# 功能

### **第三阶段需求（选出至少5个用例实现）**

完成课程学习平台的课程测验、课程讨论区等附加功能。

#### **20. 租用课程**

学生用户可以选择按照时间租用课程而不是一次性购买课程。租用课程同样需要生成订单并进行支付。租用期间学生可以查看课程内容，到期后课程内容回复到锁定状态。

#### **21. 开通会员**

学生用户可以通过开通平台会员的方式，在会员有效期内免费浏览平台所有课程。需要生成会员相关的订单。会员到期后课程内容回复到锁定状态。

#### **22. 购买学校课程包**

学生用户可以通过购买某一学校的课程包的方式，解锁对应学校的全部课程。需要生成课程包相关的订单。

#### **23. 创建题目**

教师用户可以创建题目并添加到题库。题目的内容包括题干、答案、解析、对应课程id等信息。

#### **24. 创建测试**

教师用户可以创建课程测试，从课程题库中挑选题目加入测试并设置开始时间、结束时间等内容。

#### **25. 获取测试列表**

学生用户可以根据课程查看已发布的测试列表

#### **26. 获取测试信息**

状态为进行中的测试，学生用户可以获取题目信息并进行作答。

状态为已结束的测试，学生用户可以获取得分情况，查看每一题的正确答案和解析。

#### **27. 提交测试**

学生用户可以提交测试的作答，平台会对作答进行正误判断，记录作答信息和得分情况。

#### **28. 发布帖子**

购买了课程的学生用户/创建课程的教师用户可以在课程讨论区发表帖子，帖子的内容包括主题、内容、发表时间、发表用户等信息。

#### **29. 评论帖子**

购买了课程的用户/创建课程的教师用户可以回复帖子的内容发表评论。被回复者会收到帖子/评论被回复的通知。

#### **30. 查看讨论区**

购买了课程的用户/创建课程的教师用户可以在进入讨论区后，获取讨论区的帖子列表，支持按照发表时间和最新回复时间倒序分页查看。

#### **31. 查看帖子**

购买了课程的用户/创建课程的教师用户可以查看课程讨论区内某一个帖子的具体内容，包括帖子标题、内容、发表时间、发表用户以及一个按照时间顺序排序的回复列表。每条回复包括内容、发表时间、发表用户、针对的评论内容（可选）等信息。

#### **32. 查看回帖通知**

用户发表的帖子或评论得到回复之后，需要能够及时收到通知。

# 2. 代码

|  |  |  |
| --- | --- | --- |
| **编号** | **检查点** | **检查点描述** |
| **Code01** | 代码结构 | 代码的布局是否能够清晰地体现程序的逻辑结构 |
| **Code02** | 代码命名 | 代码中包、接口、类、方法、变量等的命名是否合理 |
| **Code03** | 代码注释 | 文档注释和内部注释，复杂类和方法一定要有注释 |
| **Code04** | 代码可靠性 | 是否利用异常与断言进行防御式编程或契约式设计（可选） |
| **Code05** | 单元测试代码 | 检查是否有单元测试代码 |
| **Code06** | 集成测试代码 | 是否有桩和驱动代码 |
| **Code07** | 系统架构 | 体系结构设计是否符合某种体系结构风格 |
| **Code08** | 接口设计 | 接口设计是否合理，是否通过接口进行调用 |
| **Code09** | 控制风格 | 是否使用合适的控制器风格 |
| **Code10** | 设计模式 | 是否合理使用设计模式 |
| **Code11** | 其他 | 架构设计部分是否有其他特点（可选） |

**3. 过程**

|  |  |  |
| --- | --- | --- |
| 编号 | 检查点 | 检查点描述 |
| Proc01 | 版本控制 | 是否使用版本控制工具，使用工具证明（更新记录截图等） |
| Proc02 | 持续集成 | 是否使用持续集成工具，与git，maven等工具整合（自动构建配置，持续集成记录截图等） |
| Proc03 | 团队合作 | 检查团队合作的方式（SVN,git等） |
| Proc04 | 团队交流 | 检查团队交流方式（会议记录，聊天截图等） |
| Proc05 | 开发模式 | 检查团队开发模式（主程序员，协同开发等）该项与过程得分无关。  注：前来检查过程部分的同学需要说明队员分值（整个大作业，不仅仅是过程）分配，一般来讲为均分，如全部80分，若提出有人工作量多有人工作量少，则按割补法，从工作量少的人扣除分数加到工作量多的人身上，最多10分，如80均分改成70,90,80,80.  如老师发现有队员在团队中贡献过低，将由由老师直接打分。 |

