实验三说明文档

杨天悦 2018302140018

陈宥伊 2018302100123

# 实现功能

1. 用户输入学工号、姓名、密码，实现注册或登录功能，进入主界面。
2. 若用户身份为学生，则主界面显示该学生所选课程列表；

若用户身份为教师，则主界面显示该教师所教授的课程列表。

1. 学生点击所选课程列表中任一课程，可进行撤课操作；

教师点击授课课程列表中任一课程，可修改课程信息，或删除该课程。

1. 点击主界面添加课程按钮，

学生可输入课头号添加课程；

教师可输入课程名、课程类型、课程进度、课代表，添加一项新的课程。

# 实现方法

1.利用装饰器模式，将数据库不同权限的操作分别装饰给学生和老师这两种不同的用户身份。

①创建接口UserOp，定义用户行为LogIn()和LogOut()。

②创建抽象类User类和UserDec类，均继承自UserOp。

③创建Student类和Teacher类，继承自抽象类User；创建StudentDec类和TeacherDec类，继承自抽象类UserDec。

④在StudentDec类中，添加方法ApplyForRepre()，进行课代表申请操作；在TeacherDec类中，添加方法SetRepre()，进行课代表设置操作。

1. 用StudentDec和TeacherDec装饰User类的对象，以实现学生与老师的不同操作。

先在CourseService中设置共同的增删改查操作接口，再将具体的操作实现在MysqlClient中，再分别在继承于CourseService的StudentCS和TeacherCS中调用MysqpClient中不同的函数。接着建立CourseServiceDecorator类，设置一个CourseService类型的字段，并自定义构造器，在每次实例化这个类的时候决定是学生权限还是老师权限。这样在用户对课程的相关数据库进行操作的时候，只需要知道他的具体身份，就会拥有对应的权限。这在一定程度上也达到保障数据库数据安全的目的。

2.设计过滤器模式，筛选出满足一定条件的课程，便于之后的查询。这样提高了查询操作的效率，不必在每次查询时，都要进行数据库的访问。

① 创建抽象类Criteria类，定义方法MeetCriteria()。

② 创建CriteriaCompulsory、CriteriaOptional、CriteriaPublic、CriteriaSpecialized四个类，继承自Criteria类，使用方法MeetCriteria()实现具体的过滤操作。

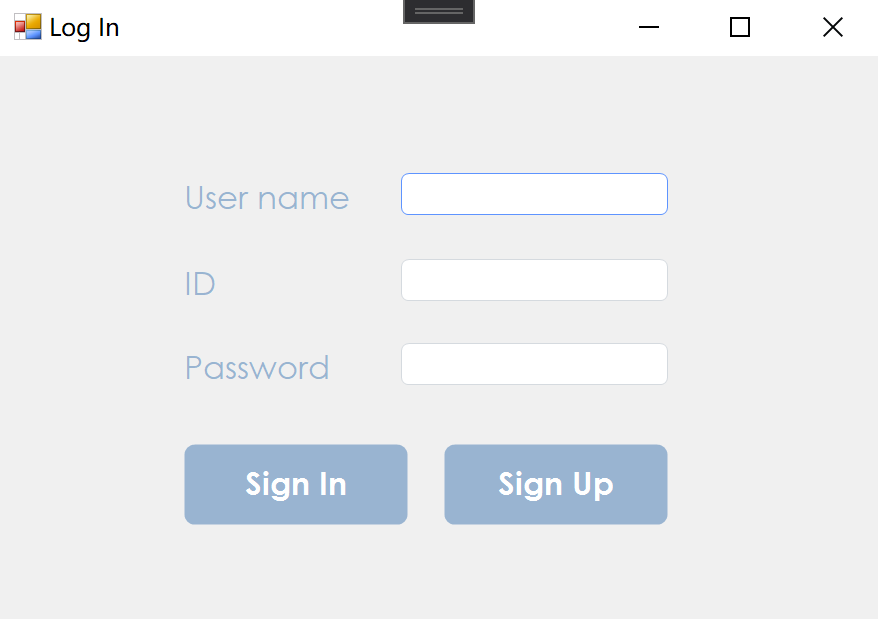
③ 在查询方法中，根据过滤标准，选择用不同的构造方法初始化Criteria类的对象，对课程列表进行过滤筛选。

