**新**

**高**

**考**

**课**

**表**

**安**

**排**

**器**

**项目策划草案**

--育星项目组

**目录**

**一.项目要求与背景**： ………………………………………………………… 3

1. 目标：……………………………………………………… 3
2. 新高考模式简介…………………………………3
3. 项目提供的服务…………………………………3
4. 需求：………………………………………………………6
5. 可用的资源：……………………………………………6

**二.设计主要原则**：………………………………………………………………7

**三应用程序框架**：…………………………………………………………………8

1. UML……………………………………………………………8
2. UML阐释……………………………………………………8
3. 设计模式……………………………………………………9

**四.实现**：…………………………………………………………………………………9

1.时间线……………………………………………………9

2.平台与其他细节………………………………………9

**五.引用：**…………………………………………………………………………………………9

**一.项目要求与背景**：

1. 目标

本项目旨在开发一款可以帮助高中教务处拟定课表的APP。新高考选课模式导致选择物化生政史地的学生人数不一，有学校可能在排课上面有一定的麻烦，开发的目的旨在自动根据学生的课程信息生成互不冲突的课表。

1. 新高考模式简介

\*高中课程必考科目包括语文、数学、英语三门。

\*选考科目根据新高考的政策要求，要求考生在历史，地理，政治，生物，化学，物理六门课中任选三门。选考课实际教学过程采用“大走班”模式。所谓“大走班”模式,是指语文、数学、外语三门必考科目可以保持高一行政班不变,三门选考科目所有学生均通过走班完成教学。这种模式的优点是可以全部满足学生的选择进行走班，由于语文、数学、外语三门必考科目可以保持原高一行政班不变,教师从高一开始任教至毕业的学生不变,有利于学校对这三门学科教师进行评价,调动他们的工作积极性。（**\*特注**：关于班级，分行政班与教学班。在同一个行政班的语数外课相同。然后选课分教学班，三个行政班组成一个部分，一个部分里面选相同的科目的放在一起组成一个教学班。比如一二三行政班（CL1,CL2,CL3）组成部分1，四五六行政班组成部分2，然后部分1里面选物理的组在一起为物理教学班1 (PH1)，选化学的组成化学教学班1(CH1), 部分二中选生物的组成生物教学班2(BI2)。一个教学班的学生在一起上课。其他的以此类推。

）

\*在选考课和必考课之外，还有文学艺术选修和体育的课程的必修要求。

\*该模式在上海和浙江已经施行，施行的结果被证明导致了一些抱怨和混乱，然而改革面向自由的初心是好的。截至今年，北京、山东、天津、海南、湖南、江西、广东、江苏、黑龙江、贵州、四川、吉林、重庆、河北、湖北、山西、西藏、福建、内蒙古、河南、辽宁已经正式启动该模式。关于试点的详细情况在本文档中的2.中进行了介绍。

1. 项目提供的服务

本项目面向三个群体开放功能，学生，教师，和管理员。三个群体应该可以通过本平台实现下述任务：

学生

**（1）Upload:可以通过项目APP将自己的姓名，学号，性别，行政班号以及选择的三门课程提交给后台并完成注册，并可以通过该方法修改。**

**（2）Get:可以在一定时间后得到属于自己的课表，课表中每节课应该清晰地指明课程起止时间，时长，教室，教师以及上课人数。**

\*（3）（\*根据项目完成情况可选的附加功能）View:更改课表视图；

Analyze:录入成绩和偏好帮助分析安排选课；

Teacher:得到上课教师的信息和联系方式……

教师

**（1）Upload:可以通过项目APP将自己的姓名，教职工号，性别，教授科目提交给后台并完成注册，并可以通过该方法修改。**

**（2）Get:可以在一定时间后得到属于自己的课程安排，课程安排中应该清晰地指明课程起止时间，时长，教室，上课人数。**

\* （3）（\*根据项目完成情况可选的附加功能）View:更改课程安排视图；

Analyze:录入教龄和班级偏好帮助优先安排选课；

Student:可以在另一个视图连接中得到自己所教的指定班级所有的学生信息……

管理员

**（1）Upload:可以在APP中通过输入或文件操作更改和添加后台数据。鉴于输入可能带来矛盾，我们只给予管理员输入分项数据的权限。文件的格式必须有严谨的前提下尽量宽松的要求。并且保证数据元素唯一（static），防止教师以及学生数据与管理者数据产生的冲突。**

