1. Background

随着当今社会经济发展，各种教育机构越来越多，人们对各类学习软件的需求也越来越明显，本产品是一个学习类的app，主要针对学生学习，让选择参加课程，观看视频，相互交流进行学习。

角色与定位：

|  |  |
| --- | --- |
| **用户角色** | **目标人群定位** |
| **学生** | 选择课程、评论课程、浏览课程 |
| **教师** | 发布课程视频、回答同学的评论或提问 |
| **系统管理员** | 管理老师学生账户、创建老师账户 |

技术路线：

|  |  |  |
| --- | --- | --- |
| 技术路线 | 客户端 | 客户端主要是针对安卓进行开发，包括用户界面的设计实现，以及界面间的联系、跳转。 |
| 前后端交互 | 客户端通过 HTTP 请求服务器接口，服务器接口根据对应参数，返回 JSON 格式数据。客户端端解析 JSON 数据，进行数据显示。 |
| 后台服务器 | 使用 Spring MVC 框架，严格遵守接口分离原则。Dao 层向Services层提供数据库操作相关接口，由Controller  层调用 Services 层接口完成业务逻辑操作，并将数据  以 Json 格式返回给客户端。 |

1. Main Functions

Functional demands:

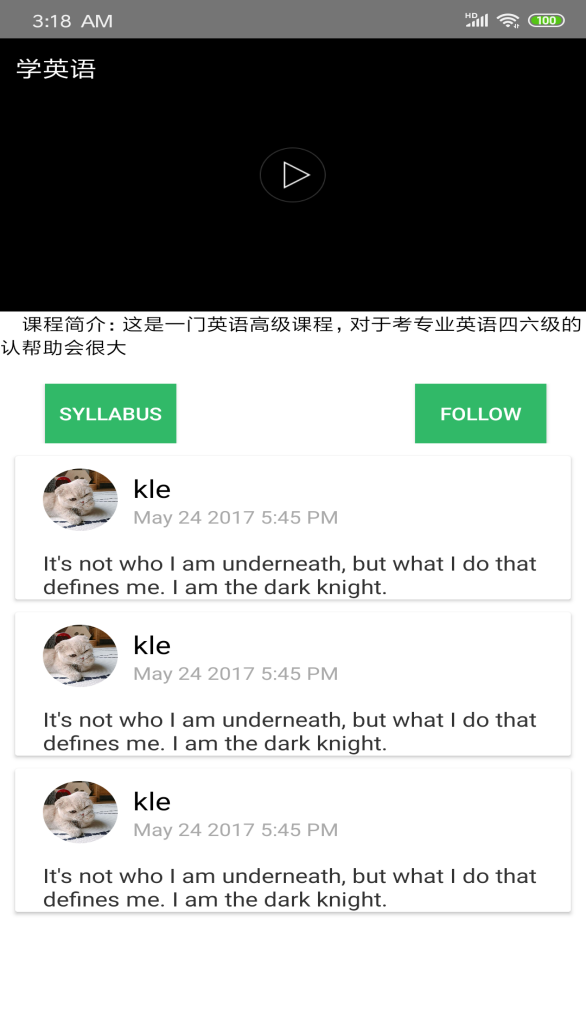
|  |  |  |
| --- | --- | --- |
| 功能需求 | 学生 | 登录  注册  找回密码  课程列表查看  课程详细信息查看  课程视频观看  课程评论  接收重要事项通知 |
| 老师 | 发布课程、读取消息、更改密码、登录、回答问题、修改信息 |
| 管理员 | 注册教师用户  管理学生、教师  管理用户密码 |

