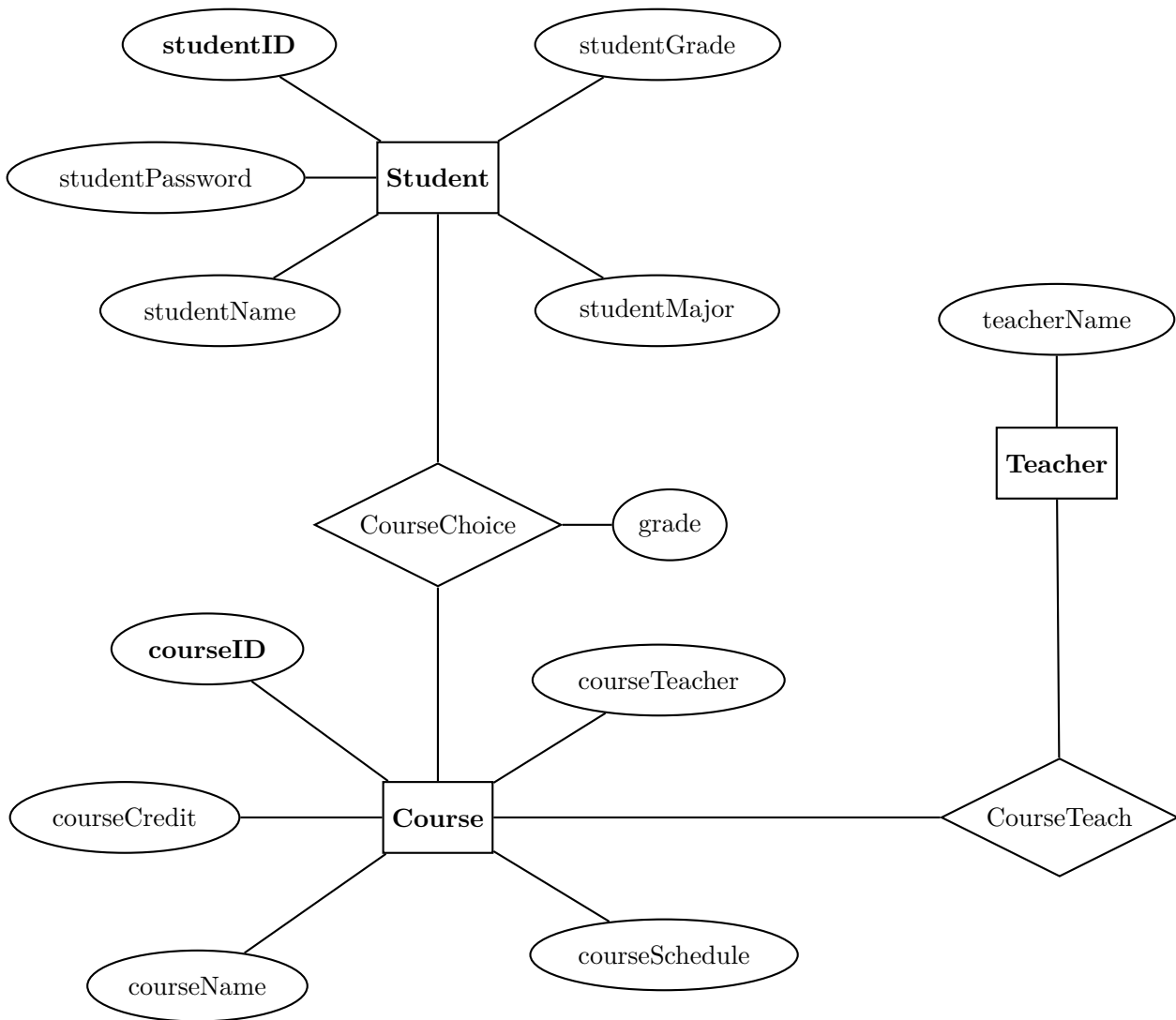


学生选课系统

5 组

2024-12-07

1 ER 图




2 组员及贡献

学号	姓名	代码	文档写作
1210604006	胡 xx	23%	22%
1221005172	李 xx	18%	22%
1221005174	刘 xx	18%	22%
1221005176	汤 xx	23%	17%
1221005177	王 x	18%	17%

表 1: 分工表

1210604006 胡 xx 代码：23% 写作：22%
1221005172 李 xx 代码：18% 写作：22%
1221005174 刘 xx 代码：18% 写作：22%
1221005176 汤 xx 代码：23% 写作：17%
1221005177 王 x 代码：18% 写作：17%

 <https://github.com/AlionFan/StudentSelectCourse>

3 工具、平台以及参考资料

在本项目中，我们使用以下编程环境和技术栈：

- 数据库：MySQL 9.0
- 后端：Python 3.12
- 前端：PySide6
- 文档撰写：LaTeX
- 参考资料：数据库系统原理及 MySQL 应用教程（第二版）实验指导书

摘要

通过 MySQL 与 python 结合，在主菜单输入学号和密码登录选课界面，进行选课。系统通过简洁直观的图形化界面，实现了学生学号和密码验证登录，以及通过拖拽上下表格选择操作完成选课的功能。最精彩之处在于系统的表格选择选课功能，它不仅提高了用户体验，而且通过前后端的高效协同，实现了实时的选课状态更新，展现了一种新颖的交互设计思路。

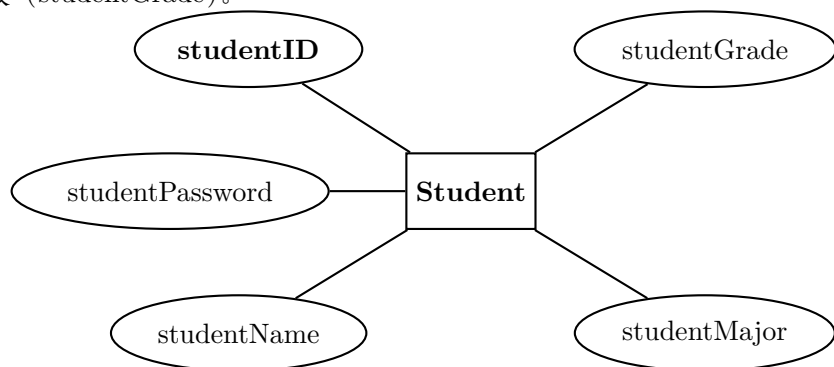
需求分析

1. 为了保证系统的安全性，学生需要**输入学号与密码登录**系统。
2. 为了保证选课系统较高的可读性，选课系统采用菜单栏的方式进行操作，保证学生对系统操作的顺利。
3. 系统需要能够**检查课程时间冲突**，避免学生选择时间冲突的课程。
4. **检查是否选课**，避免重复选择同一门课。
5. 设置最大选课人数，并实时统计每门课程选择人数，当选课人数达到最大人数，不允许选课。