**（2）Get:可以在一定时间后得到全局的课程排布。排布中应该显示的(Main Activity)第一视图是行政班与教学班的人数交叉情况。在每一个交叉项中应该有一个可以展示详细信息的接口(响应函数)，通过(调用Details)点击Button可以看到该课程的上课教师，上课教室，以及所有的上课时间这三类信息。Main Activity的框架大概由下表描绘。**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 教学班：  人数  科目  行政班（CL） | **一** | **二** | **三** | **四** | **五** | **六** |
| **物理（PH）** | **PH1：(sum of down 3)67**  For Details | | | **PH2：6**  For Details | | |
| **分项人数** | **1** | **15** | **51** | **1** | **2** | **3** |
| **化学(CH)** | **CH1：42**  For Details | | | **CH2：42**  For Details | | |
| **分项人数** | **20** | **20** | **2** | **14** | **14** | **14** |
| **历史（HI）** | **HI1：11** | | | **HI2：26** | | |
| **……** | | | | | | |
| **分项人数** | **…** | | | | | |
| **数学** | **55**  For Details | **54**  For Details | **56**  For Details | **…** | | |
| **语文** | **55**  For Details | **54**  For Details | **56**  For Details |
| **表一-1-b** | | | | | | |

\*（3）（\*根据项目完成情况可选的附加功能）View:更改排布视图及其优先级；

Adjust(应该被实现的重要方法)：我们可以观察到，横向上，上表中PH1的人数以及PH2的人数产生了巨大的差距（虽说出现的概率不大），为了避免这种安排的真实应用，一种算法应该被实现来给管理员提供调整人数至均衡的手段

(下面是一种可能的示例:

从学生统计数据，比如张三下午想要先上物理再上历史，再上化学，那么这就是张三下午的信息，然后把所有人的信息统计起来，比如下午第一节有124人选物理，124人选化学，156人选生物，89人选历史，56人选政治，66人选地理，那么我们根据老师人数与学生人数**综合**考虑分班情况，比如物理分两个班，化学分两个班，生物两个班，政史地各一个班，然后统计第二节参加各科的人数，然后统计第三节参加各科的人数，根据学生需求来分班。即在下午三节六科在第一节第二节第三节都同时开设，根据选课的要求设置班与老师)

Student/Teachers:可以看到全校所有老师或者学生的信息

Analyze:数据很贵的，浪费就是犯罪……

1. 需求

如同1.a中介绍的，新高考选课模式在浙江和上海施行一年（2018）后，结果确实被证明有较大的可变性。下列统计数据列举了的上海地区选课情况（浙江由于是7选3模式，数据不具有普遍性，便没有列举）。

归因于各科人数的不平衡，要想较为科学的，平均的安排教务处的各个教师和学生是较为困难的，人力的数据安排已经不能满足排课的需要，因此一个可以进行自主课表安排和推荐的系统是必要的。

3.可用的资源

为了更好的开发这款应用，笔者查阅了前人所做的项目，在这里仅列举其中的两个，希望得到一些启发。

a.（GitHub download URL: <https://github.com/MYWProgram/Choose-Lesson-System.git>）

这是一个典型的学生管理系统（Java, Web），而且相对来说会比较小。

public Student(int sNo, String sname,String sclass,String ssex,float scredit) {

SNo = sNo;

Sname = sname;

Sclass = sclass;

Ssex = ssex;

Scredit = scredit;

}

以上是从文档中截取的构造函数的代码。我们发现，大部分的学生系统的结构大同小异，但是针对课程和班级系统方面的结构却难以找到较为简单明了的参考资料。这将是我们编程的重点。

|  |
| --- |
|  |

b. (GitHub download URL: <https://github.com/mainxml/CourseTable.git>)