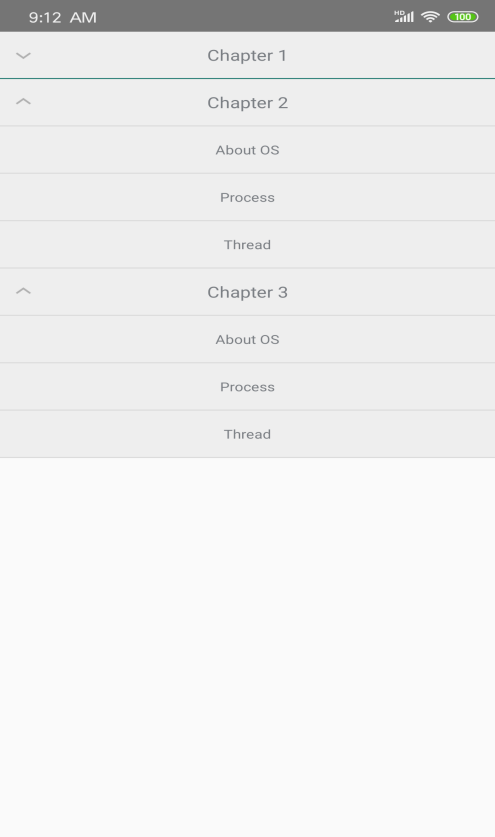
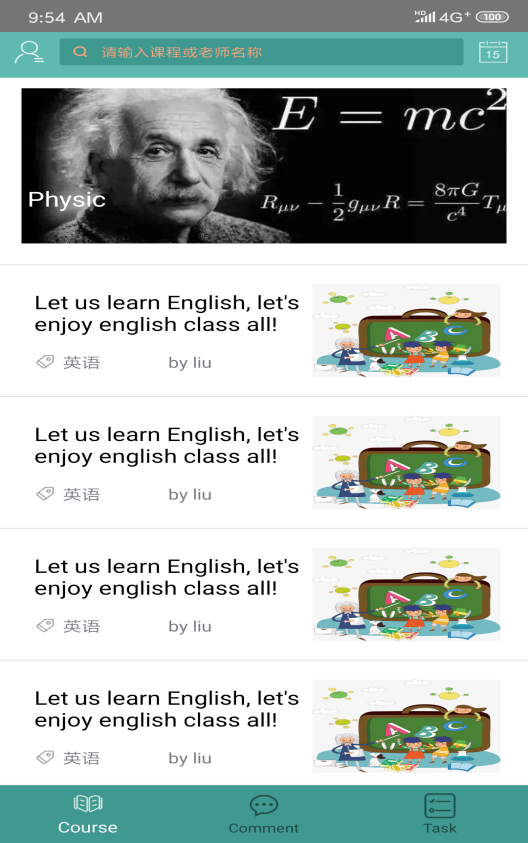
## Non-functional demands

