**Student Management System**

黄美祺HuangMeiQi Fiona

付珂FuKe Simple

刘瑞哲LiuRuiZhe Andrew

陈中ChenZhong Isabella

**1 Requirements Overview**

Students and administrators log in and enter the system. They will respectively enter the student interface and administrator interface. Students can view and modify personal information, but the student ID is refused to be modified, and they can also apply for leave. The administrator can conduct comprehensive management of students, including viewing all students and thinformation, modifying students’ basic information, but the studentID is refused to be modified, adding new students, deleting students, viewing all students’ attendance information and approving student leave application.

**2 Module Description**

**2.1 Login Module**

This module provides the functionality of login process. If correct, an interface will appear according to the user type. if incorrect, proper error message will be displayed.

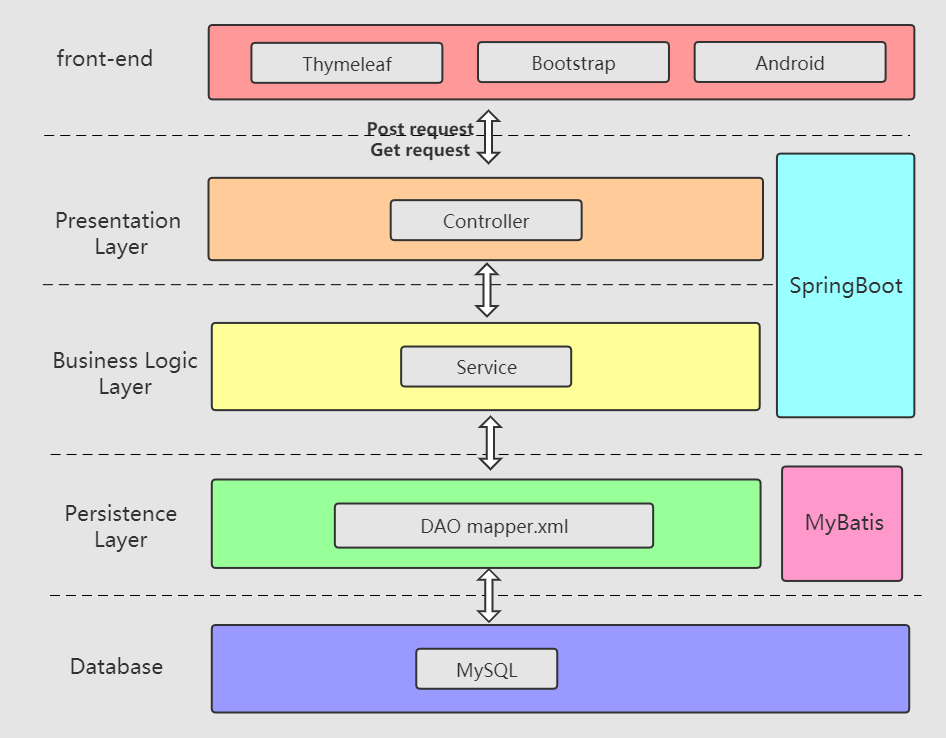
**2.2 Admin Home Module**

This module provides the functionalities of viewing all students’ basic information and leave information, adding new students, modifying student basic information and deleting students. While modifying student basic information, administrators cannot modify studentID which is the fixed information and administrator cannot leave any information blank or type error before clicking the submit button, otherwise it will appear a reminder message. If administrator delete a student, the student leave information will also be deleted in database. While approving a student leave, administrator need to check carefully and can choose to agree or reject the leave application.

**2.3 Student Home Module**

This module provides the functionalities of viewing and modifying his/her basic information and asking for a leave. While modifying his/her basic information, he/she cannot modify studentID which is the fixed information and he/she cannot leave any information blank or type error before clicking the submit button, otherwise it will appear a reminder message and failed to submit. While asking a leave, studentID field will appear but cannot be edited and if other leave information fields incomplete, it will appear a reminder message to make sure leave necessary information be filled in. After click the submit button, then he/she can wait for the administrator’s check.

**3 System Architecture Design**

****

**4 API**

# 4.1Page Request Parameters

# 4.1.1Users Login

### Basic Information

**Path：** /User/login

**Method：** POST

**Interface Description：**Verify user name and password. Administrators and students verify login uniformly.

Required Parameter

**Body**

User.{*//Administrator User id and Password*  
 "loginid": "000",  
 "password": "123"  
}OR  
User.{*//Student User id and Password*  
 "loginid": "001",  
 "password": "1234"  
}

Return Data

Admin.{  
"adminid":"000",  
"name":"Marry"  
}  
OR  
Student.{  
"stuid":"001",  
"name":"Tom",  
"batch":"one",  
"age":"21",  
"address":"HaiNan University",  
"contactnumber":"13088927313",  
"email":"13088927313@qq.com"  
}

# 4.1.2Administrators View Student Basic Information

**Path：** /admin/to-home

**Method：**GET

**Interface Description：**The administrator can view the basic information of all students directly on the home page after successful login.

