START! 软件架构设计

朗伟临、田亚博、万新熠、喻智勇、张世康

1. 引言

1.1. 编写目的

本文档的编写目的是对 Study Together And Review Together! 在线学习与复习单词系统（以下简称 START!）软件的系统结构进行描述和定义。本文档在软件需求规约、软件系统分析的基础上，利用各种模型与视图，详细展示系统的结构，为后续的软件实现工作奠定基础。本文档定义系统架构的设计目标，描述系统的结构、子系统定义、软硬件部署、数据管理、软件控制、边界条件等内容，帮助开发团队明确系统的架构和设计。

1.2. 适用范围

本文档适用于的软件：START!。
与该软件相关的特性、子系统、模型等均符合本文档中的内容。

1.3. 定义

本文档中涉及的术语定义在项目词汇表（START! 词汇表.pdf）中给出。

1.4. 参考资料

《面向对象软件工程：使用 UML、模式与 Java》（第 3 版），清华大学出版社，2011。
《面向对象软件工程实践指南》，上海交通大学出版社，2016。

1.5. 概述

本文档包括引言、目前系统架构、系统架构设计目标、建议的软件系统架构四部分。目前系统部分对当前主流用户背单词的方式进行分析并指出其不足。系统架构设计目标部分结合软件需求，列举出系统设计的目标。建议的软件系统架构给出系统的架构和子系统的分解，并以文字表述和模型图相结合的方式展示系统的对象设计、软硬件部署、数据管理、软件控制、边界条件等设计。本文件的各部分内容联系紧密，互为补充和对照，共同呈现本系统的软件架构。

2. 目前软件系统体系架构

目前存在的系统主要可以分为两类：传统的纸质单词书学习、主流的手机背单词 App 学习。传统的纸质单词书学习指用户购买纸质单词书，自行制定学习计划独立学习新单词、或独立复习已经学习过的单词；手机背单词 App 学习指用户在 App 中选定单词列表，按照 App 给定的计划独立学习新单词、或独立复习已经学习过的单词。新的系统是网页端互动式单词学习 App，使用便捷且社交性更强。

3. 软件系统架构设计目标

系统架构的设计目标如下：

（1）高可用性：本软件作为一个互动式背单词平台，如果在学习或复习单词时发生系统不可用的情形，可能导致用户学习进度丢失，严重影响用户学习体验，因此系统需要保证较高的可用性。

（2）高性能：本软件作为一个互动式背单词平台，需要在用户学习或复习单词时实时响应用户发送来的信息，并将必要的信息返回给用户，在此过程中产生较大的流量，因此要求有较高的性能。

（3）可扩展性：本系统在初期规模较小时，不需要较高规格的硬件，但应当考虑到在用户数量增加后对系统的扩展。

4. 建议的软件系统架构

4.1. 概述

本系统的架构及采用架构模式：

（1）客户 / 服务器模式（client/server）

本系统分为前端客户端和后端服务器。前端为用户打开的网页界面，后台是实现系统功能的服务器。选择客户 / 服务器模式的原因是这种模式比较成熟，且客户、服务器分离符合软件开发的环境，并且也方便数据的集中管理。

（2）三层体系结构

本系统用三个层次来组织子系统：第一层为用户界面层， 提供与用户交互的接口，即网页的 UI。第二层为应用逻辑层。这一层主要负责控制并实现系统功能，包含 UserManagement、StudyManagement、ReviewManagement 三个子系统， 分别负责用户信息、学习功能和复习功能的控制。第三层为存储层。主要负责系统数据的存储、 检索和查询。

三层体系结构比较成熟，将接口和应用逻辑分开，耦合度低，灵活性和扩展性强，方便开发过程的分工。因此本软件采用这个结构。

（3）模型视图控制器模式（ model,view and controller,MVC）MVC 模式适用于交互系统。在上述的三层体系结构中已经将系统分为视图（第一层)、控制器（第二层）和模型（第三层）三部分。

本系统包含以下子系统：

a. UI：负责用户界面部分。

b. UserManagement：负责用户信息的管理，控制登录、注册、个人资料修改等功能。

c. StudyManagement：负责创建学习房间，给出学习单词、选择前一个单词、后一个单词等功能。

d. ReviewManagement：负责创建复习房间，给出复习题、选择答案、给出正确答案、显示他人正确率等功能。

e. CommonService：负责客户机与服务器通信以及数据存取功能。

4.2. 用例视图

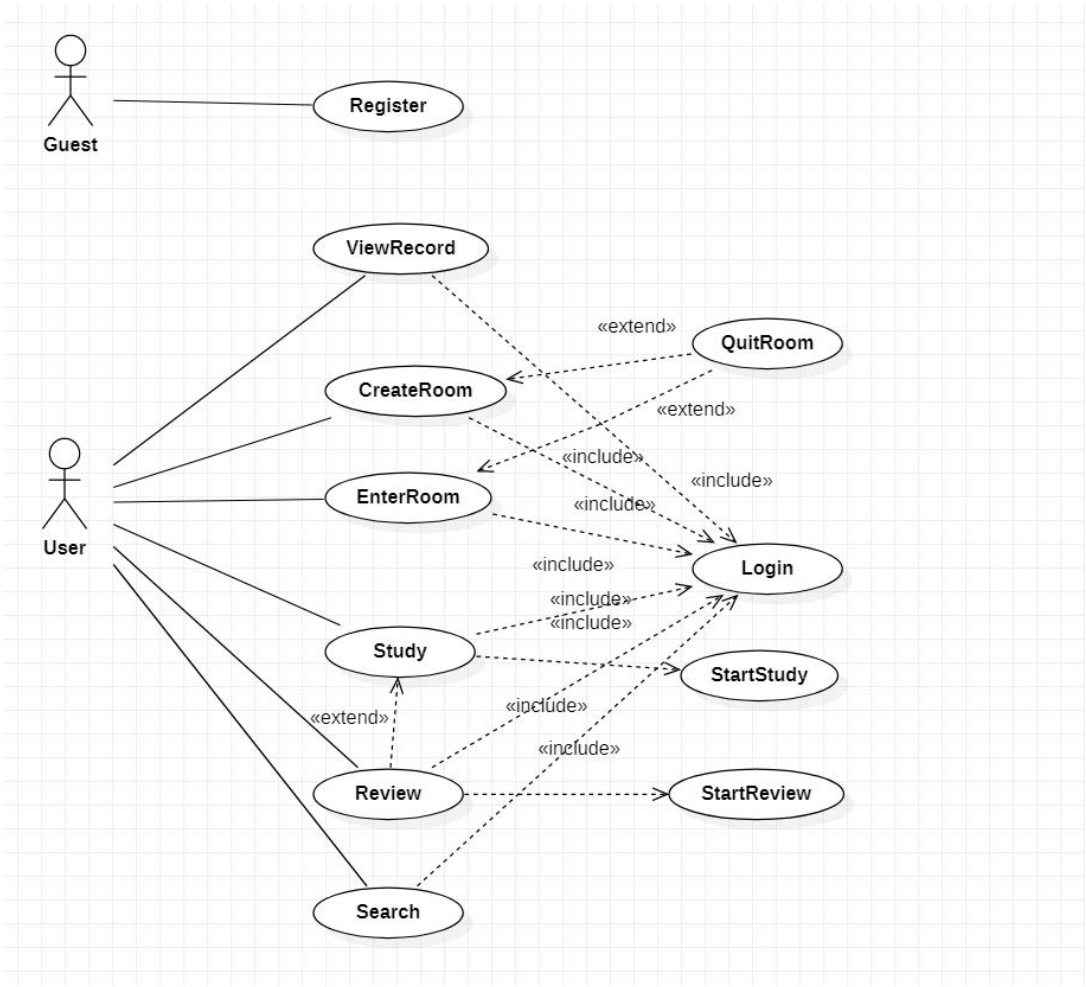
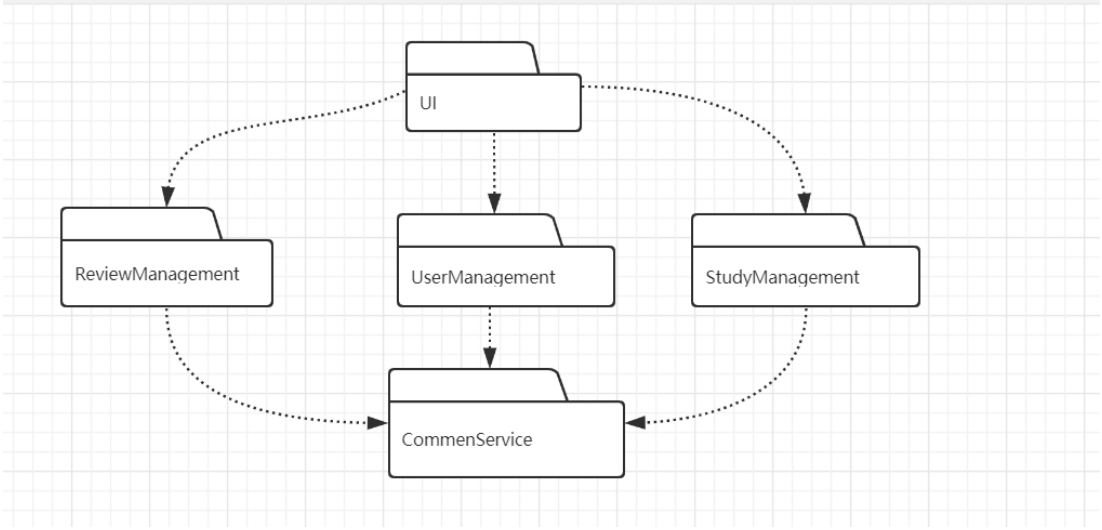