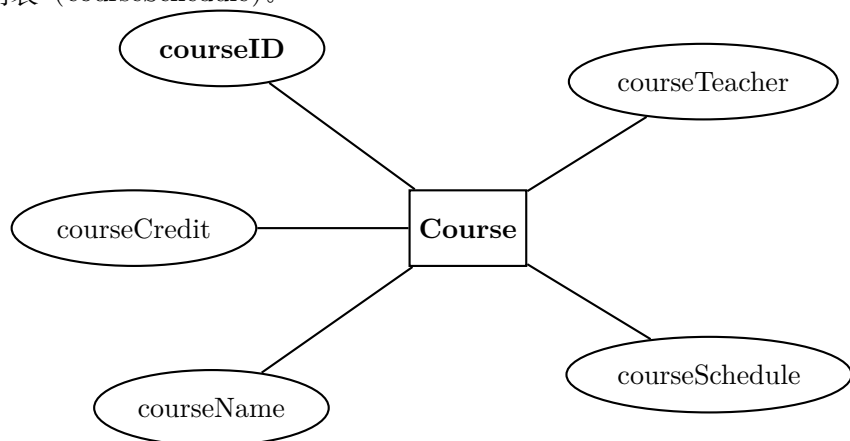
概念结构设计

我们定义了以下实体和属性：

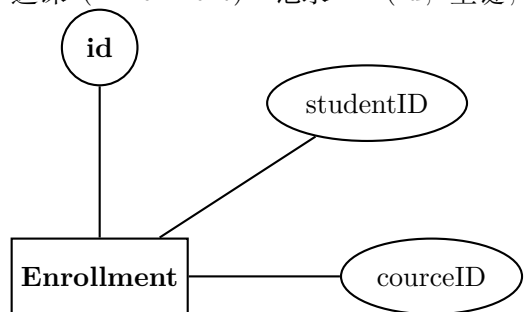
学生 (Student)：学号 (studentID, 主键)、姓名 (studentName)、密码 (studentPassword)、专业 (studentMajor)、年级 (studentGrade)。



课程 (Course)：课程 ID (courseID, 主键)、课程名称 (courseName)、学分 (courseCredit)、教师 (courseTeacher)、时间表 (courseSchedule)。



选课 (Enrollment)：记录 ID (id, 主键, 自增)、学生 ID (studentID)、课程 ID (courseID)。



4 逻辑结构设计

4.1 关系模式设计

1. 学生表 (students)

studentID(INT): 学生的唯一标识符，作为主键。

studentName(VARCHAR): 学生的姓名。

studentPassword(VARCHAR): 学生的密码。

studentMajor(VARCHAR): 学生的专业。

studentGrade(VARCHAR): 学生的年级。

2. 课程表 (courses)

courseID(INT): 课程的唯一标识符, 作为主键。

courseName(VARCHAR): 课程的名称。

courseCredit(INT): 课程的学分。

courseTeacher(VARCHAR): 教授该课程的教师。

courseSchedule(VARCHAR): 课程的时间表。

3. 选课表 (enrollments)

id(INT): 选课记录的唯一标识符, 作为主键, 自增。

studentID(INT): 选课学生的学号, 外键关联学生表。

courseID(INT): 所选课程的课程 ID, 外键关联课程表。

学生选课视图: 结合 students、courses 和 enrollments 表, 提供一个包含学生信息和他们所选课程信息的视图, 方便查询学生选课情况。

课程冲突视图: 查询学生已选课程的时间表, 与新选课程的时间表进行比较, 检查是否存在时间冲突。

优化:

索引: 为 studentID (选课表)、courseID (选课表) 等外键字段创建索引, 提高查询效率。

约束: 确保数据完整性, 如使用外键约束、唯一约束等。

查询优化: 通过分析查询模式, 优化 SQL 查询语句, 减少全表扫描, 提高查询性能。

数据库设计:

students 表: 存储学生信息。

courses 表: 存储课程信息。

enrollments 表: 存储学生的选课信息, 通过外键与学生表和课程表关联。

图形化界面设计:

图形化界面设计包括一个登录界面和主界面。登录界面包含学号和密码输入框以及登录按钮, 主界面包含课程列表、已选课程列表以及选课和退课按钮。界面简洁直观, 易于操作。

5 结论

本文总结了学生选课系统的设计与实现过程, 强调了系统的表格选择选课功能在提升用户体验和操作效率方面的重要性。一个优秀的系统设计不仅需要满足功能需求, 更应该注重用户体验和操作的便捷性。

