**SOFTWARE REQUIREMENT SPECIFICATION**

**DOCUMENT**

**Version:** Version 1.4

**ABSTRACT**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | **Course management** | | |
| **Project Mentor** | **Nguyen Quang Bao** | | |
| **Start Date** | 04/04/2025 | **End Date** |  |

Table of Contents

[Revision History 4](#_Toc195990090)

[1. Introduction 5](#_Toc195990091)

[1.1. Purpose 5](#_Toc195990092)

[1.2. Intended Audience and Reading Suggestions 5](#_Toc195990093)

[1.3. References 5](#_Toc195990094)

[2. Project Overview 6](#_Toc195990095)

[2.1. Project Description 6](#_Toc195990096)

[2.2. Business Problems 6](#_Toc195990097)

[2.3. Business Need 6](#_Toc195990098)

[2.4. Project Analyst 6](#_Toc195990099)

[2.4.1. Business Function Diagram 6](#_Toc195990100)

[2.4.2. System Context Diagram 7](#_Toc195990101)

[2.4.3. Data Flow Diagram 7](#_Toc195990102)

[2.5. Software Requirement Specification 8](#_Toc195990103)

[2.5.1. High level Functional Requirement (FR) 8](#_Toc195990104)

[2.5.2. Stakeholders 9](#_Toc195990105)

[2.5.3. Use case 9](#_Toc195990106)

[2.5.3.1. List of use cases 9](#_Toc195990107)

[2.5.3.2. <<Course Management>> Use Case Diagram Overview 10](#_Toc195990108)

[2.5.4. Use Case Specification 10](#_Toc195990109)

[*UC 01: Create a New Place* 10](#_Toc195990110)

[*UC 02: Edit a Place’s Information* 11](#_Toc195990111)

[2.5.5. Software Quality Attributes 14](#_Toc195990112)

[2.5.5.1. Usability 14](#_Toc195990113)

[2.5.5.2. Reliability 15](#_Toc195990114)

[2.5.5.3. Security 15](#_Toc195990115)

[2.5.5.4. Maintainability 16](#_Toc195990116)

[2.5.5.5. Portability 16](#_Toc195990117)

[2.5.5.6. Performance 16](#_Toc195990118)

[2.6. Software Design Description 17](#_Toc195990119)

[2.6.1. Sequence Diagrams 17](#_Toc195990120)

[2.6.1.1. Cập nhật thông tin nhân viên 17](#_Toc195990121)

[2.6.1.2. Quan hệ Gia đình 18](#_Toc195990122)

[2.6.2. Activity Diagrams 19](#_Toc195990123)

[2.6.2.1. Cập nhật thông tin nhân viên 19](#_Toc195990124)

[2.6.3. State Diagram 19](#_Toc195990125)

[2.6.4. Class Diagram 20](#_Toc195990126)

[2.6.5. Allocation Diagram 20](#_Toc195990127)

[Appendix A: Glossary 20](#_Toc195990128)

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Change Iterm | Description | by | Version |
| 16/04/2025 | Draft |  | Bao | 1.0 |
| 18/04/2025 | Update BFD, DFD level\_0, DFD level\_1 |  | Bao | 1.1 |
| 19/04/2025 | Update DFD level\_1, DFD level\_2 | Adjust DFD level\_1 | Bao | 1.2 |
| 19/04/2025 | Update Use Case Diagram, Use Case Specification | Create function | Bao | 1.3 |
| 21/04/2025 | Update Use Case Specification | Read function | Bao | 1.4 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

## Purpose

This documentation describes a course management application including all the necessary information and feature materials in detail for implementation. It helps administrators to track, update, delete and create new courses.