图 1 用例图

4.3. 系统逻辑视图

4.3.1. 系统架构

本系统采用了三层体系结构和客户机 / 服务器模式。最顶层是 UI 层，是用户直接交互的客户机部分。中间层是应用逻辑层，包含实现系统功能的各子系统。第三层为公共服务层，实现对数据的存储和管理等功能。



4.3.2. 子系统

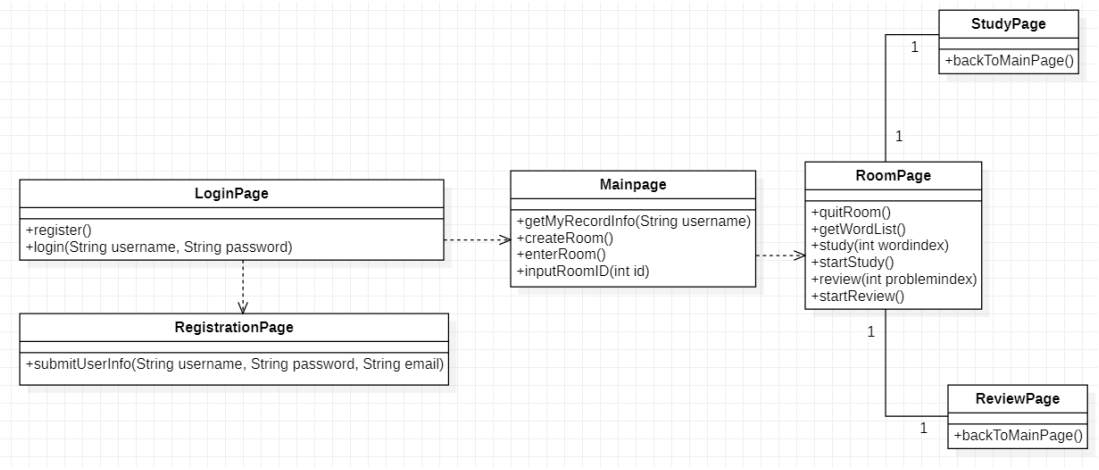
（1）UI 子系统。

功能：负责与用户的交互。UI 层主要提供各种输入框、按钮、文本框等进行信息的显示和与用户的交互的功能。

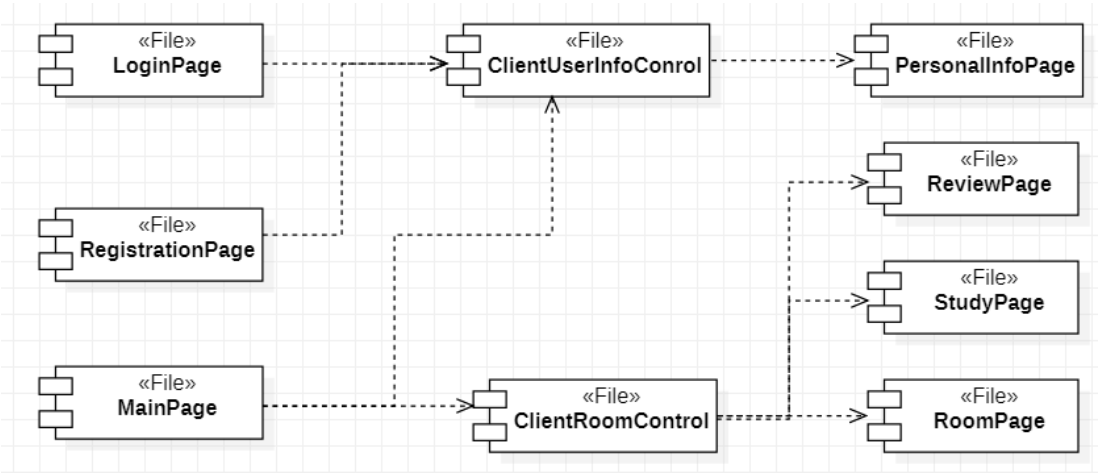
服务：

public void backToMainPage()
public void register()
public int login(String username, String password)
public void getMyRecordInfo(String username)
public int creatRoom()
public int enterRoom()
public bool inputRoomID(int id)
public void quitRoom()
public list getWorldList()
public void study(int wordindex)
public void startStudy()
public void review(int problemindex)
public void startReview()
public int submitUserInfo(String username,String password,String email)
public void backToMainPage()

类图：



组件图：



（2）UserManagement 子系统。

功能：负责用户信息的管理，控制登录、注册、个人资料修改等。

服务：

public int updatePersonalInfo()

public int editPersonalInfo()

public void register()

public void showPersonalInfo()

