

# 趣学+软件设计文档

2017-5-1~2017-6-24

状态： [ <input checked="" type="checkbox"/> ] 草稿 [ <input type="checkbox"/> ] 修改中 [ <input type="checkbox"/> ] 定稿	文件标签：	软件设计文档、趣学
	版本：	2017(1.0)
	作者：	王晓军
	日期：	2017-6-24

# 编辑历史

文件名称：				
趣学软件设计文档				
文件说明：				
本文档是对趣学 app 的项目设计说明				
编辑历史：				
编辑时间	版本	作者	编辑内容	标记
2017.6.24	1.0	王晓军	建立文档	正常

# 目录

- 1 前言.....3
  - 1.1 项目的背景和意义.....3
  - 1.2 项目的目标和范围.....3
  - 1.3 实验团队成员.....3
  - 1.4 本文结构简介.....3
- 2 技术与原理.....5
  - 2.1 前端-Android 客户端.....5
    - 2.1.1 框架..... 5
    - 2.1.2 功能实现..... 5
  - 2.2 服务端-Django.....5
    - 2.2.1 框架..... 5
    - 2.2.2 功能实现技术..... 5
    - 2.2.3 服务端代码详情..... 6
- 3 架构设计.....8
  - 3.1 登录注册.....9
  - 3.2 试题类型选择.....9
  - 3.3 进行测评并提交用户成绩.....10
  - 3.4 完整的架构设计.....10
- 4 总结..... 10

# 1 前言

## 1.1 项目的背景和意义

随着百词斩等应用的流行，人们开始将一些学习行为从线下迁移到线上。

线上学习行为相对于线下学习行为的好处是不言而喻的，一方面省去了大量使用纸质材料的繁琐流程，另一方面，线上测评应用的出现使得用户可以及时了解自己的学习成果，及时获取学习反馈信息，用户可以了解自己的学习情况，根据自己的学习进步情况适时调整试题的难度，减少了用户无意义的反复刷题过程，让用户能不断通过做题提高自己的水平，高效地发挥了试题库的作用。这些好处使得线上测评应用成为 k12 教育领域里的一个风口。

## 1.2 项目的目标和范围

本项目将实现如下的功能：

- ①注册：用户用手机号注册；
- ②登录：用户名、密码登录；
- ③试题难度选择：分为三个难度等级:easy、mediu、hard
- ④试题类型选择：选择题与判断题
- ⑤用户成绩提交；
- ⑥支持用户自己选择测评的学科：通用知识、娱乐、科学、体育、政治、历史、艺术等。

## 1.3 实验团队成员

项目成员					
姓名	学号	项目角色	身份	Github 权限	email
陈浩森	14332003	developer	组员	developer	1034121839@qq.com
冯爽	14331064	developer	组员	developer	1151009747@qq.com
谷溢	14332006	developer	组员	developer	guyi3@mail2.sysu.edu.cn
李泽龙	14331144	administrator	组长	administator	731986844@qq.com
王晓军	14331266	developer	组员	developer	820640069@qq.com

## 1.4 本文结构简介

本文的结构将以软件工程的结构来组织。

第二章：技术与原理部分，本文将分析为完成上述的项目目的和需求所需要的设计模式、编码原理和技术，将检索和参阅的技术资料和文献中的相关部分进行简要说明，并分析其在实现本工程设计上的意义。

第三章：架构设计部分，本文会在技术原理的基础上，分析整个工程系统架构的选择、设计的思路和相关技术的说明，同时对主要业务用例从技术角度进行详细分析说明。

第四章：总结部分，本文将总结本次项目所有工作成果，对项目的优势和不足、可拓展性、可维护性进行评估，对项目的发展进行展望。

## 2 技术与原理

本章将对项目中实施的设计模式和相关技术手段进行阐述，对在项目中采用的具体设计和实现方式进行说明，并分析其在实现本工程设计上的存在意义。

### 2.1 前端-Android 客户端

#### 2.1.1 框架

开发环境：Android Studio 作为开发 IDE；

- 1.初始页面，使用 Android 自带的前端框架：Editview、Button；
- 2.试题信息页面，使用 Android ListView 呈现内容；
- 3.测评页面，使用 Android ListView 呈现内容；
- 4.底部分栏，类似微信 QQ 效果，使用 Android 框架里的 Fragment 实现。

