作业格子系统数据库设计

--------数据库课程大作业

组号: 15

组长: 黄洁莹 学号: 15352127

组员 1: 蔡荣裕 学号: 15352008

组员 2: 张镓伟 学号: 15352408

组员 3: 张子豪 学号: 15352427

组长邮箱: huangjy85@mail2.sysu.edu.cn

学院: 数据科学与计算机学院

专业: 软件工程(移动信息工程)

目录

Notice：可通过 PDF 阅读器侧边目录栏跳转到相应版块

一、背景

与过去中学时期纯粹的学习不同， 当代大学生除学习外，常常还要兼顾社团生活与个人的兴趣爱好发展，这一变化使得许多大学生初入大学门槛时感到种种不适应。众多的课程、繁重的学习作业和其不同的截止日期往往令学生们眼花缭乱，容易错过作业提交的截止日期，导致不必要的扣分。除此之外，大学中教师布置作业的方式往往与中学时期的并无太大区别，与学生之间缺乏更高层次的时刻的互动。

观察当下安卓以及 IOS 的手机应用商店平台，绝大多数与学生作业 相关的手机应用的主要面对对象都是小学生与中学生，而且主要功能是学生进行作业交流、辅导以及老师进行在线批改作业。

此外，虽然大学中 多教师会安排其课程主页或者公共邮箱，但是这种方式的便携性、交互性并不能比得上手机应用

基于以上背景和本学期在数据库课程中已经学习到的的内容，结合大学生日常学习生活中的需求，我们小组决定开发一个类似超级课程表此类 APP 的用于作业发布与查询的手机应用系统，其中数据库系统是本应用的最重要的组成部分。

二、应用需求

本应用意在为广大师生之间（尤指大学师生）提供一个便利的课程作业通知发布、收集与评分查分的交流互动友好的平台，其中的数据库系统采用 MySQL 进行开发。

由此，该应用系统的用户将被分为教师账号与学生账号两大类。

对于教师账号而言，其可以：

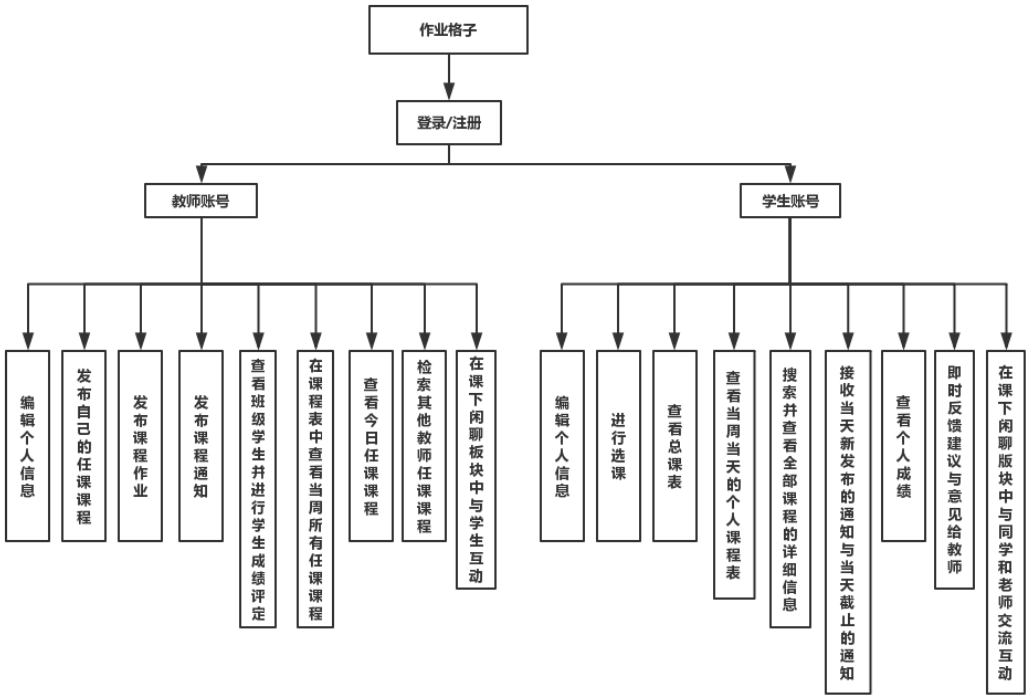
1) 编辑个人信息
2) 发布自己的任课课程
3) 发布课程通知（如上课内容、考试时间等信息）
4) 发布课程作业
5) 查看班级学生并进行学生成绩评定
6) 在课程表中查看当周所有任课课程
7) 查看今日任课课程
8) 检索其他教师任课课程
9) 在课下闲聊版块中与学生互动

而对于学生账号而言，其可以：

1) 编辑个人信息
2) 进行选课
3) 查看总课程表
4) 查看当周当天的个人课程表
5) 搜索并查看全部课程的详细信息
6) 接收当天新发布的通知与当天截止的通知
7) 查看个人成绩

8) 即时反馈建议与意见给教师
9) 在课下闲聊版块中与同学和老师交流互动

由此得到一个系统的需求顶层用例图如下:



三、数据库设计

本报告**主要说明数据库部分的设计**，app 其他部分的具体设计不详细说明。我们的数据库系统为 **MySQL**。

1、基本思路

根据需求，我们为我们的应用提出了以下几张表及其功能：

表名	功能
TEACHERS	存储教师个人信息
STUDENTS	存储学生个人信息
COURSES	存储课程信息
HOMEWORKS	存储作业信息
HOMEWORKSCORE	存储作业分数
NOTICES	存储课程相关通知信息
CHOICES	选课表
COMMENTS	课下闲聊板块中的评论信息表
COMMENTSUP	课下闲聊板块中的点赞次数表

COMMENTSDOWN	课下闲聊板块中的差评次数表
COMMENTSREPORT	课下闲聊板块中的举报次数表

2、数据库表具体设计

表名：STUDENTS			
字段名	中文意义	类型	说明
sid	学生 id	INT	约束：非空、自增、主键； STUDENTS 表每一行数据的编号
sum	学号	VARCHAR (20)	约束：UNIQUE、非空
passwd	密码	VARCHAR (20)	约束：非空
sname	学生姓名	VARCHAR (20)	约束：非空
sex	性别	VARCHAR (10)	约束：default 值为 ‘unknow’
nickname	昵称	VARCHAR (20)	约束：default 值为 ‘STUDENT’
avatar	头像 url	VARCHAR (500)	约束：default 值为系统默认头像 url
college	学院	VARCHAR (100)	约束：default 值为 ‘还没有填写学院’
introuduction	简介	VARCHAR (1000)	约束：default 值为 ‘还没有填写简介’
表名：TEACHERS			
字段名	中文意义	类型	说明
tid	教师 id	INT	约束：非空、自增、主键； TEACHERS 表每一行数据的编号
tnum	教工号	VARCHAR (20)	约束：UNIQUE、非空
passwd	密码	VARCHAR (20)	约束：非空
tname	教师姓名	VARCHAR (20)	约束：非空
sex	性别	VARCHAR (10)	约束：default 值为 ‘unknow’
nickname	昵称	VARCHAR (20)	约束：default 值为 ‘Teacher’
avatar	头像 url	VARCHAR (500)	约束：default 值为系统默认头像 url
college	学院	VARCHAR (100)	约束：default 值为 ‘还没有填写学院’
introduction	简介	VARCHAR (1000)	约束：default 值为 ‘还没有填写简介’
email	邮箱地址	VARCHAR (50)	约束：default 值为 ‘还没有填写邮箱’
phone	联系电话	VARCHAR (30)	约束：default 值为 ‘还没有填写联系方式’
office	办公地点	VARCHAR (30)	约束：default 值为 ‘还没有填写办公地点’
position	职称	VARCHAR (10)	约束：default 值为 ‘讲师’
表名：COURSES			
字段名	中文意义	类型	说明
cid	课程 id	INT	约束：非空、自增、主键； COURSES 表每一行数据的编号
tid	开课教师 id	INT	约束：非空、TEACHERS 表中 tid 的外键，设置级联删除