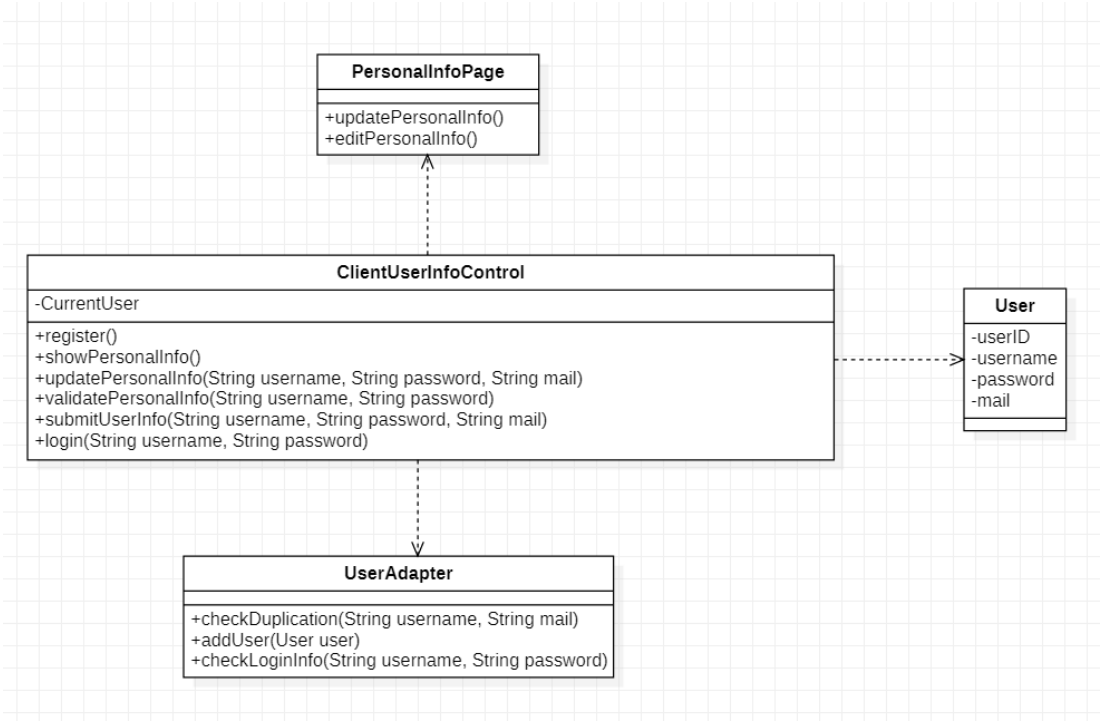
public int updatePersonalInfo(String username,String password,String mail)

public bool validatePersonalInfo(String username,String password)

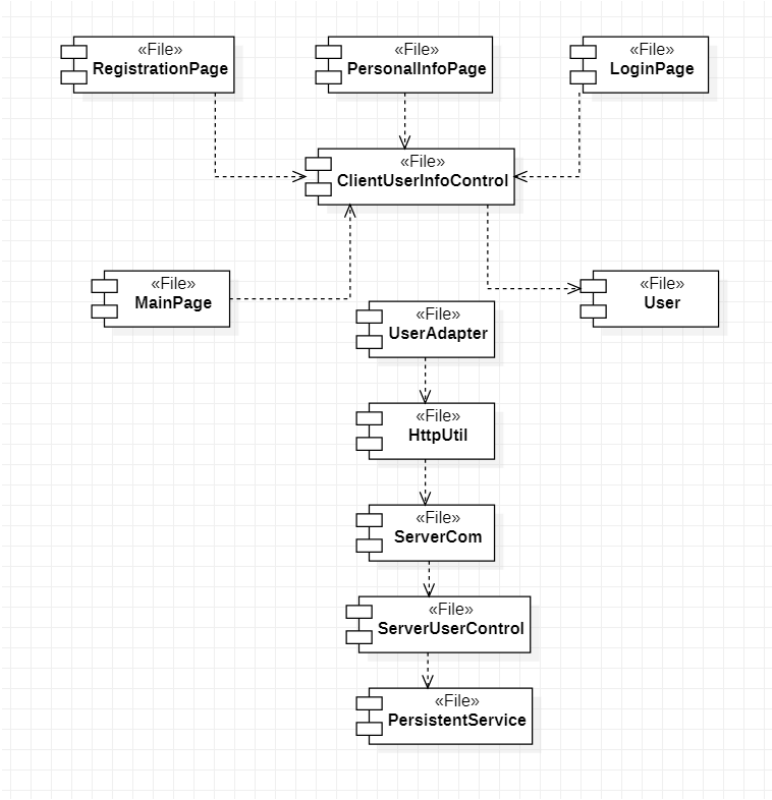
public int submitUserInfo(String username,String password,String mail)

public int login(String username,String password)

类图：


PersonalInfoPage
+updatePersonalInfo()
+editPersonalInfo()
ClientUserInfoControl
-CurrentUser
+register()
+showPersonalInfo()
+updatePersonalInfo(String username, String password, String mail)
+validatePersonalInfo(String username, String password)
+submitUserInfo(String username, String password, String mail)
+login(String username, String password)
User
-userID
-username
-password
-mail
UserAdapter
+checkDuplication(String username, String mail)
+addUser(User user)
+checkLoginInfo(String username, String password)

组件图：


«File»
RegistrationPage
«File»
PersonalInfoPage
«File»
LoginPage
«File»
ClientUserInfoControl
«File»
MainPage
«File»
User
«File»
UserAdapter
«File»
HttpUtil
«File»
ServerCom
«File»
ServerUserControl
«File»
PersistentService

（3）StudyManagement 子系统

功能：负责学习页面的创建，给出学习单词及释义，选择学习单词等。

服务：

public void startStudy()

public int creatStudyRoom()

public Word getWords()

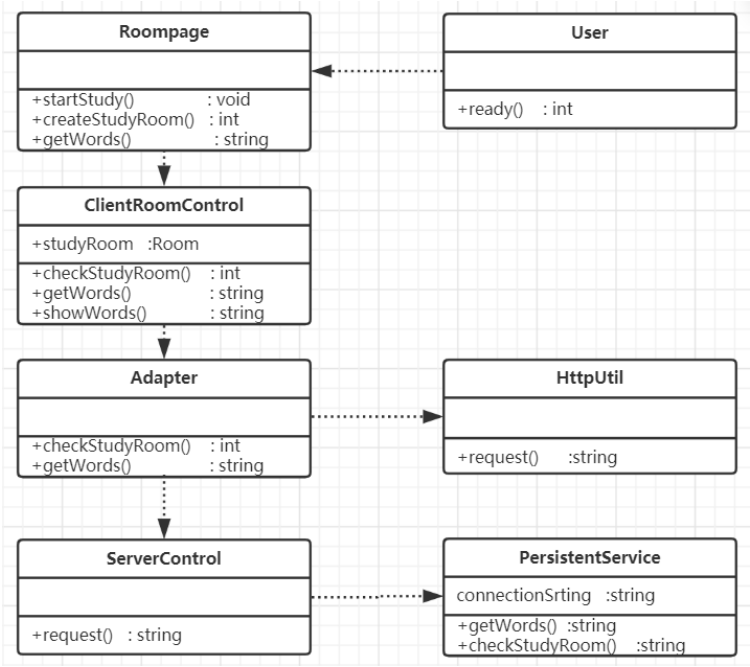
public int ready()

public int cheakStudyRoom()