Required Parameter

**Body**

null

Return Data

{  
 "status": 1,  
 "msg": "successfully get data!",  
 "data": [  
 {  
 "stuid": 3,  
 "name": "ChenZhong",  
 "batch": "two",  
 "age": "18",  
 "address": "HaiNan University",  
 "contactnumber": "13088927312",  
 "email": "1308892@126.com"  
 },  
 {  
 "stuid": 4,  
 "name": "Fiona",  
 "batch": "one",  
 "age": "18",  
 "address": "HaiNan ",  
 "contactnumber": "13088927314",  
 "email": "1308892@qq.com"  
 },  
 {  
 "stuid": 5,  
 "name": "AZhe",  
 "batch": "one",  
 "age": "21",  
 "address": "HaiNan University",  
 "contactnumber": "13088927315",  
 "email": "13088927315@qq.com"  
 },  
 {  
 "stuid": 6,  
 "name": "Tom",  
 "batch": "three",  
 "age": "19",  
 "address": "HNU",  
 "contactnumber": "12345674432",  
 "email": "1053825066@qq.com"  
 },  
 {  
 "stuid": 7,  
 "name": "FuKe",  
 "batch": "two",  
 "age": "19",  
 "address": "HNU",  
 "contactnumber": "12345674432",  
 "email": "1308892@126.com"  
 }  
 ]  
}

# 4.1.3Administrator Changes Student Basic Information

**Path：**/Admin/modify*students*info

**Method：**POST

**Interface Description：**After the administrator changes save a student's basic information, but can not modify the id

Required Parameter

**Body**

Student.{  
"stuId":"001",  
"name":"Tom",  
"batch":"one",  
"age":"21",  
"address":"HaiNan University",  
"contactNumber":"13088927313",  
"email":"13088927313@qq.com"  
}

Return Data

CODE.{  
 "code":"200"  
 "reply":"successfully update!"  
}  
OR  
CODE.{  
 "code":"400"  
 "reply":"fail to update!"  
}

# 4.1.4 Admin Adds New Students

**Path：** /Admin/add*new*students

**Method：**POST

**Interface Description：**Teachers view basic information for all students

Required Parameter

**Body**

StudentDetail.{  
"name":"xxx",  
"batch":"three",  
"password":"1234"  
"age":"21",  
"address":"HaiNan University",  
"contactNumber":"13088927316",  
"email":"13088927316@qq.com"  
}

Return Data

CODE.{  
 "code":"201"  
 "reply":"successfully add!"  
}  
或  
CODE.{  
 "code":"401"  
 "reply":"fail to add!"  
}

# 4.1.5Administrators delete students

**Path：** /Admin/delete*new*students

**Method：**GET

**Interface Description：**Teachers view basic information for all students

Required Parameter

**Body**

{"stuId":"001"}

Return Data

CODE.{  
 "code":"202"  
 "reply":"successfully delete!"  
}  
OR  
CODE.{  
 "code":"402"  
 "reply":"fail to delete!"  
}

# 4.1.6Administrators Check Student Attendance

**Path：** /Admin/look*students*leave

**Method：** GET

**Interface Description：**Teachers check attendance information for all students

Required Parameter

**Body**

NULL

Return Data

List<StudentLeaveDTO>.{  
 StudentLeaveDTO.{  
 "name":"Tom",  
 "batch":"one",  
 "age":"21",  
 "leaveDate":"22/12/2020",  
 "reason":"Sick"},  
 StudentLeaveDTO.{  
 "name":"AZhe",  
 "batch":"one",  
 "age":"21",  
 "leaveDate":"23/12/2020",  
 "reason":"go home"},  
 StudentLeaveDTO.{  
 "name":"Fuke",  
 "batch":"two",  
 "age":"21",  
 "leaveDate":"24/12/2020",  
 "reason":"visit wedding"}   
}

# 4.1.7Students Check Their Information

**Path：** /Student/look\_detail

**Method：**POST

**Interface Description：**Students check their basic information

Required Parameter

**Body**

{  
 "student": {  
 "stuid": 1,  
 "name": "Tom",  
 "batch": "one",  
 "age": "99",  
 "address": "HaiNan University",  
 "contactnumber": "13088927313",  
 "email": "13088927313@qq.com"  
 },  
 "user": {  
 "loginid": 1,  
 "password": "1234"  
 }  
}