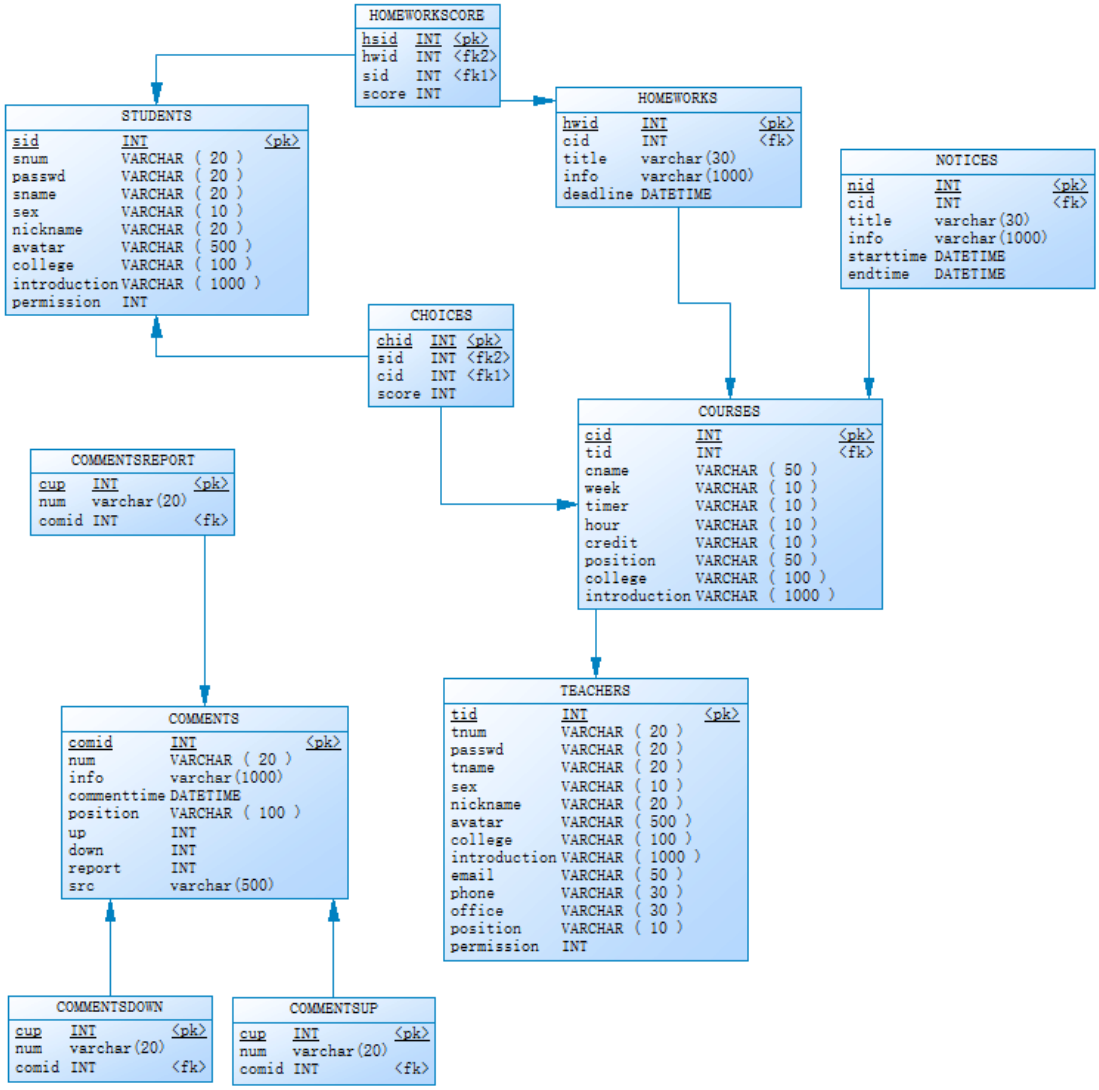
cname	课程名称	VARCHAR (50)	约束：非空
week	星期几上课	VARCHAR (10)	约束：非空
timer	第几节上课	VARCHAR (10)	约束：非空
hour	学时	VARCHAR (10)	约束：default 值为 0
credit	学分	VARCHAR (10)	约束：default 值为 0
position	上课地点	VARCHAR (50)	约束：default 值为‘未知’
college	开课学院	VARCHAR (100)	约束：default 值为‘未知’
introduction	课程简介	VARCHAR (1000)	约束：default 值为‘此课程还未填写简介’
表名：COMMENTS			
字段名	中文意义	类型	说明
comid	评论 id	INT	约束：非空、自增、主键； COMMENTS 表每一行数据的编号
num	评论者的学号/教工号	VARCHAR (20)	约束：非空
info	评论内容	VARCHAR (1000)	
commenttime	评论时间	DATETIME	
position	评论者位置	VARCHAR (100)	约束：default 值为‘无法获取位置’
up	被赞的次数	INT	约束：非空，default 值为 0
down	被差评次数	INT	约束：非空，default 值为 0
report	被举报次数	INT	约束：非空，default 值为 0
src	评论图片 src	varchar(500)	
表名：HOMEWORKS			
字段名	中文意义	类型	说明
hwid	作业 id	INT	约束：非空、自增、主键； HOMEWORKS 表每一行数据的编号
cid	课程 id	INT	约束：非空、COURSES 表中 cid 的外键，设置级联删除
title	作业标题	varchar(30)	
info	作业具体内容	varchar(1000)	
deadline	作业截止时间	DATETIME	
表名：HOMEWORKSCORE			
字段名	中文意义	类型	说明
hsid	作业评分 id	INT	约束：非空、自增、主键； HOMEWORKSCORE 表每一行数据的编号
hwid	作业 id	INT	约束：非空、HOMEWORKS 表中 hwid 的外键，设置级联删除
sid	学生 id	INT	约束：非空、STUDENTS 表中 sid 的外键，设置级联删除
score	作业得分	INT	约束：非空、default 值为 0
表名：NOTICES			
字段名	中文意义	类型	说明
nid	通知 id	INT	约束：非空、自增、主键；

			NOTICES 表每一行数据的编号
cid	课程 id	INT	约束：非空、COURSES 表中 sid 的外键，设置级联删除
title	通知标题	varchar(30)	
info	该课程通知具体内容	varchar(1000)	
starttime	发布时间	DATETIME	
endttime	销毁时间	DATETIME	这个时间的作用是我们的 app 会在该时间自动删除这条通知记录，以免记录数过多
表名：CHOICES			
字段名	中文意义	类型	说明
chid	选课 id	INT	约束：非空、自增、主键； CHOICES 表每一行数据的编号
sid	学生 id	INT	约束：非空、STUDENTS 表中 sid 的外键，设置级联删除
cid	课程 id	INT	约束：非空、COURSES 表中 sid 的外键，设置级联删除
score	该学生在该课程的得分	INT	约束：非空、default 值为 0
表名：COMMENTSUP			
字段名	中文意义	类型	说明
cup	id	INT	约束：非空、自增、主键； COMMENTSUP 表每一行数据的编号
num	点赞者的学号/学工号	varchar(20)	
comid	评论 id	INT	约束：非空、COMMENTS 表中 comid 的外键，设置级联删除
表名：COMMENTSDOWN			
字段名	中文意义	类型	说明
cup	id	INT	约束：非空、自增、主键； COMMENTSDOWN 表每一行数据的编号
num	点差评者的学号/学工号	varchar(20)	
comid	评论 id	INT	约束：非空、COMMENTS 表中 comid 的外键，设置级联删除
表名：COMMENTSREPORT			
字段名	中文意义	类型	说明
cup	id	INT	约束：非空、自增、主键； COMMENTSREPORT 表每行数据的编号
num	点差评者的学号/学工号	varchar(20)	
comid	评论 id	INT	约束：非空、COMMENTS 表中 comid 的

			外键，设置级联删除

3、数据库总 ER 图

根据上面的数据表的设计，我们得出 ER 图如下：



4、数据库实现

下面是我们创建数据库使用的 sql 语句

CHOICES 表：

```
CREATE TABLE CHOICES (
    chid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    sid INT not null,
    cid INT not null,
    score INT NOT NULL DEFAULT 0,
    foreign key (cid) references COURSES(cid) on delete cascade,
```

```
    foreign key (sid) references STUDENTS(sid) on delete cascade
) ENGINE = INNODB DEFAULT charset = utf8;
```

COMMENTS 表：

```
create table COMMENTS(
        comid INT not null AUTO_INCREMENT PRIMARY KEY,
        num VARCHAR ( 20 ) not null,
        info varchar(1000),
        commenttime DATETIME,
        position VARCHAR ( 100 ) DEFAULT '无法获取位置',--评论者位置
        up INT NOT NULL DEFAULT 0,
        down INT NOT NULL DEFAULT 0,
        report INT NOT NULL DEFAULT 0,
        src varchar(500)
)ENGINE = INNODB DEFAULT charset = utf8;
```

COMMENTSDOWN 表：

```
CREATE TABLE COMMENTSDOWN (
    cup INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    num varchar(20),
    comid INT NOT NULL,
    foreign key (comid) references COMMENTS(comid) on delete cascade
) ENGINE = INNODB DEFAULT charset = utf8;
```

COMMENTSREPORT 表：

```
CREATE TABLE COMMENTSREPORT (
    cup INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    num varchar(20),--举报者的学号教工号
    comid INT NOT NULL,
    foreign key (comid) references COMMENTS(comid) on delete cascade
) ENGINE = INNODB DEFAULT charset = utf8;
```

COMMENTSUP 表：

```
CREATE TABLE COMMENTSUP ( --评论赞数
    cup INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    num varchar(20),
    comid INT NOT NULL,
    foreign key (comid) references COMMENTS(comid) on delete cascade
) ENGINE = INNODB DEFAULT charset = utf8;
```

