课程表助手

2019

Contents

1 项目简介

由于高职院校普遍存在教室资源和教师数量不足的问题，导致教师的课程安排比较杂乱，没有规律，学生也经常因此而忘记去上课，由此经常造成教学事故的发生。为有效避免教学事故的发生，我们团队预备设计一个基于 Android 平台的课表提醒实用 APP 程序。

未来的时代必定是个更加信息化的时代, 手机正成为我们生活必不可少的部分, 针对日常快节奏的生活方式, 许多手机应用软件应运而生, 潜移默化的改变了人们的生活, 同时, 网上交易也日渐成为一种常见的交易方式。利用手机解决各种学生活上的烦恼, 已成为一种社会潮流和风尚。我们根据实际生活遇到的问题, 并提出了这样的想法。我们这款产品的主要对象是大学生, 大学生普遍都有网上搜索和购物的经历, 因此他们对这款软件不会觉得陌生, 而且更加熟悉它的操作流程, 而且这款软件是以解决学生迫切的学习生活问题为出发点的。我们的核心竞争力是立足于学生, 服务于学生, 更容易取得学生的共鸣

2 功能设想

这个 APP 程序计划包含课程表导入模块、定时提醒模块。程序将教师课表导入后，自动提取课表的上课日期、上课时间、上课地点和任课课程，在上课前 20 分钟给任课教师的手机发送上课提醒信息，提醒内容包括上课时间、地点和课程名称等，实现了对教师的课前即时提醒。这个程序还考虑到了课程表中课程安排的某些特殊性，如开学日期设置等，可以满足任课教师和学生的特定需求。

3 功能实现

3.1 技术选型

由于项目目的为学习与应用，故采用了跨平台框架flutter，兼顾 iOS 平台，而非单纯的 *java*、*kotlin*、*objective-c*或 *swift*。

3.2 开学日期设置

暂未实现

3.3 课程导入

暂未实现

3.4 课程创建

点击 + 图标进入 添加课程 界面新建课程。支持修改课程名称、教室、教师、星期、周数和上课时间段。对于时间段可能交叉的课程，时间段相同的可重叠，否则不能添加。

3.5 课程修改

点击已经存在的课程，根据当前时间段节数:

- **只有一节。直接进入 修改课程 界面。**
- **有多节。显示当前课程列表，选中相应课程后进入 修改课程 界面。**

3.6 定时提醒

3.6.1 时间设置

点击课程列表左列数字，可设置相应课时的铃声提醒时间。

3.6.2 提醒

到设置时间出现提醒界面 (时间、地点和课程名称等) 并响铃

3.6.3 * 铃声设置

进入设置界面可设置提醒铃声

3.7 周数选择

点击主界面上方周数后，可设置当前所在周数

3.8 * 主题设置

进入设置界面可设置界面主题

4 界面

4.1 主界面

月份与星期在同一行。考虑到由于时间段问题，课程会有跨行行为，为便于实现，课程显示部分均采用列布局。

4.2 课程创建

输入和选择项都置于ExpansionTile，用于选择性展开，节省空间。取消按钮使用浅色扁平类型，确认按钮使用突出类型，以符合正常使用中的用户视觉交互。

4.3 课程修改

复用课程创建界面，删除按钮改为红色，起警示作用。

5 过程中的问题与解决方案

5.1 已解决的问题

1. 布局溢出[1]。布局时，子组件可能会由于大小问题溢出父组件/设备屏幕。这种情况在子组件有固定宽/高时最明显。

   方案分析：一般在溢出的组件外包装一层 Expanded[2] 或 Container[3] 。

2. 课程交叉检测。添加课程时，可能与已有课程时间段相交。

   方案分析：需检测课程起始时间段与所跨行数。

Figure 1: 主界面

Figure 2: 设置

Figure 3: 添加课程

Figure 4: 修改课程

```
...
// 返回是否相交且未重叠
return list.any((course) {
  // 检查是否相交
  if ((start + step > course.start) &&
      (start < course.start + course.step)) {
    // 检查是否重叠
    if ((start == course.start) && (step == course.step)) {
      return false;
    } else {
      return true;
    }
  }
```

```
      return false;
    });
```

3. **课程重叠。**

   方案分析：将重叠课程置于Stack[4] 组件直接重叠，同时下方显示当前组的课程数

4. **课程布局。因存在无课程的空白空间，不能在一列直接放置课程。**

   方案分析：非绝对定位方案：在课程前添加足够高的空白块，撑起布局。

```
// 课程列表填充空白块
List getFilledCourses({
  // 课程列表
  @required List<CourseModel> courses,
  ...
}) {
  if (courses.length == 0) return [];
  // 最终返回的列表
  List ret = [];
  int prevStart = 0;
  int prevStep = 1;
  ...
  courses.forEach((course) {
    // 判断两个课程间是否有空白
    if (course.start > (prevStart + prevStep)) {
      // 两个课程间有空白则填充空白块
      ret.add(EmptyModel(
        minHeight: minHeight,
        weekday: weekday,
        start: prevStart + prevStep,
        step: course.start - (prevStart + prevStep),
      ));
      prevStart = prevStart + prevStep - (prevStart + prevStep);
      prevStep = course.start + course.step;
    } else {
      prevStart = course.start;
      prevStep = course.step;
    }
    ret.add(course);
  });
  // 如果最后有空白，也填充空白块
  if (prevStart + prevStep - 1 < getRowCount()) {
    ret.add(EmptyModel(
      minHeight: minHeight,
      weekday: weekday,
```

```
            start: prevStart + prevStep,
            step: getRowCount() - (prevStart + prevStep) + 1,
          ));
        }
        return ret;
      }
```

5.2 未解决的问题

1. **不能导入课程。**

   可能方案分析：无。

2. **不能设置开学日期。**

   可能方案分析：设置第一周，以第一周为准依次累加。

3. **周数不能随时间变化。**

   可能方案分析：获取开学周，依次累加。

4. **点击课程列表中某课程，修改后保存返回，列表不更新。**

   可能方案分析：无。

5. **一切取消设置时间的操作（点击 *cancel* 或空白处）都会使当前已设置时间清除。**

   可能方案分析：由于取消设置时程序会抛出异常，在捕获异常时根据抛出的异常类型判断 *cancel* / *dismiss*。

6. **定时提醒在熄屏后失效。**

   可能方案分析：当前使用 *dart:async*[5] 中的 *Future.delayed()*，设备熄屏后可能进入低功耗模式，此方法不可用来唤醒。应尝试使用 *dart:isolate*[6]、直接调用 native 方法[7]。

7. **铃声不能设置自定义。**

   可能方案分析：使用 *path_provider* 获取路径，*dart:io* 读取文件。

References

[1] overflow: https://stackoverflow.com/questions/50250789/expanded-widgets-must-be-placed-inside-flex-widgets

[2] Expanded: https://api.flutter.dev/flutter/widgets/Expanded-class.html

[3] Container: https://api.flutter.dev/flutter/widgets/Container-class.html

[4] Stack: https://api.flutter.dev/flutter/widgets/Stack-class.html

[5] async: https://api.flutter.dev/flutter/dart-async/dart-async-library.html

[6] isolate: https://docs.flutter.io/flutter/dart-isolate/dart-isolate-library.html

[7] PlatformChannels: https://flutter.dev/docs/development/platform-integration/platform-channels