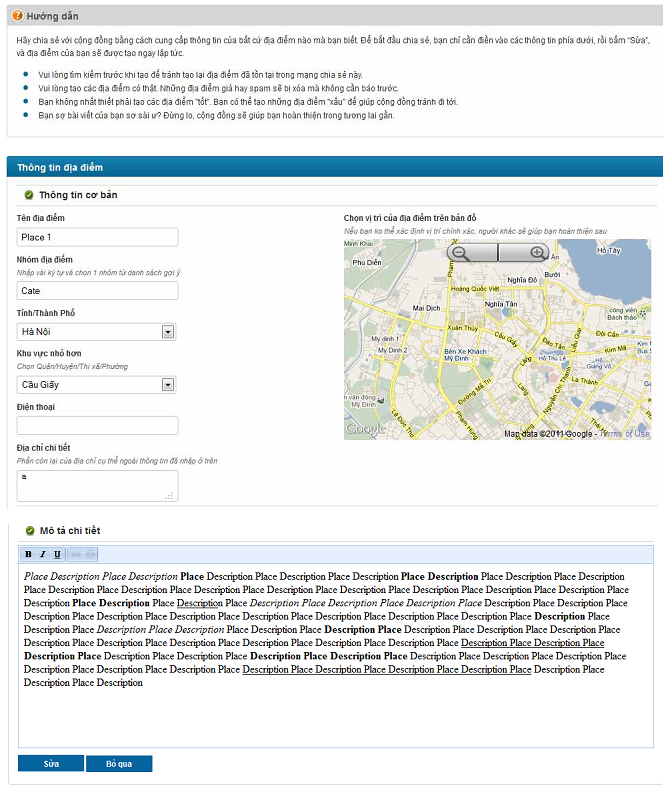
1. Use Case Diagram



1. Use Case Specìication

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | **UC.02** | | | | |
| Use case name | **Edit a Place’s Information** | | | | |
| Create by | Hung Nguyen | | Last updated by | | Hung Nguyen |
| Date created | June 20, 2010 | | Date last updated | | July 01, 2010 |
| Actor | Senior User (has more than 1000 Contribution Point) and administrator (View Business Rule for more detail) | | | | |
| Description | When some information of a place like telephone, title, address… is changed, the user with highest rank type or the admin can make it up‐to‐date by click to “**Chỉnh Sửa Thông Tin**” hyperlink in the place’s details page. | | | | |
| Goal | This use case allows the actor edit information of a place. | | | | |
| Trigger | Click to “**Chỉnh Sửa Thông Tin**” hyperlink in the place’s details page | | | | |
| Pre-condition | Logged into the system successfully and open a place’s detail page.. | | | | |
| Post-condition | Redirected back to Place Detail page | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User click to “**Chỉnh sửa thông tin**” hyperlink | | System displays the page to edit place | |
| 2 | User edit some data fields | |  | |
| 3 | User click to “**Sửa**” button | | System will redirect to the place’s details page with t e up‐to‐date data | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1 | User click to “**Chỉnh sửa thông tin**” hyperlink | | System displays the page to edit place | |
| 2 | User edit some data fields | |  | |
| 3 | User Click to “**Bỏ Qua**” button. | | System will redirect to the place’s details page and the data still is as before. | |
| Exceptions | **Step** | **Actor Action** | | **System Response** | |
| 1 | User click to “**Chỉnh sửa thông tin**” hyperlink | | System displays the page to edit place | |
| 2 | User delete or leave blank some mandatory data fields | |  | |
| 3 | User Click to “**Bỏ Qua**” button. | | System validates data in this page and shows all required messages for all missing mandatory data fie ds. | |
| Priority | High | | | | |
| Business rule | BR1, BR2, BR3 | | | | |
| Description: | Edit place is one of main focuses on the website. Sharing a place isn’t just simply by post a new place on the system. It is also about sharing information between users. And the most useful method to sharing is every user make all places in the system more detail, more precisely, increase its reliability, up‐to‐date… everything can be carry through editing places. But because editing can be contribution but also can be destruction, we can’t let every user has access to this function. So only user who has more than 1000 point can edit places.  When user has more than 1000 point, It means that user has taken quite a time using our website and he/she is trustable. The chance that user will take some destructive actions is rare.  For users with less than 1000 point, they still can go to Edit place page and edit information. But the changes they made will be taken as suggestions. These suggestions need approvals from verified users (>= 1000 point) or admin team.  To edit a place, user need go to its detail page first. After that, click to “Ch­nh s­a thông tin” hyperlink will redirect you to Edit Place page. At Edit place page, click to “S­a” button to save all changes or click to “B­ qua” button to go back to Detail page. All validations still take effect as creating a new place. | | | | |

1. Prototype



## Software Quality Attributes

## Usability

**GUI**

- All the text, image text and help documents should be in Vietnamese.

