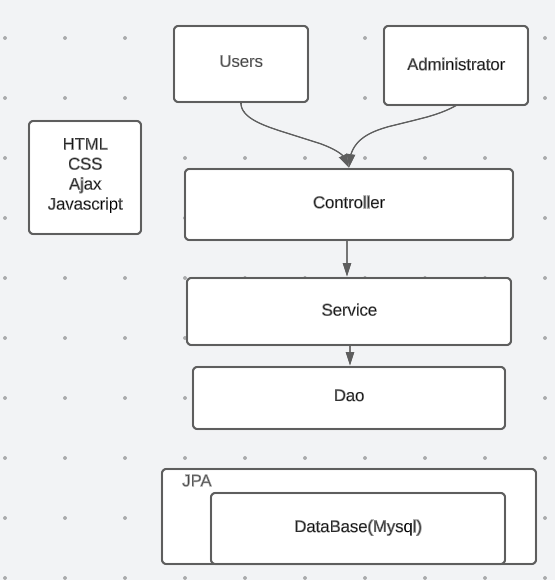
**Technical Design – Design and implementation of student information management system**

## Introduction

The original intention behind this project design is that although Canvas is powerful and comprehensive, I believe that there are some shortcomings in certain features, such as the handling of student leave requests and attendance scores. Also, it's overly functional and not precise enough.0

## 1. System Architecture

The student information management system is developed using the servlet+jsp framework, which follows the standard MVC pattern, dividing the entire system into four layers: View layer, Controller layer, Service layer, and DAO layer. Among them, data retrieval and display are done in the View layer, business object management is implemented in the Controller layer using servlets, and JSP is used as the persistence engine for data objects. The entire system structure and operation process is shown in Figure 2-1:



JSP

**Figure 2-1**

View layer：It is closely combined with the Controller layer and needs to work together for front-end JSP page presentation.

Controller layer：The controller imports the Service layer because the methods in Service layer are the ones we use. The controller receives parameters passed from the front-end to perform operations and returns a specified path or data table.

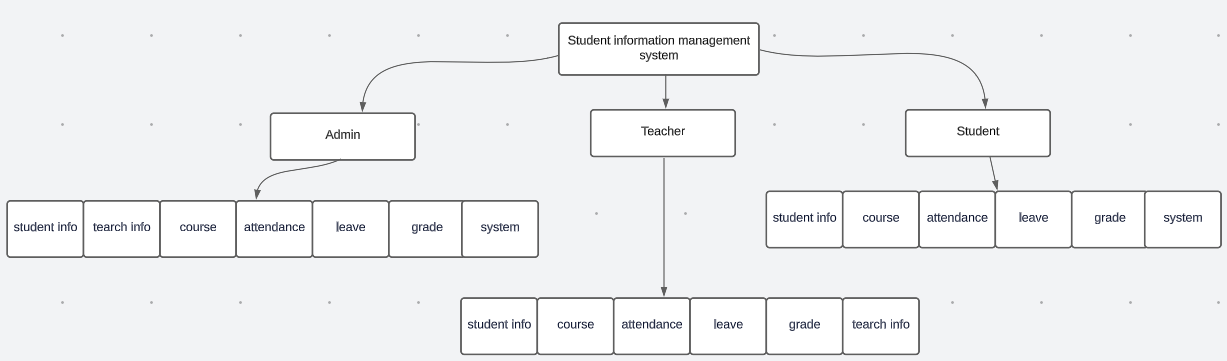
Service layer：Regarding database operations, they are not directly handled with the database. There are interfaces and implementation methods for the interfaces. In the implementation methods of the interface, the Dao layer needs to be imported. The Dao layer directly interacts with the database and it is also an interface that only has method names. The specific implementation is in the mapper.xml file. The Service layer provides methods for our use.

Dao layer: Responsible for the operations of adding, deleting, modifying, and querying data to the database.

Persistence Layer: JSP here is used to persist entity objects into the database, eliminating the need for complex JDBC and SQL statements. In the Dao layer, JSP syntax can be used directly to execute the desired SQL.

## 2.2 Design of System Functional Modules.

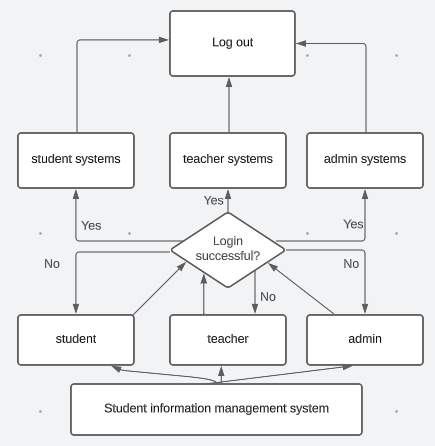
The student information management system has a relatively high degree of comprehensiveness and complexity. It can make full use of existing software for system design and planning. Building a complete and mature student information management system involves the front-end pages, processing programs, MySQL back-end database system, etc. The processing program is actually responsible for processing user-submitted forms and related operations. Information stored in the back-end database includes grade data, student data, etc.



