**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](#_Toc212408687)

[1. Introduction 5](#_Toc212408688)

[1.1. Purpose 5](#_Toc212408689)

[1.2. Intended Audience and Reading Suggestions 5](#_Toc212408690)

[1.3. References 5](#_Toc212408691)

[2. Project Overview 6](#_Toc212408692)

[2.1. Project Description 6](#_Toc212408693)

[2.2. Business Problems 6](#_Toc212408694)

[2.3. Business Need 6](#_Toc212408695)

[2.4. Project Analyst 7](#_Toc212408696)

[2.4.1. Business Function Diagram 7](#_Toc212408697)

[2.4.2. System Context Diagram 7](#_Toc212408698)

[2.4.3. Data Flow Diagram 8](#_Toc212408699)

[2.5. Software Requirement Specification 10](#_Toc212408700)

[2.5.1. High level Fucntional Requirement (FR) 10](#_Toc212408701)

[2.5.2. Stakeholders 11](#_Toc212408702)

[2.5.3. Use case 11](#_Toc212408703)

[2.5.3.1. List of use cases 11](#_Toc212408704)

[2.5.3.2. <<Student Management System>> Use Case Diagram Overview 12](#_Toc212408705)

[2.5.4. Use Case Specification 13](#_Toc212408706)

[*UC 01: User Authentication* 13](#_Toc212408707)

[*UC 04: Manage Profile* 17](#_Toc212408708)

[2.5.5. Software Quality Attributes 20](#_Toc212408709)

[2.5.5.1. Usability 20](#_Toc212408710)

[2.5.5.2. Reliability 20](#_Toc212408711)

[2.5.5.3. Maintainability 20](#_Toc212408712)

[2.5.5.4. Portability 21](#_Toc212408713)

[2.5.5.5. Performance 21](#_Toc212408714)

[2.5.5.6. Scalability 21](#_Toc212408715)

[2.5.5.7. Availability 21](#_Toc212408716)

[2.6. Software Design Description 21](#_Toc212408717)

[2.6.1. Sequence Diagrams 21](#_Toc212408718)

[2.6.1.1. User Authentication 21](#_Toc212408719)

[2.6.1.2. Manage Profile 21](#_Toc212408720)

[2.6.2. Activity Diagrams 21](#_Toc212408721)

[2.6.2.1. User Authentication 21](#_Toc212408722)

[2.6.2.2. Manage Profile 21](#_Toc212408723)

[2.6.3. Class Diagram 21](#_Toc212408724)

# 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 |
| 26/10/2025 | Update | Updated Use Case Diagram, Use Case Specification (UC.01, UC.04), Software Quality Attribute | Bao | 1.2 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 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

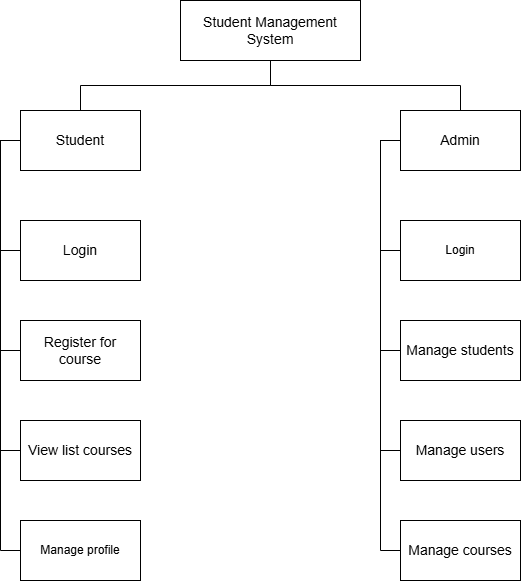
The Student Management System is needed to improve the efficiency and accuracy of managing academic information.  
Currently, most of the operations, such as course registration, user management, and student information updates, are performed manually.  
This manual process is time-consuming, prone to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

By developing this system:

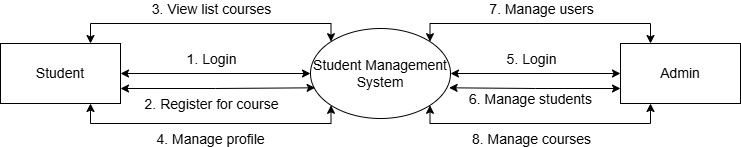
* Administrators can easily manage courses, users, and students through an integrated digital platform.
* Students can register for courses, view their profiles, and access course information conveniently.
* The system ensures data consistency, reduces workload, and improvesoverall productivity for educational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