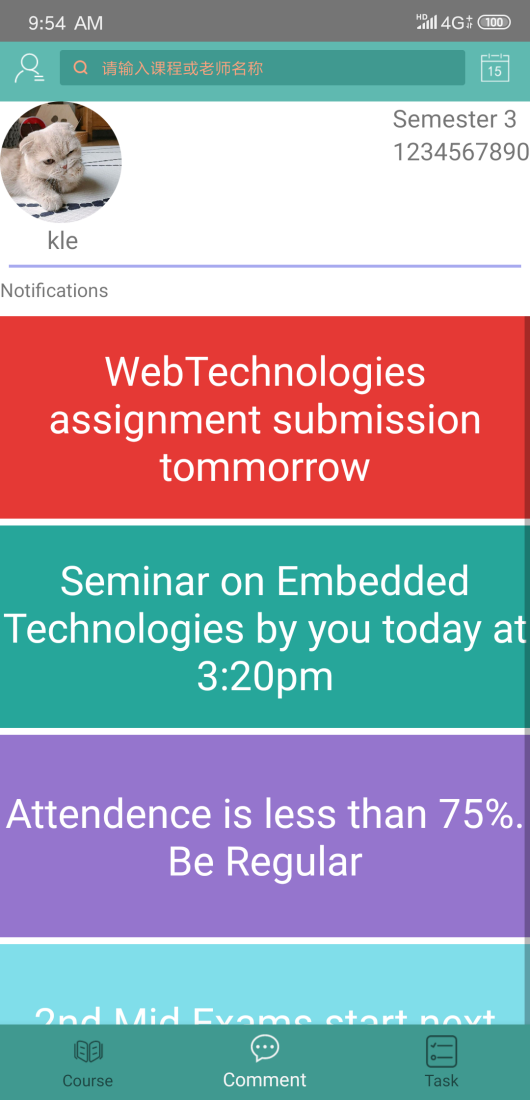
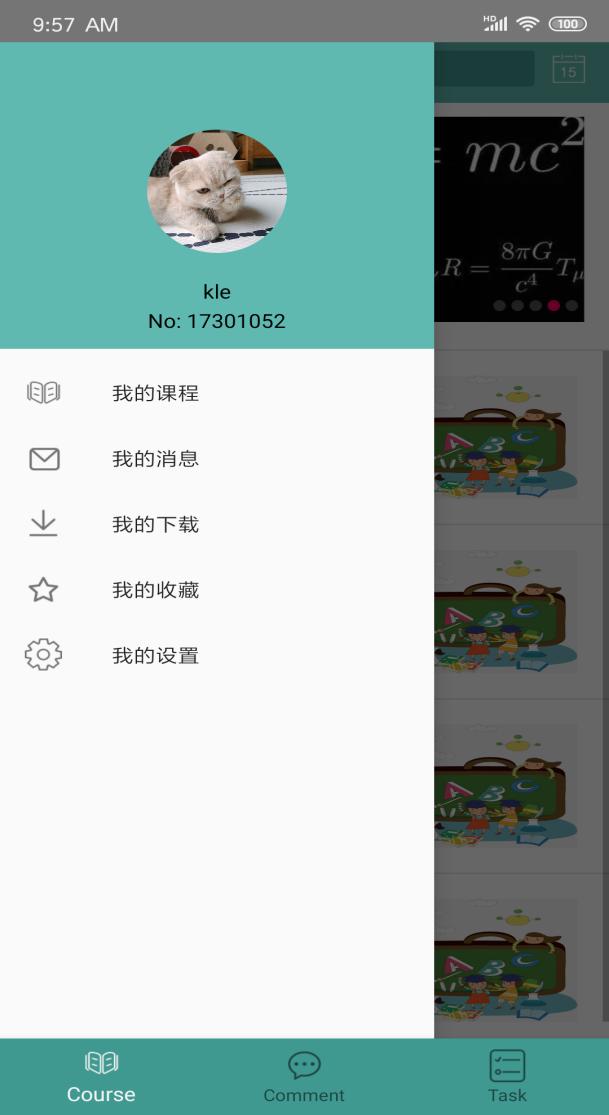
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 类别 | 需求侧重 | 具体场景描述及量化指标 | | | |
|
|
| 非功能需求的量化指标 | Usability | 用户体验（UE） | 感官冲击 | | 用户界面风格的统一（包括颜色、外观等） | |
| 页面布局与美工设计赏心悦目 | |
| 使用性 | | 操作方便，规范简洁，无业务操作冗余 | |
| 卓然有效的提醒说明（如Dialog及规范显示） | |
| Reliability | 包括故障周期、可恢复性、可预测性、有效性准确性 | 针对系统核心功能模块服务中断时间不超过3s | | | |
| 网络中断后，用户的操作可以被记忆和恢复。 | | | |
| 在给定时间内，服务中断平均次数不超过2.5次/s | | | |
| 缺陷故障率不得超过10% | | | |
|  | 对于登录及查询时非法输入（如非法输入或超越字符数量限制等操作）进行有效判断，有效率100% | | |
| Performance | 包括响应时间、吞吐量、资源利用率等体现系统容量、工作能力及效率 | 后台处理 | 对于计算机下棋思考时间响应时间不超过30s | |
| 系统支持并发用户数不低于2 | |
| Security | 系统安全（与可靠性有一定交叉） | 应用安全 | 确保应用不会被非法篡改 | |
| 没有授权的外部系统试图访问本系统的资源文件和数据（代码、图片、配置文件等），系统能识别此外部系统没有被授权，并阻止它的访问。该外部系统的操作被拒绝。 | |
| 信息安全 | 有外部未知的病毒恶意攻击系统（插入代码、逆向分析、劫持密码等），使系统瘫痪或用户的信息泄露，系统能预防和抵抗该 80%病毒，限制该病毒的操作，并在之后能在 5 个小时内正确和及时的恢复用户的数据 | |
| 该系统保证用户的密码等重要信息不被泄露 | |
| Maintainability | 针对系统维护方面：安装、数据备份、系统健康情况监控及报警（异常）、时间灵活度 | 系统主要功能模块数据每日自动执行备份 | | |
| 棋谱记录日志 | | |
| Scalability | 可扩展性审查系统架构，扩展方案及已有技术确认，针对功能的可扩展。 | 未来用户增长的可扩展性，在下一阶段在原有基础上预计提高15% | | |
| 未来系统功能的可变性提高20% | | |
| Interoperability | 系统内部交互的统一性，系统与外部接口交互的简易性。 | 系统以JSON格式数据进行交互，操作简单，通用性高，便于与其他服务于系统进行数据交互。 | | |
|  | Expandablilty | 根据实际的要求,系统可被方便地载减和灵活的扩展,使系统能适应变化和新情况。 | 系统架构设计时，采用多种设计模式，严格遵循了OCP原则，整体结构高内聚、低耦合，可扩展性高。采用前后端分离模式，系统耦合性低，便于日后再其他平台上进行开发。 | | |