#### 2.1.2 功能实现

- 1.选择学科
- 2.选择难度
- 3.选择试题类型

功能说明，由于试题类型字段已存储于服务端数据中，因此此处的试题选择，前端仅需要提供相应的试题类型参数，并向后端发送带试题类型参数的 url 请求即可。（http get 请求）

### 2.2 服务端-Django

#### 2.2.1 框架

1. 主体框架使用 Django 框架。
2. MVC 框架使用 Django 集成的 MVC 框架；

#### 2.2.2 功能实现技术

1. 使用 JSON 格式的数据进行前后端的数据交互。

2. 使用 SQLite 作为后端数据库，存储试题库，以及用户的测评分值信息等。
3. 使用 Android 的本地 SQLite 数据库暂存用户账户信息

### 2.2.3 服务端代码详情

服务端模块划分

服务端目录是 Server 目录，Server 目录的内容如下：

```
├─MathQuestion          数学计算问题模块。因某些原因而暂停开发
|   └─migrations
|   |   └─__pycache__
|   └─__pycache__
├─OpenTrivia            选择题和判断题模块。题库来源于
OpenTriviaDB
|   └─migrations
|   |   └─__pycache__
|   └─__pycache__
├─Server                主模块
|   └─__pycache__
├─SubmitQuestion        用户原创问题模块。
|   └─migrations
|   |   └─__pycache__
|   └─__pycache__
└─User                  用户模块
    └─migrations
    |   └─__pycache__
    └─__pycache__
```

服务端总路由：（代码位于 Server/urls.py）

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^MathQuestion/', include('MathQuestion.urls')),
    url(r'^OpenTrivia/', include('OpenTrivia.urls')),
    url(r'^User/', include('User.urls')),
    url(r'^SubmitQuestion/', include('SubmitQuestion.urls')),
]
```

### Object-Oriented Programming:

在每个模块中，都有一个 `models.py` 文件，里面定义着该模块所负责的类。每个类会映射到 SQLite 数据库中的一个表。

例如，在 `OpenTrivia/models.py` 中，定义了 2 个类 `MultipleQuestion` 和 `BooleanQuestion`，对应着业务中的“选择题”和“判断题”。

### Structure Programming:

服务端使用的是 Python。Python 的导入与管理模块的方式，可以很方便地进行结构化编程。

例如：

我在 `User/views.py` 中定义了一个 `isUserAndPasswordCorrect` 函数来验证用户。其他模块可以 `import` 这个函数来进行用户验证。

```
1  from django.shortcuts import render
2  from django.http import HttpResponse, JsonResponse
3  from django.views.decorators.csrf import csrf_exempt
4
5  from User.views import isUserAndPasswordCorrect
6  from .models import SubmitMultipleQuestion, SubmitBooleanQuestion
7  from OpenTrivia.views import getCategoryByCode, convertCategoryCode
8
```