Return Data

Student.{  
"stuId":"001",  
"name":"Tom",  
"batch":"one",  
"age":"21",  
"address":"HaiNan University",  
"contactNumber":"13088927313",  
"email":"13088927313@qq.com"  
}  
User.{*//Student User id and Password*  
 "loginId": "001",  
 "password": "1234"  
}

# 4.1.8Students Modify their Information

**Path：** /Student/modify\_detail

**Method：**POST

**Interface Description：**A student looks at his basic information, but can not modify his id.

Required Parameter

**Body**

{  
"stuID":1,  
"name":"Tom",  
"batch":"one",  
"password":"123",  
"age":"88",  
"address":"HaiNan University",  
"contactNumber":"13088927313",  
"email":"13088927313@qq.com"  
}

Return Data

CODE.{  
 "code":"200"  
 "reply":"successfully update!"  
}  
OR  
CODE.{  
 "code":"400"  
 "reply":"fail to update!"  
}

# 4.1.9Student Leave

**Path：** /Student/ask\_leave

**Method：**POST

**Interface Description：**Student leave

Required Parameter

**Body**

{  
 "leaveid": 1,  
 "stuid": 1,  
 "leavedate": "2020-12-22",  
 "reason": "Sick"  
}

Return data

CODE.{  
 "code":"203"  
 "reply":"successfully ask leave!"  
}  
OR  
CODE.{  
 "code":"403"  
 "reply":"fail to ask leave!"  
}

# 4.1.10Administrator Approves Leave

**Path：** /Admin/agree\_leave

**Method：**POST

**Interface Description：**Approval of leave

Required Parameter

Leaveisagreed.{  
leaveid:1,  
isagreed:-1OR 1  
}

# 4.2 util Layer Interface

# 4.2.1 ServerResponse-SpringBoot&Android

ServerResponse

{ "status":"201", "msg":"fail to update", "data":null }

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| status-int | Status |
| msg-String | Message |
| leaveDate-date-[now()] | Data |

# 4.2.2 OkHttpUtil-Android

OkHttpUtil

{ "obj":student, "msg":null, "url":"https://native ip address:8086/restful/..." }

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| obj-Object | Body passed |
| msg-Message | Message |
| url-String | Request address |

# 4.2.3 requestUrl-Android

requestUrl

{ "prefix":"http://192.168.43.134:8086/restful", "suffix":{serLogin("/User/login"), AdminToHome("/admin/to-home"), AdminAddNewStudent("/Admin/add*new*students"), AdminDeleteStudent("/Admin/delete*new*students?stuid="), AdminLookStudentLeave("/Admin/look*students*leave"), StudentModifyDetail("/Student/modify*detail"), StudentAskLeave("/Student/ask*leave"), StudentLookDetail("/Student/look*detail"), AdminModifyStuInfo("/Admin/modify*students*info"), AdminAgreeLeave("/Admin/agree*leave");} }

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| prefix-String | Url prefix |
| suffix-String | Url suffix |

# 4.3 Service Layer Interface

# 4.3.1 Student Service

## Name: StudentService

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| int updateStudent(StudentDetail student) | Update student information |
| int deleteStudent(Integer stuId) | Delete students |
| Student selectStudent(Integer stuId) | Enquiries about a student's details |
| List selectAllStudent() | Check all students and return list form |
| int addStudent(StudentDetail student) | One new student |

# 4.3.2Leave Service

## Name: LeaveService

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| int addLeaveInfo(Leave leave) | New leave information |
| List selectAllStudentLeave() | Check all student leave details and return to list form |

# 4.3.3Admin Service

## Name: AdminService

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| Admin selectAdmin(Integer adminId) | Returns Admin basic information |

# 4.3.4UserService

## Name:UserService

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| int checkUser(User user) | Check user legitimacy, return 0 for illegal, return 1 for administrator, return 2 for student. |

# 4.4 mapper Layer Interface

# 4.4.1Student Mapper

## Name:StudentMapper

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| int updateStudent(Student student) | Update student information |
| int deleteStudent(Integer stuId) | Delete students |
| Student selectStudent(Integer stuId) | Enquiries about a student's details |
| List selectAllStudent() | Check all students and return list form |
| int addStudent(Student student) | One new student |