COURSES 表：

```
CREATE TABLE COURSES (
    cid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tid INT NOT NULL,
```

```
    cname VARCHAR ( 50 ) NOT NULL,
    week VARCHAR ( 10 ) NOT NULL,
    timer VARCHAR ( 10 ) NOT NULL,
    hour    VARCHAR ( 10 ) DEFAULT '0',
    credit VARCHAR ( 10 ) DEFAULT '0',
    position VARCHAR ( 50 ) DEFAULT '未知',
    college VARCHAR ( 100 ) DEFAULT '未知',
    introduction VARCHAR ( 1000 ) DEFAULT '此课程还未填写简介',
    foreign key (tid) references TEACHERS(tid) on delete cascade
) ENGINE = INNODB DEFAULT charset = utf8;
```

HOMEWORKS 表：

```
create table HOMEWORKS(
        hwid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        cid INT NOT NULL,
        title varchar(30),
        info varchar(1000),
        deadline DATETIME,
        foreign key (cid) references COURSES(cid) on delete cascade
)ENGINE = INNODB DEFAULT charset = utf8;
```

HOMEWORKSCORE 表：

```
create table HOMEWORKSCORE(
        hsid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        hwid INT NOT NULL,
        sid INT NOT NULL,
        score INT NOT NULL DEFAULT 0,
        foreign key (sid) references STUDENTS(sid) on delete cascade,
        foreign key (hwid) references HOMEWORKS(hwid) on delete cascade
)ENGINE = INNODB DEFAULT charset = utf8;
```

NOTICES 表：

```
create table NOTICES(
        nid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        cid INT NOT NULL,
        title varchar(30),
        info varchar(1000),
        starttime DATETIME,
        endtime DATETIME,
        foreign key (cid) references COURSES(cid) on delete cascade
)ENGINE = INNODB DEFAULT charset = utf8;
```

STUDENTS 表：

```
CREATE TABLE STUDENTS (
```

```
    sid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    snum VARCHAR ( 20 ) UNIQUE NOT NULL,
    passwd VARCHAR ( 20 ) NOT NULL,
    sname VARCHAR ( 20 ) NOT NULL,
    sex VARCHAR ( 10 ) DEFAULT 'unknow',
    nickname VARCHAR ( 20 ) DEFAULT 'STUDENT',
avatar VARCHAR ( 500 ) DEFAULT 'https://chonor.cn/Android/Avatar/defaultPhoto.png',
    college VARCHAR ( 100 ) DEFAULT '还没有填写学院',
    introduction VARCHAR ( 1000 ) DEFAULT '还没有填写简介',
    permission INT NOT NULL DEFAULT 0
) ENGINE = INNODB DEFAULT charset = utf8;
```

TEACHERS 表：

```
CREATE TABLE TEACHERS (
    tid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tnum VARCHAR ( 20 ) UNIQUE NOT NULL,
    passwd VARCHAR ( 20 ) NOT NULL,
    tname VARCHAR ( 20 ) NOT NULL,
    sex VARCHAR ( 10 ) DEFAULT 'unknow',
    nickname VARCHAR ( 20 ) DEFAULT 'Teacher',
avatar VARCHAR ( 500 ) DEFAULT 'https://chonor.cn/Android/Avatar/defaultPhoto.png',
    college VARCHAR ( 100 ) DEFAULT '还没有填写学院',
    introduction VARCHAR ( 1000 ) DEFAULT '还没有填写简介',
    email VARCHAR ( 50 ) DEFAULT '还没有填写邮箱',
    phone VARCHAR ( 30 ) DEFAULT '还没有填写联系方式',
    office VARCHAR ( 30 ) DEFAULT '还没有填写办公地点',
    position VARCHAR ( 10 ) DEFAULT '讲师',
    permission INT NOT NULL DEFAULT 0
) ENGINE = INNODB DEFAULT charset = utf8;
```

四、安卓系统中 java 代码的数据库部分

1、数据库操作底层封装类 MySqlHelper

这个类包括以下几个功能：

1. 连接数据库
2. 关闭数据库连接
3. 根据传入的 sql 语句执行 select 操作
4. 根据传入的 sql 语句执行 insert 操作
5. 根据传入的 sql 语句执行 update 操作
6. 根据传入的 sql 语句执行 delete 操作

注释代码如下：

```java
public class MySqlHelper {
    Connection conn = null;
    Statement stmt = null;

    //关闭数据库连接
    public void Close() throws SQLException {
        if (stmt!=null)stmt.close();
        if (conn!=null) conn.close();
        stmt = null;
        conn = null;
    }
    //获取数据库连接的类
    public Connection getConnection() throws SQLException,
            InstantiationException, IllegalAccessException,
            ClassNotFoundException {

        Connection Newconn = null;
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        String connectString = "jdbc:mysql://119.29.145.13:3306/android"
                + "?autoReconnect=true&useUnicode=true"
                + "&characterEncoding=UTF-8";
        String user = "android";
        String password = "123";
        Newconn = (Connection) DriverManager.getConnection(connectString,user,password);
        return Newconn;
    }
    //查
    public ResultSet select(String sql) throws Exception {
        conn = null;
        stmt = null;
        ResultSet rs = null;
        try {
            conn = getConnection();
            stmt = (Statement) conn.createStatement();
            rs = stmt.executeQuery(sql);
            return rs;
        } catch (SQLException sqle) {
            throw new SQLException("select data exception: "
                    + sqle.getMessage());
        } catch (Exception e) {
            throw new Exception("System e exception: " + e.getMessage());
        }
    }

    //改
    public void update(String sql) throws Exception {
        conn = null;
        PreparedStatement ps = null;
        try {
            conn = getConnection();
            ps =  (PreparedStatement)conn.prepareStatement(sql);
            ps.executeUpdate();
        } catch (SQLException sqle) {
            throw new Exception("update exception: " + sqle.getMessage());
        } finally {
            try {
                if (ps != null) {
                    ps.close();
                }
            } catch (Exception e) {
                throw new Exception("ps close exception: " + e.getMessage());
            }
        }
        try {
            if (conn != null) {
                conn.close();
            }
        } catch (Exception e) {
            throw new Exception("connection close exception: " + e.getMessage());
        }
    }
```

```
    //增
    public void insert(String sql) throws Exception {
        conn = null;
        PreparedStatement ps = null;
        try {
            conn = getConnection();
            ps = (PreparedStatement) conn.prepareStatement(sql);
            ps.executeUpdate();
        } catch (SQLException sqle) {
            throw new Exception("insert data exception: " + sqle.getMessage());
        } finally {
            try {
                if (ps != null) {ps.close();}
            } catch (Exception e) {
                throw new Exception("ps close exception: " + e.getMessage());
            }
        }
        try {
            if (conn != null) {conn.close();}
        } catch (Exception e) {
            throw new Exception("connection close exception: " + e.getMessage());
        }
    }

    public void delete(String sql) throws Exception {
        conn = null;
        PreparedStatement ps = null;
        try {
            conn = getConnection();
            ps =  (PreparedStatement) conn.prepareStatement(sql);
            ps.executeUpdate();
        } catch (SQLException sqle) {
            throw new Exception("delete data exception: " + sqle.getMessage());
        } finally {
            try {
                if (ps != null) {ps.close();}
            } catch (Exception e) {
                throw new Exception("ps close exception: " + e.getMessage());
            }
        }
        try {
            if (conn != null) {conn.close();}
        } catch (Exception e) {
            throw new Exception("connection close exception: " + e.getMessage());
        }
    }
}
```

2、教师信息相关操作 Teacherdb 类

A. 首先先获得一个数据库操作类

```
private MySqlHelper db;

public Teacherdb() { db = new MySqlHelper(); }
```

B. 根据教工号 num 查询信息

```
public ResultSet queryByNum(String Num)throws Exception {
    String sql = "select * from TEACHERS where tnum='" + Num + "'";
    ResultSet rs =db.select(sql);
    return rs;
}
```

```
public Teacher selectByNum(String Num)throws Exception {
    Teacher teacher=null;
    ResultSet rs =queryByNum(Num);
    if(rs.next()){
        teacher=new Teacher();
        teacher.setId(rs.getInt("tid"));
        teacher.setNum(rs.getString("tnum"));
        teacher.setSex(rs.getString("sex"));
        teacher.setNickname(rs.getString("nickname"));
        teacher.setPasswd(rs.getString("passwd"));
        teacher.setName(rs.getString("tname"));
        teacher.setAvatar(rs.getString("avatar"));
        teacher.setCollege(rs.getString("college"));
        teacher.setInfo(rs.getString("introduction"));
        teacher.setEmail(rs.getString("email"));
        teacher.setPhone(rs.getString("phone"));
        teacher.setOffice(rs.getString("office"));
        teacher.setPosition(rs.getString("position"));
    }
    rs.close();
    db.Close();
    return teacher;
}
```