参考文献、书籍、网络内容: 数据库系统原理及 MySQL 应用教程 (第二版) 实验指导书。

6 附录

6.1 展示

6.1.1 登录界面

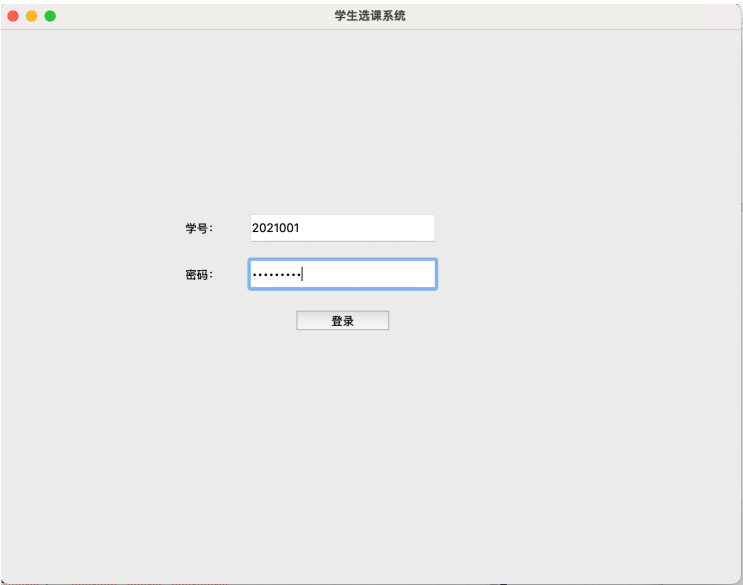


图 1: 学生登录界面



图 2: 放大



图 3: 密码正确

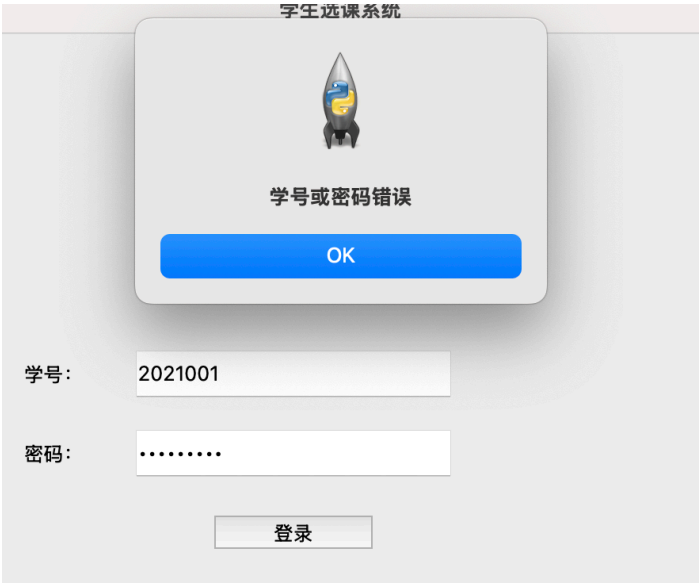


图 4: 密码错误

为了便于操作和展示，默认设置之前有过选课记录。

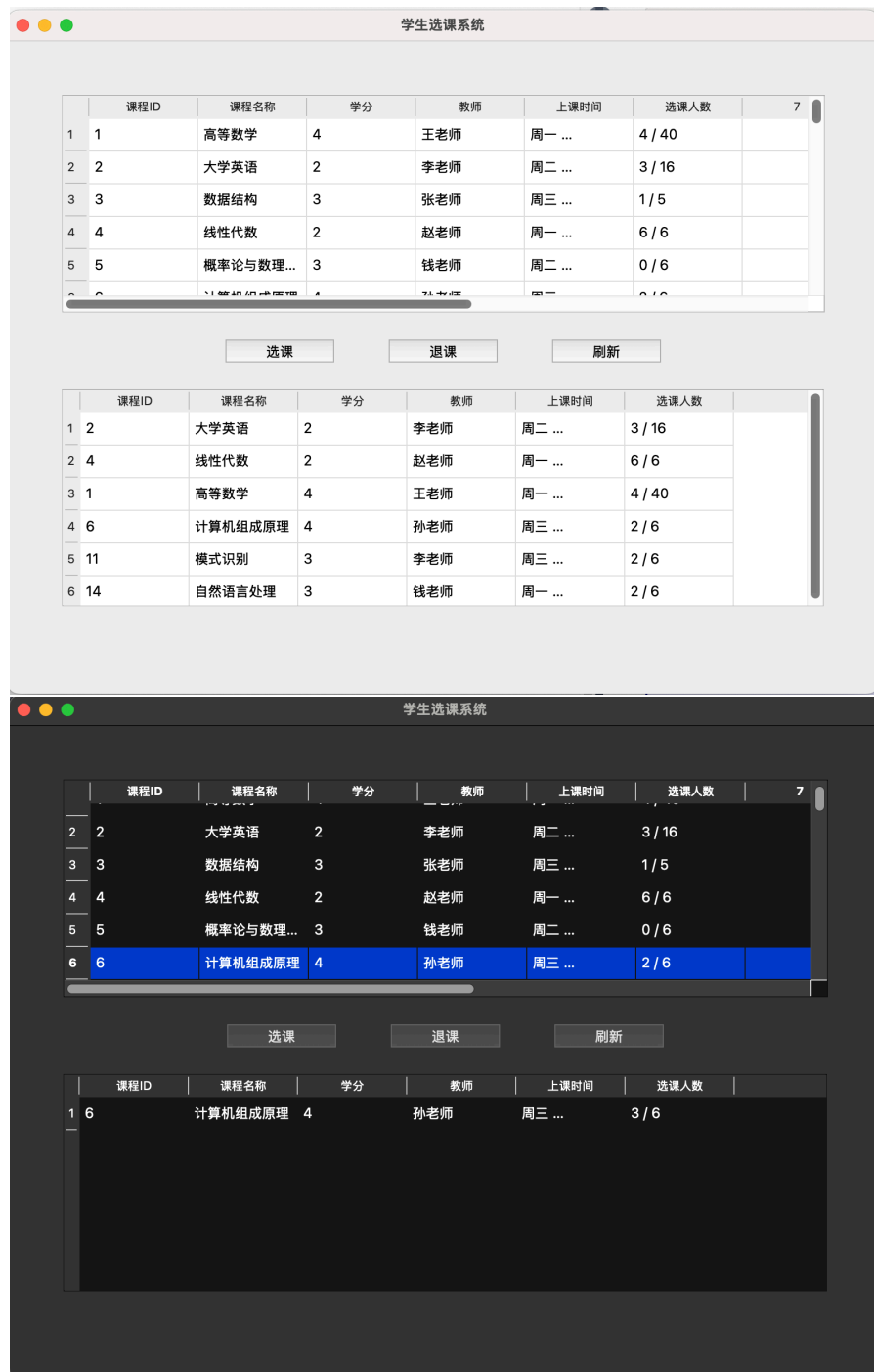
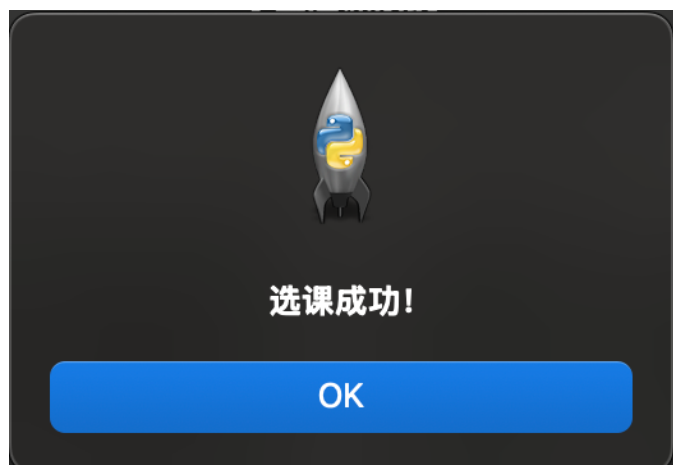
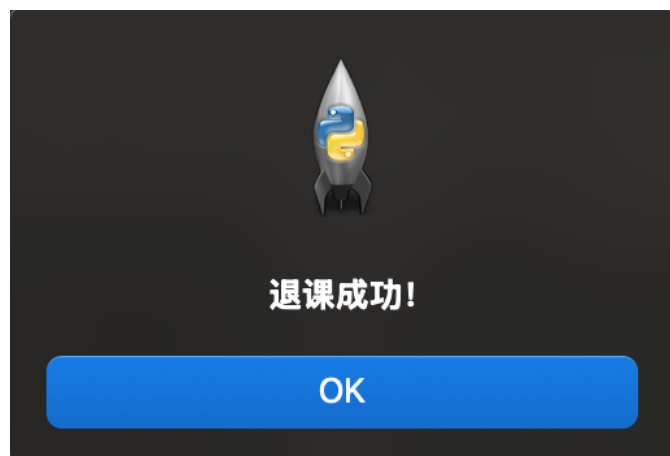