3.利用EntityFramwork框架，自动建立数据库及表。在建立CourseController类后，即可通过GET、POST、PUT、DELETE对应的url在前端使用对应不同的函数，便于教师与学生用户进行课程的增删改查操作。

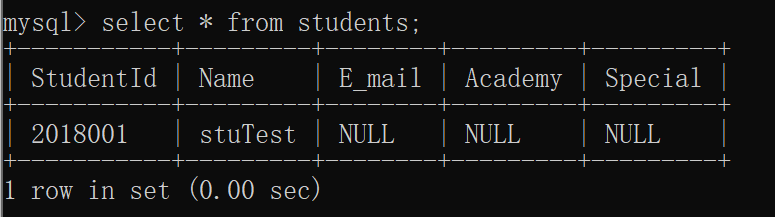
# 实现效果

1. **用户登录注册**

用户登录注册界面：

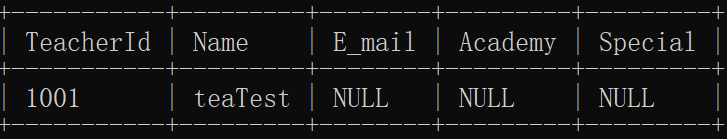


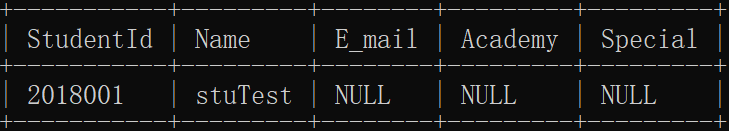
注册后数据库记录更新：



1. **主界面**

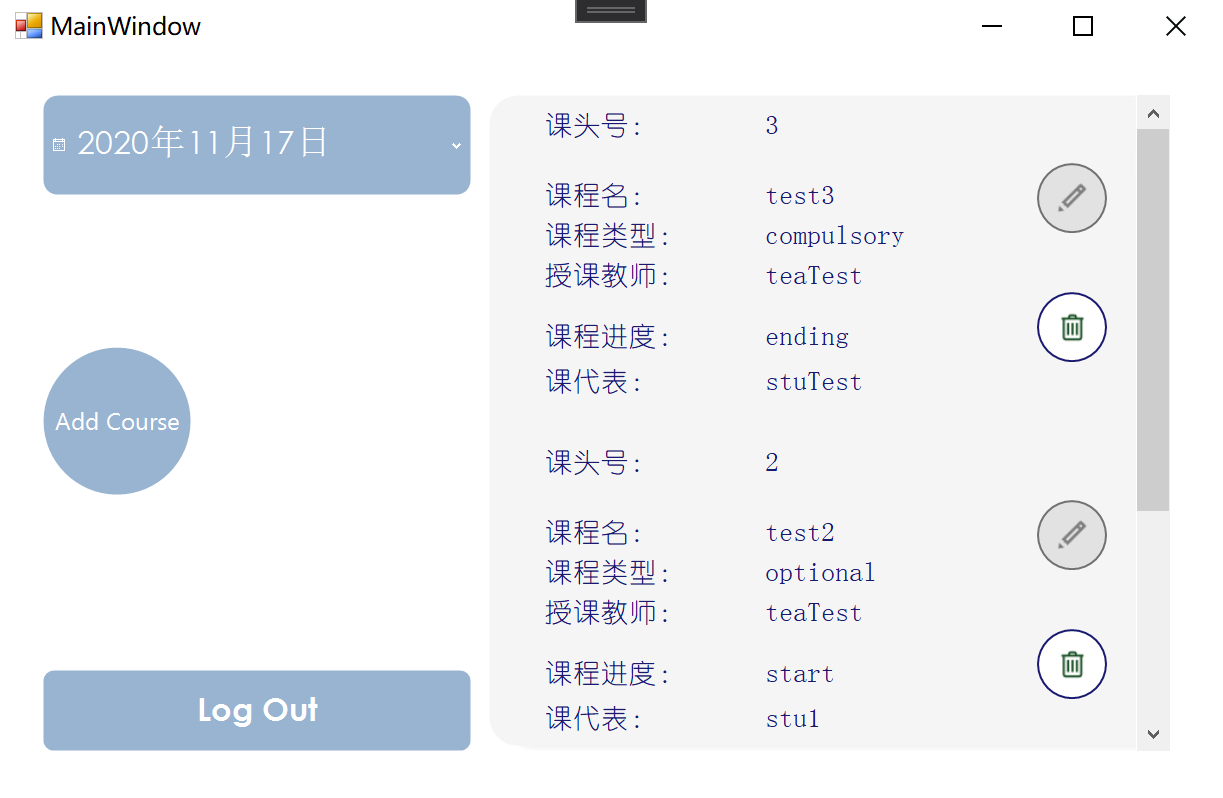
创建学生测试账号，用户名为stuTest，学号为2018001；创建教师测试账号，用户名为teaTest，学工号为1001：