## Intended Audience and Reading Suggestions

|  |  |
| --- | --- |
| Intended Audience | Reading Suggestions |
| Architect analyst and designer | Overall description and user cases to architect and design the system |

## References

|  |  |  |
| --- | --- | --- |
| No. | References | Document Information |
| 1 | Jeremy Dick. Elizabeth Hull, Ken Jackson | “*Requirements Engineering*” – Academic book |
| 2 | Nguyen Dang Quang Huy - Duy Tan university | Slide: *Requirements Engineering* |
| 3 | Nguyen Dang Quang Huy – Duy Tan university | SRS Template |
| 4 | W3Schools | https://www.w3schools.com/cs/index.php |
| 5 | TITV | *SQL server Database Connection* (Video series) |
| 6 | K Team | *WinForms C# tutorial* (Video series) |

# Project Overview

## Project Description

This project is created to help administrators manage all courses like tracking, updating, deleting and creating new courses.

## Business Problems

The administrator finds it very difficult to manage all the courses in the system, including:

* Lack of real-time course visibility
* Manual management of courses leading to operational bottlenecks

## Business Need

The administrator needs a unified, easy-to-use management system that tracks all course information:

* Ensuring seamless
* Provides a visual overview of the entire course management system

## Project Analyst

## Business Function Diagram

A diagram of course management

AI-generated content may be incorrect.

## System Context Diagram

A list of courses with text

AI-generated content may be incorrect.

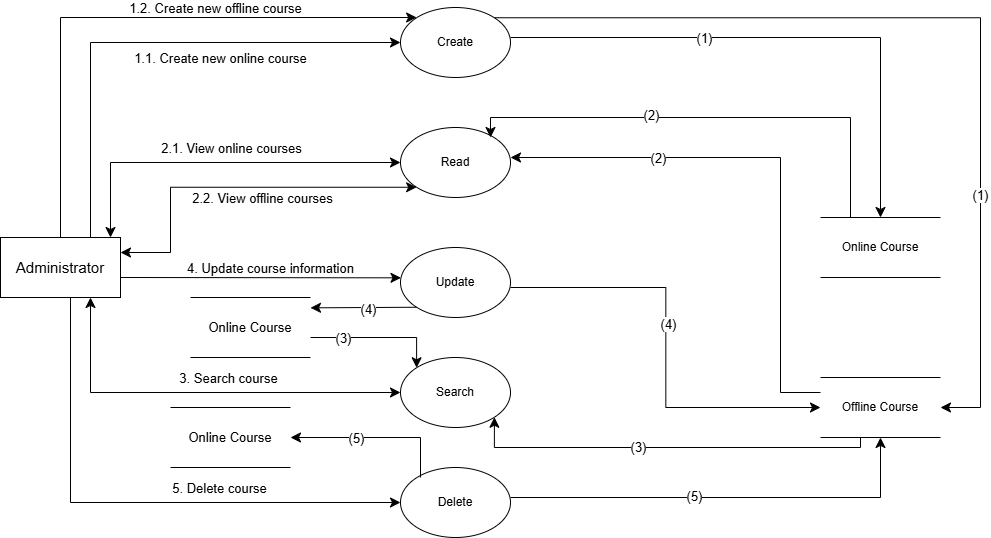
## Data Flow Diagram

* Level 1:

A diagram of course management

AI-generated content may be incorrect.

* Level 2:



## Software Requirement Specification

## High level Functional Requirement (FR)

|  |  |  |
| --- | --- | --- |
| FR1.1 | Title | Create new course |
| Stakeholder | Administrator who uses this application to create some new courses |
| Description | The system helps administrators easily create new online or offline course |
| FR1.2 | **Title** | **View course** |
| Stakeholder | Administrator who uses this application to keep track of all courses |
| Description | The system displays all online and offline courses. So, the administrator will have a visual overview of the courses and easily track it |
| FR1.3 | **Title** | **Search course** |
|  | Stakeholder | Administrator who uses this application to search for course information |
|  | Description | The administrator just needs to enter the course ID to search for course information. And the system will display the correct course that the administrator is looking for |
| FR1.4 | **Title** | **Update course information** |
|  | Stakeholder | Administrator who uses this application to adjust some course information and update it |
|  | Description | The administrator just needs to enter the course ID to search for course information. Then, the system will display the correct course. The administrator enters the necessary information to adjust and then presses the update button to update this information into the database |
| FR1.5 | **Title** | **Delete course** |
|  | Stakeholder | Administrator who uses this application to delete course |
|  | Description | The administrator just needs to enter the course ID to search for course information. Then, the system will display the correct course. The administrator presses the delete button to remove that course from the system |