## Project Analyst

## Business Function Diagram



## System Context Diagram

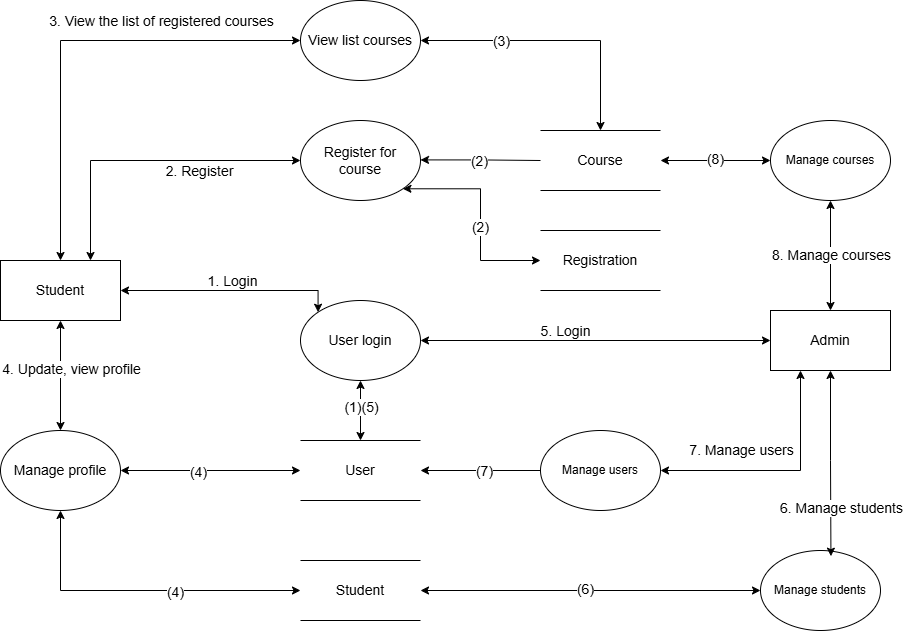


The “Student Management” system with key features that meet business requirements includes:

* For User (Student) role:
* Login to the system.
* Register for course
* View list courses
* Manage Profile
* For Admin role:
  + Manage students
  + Manage users
  + Manage courses

## 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** | **Register For Course** |
| Stakeholder | Student |
| Description | The system allows students to enroll in available courses based on their interests or requirements |
| FR3 | **Title** | **View List Courses** |
| Stakeholder | Student |
| Description | The system allows students to view the list of available and enrolled courses |
| FR4 | **Title** | **Manage 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 the admin to manage students, including viewing and deleting student accounts (Students cannot edit others) |
| 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. | Status |
| UC.01 | User Authentication | FR1 | **Implemented in Sprint 1** |
| UC.02 | Register For Course | FR2 | Planned for future sprint |
| UC.03 | View List Courses | FR3 | Planned for future sprint |
| UC.04 | Manage Profile | FR4 | **Implemented in Sprint 1** |
| UC.05 | Manage Students | FR5 | Planned for future sprint |
| UC.06 | Manage Users | FR6 | Planned for future sprint |
| UC.07 | Manage Courses | FR7 | Planned for future sprint |

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

A diagram of a company structure

AI-generated content may be incorrect.