这是一个典型的课表安排系统(Java,(Android Studio)Application)，而且相对来说会比较小。以下展示了MainActivity中的代码框架，并省略了实现，如果可能，我们会在项目框架的基础上连接云数据库，但考虑到时间等的因素，做成单机版也未尝不可。

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

…//onCreate等（启动），并从数据库初始数据

loadData();

}

private void loadData() {//从数据库加载数据 }

private void saveData(Course course) { //保存数据到数据库 }

private void createLeftView(Course course) { //创建"第几节数"视图}

private void createItemCourseView(final Course course) {//创建单个课程视图}

@Override

public boolean onCreateOptionsMenu(Menu menu) {}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {}

}

1. 关于其他

诚然，在高考改革一年后，不少公司推出了一些智能选课的项目，但是遗憾的是，诸如“基于蚁群算法的课表智能分析实现”等等的内容实在对于我们这样的短期项目没有什么参考价值（二.1）。

二.设计主要原则：

1. 结构性

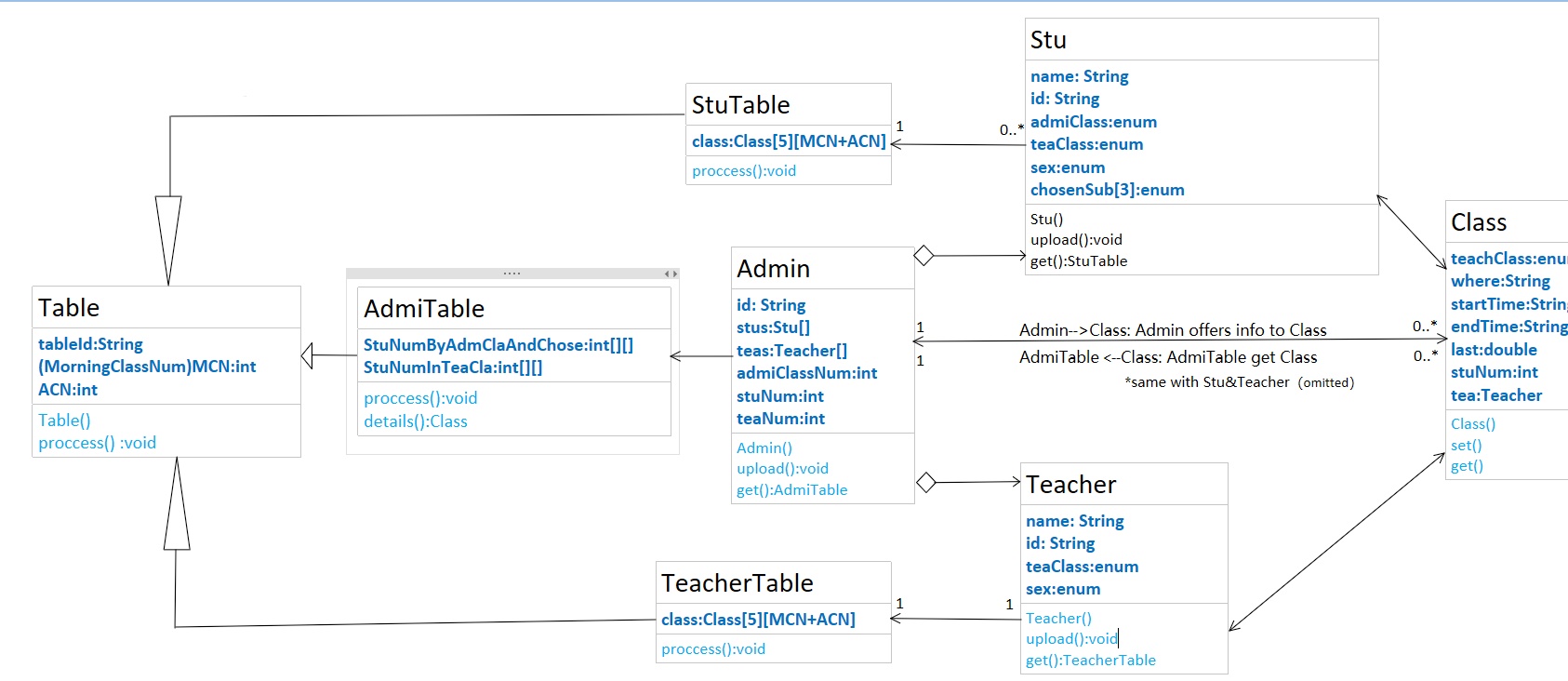
需要明确的是，育星项目组成员目前没有实战经验。所以为了能让此项目可以有效的在以后发挥更大的用处，相比于算法本身，我们可能更加注重框架。这里所说结构性的目的是更好的重用性，可拓展性与可移植性甚至可改写性。考虑到市场上有太多成熟的产品，强调开闭原则不是本项目的特征。关于方法本身，可能是不精妙的。基于以上目的，我们选择使用Java来作为开发使用的语言，少使用继承而多使用接口和组合以减少波动影响。并计划将成果应用较为简单的 Android SDK打包发布。平台的详细信息会在之后提及。