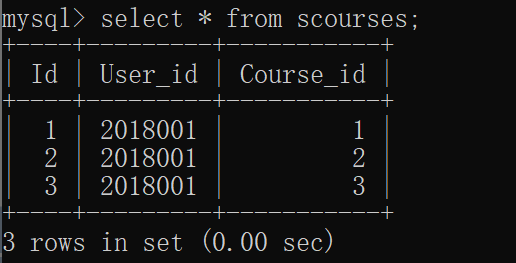




1. 用户身份为学生：

显示学生所选全部课程（学生无权限修改课程信息，编辑按钮不可点击）：





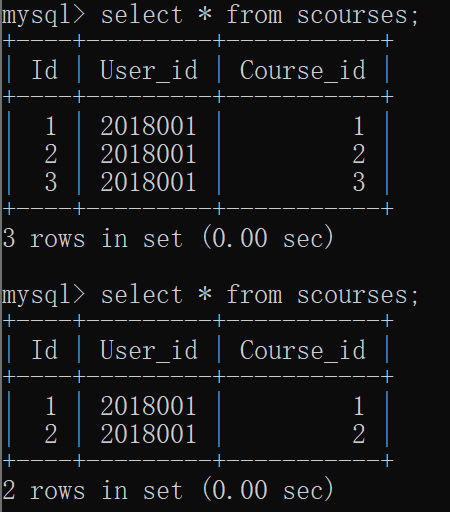
学生点击任一课程进行撤课操作：



撤课操作完成后主界面刷新：