## Use Case Specification

### *UC 01: User Authentication*

* Use case specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.01 | | | | |
| **Use Case Name** | User Authentication | | | | |
| **Created by** | Bao Nguyen | | **Last updated by** | | Bao Nguyen |
| **Date Created** | Oct 26, 2025 | | **Date last updated** | | Oct 26, 2025 |
| **Actors** | Student, Admin | | | | |
| **Brief Description** | Users log in to the system by providing valid username and password | | | | |
| **Goal** | This function allows users to login to the system | | | | |
| **Trigger** | Users access the login page and require login | | | | |
| **Pre-conditions** | * Users have a valid account. * The system is operating and having connect to the database. | | | | |
| **Post-conditions** | * Users have been successfully validated and accessed to the system. * A user session is created for the user. | | | | |
| **Main Flow** | **Step** | **Actor Action** | | **System Response** | |
| **1** | Open the system login page | | Display the login page including fields for username and password. | |
| **2** | Enter a valid username and password in the respective fields. | |  | |
| **3** | Click the "**Login**" button to submit your information. | | Verify the user input information. The system sends a request to verify the username and password to the database. | |
| **4** |  | | If username and password are valid, the system creates a session for user. | |
| **5** | Get access to the system's home page and start using other functions. | | Redirect users to the system’s home page or dashboard | |
| **Alternative Flow** | **Alternative Flow 1: Incorrect login information** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Open the system login page. | | Display the login page including fields for username and password. | |
| **2** | User entered incorrect username or password. | |  | |
| **3** | Click the "**Login**" button to submit your information. | | Verify the user input information. The system sends a request to verify the username and password to the database. | |
| **4** |  | | Verify information and find that username or password is incorrect. | |
| **5** |  | | Display an error message: "**Incorrect username or password. Please try again**” | |
| **6** | Try entering your login information again | |  | |
| **7** | Click the "**Login**" button to submit your information. | | Continue verifying the new information (return to step 3 of the main flow). | |
| **Alternative Flow 2: User clicks on forgot password link** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Open the system login page. | | Display the login page including fields for username and password. | |
| **2** | Click the “**Forgot Password**” link on the login page | | Display the forgot password page | |
| **3** | Enter your registered username and a new password in the “**New password**” field. Then re-enter the same password in the “**Confirm new password**” field to confirm | |  | |
| **4** | Click the “**Reset Password**” button to reset your password | | The system updates user’s account. Display a success message: “**Reset password successful**“ | |
| **5** |  | | Redirect users to the login page | |
| **Exception Flow** | **Exception Flow 1: The system lost connection to the database** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Open the system login page. | | Display the login page including fields for username and password. | |
| **2** | Enter a valid username and password in the respective fields. | |  | |
| **3** | Click the "**Login**" button to submit your information. | | Verify information but cannot access database due to lost connection. | |
| **5** |  | | Display an error message: "**Error connecting to server. Please try again later**" | |
| **6** | Can try again later or contact the administrator. | |  | |
| **Exception Flow 2: Users don’t enter username or password** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Open the system login page. | | Display the login page including fields for username and password. | |
| **2** | Users don’t enter username or password | |  | |
| **3** | Click the "**Login**" button to submit your information. | | Check the user information entered, if the username or password information is empty, display the message "**Request to enter User or Password**" | |
| **4** | Enter fully username and password information | |  | |
| **5** | Click the "**Login**" button to submit your information. | | Continue verifying the new information (return to step 3 of the main flow). | |
| **Priority** | High | | | | |
| **Business Rule** | * Password and username must be at least 8 characters long. * Users must enter matching passwords when resetting. * Password and username cannot be blank and must not contain whitespace | | | | |