1. 可控性

下方的UML图只实现了基本的方法，看起来可能略显单薄。但是这并不意味着需要做的就只有这些。为了能够在较短的时间内完成较有成效的工作。本次开发的原则应该是自下而上的。通过不断进行细节的填充来明确框架的工程设计，随时可以见到一些成果。

三.应用程序框架

1. UML

本应用程序考虑以下主要的类关系实现：

|  |
| --- |
| 图三-1 |

1. UML阐释

各个类应该至少通过展示的可能的基本方法实现一.1.b中功能，下面

a.首先用语言对类图做一些简单阐释（从左至右）

\*所有的类可以分为两种，客户端类（Class,Stu,Teacher,Admin 处于

图的右半部分）和处理端类（Table,AdmiTable,TeacherTable,StuTable处于图的左半部分）。客户端类获得处理端类的结果，处理端类获得客户端类的细节。

\*处理端：三种课表（学生表 管理员表 老师表）继承自同一个课表（虚）基类，其中的proccess方法原则上必须被重载来填写处理端类的内容（**核心代码，从客户端类获取适当信息生成课表并且更新Admin中的宏观信息防止发生冲突）**

\*客户端：每一种客户（学生，老师，管理员）发出请求（upload）之后，对应的处理端通过调用process方法生成处理端对象返回给客户端的接口(get),并由该接口将对应对象传递给前台。

b.之后对一些可能不是很好理解的方法名称进行解释。

\*admiClass:学生所在的行政班

\*teaClass:学生所在的教学班

\*tableId:课表ID应该可以阐明课表的应用对象和视图类型

\*AdmiTable:请参考表一-1-b理解参数

3.设计模式

采用以上简单的设计方案，代码的封装性并不好。考虑到方法的相似性和客户端命令的多样性，项目在后期实现封装时宏观上拟采用**命令设计模式**，并通过面向接口的编程来实现解耦。因此我们在实施过程中可能会对类图进一些小的重构，这是自下而上的编程中所应允许的。

四.实现：

1.时间线

需要注意的是，这是本学期项目，在繁重的课业下进行项目，时间将比较紧张。 以下是一个参考的完成时间的梳理：

———5.6完成所有的8个类的内部实现

————5.31 完善完成类的框架

————6.15完成UI编程设计与测试

————6.20完成打包发布

2.平台与其他细节

本次开发的开发者所使用的软件版本信息（Windows 10 Platform）（测试平台不在本草案中描述）。

JDK1.8.0;

Android SDK 9.0;

五.引用：

1.<上海市教育考试院> <http://www.shmec.gov.cn/web/jyzt/zsdw/jyzt_show.php?id=16>

2.<开源项目平台GitHub：具体项目见一.3><https://github.com/explore>

3.<百度百科：选课走班制>

<https://baike.baidu.com/item/%E9%80%89%E8%AF%BE%E8%B5%B0%E7%8F%AD/7060601>