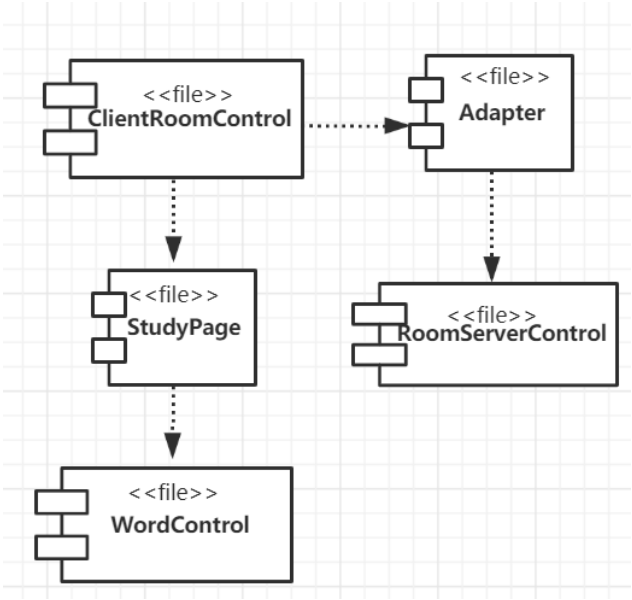
public String showWords()

public String request()

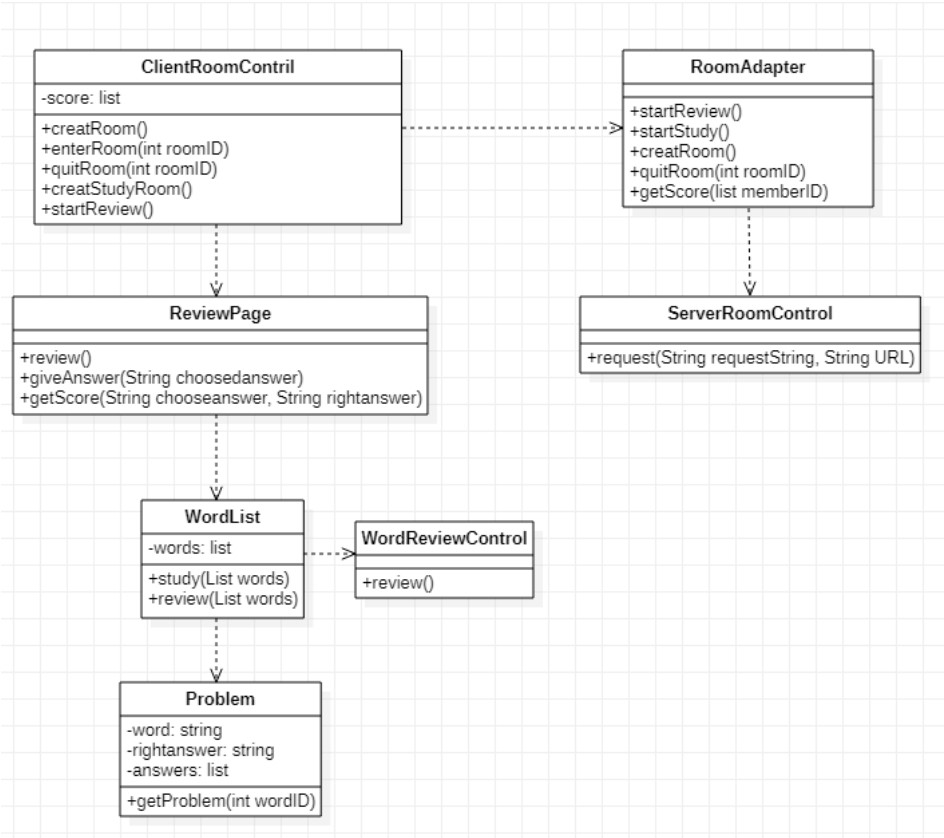
类图：



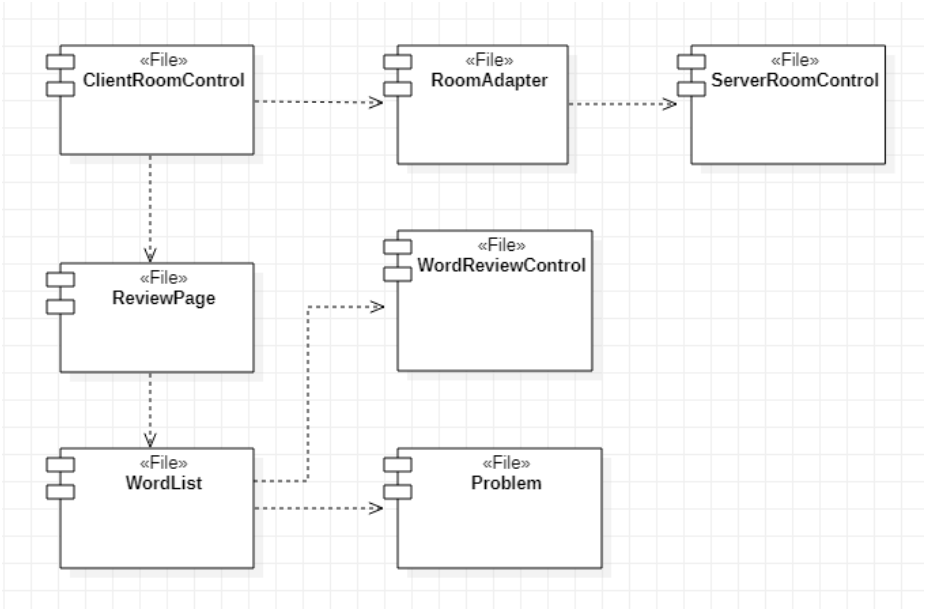
组件图：



（4）ReviewManagement 子系统
功能：负责复习页面的创建，给出复习题并判断正确与错误答案，显示所有人的正确率等。
服务：
public int createRoom()
public int enterRoom(int roomID)
public void quitRoom(int roomID)
public int creatStudyRoom()
public void startReview()
public void startStudy()
public int getScore(list memberID)
public int review()
public int giveAnswer(String choosedanswer)
public int getScore(String chooseanser,String rightanswer)
public int request(String requestString,String URL)
public void study(List words)
public void review(List words)
public Problem getProblem(int wordID)
类图：



组件图：



（5）CommonService 子系统

功能：提供数据存取管理、客户端等功能。

服务：

public int request(String requestString,String URL)

public int httpPost(String requestString,String URL)

public int checkDuplication(String sqlString)

public int intsertRecord(Object sqlString)