1. UI design

1. Package and class design

下表是本系统的的各个包的设计说明：

|  |  |
| --- | --- |
| **包名** | **设计说明** |
| **com.star.e\_learning.adapter** | **该包主要存放控件的适配器。** |
| **com.star.e\_learning.activity** | **主要存放所有具体活动** |
| **com.star.e\_learning.fragment** | **该包中主要存放了所有的碎片** |
| **com.star.e\_learning.bean** | **存放了所有的实例对象** |
| **com.star.e\_learning.view** | **自身实现的简单的view控件** |

***adapter相关类设计***

|  |  |
| --- | --- |
| **Adapter相关类** | |
| **类名** | **设计说明** |
| **CourseItemAdapter** | **匹配课程信息的适配器** |
| **CommentAdapter** | **匹配评论信息的适配器** |
| **SyllabusAdapter** | **匹配课程大纲的适配器** |

***activity相关类设计***

|  |  |
| --- | --- |
| **相关类** | |
| **类名** | **设计说明** |
| **LoginActivity** | **登录** |
| **RegisterActivity** | **注册** |
| **ForgetPwdActivity** | **忘记密码** |
| **SyllabusActivity** | **课程大纲** |
| **CourseContentActivity** | **课程详情** |
| **HomeACtivity** | **主界面** |
| **CalendarActivity** | **日历** |
| **SplashActivity** | **欢迎** |

***fragment相关类设计***

|  |  |
| --- | --- |
| **com.zeus.app.entity相关类** | |
| **类名** | **设计说明** |
| **HomeFragment** | **主界面** |
| **NotificationFragment** | **通知碎片** |
| **TaskFragment** | **任务碎片** |

***bean相关类设计***

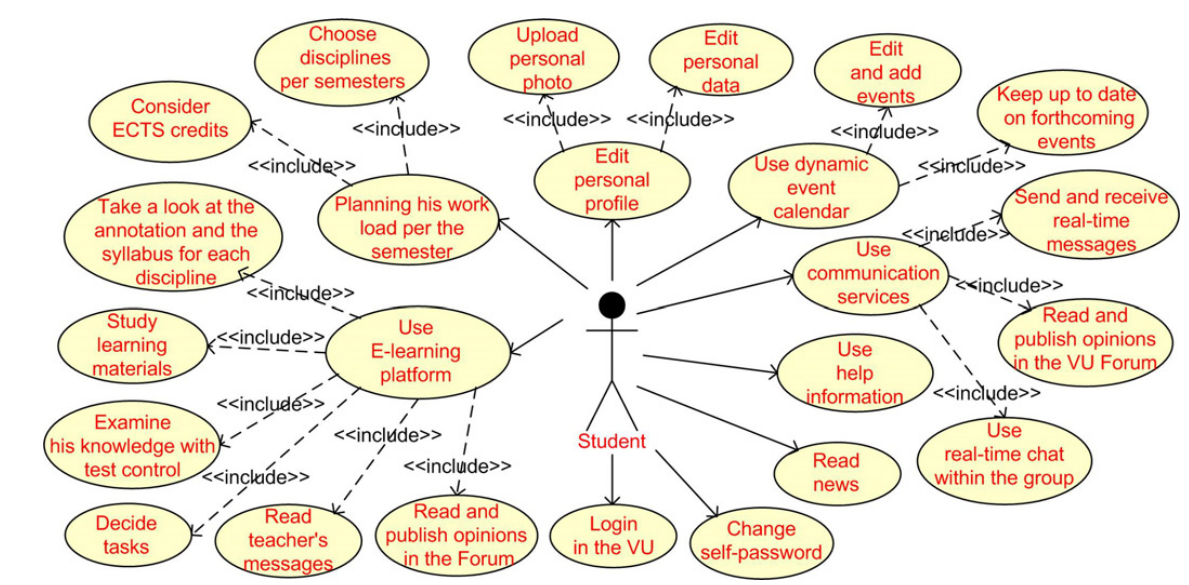
|  |  |
| --- | --- |
| **com.zeus.app.entity相关类** | |
| **类名** | **设计说明** |
| **Course** | **课程实体** |