C. 根据 tid 查询教师信息

```
public ResultSet queryById(int tid)throws Exception {
    String sql = "select * from TEACHERS where tid=" + tid;
    ResultSet rs =db.select(sql);
    return rs;
}
```

D. 查询所有教师信息，加入 limit 限制

```
public ResultSet queryAll(int st, int cnt) throws Exception {
    String sql = "select * from TEACHERS ";
    sql += "limit " + st +", " + cnt;
    ResultSet rs =db.select(sql);
    return rs;
}
```

E. 插入新的教师信息

```
public void insert(Teacher teacher)throws Exception{
    String sql="insert into " +
            " TEACHERS(tnum,passwd,tname,sex,nickname,avatar,college,introduction,email,phone,office,position) " +
            " values('"+ teacher.getNum()+"','"+
            teacher.getPasswd()+"','"+
            teacher.getName()+"','"+
            teacher.getSex() +"','"+
            teacher.getNickname()+"','"+
            teacher.getAvatar()+"','"+
            teacher.getCollege()+"','"+
            teacher.getInfo()+"','"+
            teacher.getEmail()+"','"+
            teacher.getPhone()+"','"+
            teacher.getOffice()+"','"+
            teacher.getPosition()+"')";
    db.insert(sql);
}
```

F. 更新指定教工号的教师信息

```
public void update(Teacher teacher)throws Exception{
    String sql="update TEACHERS set nickname='"+ teacher.getNickname()+"',"+
            " sex='"+ teacher.getSex()+"', "+
            " college='"+ teacher.getCollege()+"', "+
            " introduction='"+ teacher.getInfo()+"', "+
            " email='"+ teacher.getEmail()+"', "+
            " phone='"+ teacher.getPhone()+"', "+
            " office='"+ teacher.getOffice()+"', "+
            " position='"+ teacher.getPosition()+"', "+
            " avatar='"+teacher.getAvatar()+"' "+
            " where tnum='"+ teacher.getNum()+"'";
    db.update(sql);
}
```

3、学生信息相关操作 Studentdb 类

A. 首先先获得一个数据库操作类

```
private MySqlHelper db;

public Studentdb() { db = new MySqlHelper(); }
```

B. 根据学号 num 查询学生信息

```
public ResultSet queryByNum(String Num)throws Exception {
    String sql = "select * from STUDENTS where snum='" + Num + "'";
    ResultSet rs =db.select(sql);
    return rs;
}
public Student selectByNum(String Num)throws Exception{
    Student student=null;
    ResultSet rs =queryByNum(Num);
    if(rs.next()){
        student=new Student();
        student.setId(rs.getInt("sid"));
        student.setNum(rs.getString("snum"));
        student.setSex(rs.getString("sex"));
        student.setNickname(rs.getString("nickname"));
        student.setPasswd(rs.getString("passwd"));
        student.setName(rs.getString("sname"));
        student.setAvatar(rs.getString("avatar"));
        student.setCollege(rs.getString("college"));
        student.setInfo(rs.getString("introduction"));
    }
    rs.close();
    db.Close();
    return  student;
}
```

C. 根据 sid 查询学生信息

```
public ResultSet queryById(int sid)throws Exception {
    String sql = "select * from STUDENTS where sid=" + sid;
    ResultSet rs =db.select(sql);
    return rs;
}
```

D. 查询所有学生信息，加入 limit 限制

```
public ResultSet queryAll(int st, int cnt) throws Exception {
    String sql = "select * from STUDENTS ";
    sql += "limit " + st +", " + cnt;
    ResultSet rs =db.select(sql);
    return rs;
}
```

E. 插入新的学生信息

```
public void insert(Student student)throws Exception{
    String sql="insert into " +
            " STUDENTS(snum,passwd,sname,sex,nickname,avatar,college,introduction) " +
            " values('"+student.getNum()+"','"+student.getPasswd()+"','"+student.getName()+"','"+student.getSex()
            +"','"+student.getNickname()+"','"+student.getAvatar()+"','"+student.getCollege()+"','"+student.getInfo()+"')";
    db.insert(sql);
}
```

F. 更新指定学号的学生信息

```
public void update(Student student)throws Exception{
    String sql="update STUDENTS set nickname='"+student.getNickname()+"', sex='"+student.getSex()+"', " +
            "college='"+student.getCollege()+"', introduction='"+student.getInfo()+"', avatar='"+student.getAvatar()+"' " +
            "where snum='"+student.getNum()+"'";
    db.update(sql);
}
```

G. 查询选了某个课程的所有学生信息，即与 CHOICE 表做联合查询

```
public ArrayList<Student> selectByCid(int cid)throws Exception{
    ArrayList<Student>resules=new ArrayList<>();
    String sql = "SELECT STUDENTS.sid as sid,STUDENTS.snum as snum,STUDENTS.sname as sname,CHOICES.score as score" +
            " FROM STUDENTS,CHOICES" +
            " WHERE STUDENTS.sid=CHOICES.sid AND CHOICES.cid='"+cid+"'";
    ResultSet rs =db.select(sql);
    while (rs.next()){
        Student student=new Student();
        student.setName(rs.getString("sname"));
        student.setId(rs.getInt("sid"));
        student.setNum(rs.getString("snum"));
        student.setScore(rs.getInt("score"));
        resules.add(student);
    }
    return resules;
}
```

4、课程信息相关操作 Coursedb 类

A. 首先先获得一个数据库操作类

```
private MySqlHelper db;

public  Coursedb() { db = new MySqlHelper(); }
```

B. 根据课程编号 cid 查询课程信息

```
public ResultSet queryByCid(int cid)throws Exception {
    String sql = "select * from COURSES where cid=" + cid;
    ResultSet rs =db.select(sql);
    return rs;
}
```

C. 查询某个教师所开的所有课程

```
public ResultSet queryByTid(int tid)throws Exception {
    String sql = "select * from COURSES where tid=" + tid;
    ResultSet rs =db.select(sql);
    return rs;
}
```

D. 查询所有课程信息，加入 limit 限制

```
public ResultSet queryAll(int st, int cnt) throws Exception {
    String sql = "select * from COURSES ";
    sql += "limit " + st +", " + cnt;
    ResultSet rs =db.select(sql);
    return rs;
}
```

E.根据 cid 查询课程信息并使用与 teachers 表的联合查询加上教师名字信息。

```
public Course selectBycid(int cid)throws Exception {
    Course results=null;
    String sql="SELECT COURSES.cid as cid," +
            " COURSES.tid as tid," +
            " COURSES.cname as cname," +
            " COURSES.week as week," +
            " COURSES.timer as timer," +
            " COURSES.hour as hour," +
            " COURSES.credit as credit," +
            " COURSES.position as pos," +
            " COURSES.college as college," +
            " COURSES.introduction as info," +
            " TEACHERS.tname as tname " +
            " FROM COURSES,TEACHERS" +
            " WHERE COURSES.tid=TEACHERS.tid AND COURSES.cid='"+String.valueOf(cid) +"' "+
            " ORDER BY COURSES.cid DESC";
    ResultSet rs =db.select(sql);
    while (rs.next()) {
        Course course=new Course();
        course.setCid(rs.getInt("cid"));
        course.setTid(rs.getInt("tid"));
        course.setName(rs.getString("cname"));
        course.setWeek(rs.getString("week"));
        course.setTime(rs.getString("timer"));
        course.setHour(rs.getString("hour"));
        course.setCredit(rs.getString("credit"));
        course.setPos(rs.getString("pos"));
        course.setCollege(rs.getString("college"));
        course.setInfo(rs.getString("info"));
        course.setTname(rs.getString("tname"));
        course.setChoose(1);
        results=course;
    }
    rs.close();
    db.Close();
    return results;
}
```

E.根据 tid 与 teacher 表联合查询指定老师的课程信息（加入教师名字）

```
public ArrayList<Course> selectBytid(int tid)throws Exception {
    ArrayList<Course>results=new ArrayList<>();
    String sql="SELECT COURSES.cid as cid," +
            " COURSES.tid as tid," +
            " COURSES.cname as cname," +
            " COURSES.week as week," +
            " COURSES.timer as timer," +
            " COURSES.hour as hour," +
            " COURSES.credit as credit," +
            " COURSES.position as pos," +
            " COURSES.college as college," +
            " COURSES.introduction as info," +
            " TEACHERS.tname as tname " +
            " FROM COURSES,TEACHERS" +
            " WHERE COURSES.tid=TEACHERS.tid AND TEACHERS.tid='"+String.valueOf(tid) +"' "+
            " ORDER BY COURSES.cid DESC";
    ResultSet rs =db.select(sql);
    while (rs.next()){
        Course course=new Course();
        course.setCid(rs.getInt("cid"));
        course.setTid(rs.getInt("tid"));
        course.setName(rs.getString("cname"));
        course.setWeek(rs.getString("week"));
        course.setTime(rs.getString("timer"));
        course.setHour(rs.getString("hour"));
        course.setCredit(rs.getString("credit"));
        course.setPos(rs.getString("pos"));
        course.setCollege(rs.getString("college"));
        course.setInfo(rs.getString("info"));
        course.setTname(rs.getString("tname"));
        course.setChoose(1);
        results.add(course);
    }
    rs.close();
    db.Close();
    return results;
}
```