* Prototype

### *UC 04: Manage Profile*

* Use Case Specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | UC.04 | | | | |
| **Use Case Name** | Manage Profile | | | | |
| **Created by** | Bao Nguyen | | **Last updated by** | | Bao Nguyen |
| **Date Created** | Oct 26, 2025 | | **Date last updated** | | Oct 26, 2025 |
| **Actors** | Student | | | | |
| **Brief Description** | When users select “**Profile**” option, the system redirects to profile page. User can select “**Update**” button to update information. | | | | |
| **Goal** | This function allows users to view and update personal information. | | | | |
| **Trigger** | Users click “**Profile**” option. | | | | |
| **Pre-conditions** | * The user is logged in successfully. * The system is connected to the database. | | | | |
| **Post-conditions** | * The profile page is displayed with the user’s information. * If updated, the new information is stored successfully in the database. | | | | |
| **Main Flow A** | **A. View personal profile**  **Trigger:** Users click “**Profile**” option to display information  **Result:** Users can view fully personal information | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Users click “**Profile**” option | | The system displays the profile page | |
| **2** |  | | The system retrieves user’s data from the database to display information | |
| **Main Flow B** | **B. Update personal profile**  **Trigger:** Users click “**Profile**” option, then click the “**Adjust**” button to adjust personal information  **Result:** Users can update personal information and result a confirm message | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Users click “**Profile**” option | | The system displays the profile page | |
| **2** |  | | The system retrieves user’s data from the database to display information | |
| **3** | Users click the “**Adjust**” button. | | The system displays the update form including fields such as first name, last name, age…) | |
| **4** | Users enter personal information in the respective fields. | |  | |
| **5** | Click the “**Update**” button to update information | | The system checks all input fields | |
| **6** |  | | Display a confirm message: **“Are you sure to update this information?”** | |
| **7** | Users click the “**Confirm**” button to confirm | | The system saves new information to the database | |
| **8** |  | | Display a success message: “**Update successful**” | |
| **Alternative Flow** | **Alternative Flow 1:** **Users don’t want to change information** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Users click “**Profile**” option | | The system displays the profile page | |
| **2** |  | | The system retrieves user’s data from the database to display information | |
| **3** | Users click the “**Adjust**” button. | | The system displays the update form including fields such as first name, last name, age…) | |
| **4** | Users enter personal information in the respective fields. | |  | |
| **5** | Click the “**Update**” button to update information | | The system checks all input fields | |
| **6** |  | | Display a confirm message: **“Are you sure to update this information?”** | |
| **7** | Users click the “**Cancel**” button or “**Close**” button (**X**) | | The system displays the recent update form | |
| **8** | Users click the “**Exit**” button | | The system redirects to the home page | |
| **Alternative Flow 2: Users reset the form** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Users click “**Profile**” option | | The system displays the profile page | |
| **2** |  | | The system retrieves user’s data from the database to display information | |
| **3** | Users click the “**Adjust**” button. | | The system displays the update form including fields such as first name, last name, age…) | |
| **4** | Users enter personal information in the respective fields. | |  | |
| **5** | Click the “**Update**” button to update information | | The system checks all input fields | |
| **6** |  | | Display a confirm message: **“Are you sure to update this information?”** | |
| **7** | Users click the “**Cancel**” button or “**Close**” button (**X**) | | The system displays the recent update form | |
| **8** | Users click the “**Reset**” button | | The system clears all input fields and restores default values. | |
| **9** | The user may re-enter personal information (return to step 4 of main flow B) | |  | |
| **Exception Flow** | **Exception Flow 1:** **Users don’t enter the information** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| **1** | Users click “**Profile**” option | | The system displays the profile page | |
| **2** |  | | The system retrieves user’s data from the database to display information | |
| **3** | Users click the “**Adjust**” button. | | The system displays the update form including fields such as first name, last name, age…) | |
| **4** | Users leave required information blank | |  | |
| **5** | Click the “**Update**” button to update information | | The system checks all input fields. The required information is empty | |
| **6** |  | | Display an error message: “**Please fill in all required fields.**” | |
| **7** | Enter fully information | |  | |
| **8** | Click the “**Update**” button to update information | | Continue checking the input fields. (Return to step 5 of the main flow B) | |
| **Exception Flow 2: The system lost connection to the database** | | | | |
| **1** | Users click “**Profile**” option | | The system displays the profile page | |
| **2** |  | | The system retrieves user’s data but cannot access to database due to lost connection | |
| **3** |  | | Display an error message: "**Error connecting to server. Please try again later**" | |
| **4** | Can try again later or contact the administrator. | |  | |
| **Priority** | High | | | | |
| **Business Rule** | * All required input fields must not be left blank. * Age and phone number must contain numeric values only. * Email field must be in valid email format. * Data must be validated before saving to the database. | | | | |

1. Prototype

## Software Quality Attributes

## Usability

* 1. **GUI**
  + The interface should be clear and user-friendly.
  + The layout must be visually balanced and intuitive.
  + All texts, fonts, and colors shall follow a consistent style.
  1. **Installation**
  + The system must be easy to deploy, allowing other users to install and configure it successfully.
  + The documentation for installation must be included. It describes the details of step for installing or deploying the system.

## Reliability

* + The system shall automatically recover from minor connection or service interruptions.

## Maintainability

* + All source code should be clearly commented, including class and method documentation.
  + All packages, classes, and methods shall follow standardized naming conventions to simplify maintenance and future expansion.

## Portability

* + The application should be compatible with Chrome and Edge browsers and deployable in a local server environment.

## Performance

* + The system should respond to user requests within 3 seconds under normal load.

## Scalability

* + The system design shall support future addition of new modules, new features.

## Availability

* + The system shall remain available and accessible after each sprint release.

## Software Design Description

## Sequence Diagrams

## User Authentication

## Manage Profile

## Activity Diagrams

## User Authentication

## Manage Profile

## Class Diagram