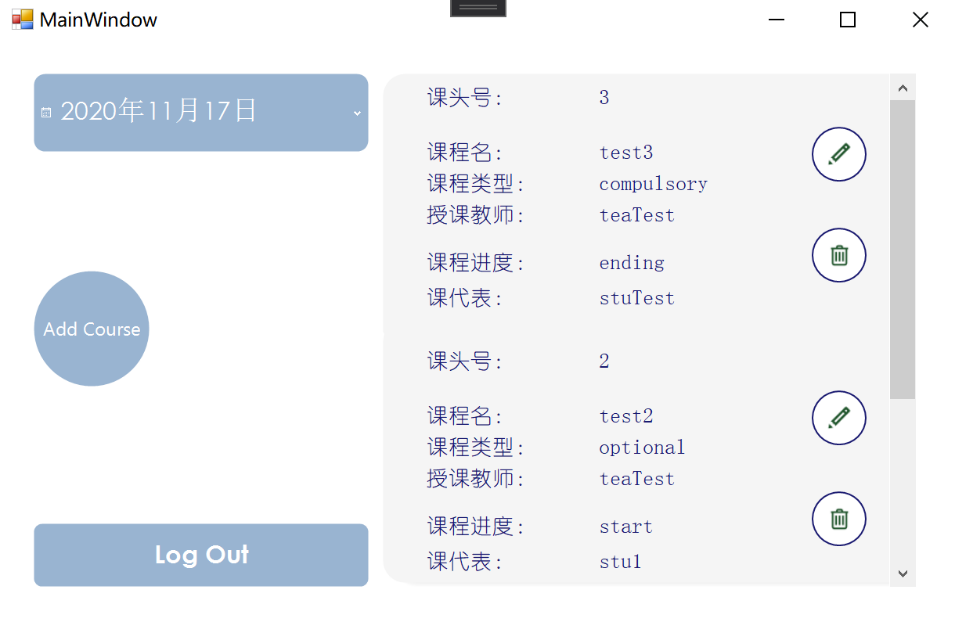


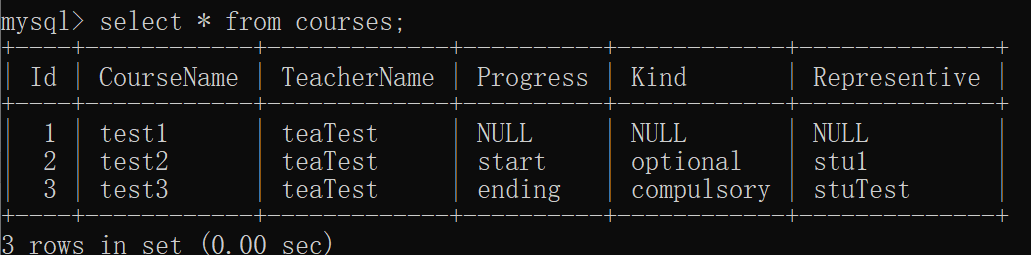
对应数据库scourses表更新：



2. 用户身份为教师：

显示教师授课课程列表：



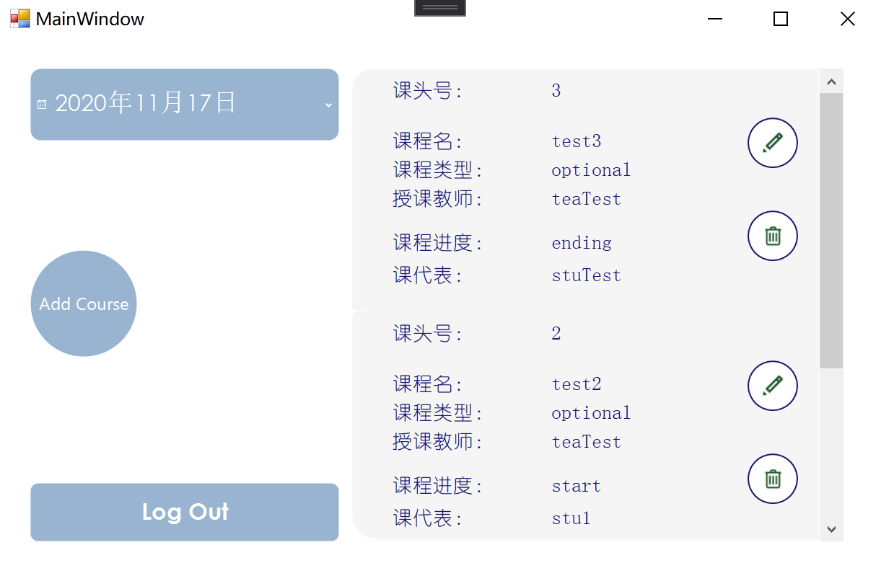


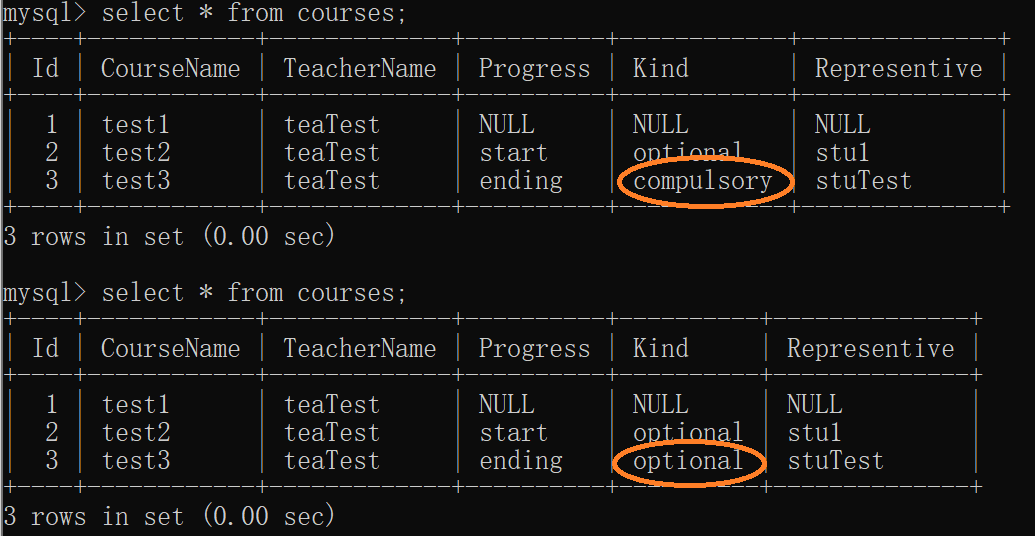