```
54      category = getCategoryByCode(category)
55      if (category == "NULL"):
56          print("category error")
57          resultJsonData["response_code"] = 2
58          return JsonResponse(resultJsonData)
59
60      # 验证用户
61      result = isUserAndPasswordCorrect(username, password)
62      if (result["response_code"] != 0):
63          resultJsonData["response_code"] = result["response_code"]
64          return JsonResponse(resultJsonData)
65
66      something, canCreate = SubmitMultipleQuestion.objects.get_or_create
```

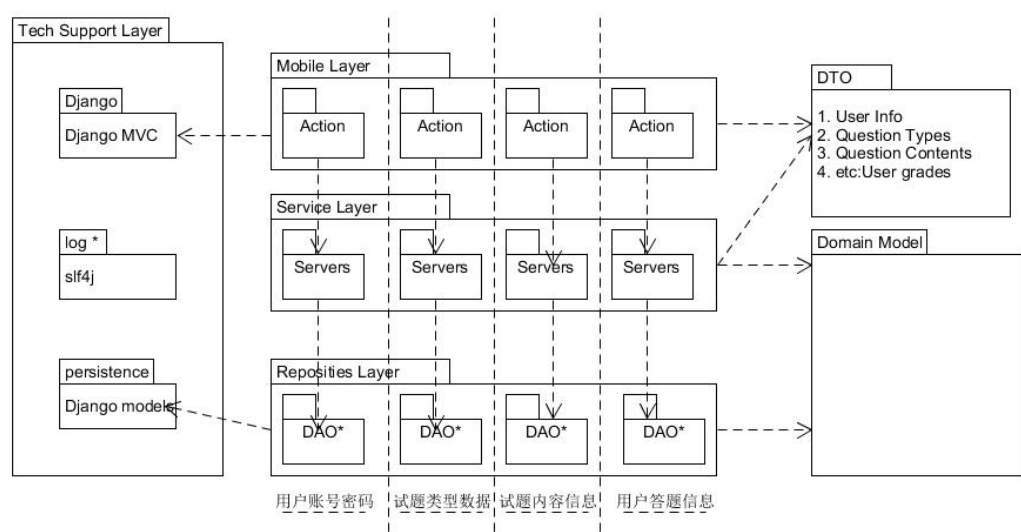
在 `SubmitQuestion/views.py` 中使用这个 `isUserAndPasswordCorrect` 函数



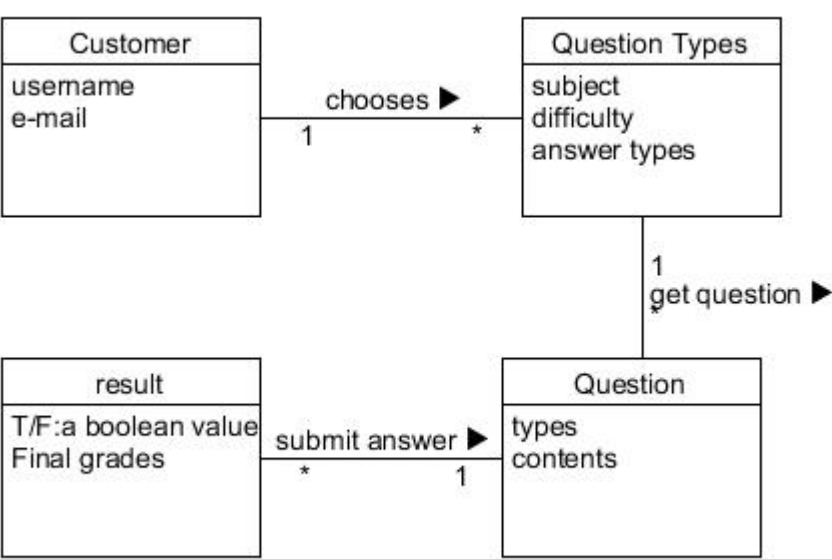
### 3 架构设计

本章将对趣学项目的核心模块：登录注册、试题类型选择、进行测评并提交用户成绩进行架构设计，以面向对象编程的方法，分析整个工程系统架构的选择、设计的思路和相关技术的说明。

架构设计的逻辑视图如下所示：



领域模型图如下所示：



### 3.1 登录注册

1. 后端：用户的账户数据存储在服务端数据库，使用 SQLite 管理。
2. 前端：Android 客户端使用 Android 原生的 SQLite 数据库管理数据，将用户数据存储在本地，用户第二次启动客户端时客户端自动读取用户信息，简化二次登录流程。

SQLite 数据库说明：

SQLite 是一个进程内的库，实现了自给自足的、无服务器的、零配置的、事务性的 SQL 数据库引擎。它是一个零配置的数据库，这意味着与其他数据库一样，您不需要在系统中配置。

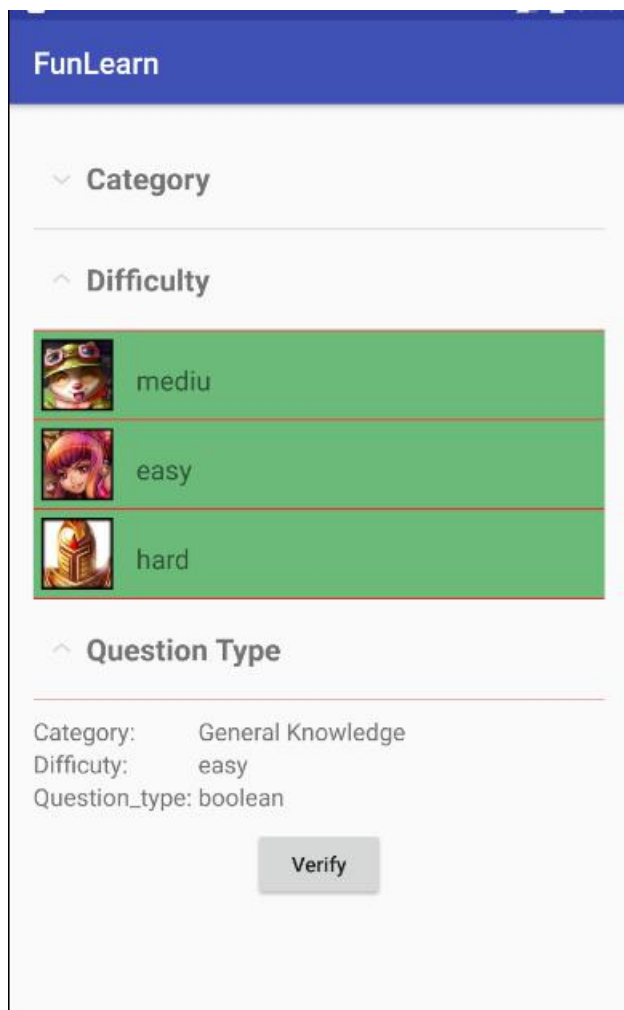
就像其他数据库，SQLite 引擎不是一个独立的进程，可以按应用程序需求进行静态或动态连接。SQLite 直接访问其存储文件。

### 3.2 试题类型选择

试题类型支持的选择包括：

1. 难度选择：easy、mediu、hard；