## Stakeholders

|  |  |
| --- | --- |
| Stakeholder | Description |
| Administrator | People who use course management systems to track and manage all courses |

## Use case

## List of use cases

|  |  |  |
| --- | --- | --- |
| Use case ID | Use case name | Functional Req. |
| UC.01 | Create new course | FR1.1 |
| UC.02 | View course | FR1.2 |
| UC.03 | Search course | FR1.3 |
| UC.04 | Update course information | FR1.4 |
| UC.05 | Delete course | FR1.5 |

## <<Course Management>> Use Case Diagram Overview

A diagram of a course

AI-generated content may be incorrect.

## Use Case Specification

### *UC 01: Create new course*

1. Use Case Specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.01 | | | | |
| **Use Case Name** | Create new course | | | | |
| **Created by** | Bao Nguyen | | **Last updated by** | | Bao Nguyen |
| **Date Created** | Apr 19, 2025 | | **Date last updated** | | Apr 05, 2024 |
| **Actors** | Administrator | | | | |
| **Brief Description** | When the administrator wants to create new online or offline course, simply enter the information, click on the “**Online**” or “**Offline**” radio button, and then press the “**Create**” button to create a new course. | | | | |
| **Goal** | This Use Case allows the actor to create a new course | | | | |
| **Trigger** | User Click to “**Online**” or “**Offline**” radio button, and then press the “**Create**” button. | | | | |
| **Pre-conditions** | Choose “**Create**” from menu strip | | | | |
| **Post-conditions** | Show course creation interface | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course creation interface | |
| **2** | Users enter the information | |  | |
| **3** | Users click to the “**Online**” or “**Offline**” radio button, | | The system will allow user to enter correct information according to the selected course type | |
| **4** | Users press the “**Create**” button. | | The system will save the data to the database | |
| **5** |  | | Display message: “**Created successfully**” | |
| **Alternative Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course creation interface | |
| **2** | Users enter the information | |  | |
| **3** | Users click to the “**Online**” or “**Offline**” radio button | | The system will allow user to enter correct information according to the selected course type | |
| **4** | Users press the “**Reset**” button | | All entered information will be reset | |
| **5** | Users enter the information | |  | |
| **6** | Users click to the “**Online**” or “**Offline**” radio button | | Continue working (return to step 3 in main flow) | |
| **Exception Flow** | **Exception Flow 1: System loses connection to database** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course creation interface | |
| **2** | Users enter the information | |  | |
| **3** | Users click to the “**Online**” or “**Offline**” radio button | | The system will allow user to enter correct information according to the selected course type | |
| **4** | Users press the “**Create**” button. | | The system tries to save the data to the database but loses connection | |
| **5** |  | | Display message: “**Error connecting to server. Please try again later**.” | |
| **Exception Flow 2: Incorrect data type entered** | | | | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course creation interface | |
| **2** | Users enter the information | |  | |
| **3** | Users click to the “**Online**” or “**Offline**” radio button | | The system will allow user to enter correct information according to the selected course type | |
| **4** | Users press the “**Create**” button. | | The system recognizes that there is an incorrect data type entered and displays the message: "**There is incorrect data entered. Please re-enter**" | |
| **Priority** | High | | | | |
| **Business Rule** | * Users must select the course type (Online or Offline) before the system allows creating a course. * The system must check the input data before saving. If the data is invalid (incorrect format, missing information), the system will display an error message and not save the data * Data is only saved when there is a database connection. In case of disconnection, the system must report an error and not save the data. * When the “Reset” button is pressed, all entered information should be cleared and the user can re-enter information. | | | | |