教师点击任一课程编辑按钮修改课程信息（以修改课程类型为例）：





编辑课程后主界面刷新显示：

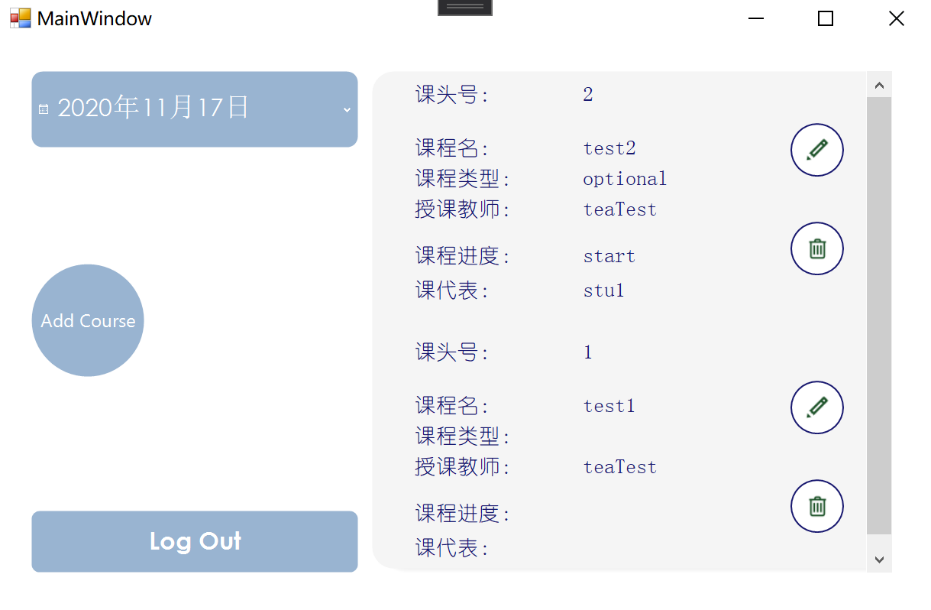




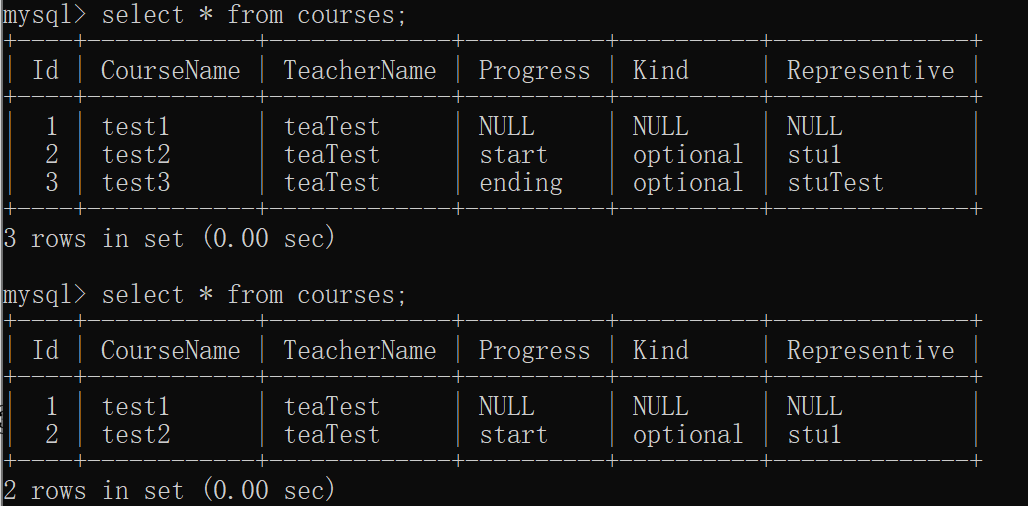
教师点击任一课程删除按钮，删除该课程：



删除课程后主界面刷新：