F.根据 sid，与 CHOICES 表、TEACHERS 表、COURSES 表联合查询指定学生的选课信息，报告课程信息和开课教师姓名。

```
public ArrayList<Course> selectBysid(int sid)throws Exception {
    ArrayList<Course>results=new ArrayList<>();
    String sql="SELECT " +
            " COURSES.cid as cid," +
            " COURSES.tid as tid, " +
            " COURSES.cname as cname," +
            " COURSES.week as week," +
            " COURSES.timer as timer," +
            " COURSES.hour as hour," +
            " COURSES.credit AS credit," +
            " COURSES.position as pos," +
            " COURSES.college as college," +
            " COURSES.introduction as info," +
            " TEACHERS.tname as tname " +
            " FROM COURSES,TEACHERS,CHOICES" +
            " WHERE COURSES.tid=TEACHERS.tid AND " +
            " COURSES.cid=CHOICES.cid AND " +
            " CHOICES.sid='"+String.valueOf(sid)+"' "+
            " ORDER BY COURSES.cid DESC";
    ResultSet rs =db.select(sql);
    while (rs.next()) {
        Course course=new Course();
        course.setCid(rs.getInt("cid"));
        course.setTid(rs.getInt("tid"));
        course.setName(rs.getString("cname"));
        course.setWeek(rs.getString("week"));
        course.setTime(rs.getString("timer"));
        course.setHour(rs.getString("hour"));
        course.setCredit(rs.getString("credit"));
        course.setPos(rs.getString("pos"));
        course.setCollege(rs.getString("college"));
        course.setInfo(rs.getString("info"));
        course.setTname(rs.getString("tname"));
        course.setChoose(1);
        results.add(course);
    }
    rs.close();
    db.Close();
    return results;
}
```

G.查询所有课程信息。这里有点不同的是还加入了是学生来查询还是教师来查询。在应用中有一个显示所有课程页面，如果是教师去查询，会标记自己开的课，如果是学生去查询，会标记已经选了的课。那么这里查询也做了标记操作。对于老师会直接设置标记。由于还需要去 CHOICE 表查询，那么所以暂时不做标记。

```
public ArrayList<Course> selectAll(int st, int cnt,int sid,int tid)throws Exception {
    ArrayList<Course>results=new ArrayList<>();
    String sql="SELECT " +
            " COURSES.cid as cid," +
            " COURSES.tid as tid, " +
            " COURSES.cname as cname," +
            " COURSES.week as week," +
```

```
            " COURSES.timer as timer," +
            " COURSES.hour as hour," +
            " COURSES.credit AS credit," +
            " COURSES.position as pos," +
            " COURSES.college as college," +
            " COURSES.introduction as info," +
            " TEACHERS.tname as tname " +
            " FROM COURSES,TEACHERS" +
            " WHERE COURSES.tid=TEACHERS.tid " +
            " ORDER BY COURSES.cid DESC " +
            " limit " + st +" , " + cnt;
    ResultSet rs =db.select(sql);
    while (rs.next()) {
        Course course=new Course();
        course.setCid(rs.getInt("cid"));
        course.setTid(rs.getInt("tid"));
        course.setName(rs.getString("cname"));
        course.setWeek(rs.getString("week"));
        course.setTime(rs.getString("timer"));
        course.setHour(rs.getString("hour"));
        course.setCredit(rs.getString("credit"));
        course.setPos(rs.getString("pos"));
        course.setCollege(rs.getString("college"));
        course.setInfo(rs.getString("info"));
        course.setTname(rs.getString("tname"));
        if(sid!=-1){
            course.setChoose(0);
        }
        else if(tid!=-1){
            if(course.getTid()==tid)course.setChoose(1);
            else course.setChoose(0);
        }
        else course.setChoose(0);
        results.add(course);
    }
    rs.close();
    db.Close();
    return results;
}
```

H. APP 中有搜索课程功能，可以根据课程名或开课学院或开课教师做模糊条件搜索。这里将这三个模糊条件查询合在一起写。因为不需要使用某个条件只要将该条件置空那么查询出来的结果就是所有信息。

```
public ArrayList<Course> selectBySearch(String info)throws Exception {
    ArrayList<Course>results=new ArrayList<>();
    String sql="SELECT " +
            " COURSES.cid as cid," +
            " COURSES.tid as tid, " +
            " COURSES.cname as cname," +
            " COURSES.week as week," +
            " COURSES.timer as timer," +
            " COURSES.hour as hour," +
            " COURSES.credit AS credit," +
            " COURSES.position as pos," +
            " COURSES.college as college," +
            " COURSES.introduction as info," +
            " TEACHERS.tname as tname " +
            " FROM COURSES,TEACHERS" +
            " WHERE COURSES.tid=TEACHERS.tid AND " +
            " (COURSES.cname LIKE '%"+info+"%' OR " +
            " COURSES.college LIKE '%"+info+"%' OR " +
            " TEACHERS.tname LIKE '%"+info+"%') " +
            " ORDER BY COURSES.cid DESC ";
```

```
    ResultSet rs =db.select(sql);
    while (rs.next()) {
        Course course=new Course();
        course.setCid(rs.getInt("cid"));
        course.setTid(rs.getInt("tid"));
        course.setName(rs.getString("cname"));
        course.setWeek(rs.getString("week"));
        course.setTime(rs.getString("timer"));
        course.setHour(rs.getString("hour"));
        course.setCredit(rs.getString("credit"));
        course.setPos(rs.getString("pos"));
        course.setCollege(rs.getString("college"));
        course.setInfo(rs.getString("info"));
        course.setTname(rs.getString("tname"));
        course.setChoose(0);
        results.add(course);
    }
    rs.close();
    db.Close();
    return results;
}
```

I. 插入新的课程信息

```
public void insert(Course course)throws Exception{
    String sql="insert into " +
            " COURSES(tid,cname,week,timer,hour,credit,position,college,introduction) " +
            " values('"+ course.getTid()+"','"+
            course.getName()+"','"+
            course.getWeek()+"','"+
            course.getTime()  +"','"+
            course.getHour()+"','"+
            course.getCredit()+"','"+
            course.getPos()+"','"+
            course.getCollege()+"','"+
            course.getInfo()+"')";
    db.insert(sql);
}
```

J. 更新课程信息

```
public void update(Course course)throws Exception{
    String sql="update COURSES set " +
            " tid='"+ course.getTid()+ "'," +
            " cname='"+ course.getName()+"', " +
            " week='"+ course.getWeek()+"', " +
            " timer='"+ course.getTime()+"' " +
            " hour='"+ course.getHour()+"' " +
            " credit='"+ course.getCredit()+"' " +
            " college='"+ course.getCollege()+"' " +
            " introduction='"+ course.getInfo()+"' " +
            "where cid='"+ course.getCid()+"'";
    db.update(sql);
}
```

K. 更新课程信息

5、选课信息相关操作 Choicedb 类

A. 根据 sid 查询指定学生的选课信息

```
public ResultSet querySid(int sid) throws Exception {
    String sql = "select * from CHOICES "+
            " where sid ='"+sid+"' ";
    ResultSet rs =db.select(sql);
    return rs;
}
```

B.根据 sid 和 cid 查询指定学生某门课程的分数

```
public Integer queryscore(int sid,int cid) throws Exception {
    String sql = "select * from CHOICES "+
            " where sid ='"+sid+"' AND cid='"+cid+"' ";
    ResultSet rs =db.select(sql);
    int score=-1;
    if(rs.next()) {
        score=rs.getInt("score");
    }
    rs.close();
    db.Close();
    return score;
}
```

C.根据 sid 和 cid 查询某个学生是否选择了某门课程

```
public boolean querySelect(int cid,int sid)throws Exception{
    String sql = "select * from CHOICES "+
            " where sid ='"+sid+"' AND cid='"+cid+"' ";
    ResultSet rs =db.select(sql);
    boolean flag=false;
    if(rs.next())flag=true;
    rs.close();
    close();
    return flag;
}
```

E.插入新选课信息、更新课程分数、删除选课信息

```
public void insert(int cid,int sid)throws Exception{
    String sql="insert into " +
            " CHOICES(cid,sid) " +
            " values('"+ cid+"','"+ sid+"')";
    db.insert(sql);
}

public void update(int cid,int sid,int score)throws Exception{
    String sql="update CHOICES set " +
            " score='"+score +"' " +
            " where cid='"+ cid +"'AND sid='"+sid+"' ";
    db.update(sql);
}

public void delete(int cid,int sid) throws Exception{
    String sql="delete from CHOICES "+
            " where cid='"+ cid +"'AND sid='"+sid+"' ";
    db.delete(sql);
}
```