- The interface should be elegant, simple and out‐standing.

- All images must also provide alt attribute.

**Usability for end‐users**

- Searching tool should be easy to use.

- The website must provide a help page to support novice users.

- Users can use main functionality of the system without logging in to the system.

- The system should remember information (but not confidential) that users have to

provide regularly. For example: name, address…

**Usability for admin and staff**

- Website admin and shop’s staff should need no more than one day of training to be

productive with the system.

- Detailed help must be available for the admin and staff, both in web pages and separate

documentations.

**Installation**

- The system must be easy to deploy. Customer can deploy successfully and learn to

configure, maintain the system within one day of training.

- The documentation for installation must be included. It describes detailed steps for

installing or deploying the system. The customer can follow the steps without direct help from

the developing team.

## Reliability

- Initial data must be collected carefully and correctly

- The database must be backed up regularly and can be recovered if necessary

- Have a good spam filter

- Rating system must be reliable

- Not conflict with other software

- Rate of fault occurrence (ROFO): a number of 0.005 is acceptable. It means that it is

acceptable to have 5 failures happen in each 1000 operational time units (e.g. 5 failures per

1000 hours of operation).

- Mean Time Between Failures (MTBF): 1 month

- Mean Time To Repair (MTTR): immediately when admin finds out problem or website is attacked by someone. Average 1 day.

- Accuracy: precision of floating number should be rounded to 0.01. All currency units must be displayed (e.g. $, VND…)

- Maximum Bugs or Defect Rate: 5 bugs / KLOC.

- Bugs or Defect Rate

o Minor bugs: bugs related to GUI

o Significant bugs: bugs related to minor business logic

o Critical bugs: function can’t execute correctly, completely loss of data,

disconnect to the server

## Security

- Privacy: the system should provide protection method for protecting user information from

outside or from other users. All the information of users must not be available for anyone or

software that is not part of the system. User password is also invisible for the system

administrator.

- The system must provide methods to prevent common security attacks. E.g. SQL injection,

Ddos,…

- Transformation of confidential information must be encrypted.

- The system must provide secure methods for users to recovery their password, including

the interference of system’s admin.

- Secure information of customers should not be stored on customer’s machine.

- Utilize certain cryptographic techniques for database

- Provide options for users to choose whether to share private information

- Restrict communications between some areas of the program

- Check data integrity for critical variables

- Must use secure connection (SSL) for transferring sensitive data

## Maintainability

**Coding standards and naming conventions**

- Output of the project must include coding standards and naming conventions

documentations. Implementation code must be easy to maintain.

- All code must be clearly commented, including class, method documentations.

- If some components are reused, the documentations of those components must also be

included.

**Design**

- The design of the system must be loosely coupled that chances on some module will not

affect others.

**Logging**

- All the errors should be logged, supporting for bug fixing and maintenance.

- All strange or sensitive situations should also be logged.

**Email**

- The system must provide an email address to receive customer feedback or send emails.

## Portability

Website can be used by any people with a browser and an Internet connection..

## Performance

**Load time**

- Every page should be completely displayed within: average 1.5 second and maximum 5

seconds.

- Searching should return and display the result within 2 seconds.

- Comments and rating and the like should affect immediately, without reloading the web

page.

- Frequently accessed data must be cached

- Reference data must be cached

**Mailing system**

- Mail server should send emails within 1 minute after associated events happen.

**Capacity**

- The system should serve correctly and reasonably with at least 1000 online users at a

moment.

- The system can store at least 100000 places and 10000 users without affecting the loading

speed.

- If the system is busy, it has to inform users about that.

**Compatibility**

- The system should provide elegant responses for nowadays common screen resolutions:

1024\*768, 1280\*800.