1. Prototype

### *UC 02: View course*

1. Use Case Specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.02 | | | | |
| **Use Case Name** | View course | | | | |
| **Created by** | Bao Nguyen | | **Last updated by** | | Bao Nguyen |
| **Date Created** | Apr 21, 2025 | | **Date last updated** | | Apr 21, 2025 |
| **Actors** | Admininistrator | | | | |
| **Brief Description** | When the administrator wants to see all courses, simply select the “**Online**” or “**Offline**” radio button, and then the system will display accurate information according to the selected course type | | | | |
| **Goal** | This Use Case allows the actor to see the online or offline courses | | | | |
| **Trigger** | User Click to “**Online**” or “**Offline**” radio button | | | | |
| **Pre-conditions** | Choose “**Read**” from menu strip | | | | |
| **Post-conditions** | Show course reading interface | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course reading interface | |
| **2** | Users click to the “**Online**” or “**Offline**” radio button | | System will send a requirement and extract data from the database | |
| **3** |  | | Course information will be displayed on the Data Grid View table | |
| **Alternative Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course reading interface | |
| **2** | Users enter the course information (such as ID, name, duration…) | |  | |
| **3** | Users press the “**Search**” button | | System will send a requirement and extract data from the database | |
| **4** |  | | Course information will be displayed on the Data Grid View table | |
| **Exception Flow** | **Exception Flow 1: System loses connection to database** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course reading interface | |
| **2** | Users click to the “**Online**” or “**Offline**” radio button | | System will send a requirement and extract data from the database | |
| **3** |  | | System tries to extract the data to the database but loses connection | |
| **4** |  | | Display message: “**Error connecting to server. Please try again later**.” | |
| **Exception Flow 2: Course don’t exist** | | | | |
| **1** | Users Choose “**Create**” from menu strip | | System displays the course reading interface | |
| **2** | Users enter the course information (such as ID, name, duration…) | |  | |
| **3** | Users press the “**Search**” button | | System will send a requirement and extract data from the database | |
| **4** |  | | System cannot find the course and displays message: “**Course does not exist, please re-enter**” | |
| **Priority** | High | | | | |
| **Business Rule** | * Users must select the course type (Online or Offline) for the system to display accurate course information. * Enter correct course information if users want to use search function to search. * Course information is only displayed when there is a database connection. In case of disconnection, the system must report an error and not save the data. * If users enter incorrect course information, the system must display a message reminding the user to enter correct information. | | | | |