图 5: 学生 1(上), 学生 2(下) 登录后界面



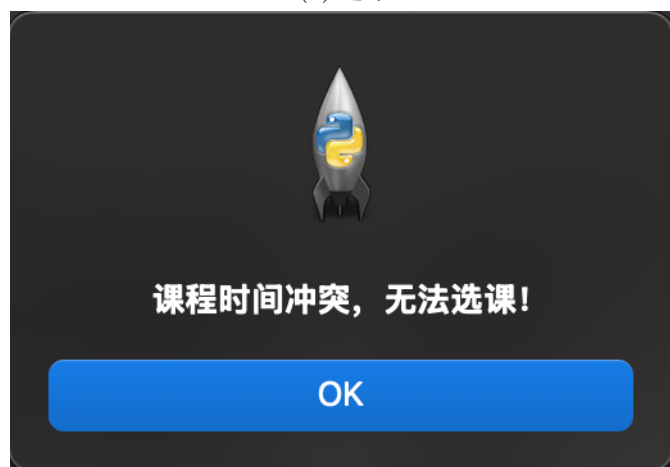
(a) 选课



(b) 退课



(c) 重复选课



(d) 课程时间冲突



(e) 满课

图 6: 操作示例

7 实验亮点

1. 学生通过账号密码登录，具有密码校验功能
2. 可以进行选课，退课操作
3. 选课检查选课时间是否冲突
4. 避免重复选课
5. 实时刷新选课人数
6. 选课人数达到上限禁止选课
7. 有 UI 界面

8 Code

8.1 StudentSelectClassSystem.sql

```
1 CREATE DATABASE IF NOT EXISTS StudentCourseSystem;
2 USE StudentCourseSystem;
3
4 -- 学生表
5 CREATE TABLE IF NOT EXISTS students (
6     studentID                INT UNIQUE PRIMARY KEY,
7     studentName              VARCHAR(50) NOT NULL,
8     studentPassword          VARCHAR(100) NOT NULL,
9     studentMajor             VARCHAR(50),
10    studentGrade              YEAR
11 );
12
13 -- 课程表
14 CREATE TABLE IF NOT EXISTS courses (
15     courseID                 INT UNIQUE PRIMARY KEY,
16     courseName               VARCHAR(100) NOT NULL,
17     courseNumber             INT NOT NULL,
18     courseCredit             INT,
19     courseTeacher            VARCHAR(50),
20     courseSchedule           VARCHAR(100)
21 );
22
23
24 -- 教师表
25 -- CREATE TABLE IF NOT EXISTS teachers (
26 --     teacherID              INT PRIMARY KEY,
27 --     teacherName            VARCHAR(50) NOT NULL,
28 --     teacherPassword        VARCHAR(100) NOT NULL,
```



```

29 --      teacherEmail          VARCHAR(100)
30 -- );
31
32 -- 选课表
33 CREATE TABLE IF NOT EXISTS enrollments (
34     ID INT PRIMARY KEY AUTO_INCREMENT,
35     studentID INT,
36     courseID INT,
37     FOREIGN KEY (studentID) REFERENCES students(studentID),
38     FOREIGN KEY (courseID) REFERENCES courses(courseID)
39 );
40
41
42 -- 插入示例课程数据
43 INSERT INTO courses (courseID, courseName, courseNumber, courseCredit, courseTeacher,
44     courseschedule) VALUES
45 (1, '高等数学', 40, 4, '王老师', '周一_8:00-10:00'),
46 (2, '大学英语', 16, 2, '李老师', '周二_10:00-12:00'),
47 (3, '数据结构', 5, 3, '张老师', '周三_14:00-16:00'),
48 (4, '线性代数', 6, 2, '赵老师', '周一_10:00-12:00'),
49 (5, '概率论与数理统计', 6, 3, '钱老师', '周二_8:00-10:00'),
50 (6, '计算机组成原理', 6, 4, '孙老师', '周三_8:00-10:00'),
51 (7, '操作系统', 6, 3, '周老师', '周四_10:00-12:00'),
52 (8, '数据库原理', 6, 2, '吴老师', '周五_14:00-16:00'),
53 (9, '人工智能导论', 6, 2, '郑老师', '周一_14:00-16:00'),
54 (10, '机器学习', 6, 4, '王老师', '周二_14:00-16:00'),
55 (11, '模式识别', 6, 3, '李老师', '周三_10:00-12:00'),
56 (12, '深度学习', 6, 2, '张老师', '周四_8:00-10:00'),
57 (13, '计算机视觉', 6, 3, '赵老师', '周五_8:00-10:00'),
58 (14, '自然语言处理', 6, 3, '钱老师', '周一_16:00-18:00'),
59 (15, '算法分析与设计', 6, 3, '孙老师', '周二_16:00-18:00'),
60 (16, '离散数学', 6, 3, '周老师', '周三_16:00-18:00'),
61 (17, '计算机网络', 6, 3, '吴老师', '周四_14:00-16:00'),
62 (18, '软件工程', 6, 2, '郑老师', '周五_10:00-12:00'),
63 (19, '编译原理', 6, 3, '王老师', '周一_10:00-12:00'),
64 (20, '计算机图形学', 6, 3, '李老师', '周二_8:00-10:00'),
65 (21, '高等数学', 6, 4, '张老师', '周三_8:00-10:00'),
66 (22, '线性代数', 6, 3, '赵老师', '周四_10:00-12:00'),
67 (23, '概率论与数理统计', 6, 3, '钱老师', '周五_14:00-16:00'),
68 (24, '计算机组成原理', 6, 2, '孙老师', '周一_14:00-16:00'),
69 (25, '操作系统', 6, 3, '周老师', '周二_14:00-16:00'),
70 (26, '数据库原理', 6, 3, '吴老师', '周三_10:00-12:00'),
71 (27, '人工智能导论', 6, 2, '郑老师', '周四_8:00-10:00'),
72 (28, '机器学习', 6, 4, '王老师', '周五_8:00-10:00'),
73 (29, '模式识别', 6, 3, '李老师', '周一_16:00-18:00'),