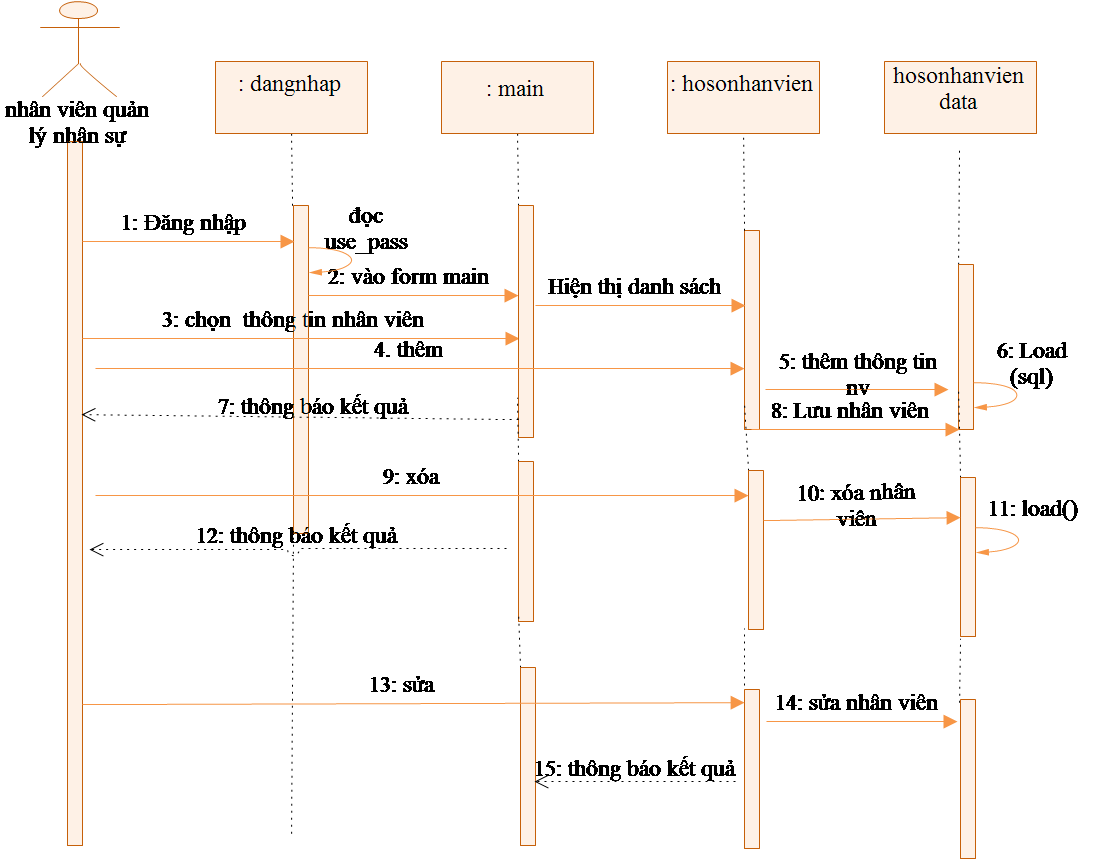
- The output must be compatible with all common browsers: Mozilla Firefox, Internet

Explorer, Google Chrome, Apple Safari, and Opera.