1. Prototype

### *UC 03: Search course*

1. Use Case Specification
2. Prototype

### *UC 04: Update course information*

1. Use Case Specification
2. Prototype

### *UC 05: Delete course*

1. Use Case Specification
2. 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.

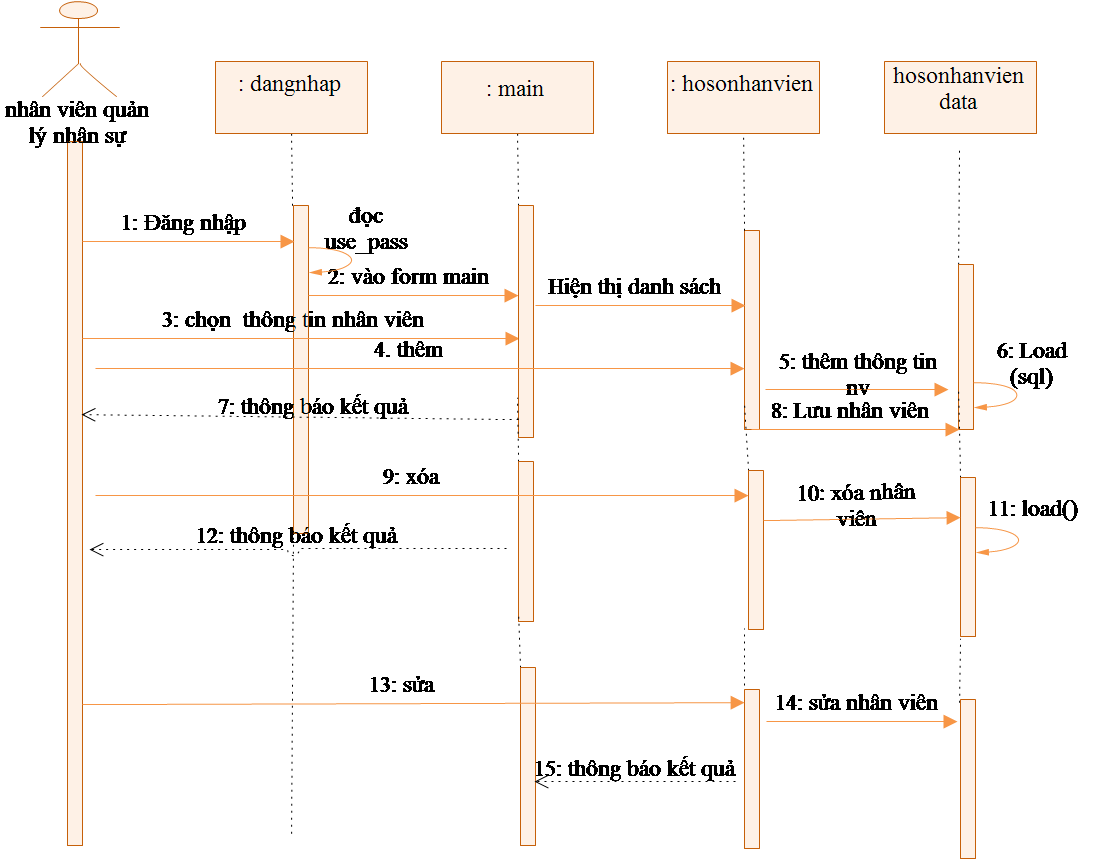
- The output must be compatible with all common browsers: Mozilla Firefox, Internet

Explorer, Google Chrome, Apple Safari, and Opera.