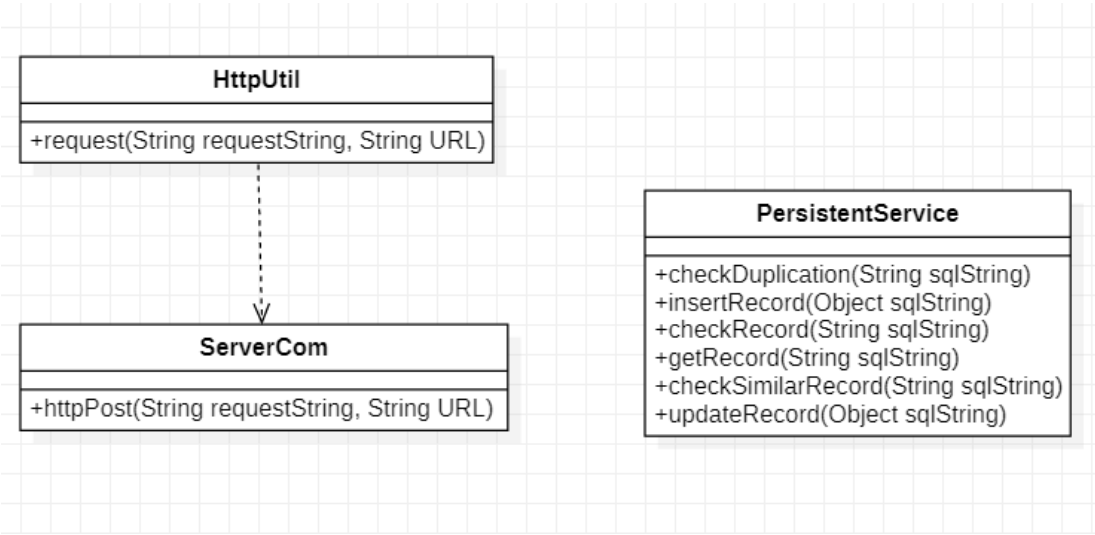
public int checkRecord(String sqlString)

public String getRecord(String sqlString)

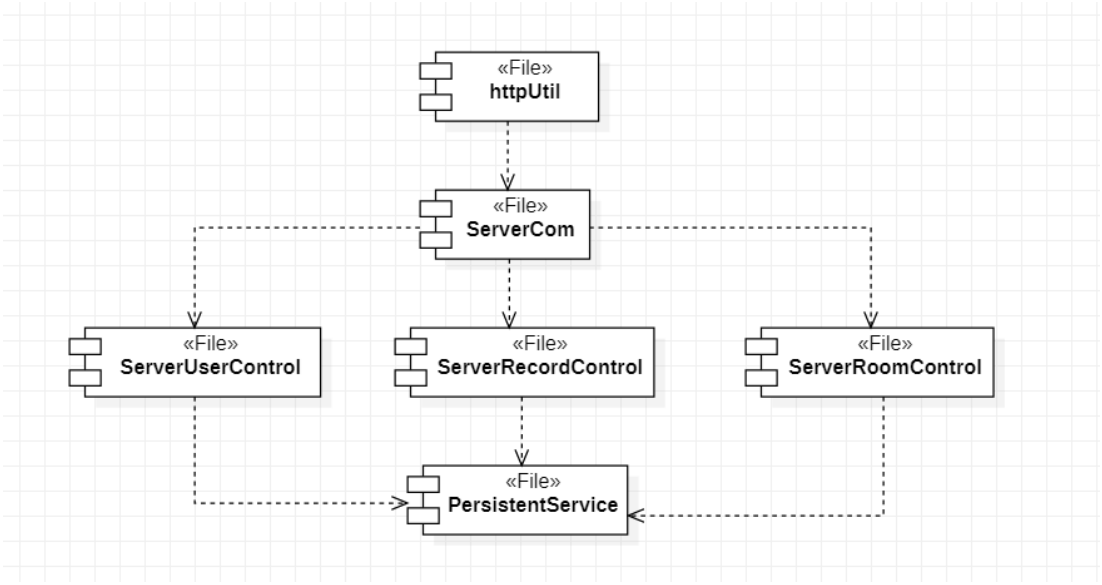
public int checkSimilarRecord(String sqlString)

public int updateRecord(Object sqlString)