```

```

73 (30, '深度学习',6, 2, '张老师', '周二_16:00-18:00'),
74 (31, '计算机视觉',6, 3, '赵老师', '周三_16:00-18:00'),
75 (32, '自然语言处理',6, 3, '钱老师', '周四_14:00-16:00'),
76 (33, '算法分析与设计',6, 3, '孙老师', '周五_10:00-12:00'),
77 (34, '离散数学',6, 3, '周老师', '周一_10:00-12:00'),
78 (35, '计算机网络', 6,2, '吴老师', '周二_8:00-10:00'),
79 (36, '软件工程', 6,3, '郑老师', '周三_8:00-10:00'),
80 (37, '编译原理', 6,2, '王老师', '周四_10:00-12:00'),
81 (38, '计算机图形学', 6,3, '李老师', '周五_14:00-16:00'),
82 (39, '高等数学', 6,4, '张老师', '周一_14:00-16:00'),
83 (40, '线性代数', 6,3, '刘老师', '周五_14:00-16:00');
84
85
86 INSERT INTO students (studentID, studentName, studentPassword, studentMajor,
      studentGrade) VALUES
87 (2022001, '张三', 'passwd001', '计算机科学', 2022),
88 (2022002, '李四', 'passwd002', '电子工程', 2022),
89 (2021001, '赵瑾瑜', 'passwd001', '计算机科学', 2021),
90 (2021002, '钱浩然', 'passwd002', '电子工程', 2021),
91 (2021003, '孙思远', 'passwd003', '人工智能', 2021),
92 (2021004, '李晓蕾', 'passwd004', '软件工程', 2021),
93 (2021005, '周芷若', 'passwd005', '计算机科学', 2021),
94 (2021006, '吴文轩', 'passwd006', '电子工程', 2021),
95 (2021007, '郑雅婷', 'passwd007', '人工智能', 2021),
96 (2021008, '王梓轩', 'passwd008', '软件工程', 2021),
97 (2021009, '冯嘉熙', 'passwd009', '计算机科学', 2021),
98 (2021010, '陈思敏', 'passwd010', '电子工程', 2021),
99 (2021011, '褚子涵', 'passwd011', '人工智能', 2021),
100 (2021012, '卫子夫', 'passwd012', '软件工程', 2021),
101 (2021013, '蒋梦婕', 'passwd013', '计算机科学', 2021),
102 (2021014, '沈涛', 'passwd014', '电子工程', 2021),
103 (2021015, '韩雪', 'passwd015', '人工智能', 2021),
104 (2021016, '杨洋', 'passwd016', '软件工程', 2021),
105 (2021017, '朱一龙', 'passwd017', '计算机科学', 2021),
106 (2021018, '秦岚', 'passwd018', '电子工程', 2021),
107 (2021019, '尤靖茹', 'passwd019', '人工智能', 2021),
108 (2021020, '许凯', 'passwd020', '软件工程', 2021),
109 (2021021, '罗云熙', 'passwd021', '计算机科学', 2021),
110 (2021022, '邓紫棋', 'passwd022', '电子工程', 2021),
111 (2021023, '韩孝周', 'passwd023', '人工智能', 2021),
112 (2021024, '谢霆锋', 'passwd024', '软件工程', 2021),
113 (2021025, '冯绍峰', 'passwd025', '计算机科学', 2021),
114 (2021026, '沈腾', 'passwd026', '电子工程', 2021),
115 (2021027, '汪峰', 'passwd027', '人工智能', 2021),
116 (2021028, '程瑶瑶', 'passwd028', '软件工程', 2021),

```

```

117 (2021029, '曹曦文', 'passwd029', '计算机科学', 2021),
118 (2021030, '严宽', 'passwd030', '电子工程', 2021),
119 (2021031, '金瀚', 'passwd031', '人工智能', 2021),
120 (2021032, '魏晨', 'passwd032', '软件工程', 2021),
121 (2021033, '陶虹', 'passwd033', '计算机科学', 2021),
122 (2021034, '姜潮', 'passwd034', '电子工程', 2021),
123 (2021035, '戚薇', 'passwd035', '人工智能', 2021),
124 (2021036, '谢楠', 'passwd036', '软件工程', 2021),
125 (2021037, '邹市明', 'passwd037', '计算机科学', 2021),
126 (2021038, '喻恩泰', 'passwd038', '电子工程', 2021),
127 (2021039, '柏雪', 'passwd039', '人工智能', 2021),
128 (2021040, '水原希子', 'passwd040', '软件工程', 2021),
129 (2021041, '窦骁', 'passwd041', '计算机科学', 2021),
130 (2021042, '章子怡', 'passwd042', '电子工程', 2021),
131 (2021043, '云朵', 'passwd043', '人工智能', 2021),
132 (2021044, '苏醒', 'passwd044', '软件工程', 2021);

```

8.2 db_operations.py

```

1 import pymysql
2
3 connection = pymysql.connect(
4     host='localhost',
5     user='root',
6     # password='ADthq123',
7     password="!1004R00Tpasswd",
8     database='StudentCourseSystem',
9     charset='utf8mb4',
10    cursorclass=pymysql.cursors.DictCursor
11 )
12
13 # 用户验证
14 def verify_user(studentID, studentPassword):
15     with connection.cursor() as cursor:
16         sql = "SELECT * FROM students WHERE studentID=%s AND studentPassword=%s"
17         cursor.execute(sql, (studentID, studentPassword))
18         result = cursor.fetchone()
19         return result is not None
20
21
22 # 获取课程列表
23 def get_courses():
24     with connection.cursor() as cursor:
25         sql = "SELECT * FROM courses"
26         cursor.execute(sql)