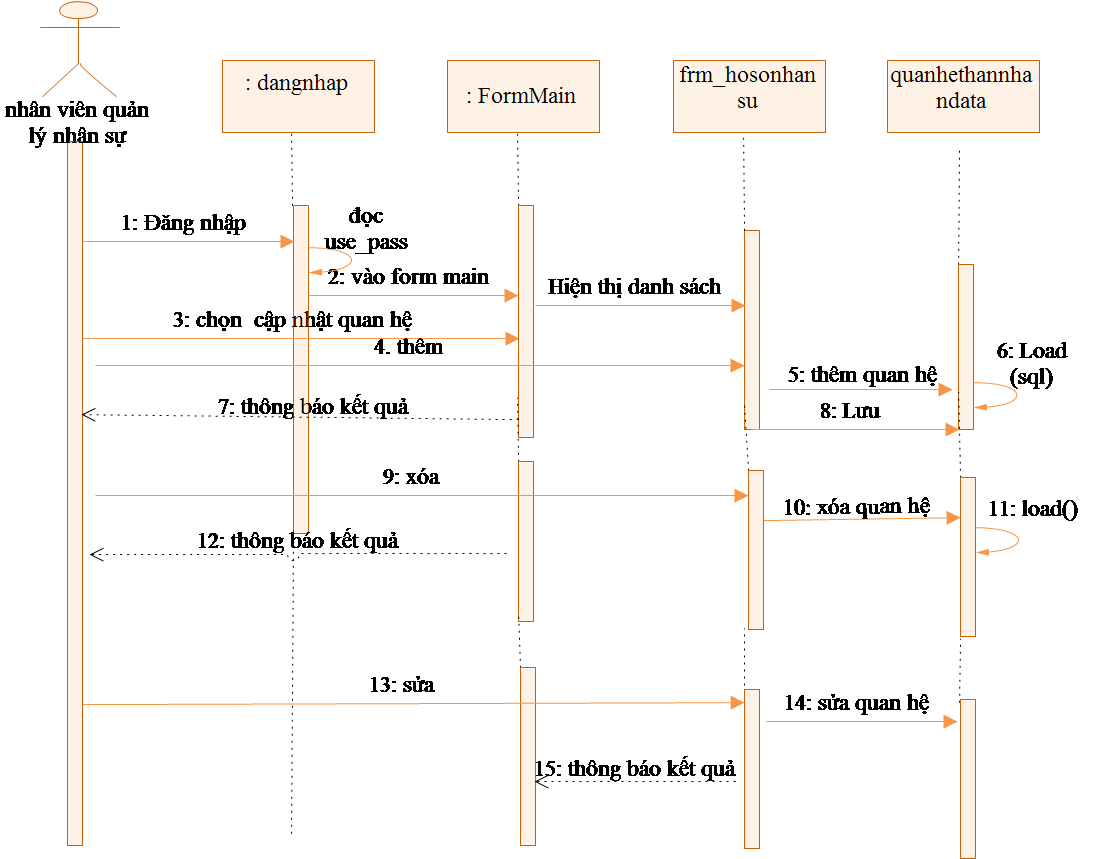
## Software Design Description

## Sequence Diagrams

## Cập nhật thông tin nhân viên

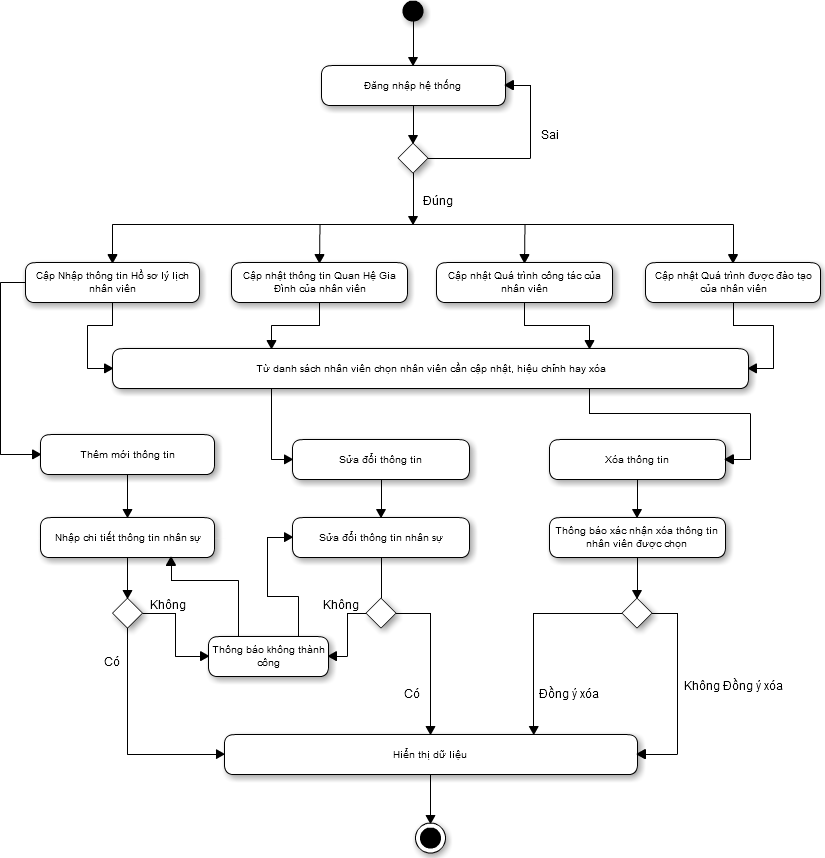