类图：



组件图：



4.3.3. 用例实现

本系统中的核心用例为 ReviewManagement（见图 2）。

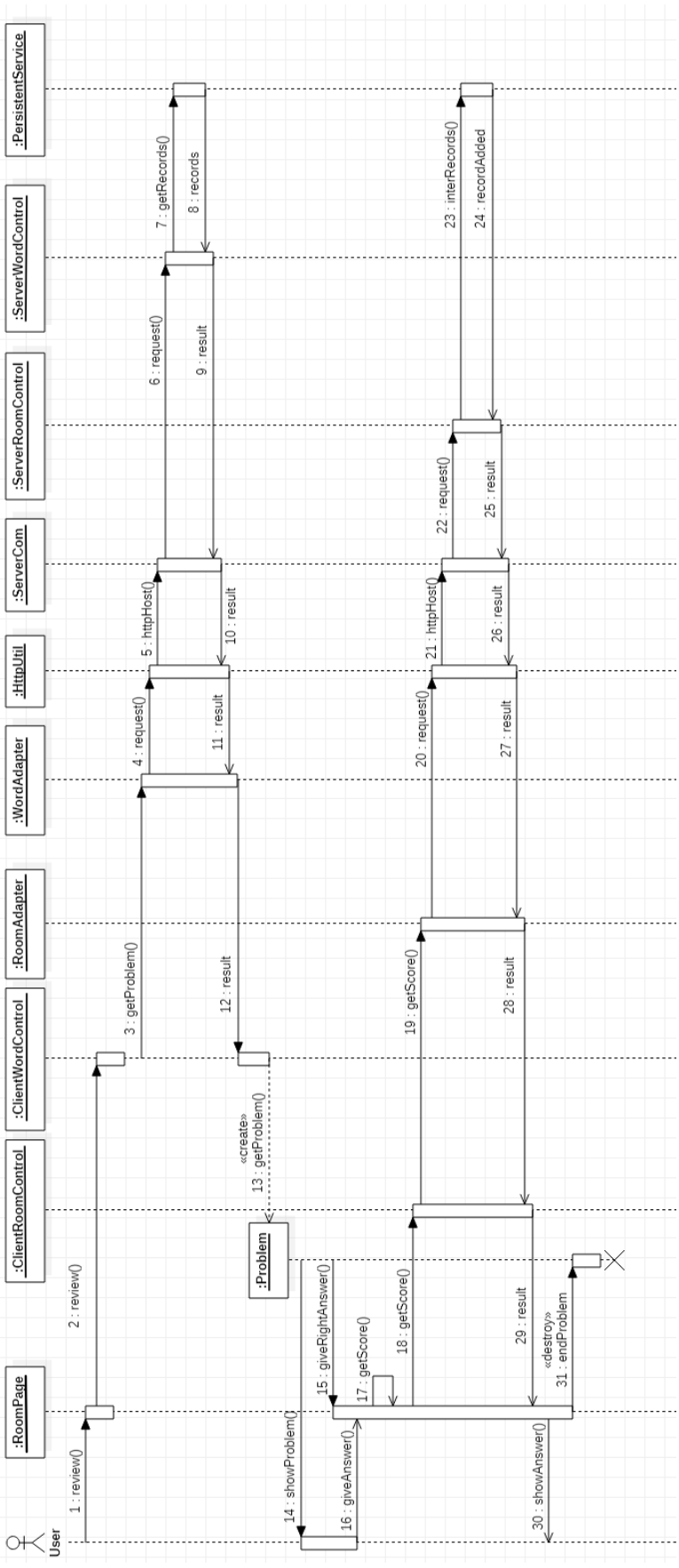


图 2 <ReviewManagement>用例实现图

4.3.4. 子系统协作

核心用例 ReviewManagement 的子系统协作图如图 3 所示。

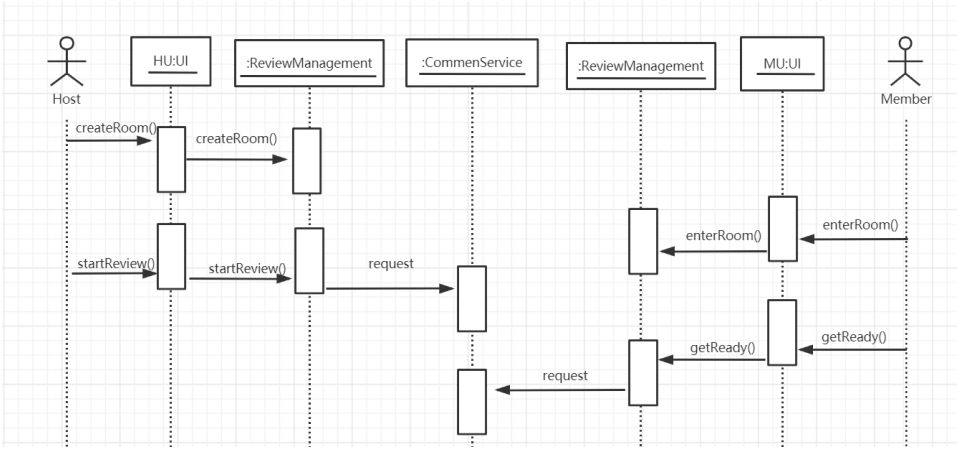


图 3 <ReviewManagement> 子系统协作图

4.4. 系统运行视图

(1) 客户端进程视图

在 Client 端，为了保证用户体验，UI 线程只负责边界对象与用户的交互。同时对每一个控制器单独启动一个线程。同时对于通信模块，也启动一个线程。这样的设计能够保证前台、功能实现、通信不会阻塞应用的运行。

(2) 服务端进程视图

服务端进程设计中，对于通信部分，将依据请求的并发性，进行多线程的管理，以响应每个用户的请求。同时，各个控制对象也有自己单独的线程。对于持久服务，我们也将采用多线程 机制保证其响应性。

4.5. 系统实现视图

(1) 系统开发环境

开发环境：JetBrains WebStorm、PyCharm。

开发语言：JavaScript、Python。

(2) 系统开发模型

本系统中按照组件图进行了相应的软件文件定义，因此系统开发模型与本文档与软件设计模型文档中的组件图一致。

4.6. 系统物理视图

此处插入软件设计模型文档中的部署视图。

硬件配置要求：

（1）用户端：一台配有 1GHz 双核处理器及 1GB 内存(或以上)的手机。为保证较为良好的用户体验，此为流畅运行大多数安卓应用程序的最低配置。

（2）服务器：初期需要一台双核心处理器，配以 4GB 内存与 100GB 存储空间的服务器，运行 Web App 以及与之配套的数据库。由于初期用户量不大，可暂时将程序与数据库运行在同一台服务器上，通过内部接口连接，但在后期用户量扩大时，应将数据库部署至另一台服务器上，并提高服务器配置。

（3）网络：50Mbps 的互联网络。本 Web App 虽然对单个用户而言数据传输要求带宽不高，但是在大量用户学习单词过程中需要实时传输进度信息，仍需要较大的带宽。

4.7. 边界条件设计

通过定义启动、关闭用例来考虑边界条件。

（1）启动服务器 StartServer（见图 4）。

a. 用户打开网页。

b. 启动 ServerCom 对象。

c. 启动持久服务 PersistentService 对象。

d. 依次启动 ServerRecordControl,ServerRoomControl,ServerUserInfoControl 对象，， 同时把 ServerCom 对象、PersistentService 对象的引用传给这些对象。

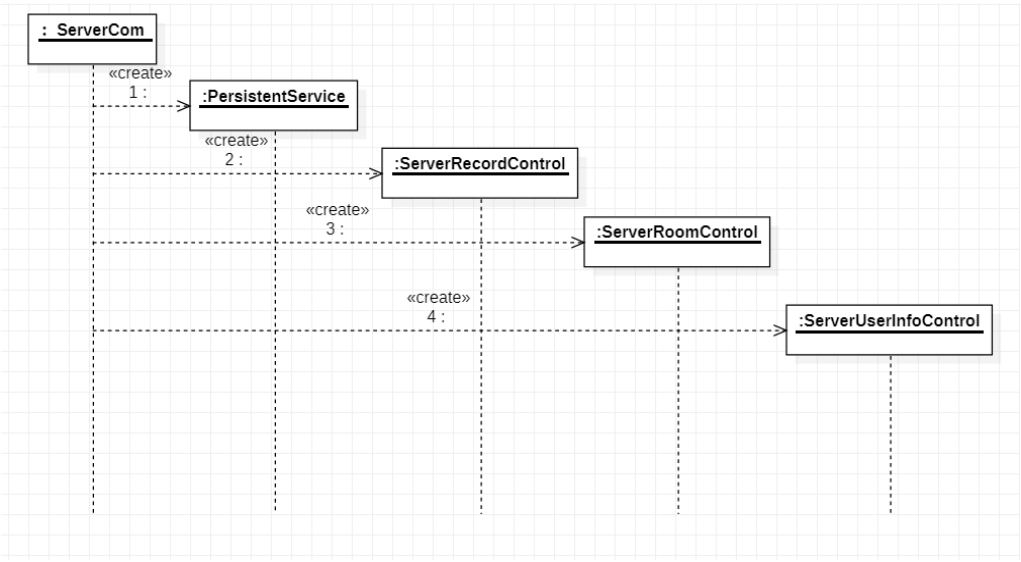


图 4 启动服务器用例实现

（2）关闭服务器 StopServer（见图 5）。

a. 用户关闭网页。

b. 系统删除 ServerUserInfoControl,ServerRoomControl,ServerRecordControl 对象。

c. 系统断开与数据库连接，删除 PersistentService 对象。

d. 系统删除 ServerCom 对象。

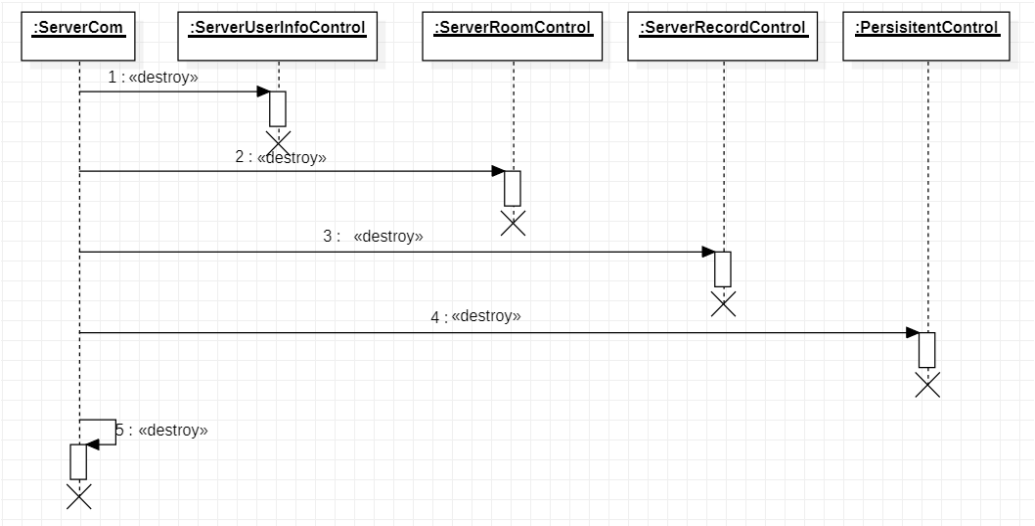


图 5 关闭服务器用例实现

(3) 打开网页用例（见图 6）

a. 用户打开网页。

b.系统启动 MainPage 对象。

c. MainPage 对象启动 HttpUtil 对象。

d. MainPage 对象启动 UserAdapter,RoomAdapter 对象，并把 HttpUtil 对象传给他们。

e. MainPage 对象启动 ClientUserInfoControl, ClientRoomControl, WordStudyControl, WordReviewControl 对象，并把 UserAdapter,RoomAdapter 对象引用传给它们。