1. Database design

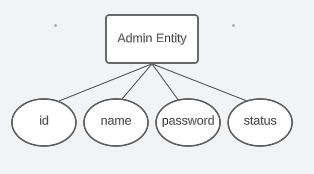
The student information management system uses MySQL as the database development tool.

3.1 System flow

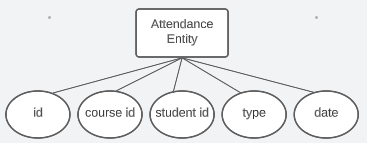


The main entities are: admin entity, attendance entity, course entity, announcement entity, teacher entity, student entity, etc. The specific description and attribute diagrams of each entity are as follows (not all entity attributes are listed in the diagram due to the large number of attributes):

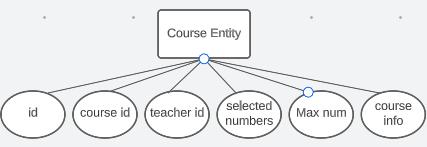
1.Admin Entity



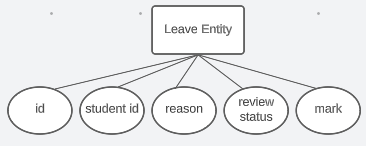
2.Attendance Entity



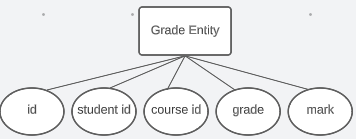
3.Course Entity



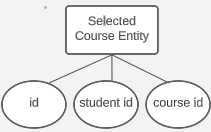
5. Leave Management Entity



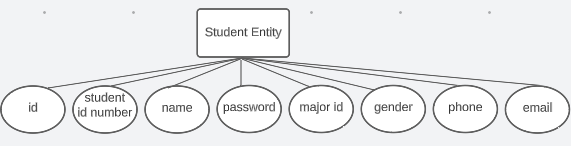
6. Grade Entity



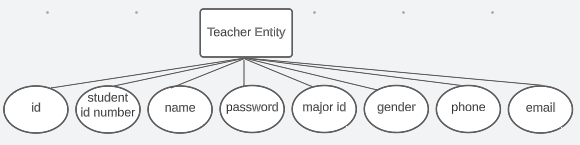
7. Selected Course Enity



8. Student Entity



9. Teacher Entity



### 3.2 Tables

The database includes:

1. Admin table: s\_admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | admin id |
| name | Varchar（32） | N | Y | admin name |
| password | Varchar（32） | N | Y | password |
| status | Tinyint（1） | N | Y | is admin |

（2）Attendance:s\_attendance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | attendance id |
| course\_id | Int（5） | N | Y | course id |
| student\_id | Int（5） | N | Y | student id |
| type | Varchar（11） | N | Y | time(am or pm) |
| date | Varchar（11） | N | Y | date |

（3）Major: s\_major

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | major id |
| name | Varchar（32） | N | Y | major name |
| info | Varchar（128） | N | Y | major info |

（4）Course: s\_course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | course id |
| name | Varchar（32） | N | Y | course name |
| teacher\_id | Int（5） | N | Y | teacher id |
| course\_date | Varchar（32） | N | Y | course date |
| selected\_num | Int（5） | N | Y | selected number |
| max\_num | Int（5） | N | Y | max number |
| info | Varchar（128） | N | Y | course info |

（5）Leave management: s\_leave

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | id |
| student\_id | Int（5） | N | Y | student id |
| info | Varchar（521） | N | Y | leave reason |
| status | Tinyint（1） | N | Y | reviewed or not |
| remark | Varchar（512） | N | Y | remark |

（6）Score:s\_score

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | id |
| student\_id | Int（5） | N | Y | student id |
| course\_id | Int（5） | N | Y | course id |
| score | Double（5） | N | Y | score |
| remark | Varchar（128） | N | Y | remark |

（7）Selected course: s\_selected\_course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | id |
| student\_id | Int（5） | N | Y | student id |
| course\_id | Int（5） | N | Y | course id |

（8）Student: s\_student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | id |
| sn | Varchar（32） | N | Y | student number |
| name | Varchar（32） | N | Y | name |
| password | Varchar（32） | N | Y | password |
| major\_id | Int（5） | N | Y | major |
| sex | Varchar（5） | N | Y | student’s gender |
| mobile | Varchar（12） | N | Y | phone number |
| email | Varchar（32） | N | Y | Email |
| photo | mediumblob | N | Y | photo |

（9）Teacher: s\_teacher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Primary | Null or not | Description |
| id | bigint(20) | Y | N | id |
| name | Varchar（32） | N | Y | name |
| password | Varchar（32） | N | Y | password |
| major\_id | Int（5） | N | Y | major |
| sex | Varchar（5） | N | Y | gender |
| mobile | Varchar（12） | N | Y | phone number |
| email | Varchar（32） | N | Y | Email |
| photo | mediumblob | N | Y | photo |
| sn | Varchar（32） | N | Y | teacher id number |