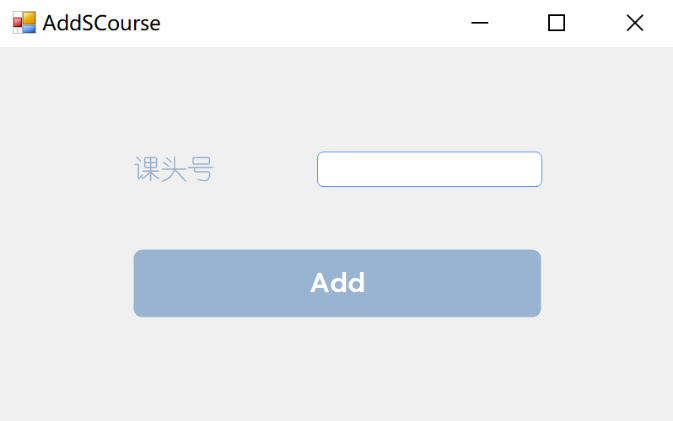
对应数据库courses表更新：



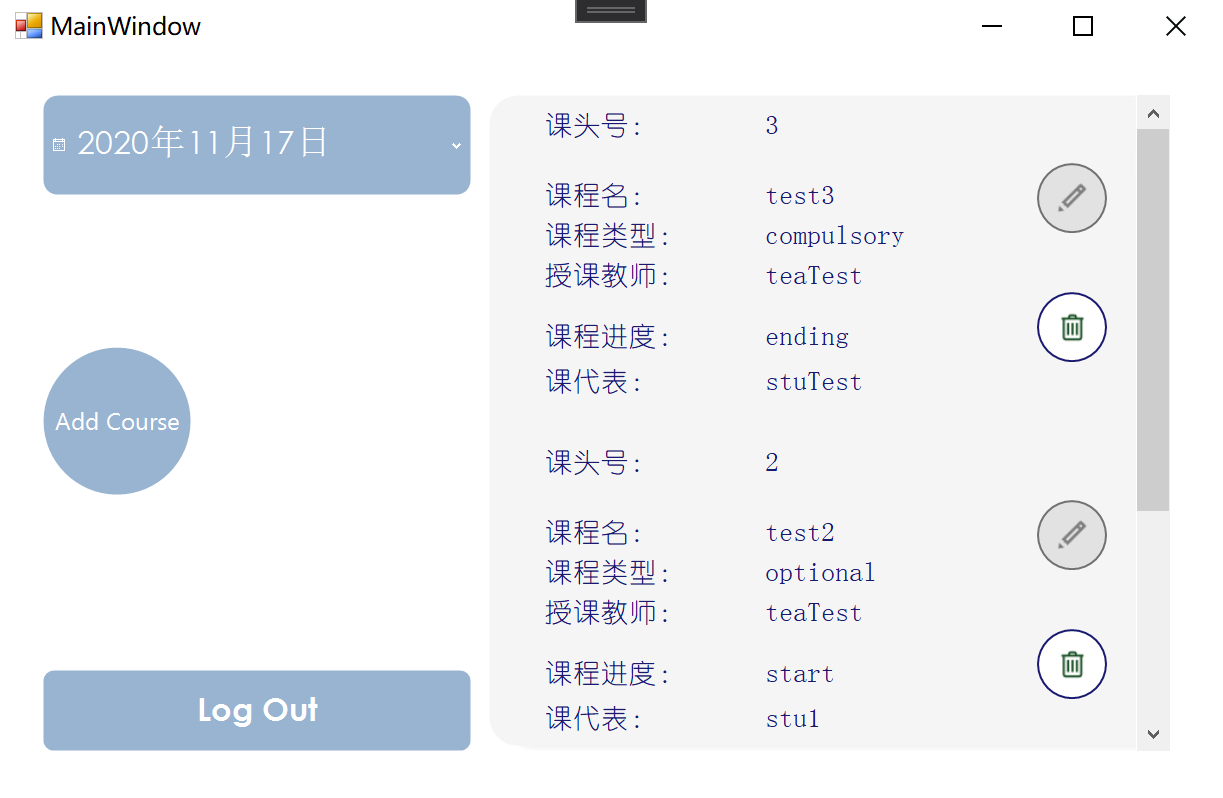
1. **添加课程**

1.用户身份为学生：

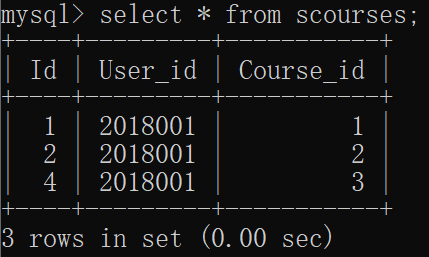
学生输入课头号，添加课程：



添加课程成功后主界面刷新显示课程列表：