2. 学科选择：generally knowledge、science、history、art...等
3. 试题类型：multiply、boolean

具体实现的 UI 图：



The screenshot shows the FunLearn app interface. At the top is a blue header with the text "FunLearn". Below the header, there are three expandable sections: "Category", "Difficulty", and "Question Type". The "Difficulty" section is currently expanded, showing three options: "mediu" (with a cartoon character icon), "easy" (with a cartoon character icon), and "hard" (with a cartoon character icon). Below these sections, there is a summary of the selected filters: "Category: General Knowledge", "Difficulty: easy", and "Question\_type: boolean". At the bottom of the form is a "Verify" button.

### 3.3 进行测评并提交用户成绩

1. 测评试题呈现：使用 java 的 URL 类获取数据（http get 方法），再使用 Android Listview 呈现试题信息；
2. 提交用户成绩：使用 java 的 URL 类提交数据（http post 方法）

### 3.4 完整的架构设计

趣学严格地按照软件工程面向对象的方法进行开发。下图是它的完整类包图。这使得各包独立成为模块，方便代码复用，又可以有机地结合和比较，提高开发效率。

## 4 总结

1. 优点：趣学作为一款可以自由选择试题难度的在线测评 APP，优点是

显的：它除了提供难度选择的功能以外，还提供了多种学科的试题选择，可以满足不同学科背景的用户需求。

2. 缺点但同时我们也看到，趣学并非完美，它也存在着短板和不足。例如目前趣学的难度选择还是由用户自己手动选择。并不能根据用户的测评数据做到自动选择，目前趣学测评应用仍然处于初步阶段，还未引入大数据分析方法。

3. 未来的改进方案：人工智能是当下最热门的研究方向之一，对智能测评应用的研究是当今 K12 教育领域的重点工作。将自适应学习如对用户数据进行线性回归分析等技术引入在线测评应用，使测评应用能根据用户的测评信息智能化地推荐试题，将是我们未来研究的方向。我们希望将趣学应用打造成一款智能化学习应用。