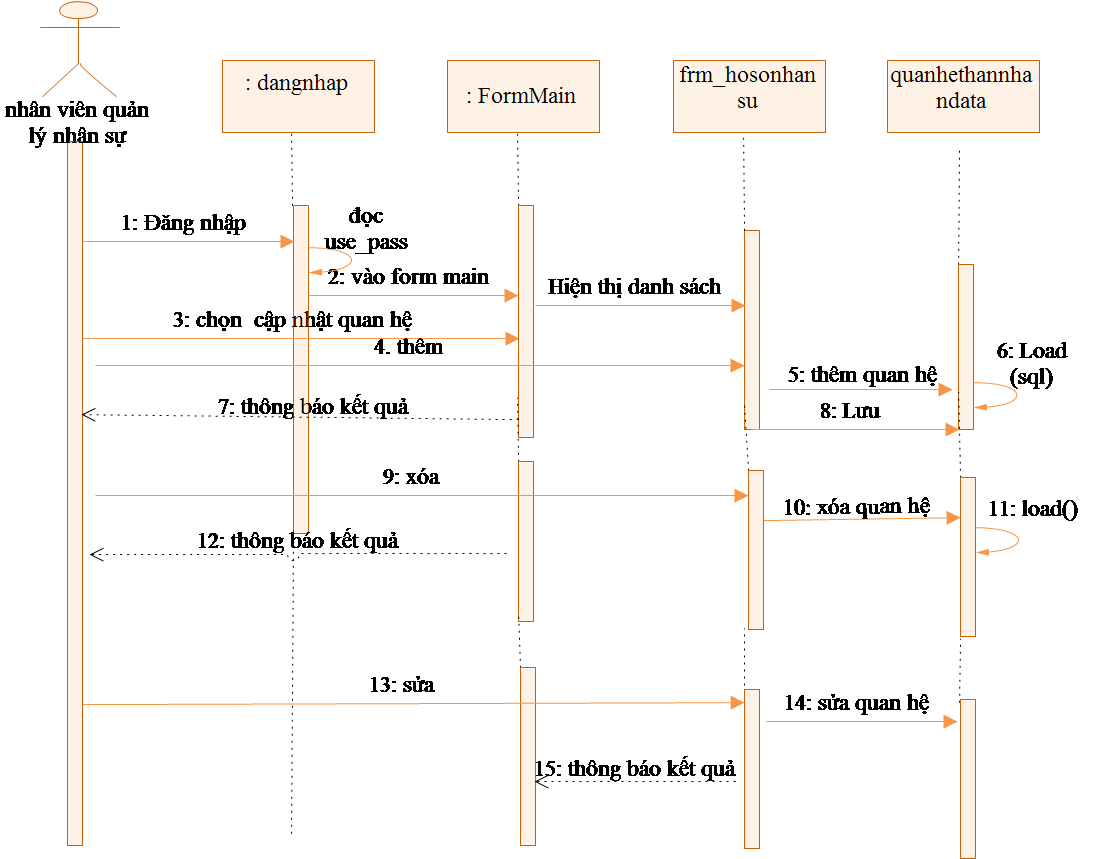
## Software Design Description

## Sequence Diagrams

## Cập nhật thông tin nhân viên

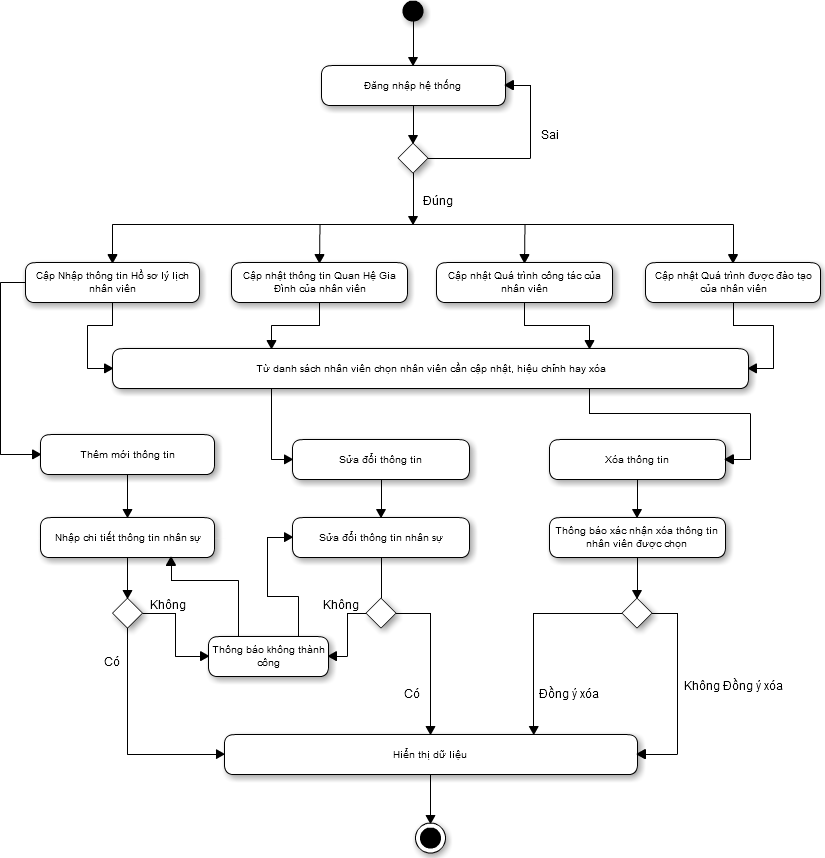


## Quan hệ Gia đình



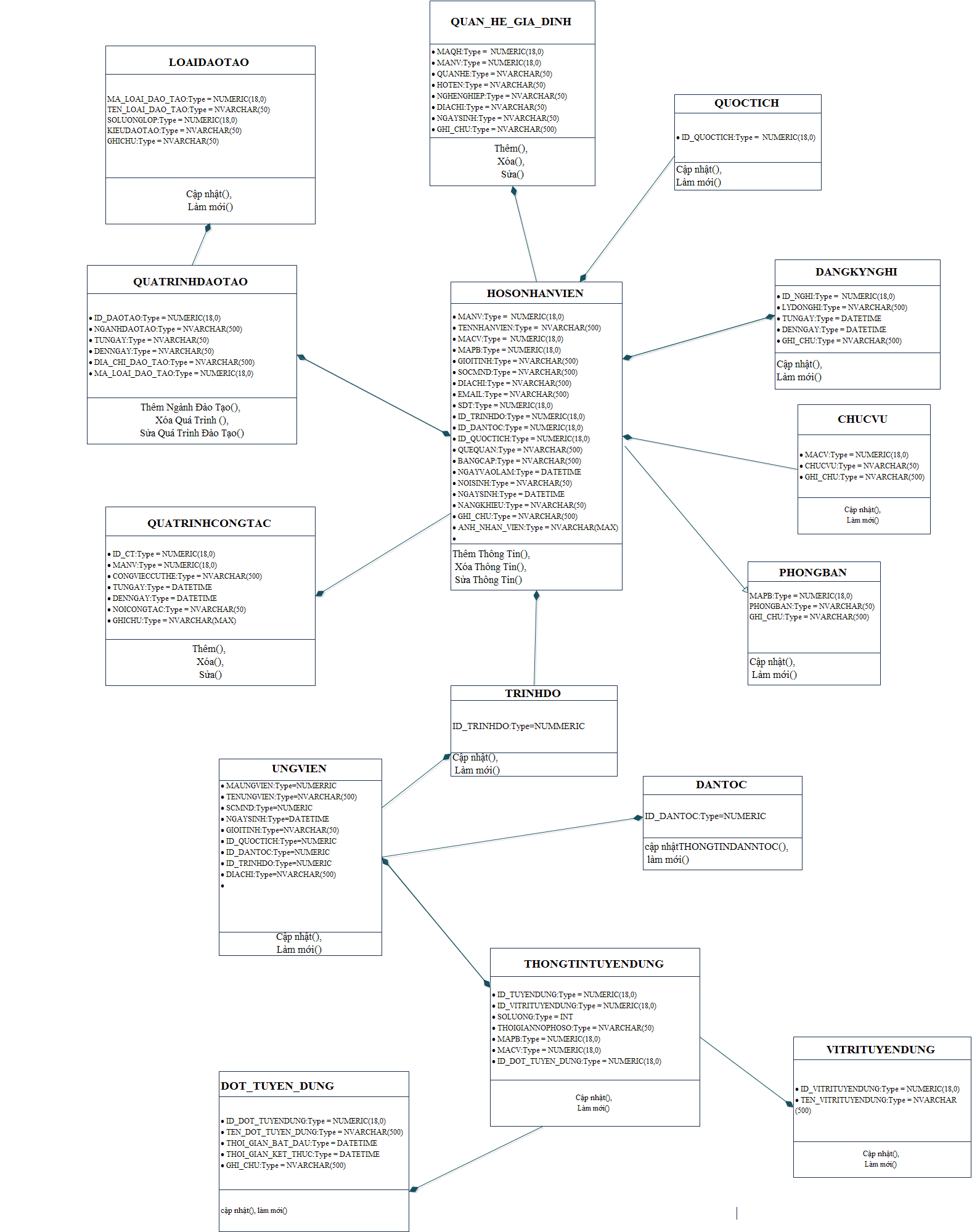
## Activity Diagrams

## Cập nhật thông tin nhân viên



## State Diagram

## Class Diagram



## Allocation Diagram

# Appendix A: Glossary

|  |  |
| --- | --- |
| BC | Business Constraint |
| TC | Technical Constraint |
| FR | Functional Requirement |
| QA | Quality Attribute |
| UC | Use case |
| BR | Business rule |