**实 验 报 告**

**课程名称：**  **模型驱动的软件开发技术**

**学 院： 计算机科学与工程学院**

**专 业：** **软件工程**  **班 级： 软件18-1班**

**姓 名：KAFLE SAMRAT学 号： 201801060933**

**2020年 11月 23日**

**山 东 科 技 大 学**

**实 验 报 告**

**页**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组 别** |  | **姓 名** | **KAFLE SAMRAT** | **同组实验者** |  |
| **实验项目**  **名称** | [综合实践报告电子版](javascript:void(0);) | | **实验日期** | **2020.11.23** | |
| **教师评语** |  | | | | |
| **实验成绩：** | | | **指导教师（签名）：**  **年 月 日** | | |
| 1. **实验过程**   在这个实验中，我们将学习UML图。UML图是基于UML(统一建模语言)的图，其目的是可视化地表示一个系统及其主要参与者、角色、操作、工件或类，以便更好地理解、更改、维护或记录有关系统的信息。并利用类图、活动图、顺序图等不同类型的UML图开发在线选课系统。  **Online course selection system**  The online course selection system of a school mainly includes the following functions:  The administrator enters through the system management interface, establishes various courses to be held this semester, saves the course information in the database, and can modify and delete the courses. Through the client browser, students can enter the course selection interface according to their student id and password. Here, students can conduct two operations: inquiring the selected courses and selecting courses. Again, through the business layer, the results of these operations are stored in the database.  **Modeling**  Actor: Registar, Student, People(Abstract), Database  Use cases: Select Course, Query Course, Add Course, Modify course, Delete Course    **Add the course event stream**  (1) The administrator selects to enter the management interface and the use case starts  (2) The system prompts for the administrator password  (3) The administrator enters the password  (4) System verification password  A1:  Wrong password  (5) Enter the management interface and the system displays all the course information currently established  (6) The administrator chooses to add courses  (7) The system prompts for new course information  (8) The administrator enters the information  9) The system verifies whether it is in conflict with existing courses  A2:  there are conflicts  (10) The system adds new courses to prompt the success of the courses  (11) The system reenters the main management interface to display all courses  (12) ends with a column  **Other event flow**  A1: Wrong password  (1) The system prompts you to input again  (2) User confirmation  (3) Three times error, refused to visit again  (4) otherwise  Add the course event stream  (1) The administrator selects to enter the management interface and the use case starts  (2) The system prompts for the administrator password  (3) The administrator enters the password  (4) System verification password  A1: Wrong password  (5) Enter the management interface and the system displays all the course information currently established  (6) The administrator chooses to add courses  (7) The system prompts for new course information  (8) The administrator enters the information  9) The system verifies whether it is in conflict with existing courses  A2: there are conflicts  (10) The system adds new courses to prompt the success of the courses  (11) The system reenters the main management interface to display all courses  (12) ends with a column  **Error stream**  A1: Verification failed  (1) The system prompts verification failure, prompting re-input (2) Three times of failure, access is denied  (3) Successful transfer of course events (5)  A2: Courses are not optional  (1) The system prompt course is not available and the reasons  (2) Students reschedule their courses  (3) Course transfer event flow (10)  **Activity Diagram**  Take the Add Course use case as an example.  (1) The administrator enters information  (2) Whether the system is in conflict with existing courses  A2:  there are conflicts  (3) New courses are added to the system to prompt for successful addition of courses. (4) The system re-enters the main management interface to display all courses  (5) End with a column  。  **object**  (2) Look for roles  Objects: Interface, Course , business layer control object role: student, database  **Select Course use case interaction sequence**  (1) Students send course selection commands through the interface  2) Bounded control-oriented object requests course information  3) The control object sends query data information to the database  4) Control the query results of the temporary database  (5) The interface object obtains all course information from the control object  6) Display all course information on the interface  (7) The interface object sends commands that require the control object to delete the course information  (8) Students choose courses  9) The interface object requires students to input their student number  (10). The interface sends information to the control object to inquire whether the student can choose a specific course  (11) The control object queries association information from the database  (12) The control object judges whether the course can be selected  (13) If courses can be selected, the associated information will be added to the database.  (14) Return information to the interface object.  .  .  .    **The class diagram**  Entity Class: Course  Boundary class: Interface class: FormView  Control class: ControlObject class ControlObject  Roles: Student, Registar, People (parent) Database  Relationship: When students and administrators interact with the system, there is an interface corresponding to it, which may be related to multiple courses. The control object is responsible for course processing, and the processing results are displayed on the street. The control object completes the operation of the database, and the interface requests the service of the control object  **Class- People**  Name String  Age int  GetName()  SetName()  Getage()  SetAge()  **Class- Student**  StudentId int  Student()  GetStudentID()  SetStudentID()     1. 实验总结   完成本课程设计主要依靠在软件架构课程中所学的相关知识，加上适当阅读相关参考文献，才能很好地完成整个软件架构课程设计。在整个课程设计过程中，也测试了整个软件从提出需求到实现整个架构建模的流程思维和处理逻辑。在设计的过程中，我加深了对课堂所学知识的理解。通过实践实习，我熟练掌握了软件架构设计的整体思路、流程和方法，并获得了丰富的收获。 | | | | | |