6、作业信息相关操作 Homeworkdb 类

A.根据 cid 查询某门课程的所有作业信息：

```
public ResultSet queryBycid(int cid) throws Exception {
    String sql = "select * from HOMEWORKS "+
            " where cid ='"+cid+"' " +
            " ORDER BY hwid DESC";
    ResultSet rs =db.select(sql);
    return rs;
}


public ArrayList<Homework> selectBycid(int cid) throws Exception {
    ArrayList<Homework>results=new ArrayList<>();
    ResultSet rs=queryBycid(cid);
    while(rs.next()){
        Homework homework=new Homework();
        homework.setCid(rs.getInt("cid"));
        homework.setTitle(rs.getString("title"));
        homework.setHwid(rs.getInt("hwid"));
        homework.setInfo(rs.getString("info"));
        homework.setDdl(rs.getString("deadline"));
        results.add(homework);
    }
    rs.close();
    db.Close();
    return results;
}
```

B.根据作业编号 hwid 查询指定作业信息

```
public Homework selectByhwid(int hwid) throws Exception {
    Homework homework=null;
    String sql = "select * from HOMEWORKS "+
            " where hwid ='"+hwid+"' ";
    ResultSet rs =db.select(sql);
    if(rs.next()){
        homework=new Homework();
        homework.setHwid(rs.getInt("hwid"));
        homework.setCid(rs.getInt("cid"));
        homework.setTitle(rs.getString("title"));
        homework.setInfo(rs.getString("info"));
        homework.setDdl(rs.getString("deadline"));
    }
    rs.close();
    close();
    return homework;
}
```

C.查询指定学生在某个 deadline 需要交的作业数，方便发通知通知该学生

```
public Integer CountToday(int sid,String date)throws Exception {
    int cnt=0;
    String sql="SELECT COUNT(*) AS cnt " +
            "FROM HOMEWORKS,CHOICES,STUDENTS " +
            "WHERE STUDENTS.sid = CHOICES.sid AND " +
            "CHOICES.cid=HOMEWORKS.cid AND " +
            "STUDENTS.sid="+sid+" AND " +
            "HOMEWORKS.deadline='"+date+"' ";
    ResultSet rs = db.select(sql);
    if(rs.next()){
        cnt=rs.getInt("cnt");
    }
    rs.close();
    close();
    return cnt;
}
```

D.插入新的作业信息

```
public void insert(Homework homework)throws Exception{
    String sql="insert into " +
            " HOMEWORKS(cid,title,info,deadline) " +
            " values('"+ homework.getCid()+"','"+homework.getTitle()+"','"+homework.getInfo()+"','"+homework.getDdl()+"')";
    db.insert(sql);
}
```

E.更新作业信息

```
public void update(Homework homework)throws Exception{
    String sql="update HOMEWORKS set " +
            " title='"+homework.getTitle() +"' " +
            " info='"+homework.getInfo() +"' " +
            " deadline='"+homework.getDdl() +"' " +
            " where hwid='"+homework.getHwid();
    db.update(sql);
}
```

F. 向作业评分表插入某个学生某个作业的分数

```
public void insertScore(int hwid,int sid,int score)throws Exception{
    String sql="insert ignore into " +
            " HOMEWORKSCORE(hsid,hwid,sid)" +
            " values('"+ (hwid+sid)+"','"+ hwid+"','"+ sid+"','"+score+"')";
    db.update(sql);
}
```

G.删除指定作业信息

```
public void delete(Homework homework) throws Exception{
    String sql="delete from HOMEWORKS "+
            " where hwid='"+ homework.getHwid() +"'";
    db.delete(sql);
}
```

7、课下闲聊板块评论相关操作 Commentdb 类

A.查询所有评论，加入 limit

```
public ResultSet queryAll(int st, int cnt) throws Exception {
    String sql = "select * from COMMENTS ";
    sql += " ORDER BY comid DESC";
    sql += " limit " + st +", " + cnt;
    ResultSet rs =db.select(sql);
    return rs;
}
```

B.插入新的评论信息

```
public void insert(Comment comment)throws Exception{
    String sql="insert into " +
            " COMMENTS(num,info,ctime,position,src) " +
            " values('"+ comment.getNum()+"'," +
            "'"+comment.getInfo()+"'," +
            "'"+comment.getTime()+"'," +
            "'"+comment.getPos()+"'," +
            "'"+comment.getSrc() +"')";
    db.insert(sql);
}
```

C.更新某条评论的点赞数、差评数以及举报数

```
//点赞数
public void updateup(int comid, int up)throws Exception{
    String sql="update COMMENTS set up='"+ up+
            "' where comid='"+comid+"'";
    db.update(sql);
}
//差评数
public void updatedown(int comid, int down)throws Exception{
    String sql="update COMMENTS set down='"+ down+
            "' where comid='"+comid+"'";
    db.update(sql);
}
//举报数
public void updatereport(int comid, int report)throws Exception{
    String sql="update COMMENTS set report='"+ report +
            "' where comid='"+comid+"'";
    db.update(sql);
}
```

D.删除评论

```
public void delete(Comment comment) throws Exception{
    String sql="delete from COMMENTS "+
            " where comid='"+comment.getCid()+"'";
    db.delete(sql);
}
public void delete(int comid) throws Exception{
    String sql="delete from COMMENTS "+
            " where comid='"+comid+"'";
    db.delete(sql);
}
```

E.对某条评论插入新的点赞、差评、举报信息

```
public void insertup(int comid , String num)throws Exception{
    String sql="insert into " +
            " COMMENTSUP(num, comid) " +
            " values('"+ num+"'" +
            ",'"+comid +"')";
    db.insert(sql);
}
public void insertdown(int comid , String num)throws Exception{
    String sql="insert into " +
            " COMMENTSDOWN(num, comid) " +
            " values('"+ num+"'" +
            ",'"+comid +"')";
    db.insert(sql);
}
public void insertreport(int comid , String num)throws Exception{
    String sql="insert into " +
            " COMMENTSREPORT(num, comid) " +
            " values('"+ num+"'" +
            ",'"+comid +"')";
    db.insert(sql);
}
```

F.判断某条评论是否有点赞、差评、举报信息

```
public boolean selectup(int comid ,String num)throws Exception{
    String sql = "SELECT * FROM COMMENTSUP WHERE comid='"+comid+"' AND num='"+num+"'";
    ResultSet rs =db.select(sql);
    boolean flag=false;
    if(rs.next())flag=true;
    rs.close();
    close();
    return flag;
}
public boolean selectdown(int comid ,String num)throws Exception{
    String sql = "SELECT * FROM COMMENTSDOWN WHERE comid='"+comid+"' AND num='"+num+"'";
    ResultSet rs =db.select(sql);
    boolean flag=false;
    if(rs.next())flag=true;
    rs.close();
    close();
    return flag;
}
public boolean selectreport(int comid ,String num)throws Exception{
    String sql = "SELECT * FROM COMMENTSREPORT WHERE comid='"+comid+"' AND num='"+num+"'";
    ResultSet rs =db.select(sql);
    boolean flag=false;
    if(rs.next())flag=true;
    rs.close();
    close();
    return flag;
}
```

8、通知相关操作 Noticedb 类

A.查询某门课程的所有通知

```
public ResultSet querycid(int cid) throws Exception {
    String sql = "select * from NOTICES "+
            " where cid ='"+cid+"'";
    ResultSet rs =db.select(sql);
    return rs;
}
```

B.根据通知编号 nid，与 COURSES 表联合查询某门课程的某个通知信息

```
public Notice selectBynid(int nid)throws Exception{
    Notice results=null;
    String sql = "SELECT NOTICES.nid as nid," +
            " NOTICES.cid as cid , " +
            " NOTICES.title as title," +
            " NOTICES.info as info," +
            " NOTICES.starttime as starttime," +
            " NOTICES.endtime as endtime," +
            " COURSES.cname as cname" +
            " FROM NOTICES,COURSES " +
            " WHERE NOTICES.cid=COURSES.cid AND NOTICES.nid ='"+nid+"'";
    ResultSet rs =db.select(sql);
    while(rs.next()){
        Notice notice=new Notice();
        notice.setCid(rs.getInt("cid"));
        notice.setNid(rs.getInt("nid"));
        notice.setTitle(rs.getString("title"));
        notice.setInfo(rs.getString("info"));
        notice.setStarttime(rs.getString("starttime"));
```

```
            notice.setEndtime(rs.getString("endtime"));
            notice.setCname(rs.getString("cname"));
            results=notice;
        }
        rs.close();
        db.Close();
        return  results;
    }
```