***view相关类设计***

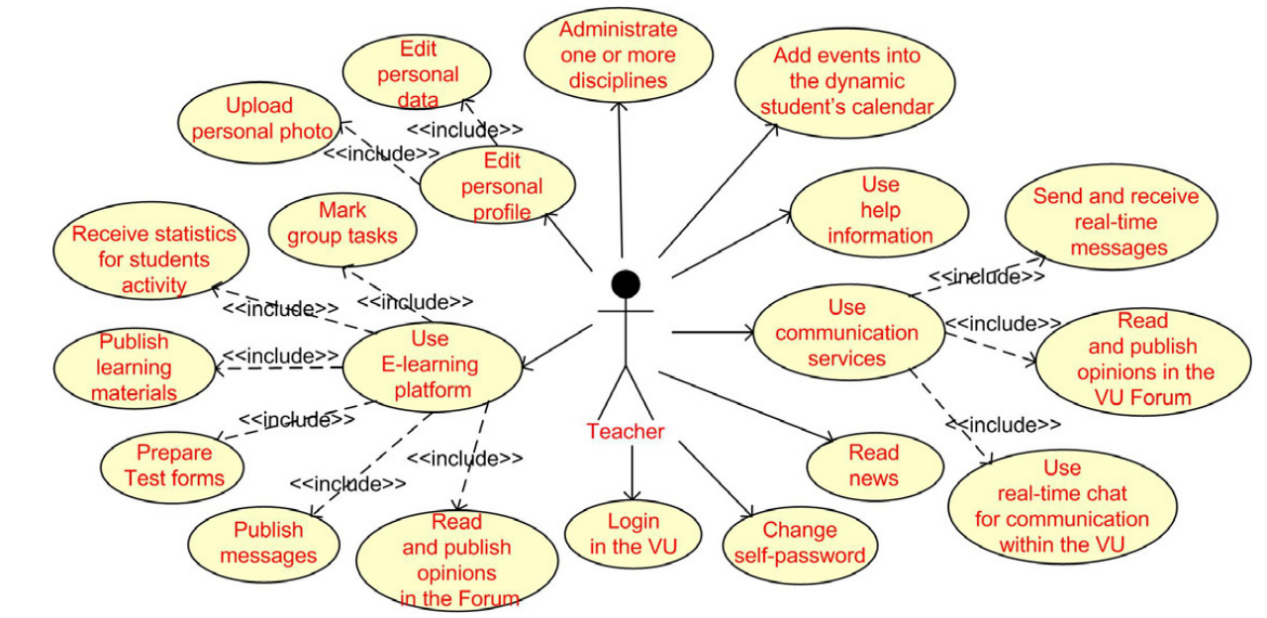
|  |  |
| --- | --- |
| **com.zeus.app.entity相关类** | |
| **类名** | **设计说明** |
| **Kanner** | **滑动项目View** |

1. Use-case

学生：



老师：



管理员：