```

```

27         return cursor.fetchall()
28
29 # 获取当前选课人数
30 def get_number(courseID):
31
32     with connection.cursor() as cursor:
33         # SQL 查询语句
34         sql = """
35         SELECT COUNT(*) AS enrollment_count
36         FROM enrollments
37         WHERE courseID = %s
38         """
39         # 执行查询
40         cursor.execute(sql, (courseID,))
41         result = cursor.fetchone()
42         # 返回结果
43         return result['enrollment_count'] if result and result['enrollment_count'] else 0
44
45
46 # 选课
47 def enroll_course(studentID, courseID):
48     with connection.cursor() as cursor:
49         # 检查是否已选该课程
50         sql = "SELECT * FROM enrollments WHERE studentID=%s AND courseID=%s"
51         cursor.execute(sql, (studentID, courseID))
52         if cursor.fetchone():
53             return "您已选过该课程！"
54
55         # 检查课程时间是否冲突
56         sql = """
57         SELECT c.courseSchedule FROM courses c
58         JOIN enrollments e ON c.courseID = e.courseID
59         WHERE e.studentID=%s
60         """
61         cursor.execute(sql, (studentID,))
62         existing_schedules = [row['courseSchedule'] for row in cursor.fetchall()]
63
64         # 获取新课程的时间
65         sql = "SELECT courseSchedule FROM courses WHERE courseID=%s"
66         cursor.execute(sql, (courseID,))
67         new_schedule = cursor.fetchone()['courseSchedule']
68
69         if new_schedule in existing_schedules:
70             return "课程时间冲突，无法选课！"

```

```

71
72 # 获取课程最大容量
73 sql = "SELECT courseNumber FROM courses WHERE courseID=%s"
74 cursor.execute(sql, (courseID,))
75 course_info = cursor.fetchone()
76 if not course_info:
77     return "课程不存在！"
78 max_capacity = course_info['courseNumber']
79
80 # 判断选课人数是否达到上限
81 current_count = get_number(courseID)
82 if current_count >= max_capacity:
83     return "选课人数已达上限，无法选课！"
84
85 # 插入新记录
86 sql = "INSERT INTO enrollments (studentID, courseID) VALUES (%s, %s)"
87 cursor.execute(sql, (studentID, courseID))
88 connection.commit()
89 return "选课成功！"
90
91 # 退课
92 def drop_course(studentID, courseID):
93     with connection.cursor() as cursor:
94         # 检查是否已选该课程
95         sql = "SELECT * FROM enrollments WHERE studentID=%s AND courseID=%s"
96         cursor.execute(sql, (studentID, courseID))
97         if not cursor.fetchone():
98             return "您未选该课程！"
99         # 删除记录
100        sql = "DELETE FROM enrollments WHERE studentID=%s AND courseID=%s"
101        cursor.execute(sql, (studentID, courseID))
102        connection.commit()
103        return "退课成功！"
104
105 # 获取已选课程
106 def get_enrolled_courses(studentID):
107     with connection.cursor() as cursor:
108         sql = """
109             SELECT c.* FROM courses c
110             JOIN enrollments e ON c.courseID = e.courseID
111             WHERE e.studentID=%s
112         """
113         cursor.execute(sql, (studentID,))
114         return cursor.fetchall()

```

8.3 main.py

```
1 import sys
2 from PySide6 import QtWidgets
3 from PySide6.QtWidgets import QMessageBox
4 from new_ui import Ui_MainWindow # 导入生成的UI类
5 import db_operations
6 import db_create
7 class MainWindow(QtWidgets.QMainWindow, Ui_MainWindow):
8     def __init__(self):
9         super(MainWindow, self).__init__()
10        self.setupUi(self)
11        self.studentID = None # 初始化学生ID
12        self.bind_events()
13        self.stackedWidget.setCurrentIndex(0) # 显示登录界面
14
15    def bind_events(self):
16        # 登录界面按钮事件
17        self.loginButton.clicked.connect(self.login)
18
19        # 选课界面按钮事件
20        self.enrollButton.clicked.connect(self.enroll_course)
21        self.dropButton.clicked.connect(self.drop_course)
22        self.refreshButton.clicked.connect(self.load_courses)
23
24    def login(self):
25        studentID = self.studentIdLineEdit.text()
26        studentPassword = self.passwordLineEdit.text()
27        if db_operations.verify_user(studentID, studentPassword):
28            QMessageBox.information(self, "成功", "登录成功")
29            self.studentID = studentID # 保存学生ID
30            self.load_courses() # 加载课程列表
31            self.load_enrolled_courses() # 加载已选课程列表
32            self.stackedWidget.setCurrentIndex(1) # 切换到主界面
33        else:
34            QMessageBox.warning(self, "错误", "学号或密码错误")
35
36    def load_courses(self):
37        courses = db_operations.get_courses()
38        self.courseTableWidget.setRowCount(len(courses))
39        for row_num, course in enumerate(courses):
40            num = db_operations.get_number(course["courseID"])
41            self.courseTableWidget.setItem(row_num, 0, QtWidgets.QTableWidgetItem(str(
42                course['courseID'])))
43            self.courseTableWidget.setItem(row_num, 1, QtWidgets.QTableWidgetItem(course
44                ['courseName']))
```

```

43         self.courseTableWidget.setItem(row_num, 2, QtWidgets.QTableWidgetItem(str(
44             course['courseCredit'])))
45         self.courseTableWidget.setItem(row_num, 3, QtWidgets.QTableWidgetItem(course
46             ['courseTeacher']))
47         self.courseTableWidget.setItem(row_num, 4, QtWidgets.QTableWidgetItem(course
48             ['courseSchedule']))
49         self.courseTableWidget.setItem(row_num, 5, QtWidgets.QTableWidgetItem(str(
50             num) + "□/□" + str(course['courseNumber'])))
51
52
53
54
55
56
57
58
59

```

```

60 def load_enrolled_courses(self):