f. MainPage 对象启动 LoginPage 对象，同时把 ClientUserInfoControl 传给 LoginPage。

(4) 关闭网页（见图 7）

a. 用户点击网页关闭按钮

b. 依次关闭各个 ClientControl 对象。

c. 依次关闭各个 Adapter 对象。

d. 关闭 HttpUtil 对象。

e. 关闭 MainPage 对象。

(5) 异常情况

a. 系统在出现异常情况下，刷新网页或者重启 Server 即可以。

b. 数据库本身靠事务特性维持其正确性。

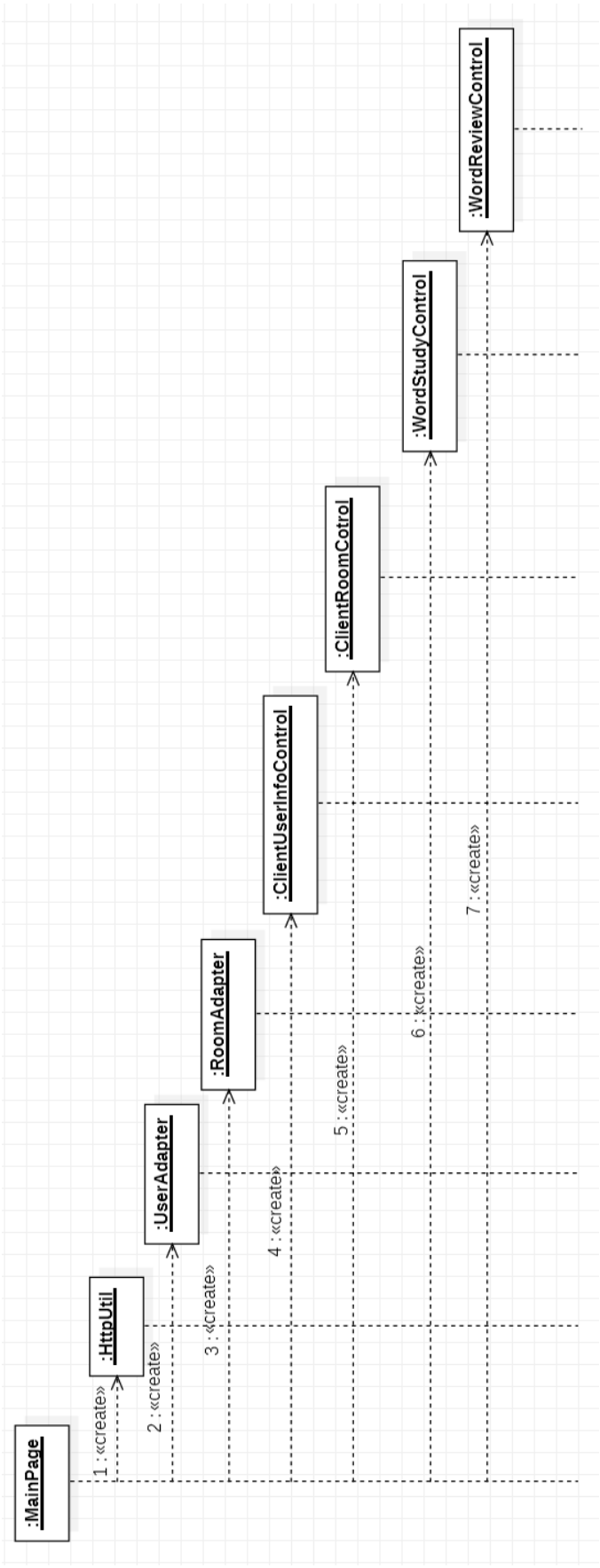

:MainPage
:HttpUtil
:UserAdapter
:RoomAdapter
:ClientUserInfoControl
:ClientRoomCotrol
:WordStudyControl
:WordReviewControl
1 : «create»
2 : «create»
3 : «create»
4 : «create»
5 : «create»
6 : «create»
7 : «create»