## Quan hệ Gia đình



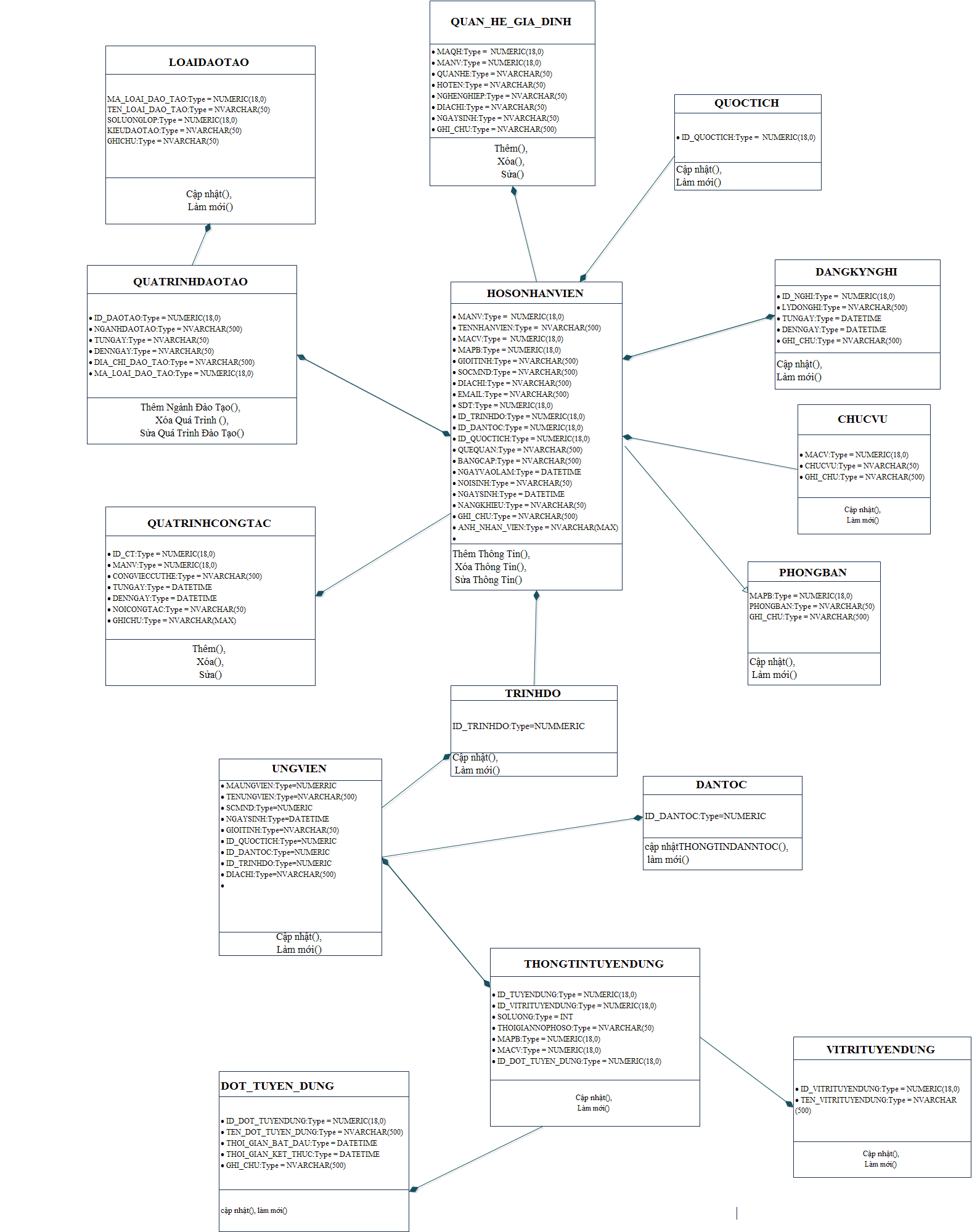
## Activity Diagrams

## 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 |