```

```

61     courses = db_operations.get_enrolled_courses(self.studentID)
62     self.enrolledCourseTableWidget.setRowCount(len(courses))
63     for row_num, course in enumerate(courses):
64         num = db_operations.get_number(course["courseID"])
65         self.enrolledCourseTableWidget.setItem(row_num, 0, QtWidgets.
66             QTableWidgetItem(str(course['courseID'])))
67         self.enrolledCourseTableWidget.setItem(row_num, 1, QtWidgets.
68             QTableWidgetItem(course['courseName']))
69         self.enrolledCourseTableWidget.setItem(row_num, 2, QtWidgets.
70             QTableWidgetItem(str(course['courseCredit'])))
71         self.enrolledCourseTableWidget.setItem(row_num, 3, QtWidgets.
72             QTableWidgetItem(course['courseTeacher']))
73         self.enrolledCourseTableWidget.setItem(row_num, 4, QtWidgets.
74             QTableWidgetItem(course['courseSchedule']))
75         self.enrolledCourseTableWidget.setItem(row_num, 5, QtWidgets.
76             QTableWidgetItem(str(num) + "□/□" + str(course['courseNumber'])))
77
78
79
80
81
82
83
84
85

```

```

86 def enroll_course(self):

```

```

87     selected_items = self.courseTableWidget.selectedItems()
88     if not selected_items:
89         QMessageBox.warning(self, "提示", "请先选择一门课程！")
90         return
91     courseID = selected_items[0].text() # 假设课程ID在第0列
92     result = db_operations.enroll_course(self.studentID, courseID)
93     QMessageBox.information(self, "选课结果", result)
94     self.load_enrolled_courses() # 更新已选课程列表
95
96
97
98
99

```

```

100 def drop_course(self):

```

```

101     selected_items = self.enrolledCourseTableWidget.selectedItems()
102     if not selected_items:
103         QMessageBox.warning(self, "提示", "请先选择一门课程！")
104         return
105     courseID = selected_items[0].text()
106     result = db_operations.drop_course(self.studentID, courseID)
107     QMessageBox.information(self, "退课结果", result)
108
109
110
111
112

```

```
self.load_enrolled_courses() # 更新已选课程列表
```

```
if __name__ == "__main__":
    # db_create.new_create(0)
    app = QtWidgets.QApplication(sys.argv)
    window = MainWindow()
    window.show()
    sys.exit(app.exec())
```

8.4 new_ui.py

```
1 # -*- coding: utf-8 -*-
2
3 #####
4 ## Form generated from reading UI file 'new_ui.ui'
5 ##
6 ## Created by: Qt User Interface Compiler version 6.8.0
7 ##
8 ## WARNING! All changes made in this file will be lost when recompiling UI file!
9 #####
10
11 from PySide6.QtCore import (QCoreApplication, QDate, QDateTime, QLocale,
12     QMetaObject, QObject, QPoint, QRect,
13     QSize, QTime, QUrl, Qt)
14 from PySide6.QtGui import (QBrush, QColor, QConicalGradient, QCursor,
15     QFont, QFontDatabase, QGradient, QIcon,
16     QImage, QKeySequence, QLinearGradient, QPainter,
17     QPalette, QPixmap, QRadialGradient, QTransform)
18 from PySide6.QtWidgets import (QApplication, QHeaderView, QLabel, QLineEdit,
19     QMainWindow, QPushButton, QSizePolicy, QStackedWidget,
20     QTableWidget, QTableWidgetItem, QWidget)
21
22 class Ui_MainWindow(object):
23     def setupUi(self, MainWindow):
24         if not MainWindow.setObjectName():
25             MainWindow.setObjectName(u"MainWindow")
26         MainWindow.resize(800, 600)
27         self.centralwidget = QWidget(MainWindow)
28         self.centralwidget.setObjectName(u"centralwidget")
29         self.stackedWidget = QStackedWidget(self.centralwidget)
30         self.stackedWidget.setObjectName(u"stackedWidget")
31         self.stackedWidget.setGeometry(QRect(0, 0, 800, 600))
32         self.loginPage = QWidget()
33         self.loginPage.setObjectName(u"loginPage")
34         self.labelStudentId = QLabel(self.loginPage)
```



```

35 self.labelStudentId.setObjectName(u"labelStudentId")
36 self.labelStudentId.setGeometry(QRect(200, 200, 60, 30))
37 self.studentIdLineEdit = QLineEdit(self.loginPage)
38 self.studentIdLineEdit.setObjectName(u"studentIdLineEdit")
39 self.studentIdLineEdit.setGeometry(QRect(270, 200, 200, 30))
40 self.labelPassword = QLabel(self.loginPage)
41 self.labelPassword.setObjectName(u"labelPassword")
42 self.labelPassword.setGeometry(QRect(200, 250, 60, 30))
43 self.passwordLineEdit = QLineEdit(self.loginPage)
44 self.passwordLineEdit.setObjectName(u"passwordLineEdit")
45 self.passwordLineEdit.setGeometry(QRect(270, 250, 200, 30))
46 self.passwordLineEdit.setEchoMode(QLineEdit.EchoMode.Password)
47 self.loginButton = QPushButton(self.loginPage)
48 self.loginButton.setObjectName(u"loginButton")
49 self.loginButton.setGeometry(QRect(320, 300, 100, 30))
50 self.stackedWidget.addWidget(self.loginPage)
51 self.mainPage = QWidget()
52 self.mainPage.setObjectName(u"mainPage")
53 self.courseTableWidget = QTableWidgetItem(self.mainPage)
54 if (self.courseTableWidget.columnCount() < 12):
55     self.courseTableWidget.setColumnCount(12)
56 __qtablewidgetitem = QTableWidgetItem()
57 self.courseTableWidget.setHorizontalHeaderItem(0, __qtablewidgetitem)
58 __qtablewidgetitem1 = QTableWidgetItem()
59 self.courseTableWidget.setHorizontalHeaderItem(1, __qtablewidgetitem1)
60 __qtablewidgetitem2 = QTableWidgetItem()
61 self.courseTableWidget.setHorizontalHeaderItem(2, __qtablewidgetitem2)
62 __qtablewidgetitem3 = QTableWidgetItem()
63 self.courseTableWidget.setHorizontalHeaderItem(3, __qtablewidgetitem3)
64 __qtablewidgetitem4 = QTableWidgetItem()
65 self.courseTableWidget.setHorizontalHeaderItem(4, __qtablewidgetitem4)
66 __qtablewidgetitem5 = QTableWidgetItem()
67 self.courseTableWidget.setHorizontalHeaderItem(5, __qtablewidgetitem5)
68 self.courseTableWidget.setObjectName(u"courseTableWidget")
69 self.courseTableWidget.setGeometry(QRect(50, 50, 701, 200))
70 self.courseTableWidget.setColumnCount(12)
71 self.enrollButton = QPushButton(self.mainPage)
72 self.enrollButton.setObjectName(u"enrollButton")
73 self.enrollButton.setGeometry(QRect(200, 270, 100, 30))
74 self.dropButton = QPushButton(self.mainPage)
75 self.dropButton.setObjectName(u"dropButton")
76 self.dropButton.setGeometry(QRect(350, 270, 100, 30))
77 self.refreshButton = QPushButton(self.mainPage)
78 self.refreshButton.setObjectName(u"refreshButton")
79 self.refreshButton.setGeometry(QRect(500, 270, 100, 30))