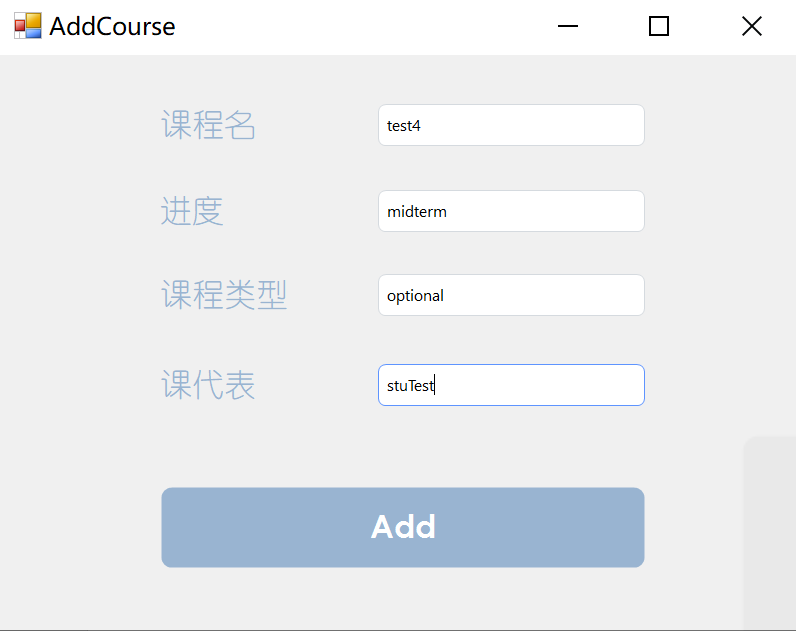


对应数据库scourses表更新：

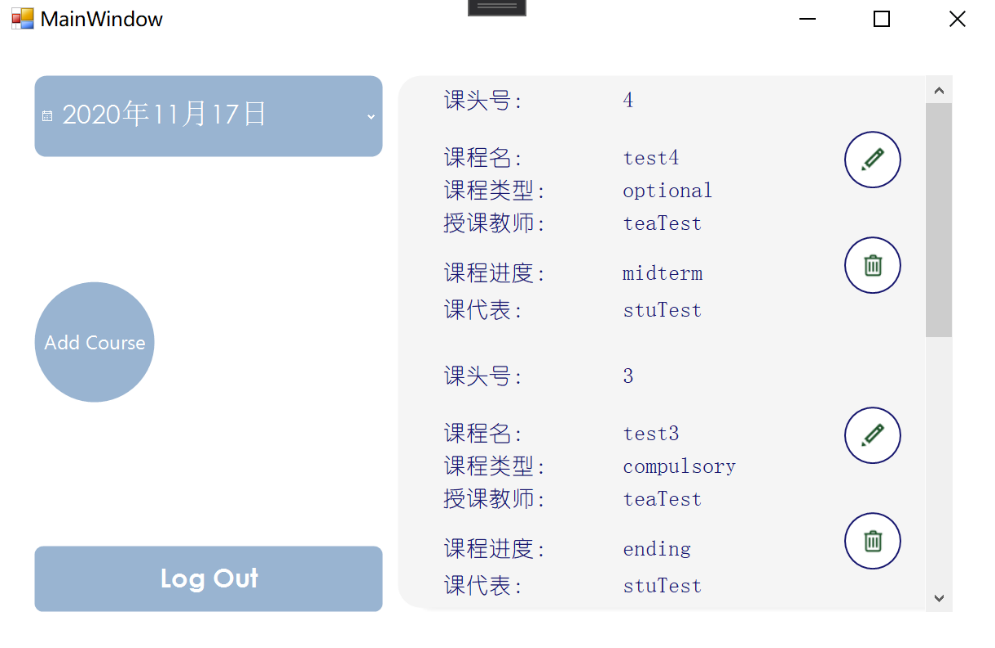


2.用户身份为教师：

输入课程名、课程类型、进度、课代表，添加课程



添加课程成功后，自动生成课头号，主界面刷新显示课程列表：



对应数据库courses表更新：