# 4.4.2LeaveisagreedMapper

## Name:LeaveisagreedMapper

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| int deleteByPrimaryKey(Integer leaveid) | Delete Leave permit information |
| int insert(Leaveisagreed record) | Insert an Leave permit |
| Leaveisagreed selectByPrimaryKey(Integer leaveid) | Search for a Leave Permit |
| List selectAll(); | Enquire all leave permit information |
| int updateByPrimaryKey(Leaveisagreed record); | Update Leave permit information |

# 4.4.3LeaveMapper

## Name:LeaveMapper

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| int addLeaveInfo(Leave leave) | New leave information |
| List selectAllLeave() | Check all student leave details and return to list form |
| Leave selectLeave(int stuId) | Enquire about a student's leave details |

# 4.4.4Adminmapper

## Name:AdminMapper

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| Admin selectAdmin(int adminId) | Returns Admin basic information |

# 4.4.5Usermapper

## Name:UserMapper

## Interface and Description

|  |  |
| --- | --- |
| Interface | Description |
| User selectUser(User user) | Query whether the user object exists in the user table. If return exists, no return null. exists |
| int updateUser(User user) | Update user password |

# 4.5Database Dictionary

# 4.5.1Student Table

student

{

"stuId":"001", "name":"Tom", "batch":"one", "age":"21", "address":"HaiNan University", "contactNumber":"13088927313", "email":"13088927313@qq.com"

}

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| stuId-int-[auto increment|primary key] | Student id, self-increase, primary key |
| name-varchar(10)-[not null] | Name, not null |
| batch-ENUM-['one','two','three'] | Class, enumeration class |
| age-char(2)-[not null] | Age, not null |
| address-varchar(20) | Ddress |
| contactNumber-varchar(20) | ContactNumber |
| email-varchar(20) | Email |

#### Trigger：student\_BEFORE\_DELETE

CREATE DEFINER=root@localhost TRIGGER springboot\_hw.student\_BEFORE\_DELETE BEFORE DELETE ON student FOR EACH ROW BEGIN delete from user where loginid=old.stuId; END

#### Trigger：student\_AFTER\_INSERT

CREATE DEFINER=root@localhost TRIGGER springboot\_hw.student\_AFTER\_INSERT AFTER INSERT ON student FOR EACH ROW BEGIN insert into user values(new.stuId,'1234'); END

# 4.5.2Admin Table

admin

{ "adminId":"000", "name":"Marry" }

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| adminId-int-[primary key] | Login id, foreign key constraints as foreign keys for students and administrators |
| name-varchar(10)-[not null] | Name |

# 4.5.3User Table

user

{ "loginId": "000", "password": "123" }

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| loginId-int-[trigger,To check whether the id exists in the student or admin table before inserting] | Login id, foreign key constraints, as a student, administrator foreign key, the main table deleted from the table cascade. |
| password-varchar(30)-[not null] | Password |

# 4.5.4Leave Table

leave

{ "stuId":"001", "leaveDate":"22/12/2020", "reason":"Sick" }

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| leaveId-int-[auto increment|primary key] | id self-increase |
| stuId-int-[foreign key references student(stuId) on delete cascade|not null] | Student id, as a student's foreign key, the main table deleted the sub-table cascade. Not null. |
| leaveDate-date-[now()] | School departure time, year-month-day. Use sql insert when inserted |
| reason-varchar(30) | Reason |

#### Trigger：leave\_AFTER\_INSERT

CREATEDEFINER=root@localhostTRIGGER springboot\_hw.leave\_AFTER\_INSERT AFTER INSERT ON leave FOR EACH ROW BEGIN insert into springboot\_hw.leaveisagreed values(new.leaveId,0); END;;

#### Trigger：leave\_BEFORE\_DELETE

CREATE DEFINER=root@localhost TRIGGER springboot\_hw.leave\_BEFORE\_DELETE BEFORE DELETE ON leave FOR EACH ROW BEGIN delete from springboot\_hw.leaveisagreed where leaveId=old.leaveId; END ;;

# 4.5.5leaveisagreed Table

leaveisagreed

{ "leaveId":1, "isAgreed":1,

}

|  |  |
| --- | --- |
| Field Name - Properties - Constraints | Description |
| leaveId-int-[auto increment|primary key] | auto increment id for leave |
| isAgreed-int | Leave permit |

# 4.6POJO Entity Class

# 4.6.1DAO

## Student

Student

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| stuId-Integer | id students |
| name-String | Name |
| batch-String | Batch |
| age-String | Age |
| address-String | Address |
| contactNumber-String | ContactNumber |
| email-String | Email |