图 6 打开网页用例实现

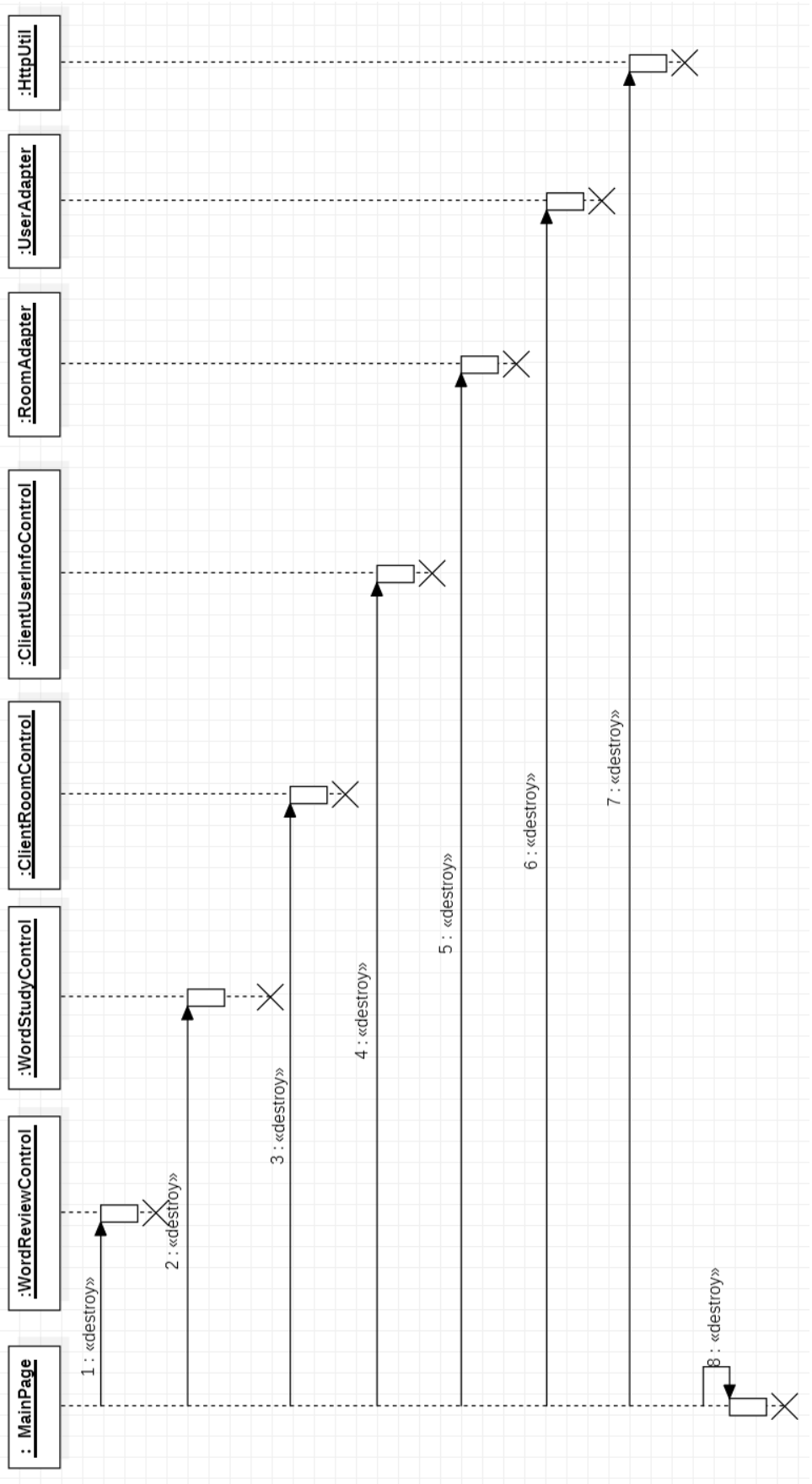


图 1-7 关闭网页用例实现

4.8. 数据管理设计

本系统需存储大量用户、单词、复习题等信息，基于经验，我们选择使用关系型数据库存储信息。

（1）用户信息表 UserInfo（见表 1）

序号	字段	说明	数据类型	允许为空	主键	单位	备注
1	username	用户名	String	N	Y		
2	userID	用户 ID	Int	N	N		
3	password	用户密码	String	N	N		
4	email	用户邮箱	String	N	N		
5	school	用户学校	String	Y	N		
6	goal	用户学习目标	String	Y	N		

表 1 用户信息表 UserInfo

（2）学习记录表 StudyRecord（见表 2）

序号	字段	说明	数据类型	允许为空	主键	单位	备注
1	date	学习日期	Date	N	Y		
2	words	学习单词列表	List	N	N		

表 2 学习记录表 StudyRecord

（3）单词信息表 Word（见表 3）

序号	字段	说明	数据类型	允许为空	主键	单位	备注
1	name	单词拼写	String	N	Y		
2	partOfSpeech	单词词性	String	N	N		
3	meaning	单词释义	String	N	N		
4	exampleSentence	单词例句	String	N	N		

表 3 单词信息表 Word

（4）复习题信息表 Problem（见表 4）

序号	字段	说明	数据类型	允许为空	主键	单位	备注
1	word	单词题	String	N	Y		
2	answerRight	正确答案	String	N	N		
3	answerWrong	错误答案	List	N	N		包含 3 个错误答案

表 4 单词信息表 Word

（5）房间信息表 AbstractRoom（见表 5）

序号	字段	说明	数据类型	允许为空	主键	单位	备注
1	roomID	房间 ID	Int	N	Y		
2	usernum	房间内用户数	Int	N	N		
3	hostname	房主名	String	N	N		

表 5 房间信息表 AbstractRoom

实体类设计类图如下图所示。

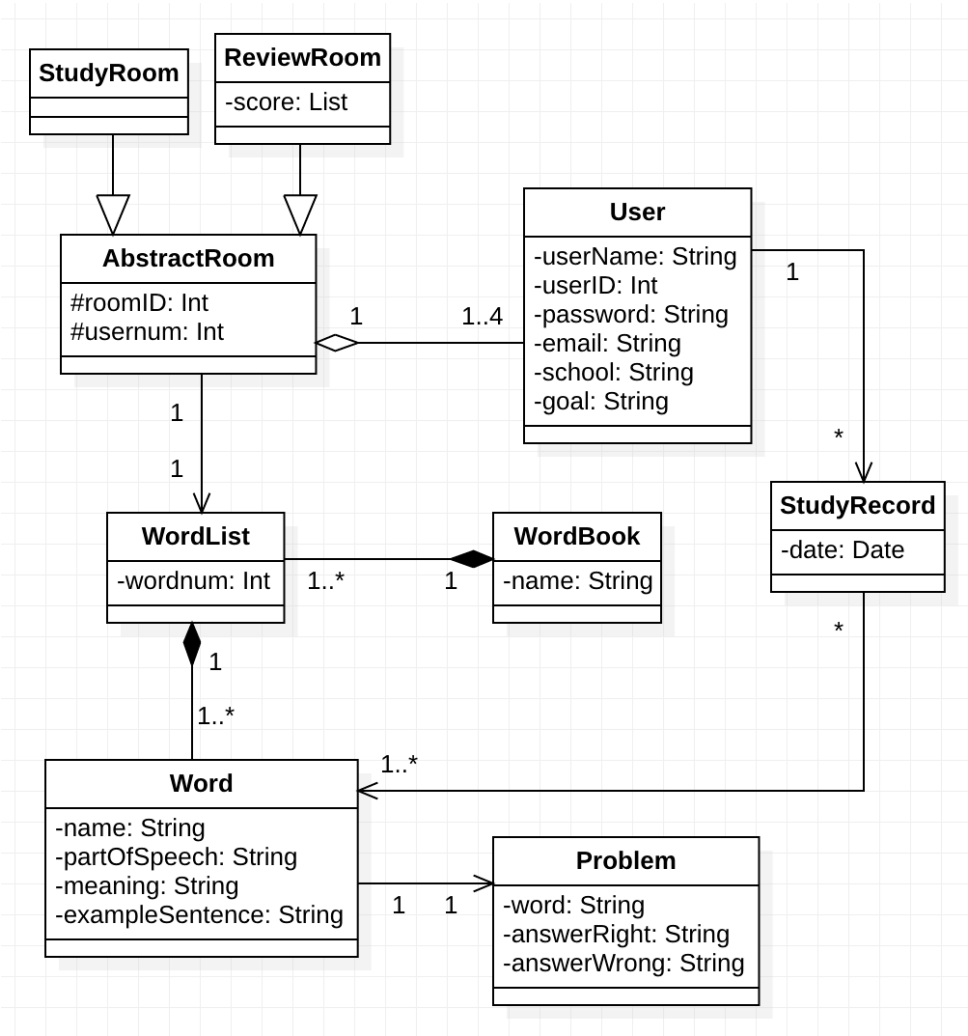


图 8 实体类设计类图

4.9. 其他设计

4.9.1. 访问控制与安全设计

（1）登录用户

登录用户在使用他们的权限之前需先提供用户名与密码，经检查是匹配的才登录成功。登录用户的权限包括：

a. 用户信息：查看与修改信息、查看学习记录；

b. 房间：创建房间与加入房间。

（2）房主

登录用户在创建房间后自动成为房主。

房主的权限包括：

a. 在房间内选择单词列表；

b. 选择房间模式：学习或复习；

c. 学习或复习单词；

d. 若为学习房间，在学习结束后可以选择是否继续复习。

（3）房间成员

登录用户在加入房间后自动成为房间成员。

房间成员的权限包括：

a. 查看房间当前单词列表；

b. 学习或复习单词。

4.9.2. 可靠性设计

为提高系统的可靠性，我们考虑采取备份方案。设置备用服务器，在主服务器发生故障时由开发团队管理人员将系统运行切换至备用服务器上；数据库每天备份至备用服务器，防止数据丢失或意外更改。

4.9.3. 可扩展性设计

初期系统运行在双核心、4GB 内存的小型第三方租赁服务器上，能满足一定数量的用户与并发请求，但是在用户规模扩大之后，可考虑增加服务器数量或性能，迁移程序与数据。