C.根据教师编号 tid，与 COURSES 表联合查询某个老师发布的所有通知信息

```
public ArrayList<Notice> selectBytid(int tid)throws Exception{
    ArrayList<Notice>results =new ArrayList<>();
    String sql="SELECT NOTICES.nid as nid," +
            " NOTICES.cid as cid," +
            " NOTICES.title as title," +
            " NOTICES.info as info," +
            " NOTICES.starttime as starttime," +
            " NOTICES.endtime as endtime," +
            " COURSES.tid as tid," +
            " COURSES.cname as cname " +
            " FROM NOTICES,COURSES" +
            " WHERE NOTICES.cid=COURSES.cid AND " +
            " COURSES.tid='" +String.valueOf(tid)+"'"+
            " ORDER BY nid DESC";
    ResultSet rs = db.select(sql);
    while(rs.next()) {
        Notice notice=new Notice();
        notice.setCname(rs.getString("cname"));
        notice.setCid(rs.getInt("cid"));
        notice.setNid(rs.getInt("nid"));
        notice.setTitle(rs.getString("title"));
        notice.setInfo(rs.getString("info"));
        notice.setStarttime(rs.getString("starttime"));
        notice.setEndtime(rs.getString("endtime"));
        results.add(notice);
    }
    rs.close();
    db.Close();
    return  results;
}
```

D.根据学生编号 sid，与 CHOICES、COURSES 表联合查询指定学生接收的所有通知信息。

```
public ArrayList<Notice> selectBysid(int sid)throws Exception{
    ArrayList<Notice>results =new ArrayList<>();
    String sql="SELECT NOTICES.nid as nid," +
            " NOTICES.cid as cid," +
            " NOTICES.title as title," +
            " NOTICES.info as info," +
            " NOTICES.starttime as starttime," +
            " NOTICES.endtime as endtime," +
            " CHOICES.sid as sid, " +
            " COURSES.cname as cname" +
            " FROM NOTICES,CHOICES,COURSES" +
            " WHERE NOTICES.cid=CHOICES.cid AND NOTICES.cid=COURSES.cid AND " +
            " CHOICES.sid='" +String.valueOf(sid)+"'"+
            " ORDER BY nid DESC";
    ResultSet rs = db.select(sql);
    while(rs.next()) {
        Notice notice=new Notice();
        notice.setCid(rs.getInt("cid"));
        notice.setNid(rs.getInt("nid"));
```

```
            notice.setTitle(rs.getString("title"));
            notice.setInfo(rs.getString("info"));
            notice.setStarttime(rs.getString("starttime"));
            notice.setEndtime(rs.getString("endtime"));
            notice.setCname(rs.getString("cname"));
            results.add(notice);
        }
        rs.close();
        db.Close();
        return  results;
    }
```

E.根据课程标号 cid，查询所有跟该课程相关的所有通知，这里课程信息可以通过调用 Coursedb 类来查询获得。

```
public ArrayList<Notice> selectBycid(int cid)throws Exception{
    ArrayList<Notice>results=new ArrayList<>();
    Coursedb coursedb=new Coursedb();
    Course course= coursedb.selectBycid(cid);
    coursedb.close();

    ResultSet rs=querycid(cid);
    while(rs.next()){
        Notice notice=new Notice();
        notice.setCid(rs.getInt("cid"));
        notice.setNid(rs.getInt("nid"));
        notice.setTitle(rs.getString("title"));
        notice.setInfo(rs.getString("info"));
        notice.setStarttime(rs.getString("starttime"));
        notice.setEndtime(rs.getString("endtime"));
        notice.setCname(course.getName());
        results.add(notice);
    }
    rs.close();
    db.Close();
    return  results;
}
```

F.某个日期发布的所有通知数量

```
public Integer CountToday(int sid,String date)throws Exception {
    int cnt=0;
    String sql="SELECT COUNT(*) AS cnt " +
            "FROM NOTICES,CHOICES,STUDENTS " +
            "WHERE STUDENTS.sid = CHOICES.sid AND " +
            "CHOICES.cid=NOTICES.cid AND " +
            "STUDENTS.sid="+sid+" AND " +
            "NOTICES.starttime='"+date+"' ";
    ResultSet rs = db.select(sql);
    if(rs.next()){
        cnt=rs.getInt("cnt");
    }
    rs.close();
    close();
    return cnt;
}
```

G.插入新通知信息、更新通知信息、删除指定通知

```
public void insert(Notice notice)throws Exception{

    String sql="insert into " +
            " NOTICES(cid,title,info,starttime,endtime) " +
            " values('"+ notice.getCid()+"','"+ notice.getTitle()+
            "','"+notice.getInfo()+"','"+notice.getStarttime()+"','"+notice.getEndtime()+"')";
    db.insert(sql);

}

public void update(Notice notice)throws Exception{
    String sql="update NOTICES set " +
            " info='"+notice.getInfo() +"' " +
            " starttime='"+notice.getStarttime() +"' " +
            " endtime='"+notice.getEndtime() +"' " +
            " where nid='"+notice.getNid();
    db.update(sql);
}

public void delete(Notice notice) throws Exception{
    String sql="delete from NOTICES "+
            " where nid='"+ notice.getNid() +"'";
    db.delete(sql);
}
```

五、应用实际运行效果

在此部分，将展示应用各功能，需要注意的是，本应用所有功能均需在联网状态下进行。

1) 登录或注册

在登录账号时，若第一次登录，则需要填写学号（或教工号）与密码，并进行验证码验证即可登录；若非第一次登录且上次未退出登录状态，则可直接进入个人账号。

在注册账号的时候，个人头像可暂不设置，需注意输入框提示为必填的选项必须填写，学号要求 8 位而教工号要求 6 位，教师账号注册时需要填写额外身份信息如联系方式、办公室地址等，并且每个输入项中输入框右下角都有字数限制的提示，若未填必填项或超出字数限制，将会被拒绝注册申请：

图 2-a 头像支持图库和拍照

图 2-b 超出限定字数报错

图 2-c 必填项未填报错

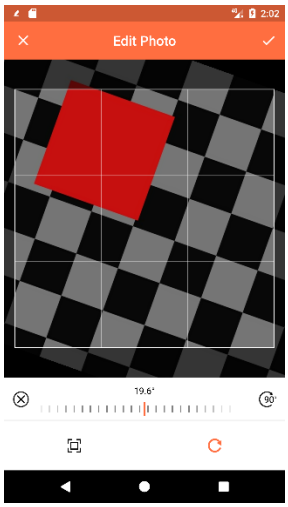


图 2-d 头像支持图片裁剪旋转

图 2-e 设置头像后效果

2) 学生选课/退课

在课程列表页面中长按课程即可进行选课/退课操作，无法选择同一时间段已有其他课程的课，已选课程会标记出来。

图 3-a 已选课程的标记

图 3-b 长按选课/退课

图 3-c 与已有课程时间冲突

而对于教师账号要进行新课程的添增的时候，需要填写课程信息，同时，要注意下列事项：

a) 课程名称不可为空；
b) 学时、学分不可为空；
c) 上课地点不可为空；

填写信息后，点击下方“添加课程”按钮即可添加新课程

图 4 自定义新课程信息

3) 搜索课程

对课程进行搜索，需要进入“课程列表”版块（图 5-a），进入页面后，可通过直接输入课程名字进行查找（图 5-b），在这里，支持部分关键字筛选，此时下拉页面进行刷新则默认取消搜索。

图 5-a 课程列表页面

图 5-b 输入课程名字进行查找

图 5-c 批量筛选

4) 下课聊介绍

在“下课聊”版块（图 6-a）中，可以对应用进行以下操作：

a) 发布新帖子（图 6-b），可上传本地图片或拍照，可对图片进行裁剪

b) 对他人的发帖进行赞或踩（图 6-c），此处为重复点赞。需要注意的

是，当帖子举报数目超过 20 将被删帖

图 6-a “下课聊”页面

图 6-b 发布新帖子

图 6-c 对他人发帖进行评价

5) 查看课程通知

查看课程的通知，首先，最为直接的是当您收到通知的时候，应用将会有对应的推送即刻通知您。而在软件中，首先可在“我”版块中，点击“查看我的通知”来查看所有通知，其次也可以通过点入某一课程来查看该课程的通知。

图 7-a 通知信息内容

图 7-b “我”版块中的入口

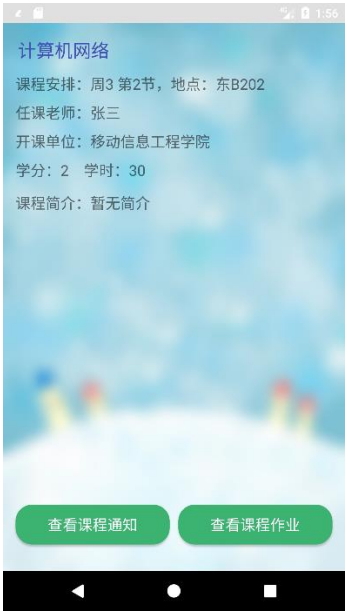


图 7-c 课程信息中的入口

图 7-d 接收到推送通知

6) 发布或查看课程作业

教师账号可在其任课课程内发布课程作业：进入该课程详情界面后，点击下方“查看课程作业”按钮后可查看过去发布作业记录，点击右下方加号图标后填写对应信息即可发布课程作业。

