****

**Information systems**

**Course project report**

**Students Name**

**2024**

**Contents**

[Task 3](#_Toc56514053)

[Work progress 4](#_Toc56514054)

[Results 5](#_Toc56514055)

# **Task**

1. Describe Business Logic of your Information System

The system is designed to provide an educational course catalog, allowing teachers to add course information and users to browse and download course outlines. The core business logic of the system includes user management, course management, and access control. Below are the main business logic components of the system:

User Management:

User Registration: Users can create a new account by filling in the necessary personal information.

User Login: Users log into the system by entering their username and password.

User Logout: Logged-in users can securely log out of the system.

Password Change: Users can change their password while logged in.

Course Management:

Teacher Adds Course: Teachers can add course information, including course name, duration of learning activities, description, and syllabus file.

Teacher Modifies Course: Teachers can modify the course information they have added.

Teacher Deletes Course: Teachers can delete the course information they have added.

User Browses Courses: Users can view the course catalog and obtain basic information about courses.

User Searches and Filters Courses: Users can search for specific courses using a search box and filters.

Paginated Display: When there are many courses, users can browse the courses using pagination.

Access Control:

Permission Management for Different Roles: The system distinguishes between user, teacher, and administrator permissions, ensuring that each role can only access authorized functions.

Administrator Functions: Administrators can manage user permissions, view all course information, and enable or disable specific user functions.

1. Describe requirements analysis (as in Software Requirement Specification)

Requirements Analysis

Functional Requirements

User Functions:

Register an account

Login and logout

Change password

Browse course catalog

Search and filter courses

Teacher Functions:

Add course

Modify course

Delete course

Administrator Functions:

Manage user permissions

View all course information

Enable and disable user functions

Non-Functional Requirements

Security:

User passwords should be stored in an encrypted format to ensure user information security.

Implement access control to ensure different roles can only access authorized functions.

Usability:

The interface should be user-friendly and easy to navigate.

The system should support access from various devices to ensure a consistent user experience.

Performance:

The system should handle a large number of user requests and ensure response times are within an acceptable range.

Database queries should be optimized to ensure quick retrieval of course information.

1. Describe ER Database diagram

1. Users (用户)

Attributes:

1.1 id (Primary Key)

1.2 username (Unique)

1.3 password (Encrypted Storage)

1.4 role (User Role: User, Teacher, Administrator)

1.5 login

1.6 status

2. Courses (课程)

Attributes:

2.1 id (Primary Key)

2.2 course\_name (Course Name)

2.3 duration (Duration of Learning Activities)

2.4 description (Course Description)

2.5 image

2.6 status

1. Describe UML Use Case Diagram

| **Use Case** | **No Permissions** | **User** | **Teacher** | **Administrator** |
| --- | --- | --- | --- | --- |
| Login | + | + | + | + |
| Register | + | + | + | + |
| Logout |  | + | + | + |
| Change Password |  | + | + | + |
| View Course List |  | + | + | + |
| Search Course |  | + | + | + |
| Add Course |  |  | + | + |
| Modify Course |  |  | + | + |
| Delete Course |  |  | + | + |
| Manage Users |  |  |  | + |
| View All Schedules |  |  |  | + |
| Enable Admin Rights |  |  |  | + |
| Disable Admin Rights |  |  |  | + |
|  |  |  |  |  |

1. Describe interface layout

The pages should be as follows:

1. \*\*index.html\*\*: The homepage has buttons for login, registration, logout, students, teachers, and manager. The login information is displayed in the bottom left corner. Unlogged users can only see the login and registration buttons, while logged-in users can see the student, teacher, and manager buttons based on their permissions. They can only see the logout button and not the login and registration buttons.

2. \*\*student.html\*\*: Displays course information, with pagination showing ten items per page. Clicking 'next page' will navigate to the next set of courses. There are also buttons to return to the homepage and logout in the bottom left corner.

3. \*\*teacher.html\*\*: Displays course information, with pagination showing ten items per page. Users can also add, modify, or delete courses. There are buttons to return to the homepage and logout in the bottom left corner.

4. \*\*manager.html\*\*: Displays user information, with pagination showing ten items per page. Users can also add, modify, or delete user information. There are buttons to return to the homepage and logout in the bottom left corner.

5. \*\*login.html\*\*: Contains a form prompting for ID, name, and password. Correct input will log in the user and redirect them to the homepage.

6. \*\*logout\*\*: Clicking logout will redirect to the homepage and remove the login information.

7. \*\*registration\*\*: Contains a form prompting for name and password. Submitting this will add the information to the users and return to the login page.

1. Describe information system architecture

Tech Stack

Frontend:

Use HTML5 and CSS to build the user interface.

Use JavaScript,Thymeleaf to handle user interactions and dynamic content.

Backend:

Use Spring Boot to build RESTful APIs for processing user requests and business logic.

Use MyBatis-Plus for database operations, streamlining CRUD operations.

Database:

Use PostgreSQL to store user and course information and design a reasonable database schema to support system functions.

# **Work progress**

# **Results**