## Admin

Admin

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| adminId-Integer | Login id |
| name-String | name |

## User

User

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| loginId-Integer | Login id |
| password-String | password |

## Leave

Leave

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| leave-Integer | Leave id |
| stuId-Integer | Student id |
| leaveDate-date-[now()] | School departure time, year-month-day. Use sql insert when inserted |
| reason-varchar(30) | Reason |

## Leaveisagreed

Leaveisagreed.{

"leaveid":"1"

"isagreed":"1"},

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| leaveid-Integer | Leave id |
| isagreed-Integer | Leave to pass through sign |

# 4.6.2DTO

## ResponseCode

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| code-string | Status code with default values |
| reply-string | Reply content with default value |

|  |  |
| --- | --- |
| code | reply |
| 200 | successfully update! |
| 201 | successfully add! |
| 202 | successfully delete! |
| 203 | successfully ask leave! |
| 204 | successfully to agree leave! |
| 1 | successfully get data! |
| code | reply |
| 400 | fail to update! |
| 401 | fail to add! |
| 402 | fail to delete! |
| 403 | fail to ask leave! |
| 404 | fail to agree leave! |
| 0 | fail to get data! |

## StudentLeave

Student\_Leave.{ "name":"Tom", "batch":"one", "age":"21", "leaveDate":"22/12/2020", "reason":"Sick"

"LeaveId":1

},

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| name-String | Student's name |
| batch-String | Student Class |
| age-String | Student age |
| LeaveDate-Date | School Leaving Date |
| reason-String | Reasons for leaving school |
| LeaveId-Integer | Leave id |

## StudentDetail

StudentDetail.{ "stuID":"001", "name":"Tom", "batch":"one", "password":"123" "age":"21", "address":"HaiNan University", "contactNumber":"13088927313", "email":"13088927313@qq.com" }

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| stuId-Integer | Student id |
| name-String | Name |
| batch-String | Class |
| password-String | Password |
| age-String | Age |
| address-String | Address |
| contactNumber-String | ContactNumber |
| email-String | Email |

## StudentLeaveDTO

StudentLeaveDTO.{ "name":"Tom", "batch":"one", "age":"21", "leaveDate":"22/12/2020", "reason":"Sick"},

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| name-String | Student's name |
| batch-String | Student Class |
| age-String | Student age |
| leaveDate-String | School Leaving Date |
| reason-String | Reasons for leaving school |
| leaveid-Integer | Leave id |
| isagreed-Integer | Leave to pass through sign |

## Batch

Bathch.{

one,two,three

}

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| Batch-enum | Student Class |

## LeaveDTO

LeaveDTO.{

"leaveid":"1"

"stuid":"1" "leaveDate":"22/12/2020", "reason":"Sick"

"isagreed":1

},

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| leave-Integer | Leave id |
| stuId-Integer | Student id |
| leaveDate-String | Leaving school |
| reason-String | Reason |
| isagreed-Integer | Leave Consent Sign |

## ResponseServer

ResponseServer.{

"status":"200"

"msg":"successfully update!"

"data":"{}"

}OR

ResponseServer.{

"status":"1"

"msg":" successfully get data!"

"data":"Student.{

"stuId":"001", "name":"Tom", "batch":"one", "age":"21", "address":"HaiNan University", "contactNumber":"13088927313", "email":"13088927313@qq.com" }"

}

|  |  |
| --- | --- |
| Field Name - Properties | Description |
| status-int | Interface Call State Code |
| msg-String | Information that the interface needs to respond to |
| data-T | Interface corresponding generic data objects |

Method

|  |  |
| --- | --- |
| Method Name | Description |
| public boolean isSuccess() | Correct response |
| public static ServerResponse createBySuccess() | Only return success code |
| public static ServerResponse createBySuccesssMessage(String msg) | Returns success code and custom corresponding messages |
| public static ServerResponse createBySuccesss(T data) | Returns success code and custom data objects |
| public static ServerResponse createBySuccesss(String msg,T data) | Returns success code and custom data objects and custom corresponding messages |
| public static ServerResponse createByError() | error code returned only |
| public static ServerResponse createBySErrorMessage(String errorMessage) | Returns error code and custom messages |
| public static ServerResponse createByErrorCodeMessage(int code,String errorMessage) | Returns custom error code and custom messages |
| public static ServerResponse createByresponseCode(ResponseCode responseCode) | 返回自定义的ResponseCode |
| Method Name | Description |
| public static ServerResponse createByresponseCodeData(ResponseCode responseCode,T data) | Returns custom ResponseCode and T |