而学生账号查看课程作业的入口基本与查看课程通知的入口一致，需要点击“查看课程作业”按钮。

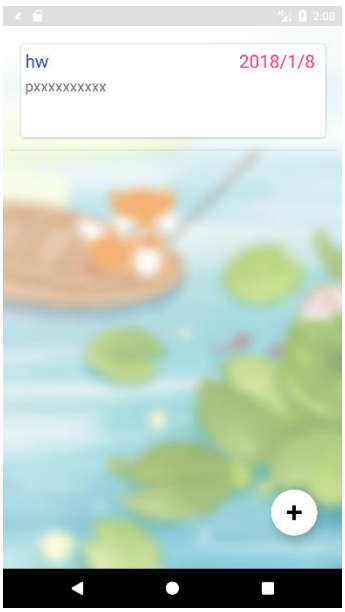


图 8-a 教师查看过往作业记录

图 8-b 教师发布课程作业

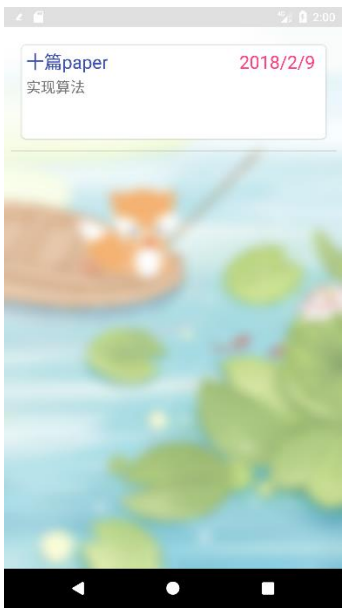


图 7-c 学生查看课程作业内容

7) 个人信息的修改

在“我”版块中，点击“编辑我的信息”即可对用户资料进行对应的修改，其中，学号/教工号一旦注册将不可修改。

图 9-a ”我”版块中的入口

图 9-b 学生账号编辑信息视图

图 9-c 教账号编辑信息视图

8) 关于课程评分

教师可长按“我的课程”以打开学生列表，并对学生进行评分，而选修了该课程的学生可在教师评分后，点击该课程详情以查看自己在该课程中所获得的分数。

图 10-a 教师评分

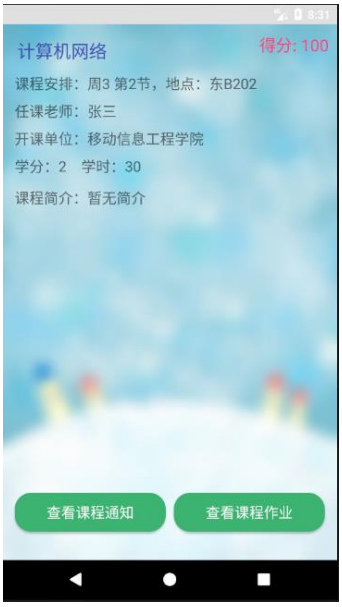


图 10-b 学生查看个人分数

六、实验总结

这次实验设计是对本学期所学数据库知识的一次应用。目的在于将我们实验课上所学知识，应用到现实中去。通过自主设计设计出一个有实用性的数据库系统。那么这次实验数据库部分最大的困难就是根据需求定数据库表。需要多少张

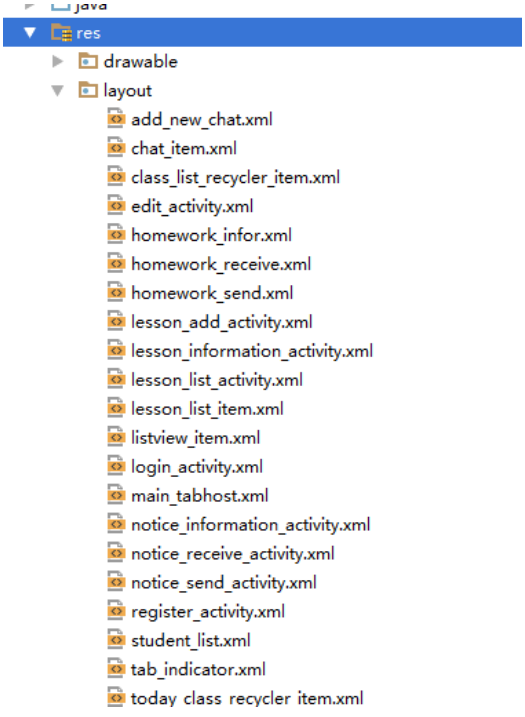
表？某种信息如何存储？需不需要将一种实体的信息拆分存储？这是我们在设计过程中讨论最多的问题。幸运地是，到最后，我们顺利地完成了设计。我们感觉自己收获了很多。感谢老师和 TA 一个学期以来地辛勤教导。

七、实验分工

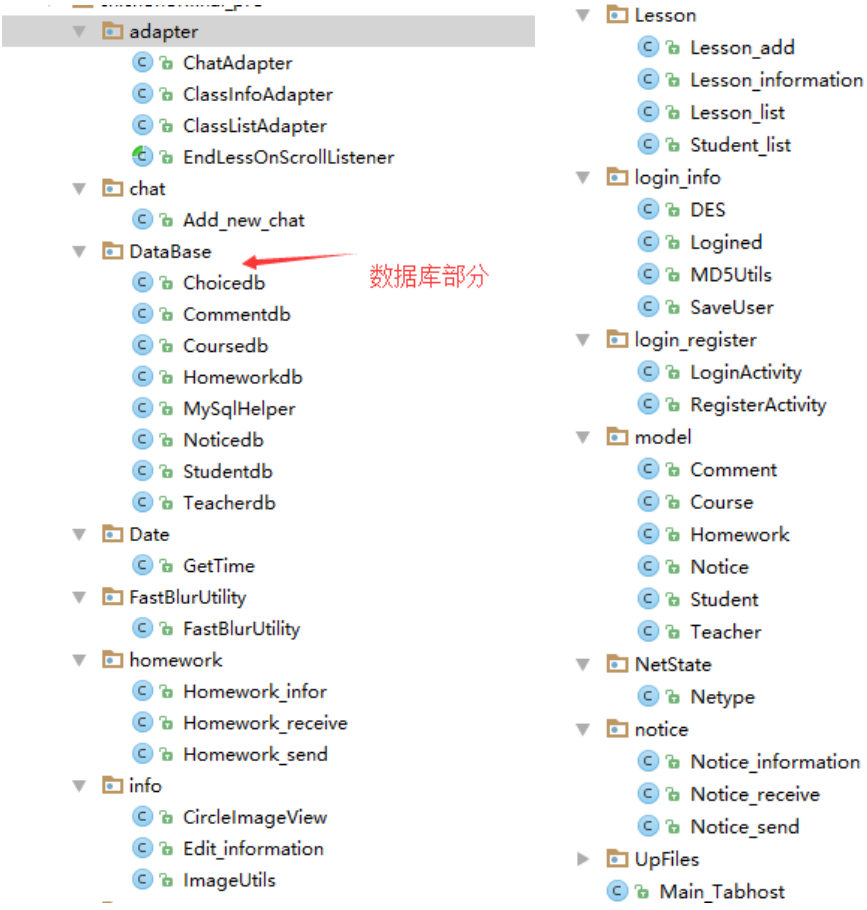
学号	姓名	工作	贡献度
15352127	黄洁莹	前端设计、数据库初表设计、部分后端设计	25%
15352427	张子豪	前端设计、数据库初表设计、部分后段设计	25%
15352008	蔡荣裕	部分后端设计、部分数据库操作接口设计	25%
15352408	张镓伟	部分后端设计、部分数据库操作接口设计、sql 语句建表实现	25%

八、附录

前端文件结构：



后端文件结构：


adapter
ChatAdapter
ClassInfoAdapter
ClassListAdapter
EndLessOnScrollListener
chat
Add_new_chat
DataBase
Choicedb
Commentdb
Coursedb
Homeworkdb
MySqlHelper
Noticedb
Studentdb
Teacherdb
数据库部分
Date
GetTime
FastBlurUtility
FastBlurUtility
homework
Homework_infor
Homework_receive
Homework_send
info
CircleImageView
Edit_information
ImageUtils
Lesson
Lesson_add
Lesson_information
Lesson_list
Student_list
login_info
DES
Logined
MD5Utils
SaveUser
login_register
LoginActivity
RegisterActivity
model
Comment
Course
Homework
Notice
Student
Teacher
NetState
Netype
notice
Notice_information
Notice_receive
Notice_send
UpFiles
Main_Tabhost