```

```

80 self.enrolledCourseTableWidget = QTableWidgetItem(self.mainPage)
81 if (self.enrolledCourseTableWidget.columnCount() < 6):
82     self.enrolledCourseTableWidget.setColumnCount(6)
83 __qtablewidgetitem6 = QTableWidgetItem()
84 self.enrolledCourseTableWidget.setHorizontalHeaderItem(0, __qtablewidgetitem6)
85 __qtablewidgetitem7 = QTableWidgetItem()
86 self.enrolledCourseTableWidget.setHorizontalHeaderItem(1, __qtablewidgetitem7)
87 __qtablewidgetitem8 = QTableWidgetItem()
88 self.enrolledCourseTableWidget.setHorizontalHeaderItem(2, __qtablewidgetitem8)
89 __qtablewidgetitem9 = QTableWidgetItem()
90 self.enrolledCourseTableWidget.setHorizontalHeaderItem(3, __qtablewidgetitem9)
91 __qtablewidgetitem10 = QTableWidgetItem()
92 self.enrolledCourseTableWidget.setHorizontalHeaderItem(4, __qtablewidgetitem10)
93 __qtablewidgetitem11 = QTableWidgetItem()
94 self.enrolledCourseTableWidget.setHorizontalHeaderItem(5, __qtablewidgetitem11)
95 self.enrolledCourseTableWidget.setObjectName(u"enrolledCourseTableWidget")
96 self.enrolledCourseTableWidget.setGeometry(QRect(50, 320, 700, 200))
97 self.enrolledCourseTableWidget.setColumnCount(6)
98 self.stackedWidget.addWidget(self.mainPage)
99 MainWindow.setCentralWidget(self.centralwidget)
100
101 self.retranslateUi(MainWindow)
102
103 self.stackedWidget.setCurrentIndex(1)
104
105
106 QMetaObject.connectSlotsByName(MainWindow)
107 # setupUi
108
109 def retranslateUi(self, MainWindow):
110     MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow", u"\u5b66\u751f\u9009\u8bfe\u7cfb\u7edf", None))
111     self.labelStudentId.setText(QCoreApplication.translate("MainWindow", u"\u5b66\u53f7\u53f7", None))
112     self.labelPassword.setText(QCoreApplication.translate("MainWindow", u"\u5bc6\u7801\u53f7", None))
113     self.loginButton.setText(QCoreApplication.translate("MainWindow", u"\u767b\u5f55", None))
114     __qtablewidgetitem = self.courseTableWidget.horizontalHeaderItem(0)
115     __qtablewidgetitem.setText(QCoreApplication.translate("MainWindow", u"\u8bfe\u7a0bID", None));
116     __qtablewidgetitem1 = self.courseTableWidget.horizontalHeaderItem(1)
117     __qtablewidgetitem1.setText(QCoreApplication.translate("MainWindow", u"\u8bfe\u7a0b\u540d\u79f0", None));
118     __qtablewidgetitem2 = self.courseTableWidget.horizontalHeaderItem(2)

```

```

119 ___qtablewidgetitem2.setText(QCoreApplication.translate("MainWindow", u"\u5b66\u5206", None));
120 ___qtablewidgetitem3 = self.courseTableWidget.horizontalHeaderItem(3)
121 ___qtablewidgetitem3.setText(QCoreApplication.translate("MainWindow", u"\u6559\u5e08", None));
122 ___qtablewidgetitem4 = self.courseTableWidget.horizontalHeaderItem(4)
123 ___qtablewidgetitem4.setText(QCoreApplication.translate("MainWindow", u"\u4e0a\u8bfe\u65f6\u95f4", None));
124 ___qtablewidgetitem5 = self.courseTableWidget.horizontalHeaderItem(5)
125 ___qtablewidgetitem5.setText(QCoreApplication.translate("MainWindow", u"\u9009\u8bfe\u4eba\u6570", None));
126 self.enrollButton.setText(QCoreApplication.translate("MainWindow", u"\u9009\u8bfe", None))
127 self.dropButton.setText(QCoreApplication.translate("MainWindow", u"\u9000\u8bfe", None))
128 self.refreshButton.setText(QCoreApplication.translate("MainWindow", u"\u5237\u65b0", None))
129 ___qtablewidgetitem6 = self.enrolledCourseTableWidget.horizontalHeaderItem(0)
130 ___qtablewidgetitem6.setText(QCoreApplication.translate("MainWindow", u"\u8bfe\u7a0bID", None));
131 ___qtablewidgetitem7 = self.enrolledCourseTableWidget.horizontalHeaderItem(1)
132 ___qtablewidgetitem7.setText(QCoreApplication.translate("MainWindow", u"\u8bfe\u7a0b\u540d\u79f0", None));
133 ___qtablewidgetitem8 = self.enrolledCourseTableWidget.horizontalHeaderItem(2)
134 ___qtablewidgetitem8.setText(QCoreApplication.translate("MainWindow", u"\u5b66\u5206", None));
135 ___qtablewidgetitem9 = self.enrolledCourseTableWidget.horizontalHeaderItem(3)
136 ___qtablewidgetitem9.setText(QCoreApplication.translate("MainWindow", u"\u6559\u5e08", None));
137 ___qtablewidgetitem10 = self.enrolledCourseTableWidget.horizontalHeaderItem(4)
138 ___qtablewidgetitem10.setText(QCoreApplication.translate("MainWindow", u"\u4e0a\u8bfe\u65f6\u95f4", None));
139 ___qtablewidgetitem11 = self.enrolledCourseTableWidget.horizontalHeaderItem(5)
140 ___qtablewidgetitem11.setText(QCoreApplication.translate("MainWindow", u"\u9009\u8bfe\u4eba\u6570", None));
141 # retranslateUi

```