提供相应证据文件。

1. **文档**

|  |  |  |
| --- | --- | --- |
| **编号** | **检查点** | **检查点描述** |
| Doc01 | 文档完整性 | 包括用例文档、软件需求规格说明文档、体系结构文档、详细设计文档。 |
| Doc02 | 文档前后一致性 | 随意抽一个用例，然后要求在用例文档、需求规格说明文档、体系设计文档、详细设计文档中都有对应的匹配，并且各部分要一致（用例在登陆注册、预定酒店、维护酒店信息、制定酒店促销策略、更新入住信息、制定网站促销策略、浏览异常订单、信用充值用例中进行选择） |
| Doc03 | 文档是否纳入配置管理 | 各个文档是否有版本号，若有，是否有修订记录 |
| Doc04 | 是否符合技术文档的规范 | 是否有清晰的目录结构；是否包含作者、引用文献及引用文档 |
| Doc05 | 详细设计文档是否和代码匹配 | 抽一个功能点，检查详细设计中的接口（从界面到逻辑到数据访问）和实际代码中的接口是否匹配 |
| Doc06 | 需求文档中假设与依赖书写是否正确 | 查看文档中的依赖和假设书写是否正确 |
| Doc07 | 需求文档中对于功能性需求是否得当 | 抽取一个需求场景，查看需求文档的相应功能需求描述是否详尽 |
| Doc08 | 需求文档中对于非功能需求是否得当 | 抽取一个需求场景，查看需求文档的相应非功能性需求描述是否详尽 |
| Doc09 | 需求文档中用例描述是否等当 | 抽取若干用例，查看用例类图、用例描述等是否符合要求 |
| Doc10 | [需求文档](http://seec.yangshan.tech/course-teacher/2/document/666)中非功能需求的约束是否正确 | 查看文档中非功能需求的定义部分，查看约束是否理解为正确约束 |
| Doc11 | [需求文档](http://seec.yangshan.tech/course-teacher/2/document/666)中对于数据需求的定义是否明确 | 查看文档中关于数据的定义部分，查看是否缺少相应数据库的设定 |
| Doc12 | 需求文档中假设与依赖书写是否正确 | 查看文档中的依赖和假设书写是否正确 |
| Doc13 | 需求文档中对于功能性需求是否得当 | 抽取一个需求场景，查看需求文档的相应功能需求描述是否详尽 |
| Doc14 | 架构设计文档中逻辑视图是否规范 | 抽取其中一个逻辑视图，查看是否满足标准的逻辑过程 |
| Doc15 | 架构设计文档中包图是否规范 | 抽取其中一部分包图，查看类之间的继承关系、组合关系等是否满足相应的设计模式 |
| Doc16 | 架构设计文档中对应的模块功能是否正确 | 抽取其中若干模块，查看是否满足相应的模块功能 |
| Doc17 | 架构设计文档中使用的设计模式 | 查看文档中使用的架构是否搭配相应的设计模式完成相应的任务 |
| Doc18 | 架构设计文档中逻辑视图是否规范 | 抽取其中一个逻辑视图，查看是否满足标准的逻辑过程 |
| Doc19 | 架构设计文档中逻辑视图是否规范 | 抽取其中一个逻辑视图，查看是否满足标准的逻辑过程 |
| Doc20 | 架构设计文档中包图是否规范 | 抽取其中一部分包图，查看类之间的继承关系、组合关系等是否满足相应的设计模式 |
| Doc21 | 详细设计文档等需要指出具体设计细节处是否解释得当 | 抽取一个需求场景，对具体问题进行提问，查看解决方法、异常处理等是否得当 |
| Doc22 | 详细设计文档接口描述是否规范 | 抽取设计文档中的若干个接口，查看接口描述和接口命名是否规范 |
| Doc23 | 详细设计文档中顺序图书写是否规范 | 抽取一个业务场景，查看顺序图的格式、标号等书写是否满足要求 |
| Doc24 | 测试用例文档是否完成 | 是否完成测试用例文档 |
| Doc25 | 测试用例覆盖种类是否齐全 | 单元测试，集成测试，系统测试 |
| Doc26 | 测试用例是否描述测试结果 | 是否有输入输出，重点关注测试结果的记录 |
| Doc27 | 测试界面时是否使用脚本 | 是否使用e2e自动化测试 |

测试文档要求为加分项。

文档只涉及第三阶段需求。