基于跨平台Flutter的校园交友平台设计与实现

作者姓名：肖骏 专业班级：软件工程4班 指导老师：邓飞

摘 要

近几年，随着物联网时代到来、移动互联网高歌猛进，日新月异，在很多业务场景中，传统的纯原生开发已经不能满足日益增长的业务需求。主要表现在：动态化内容需求增大；当需求发生变化时，纯原生应用需要通过版本升级来更新内容，但应用上架、审核是需要周期的，这对高速变化的互联网时代来说是很难接受的，所以，对应用动态化(不发版也可以更新应用内容)的需求就变的迫在眉睫。业务需求变化快，开发成本变大；由于原生开发一般都要维护Android、iOS两个开发团队，版本迭代时，无论人力成本，还是测试成本都会变大。原生应用程序是指某一个移动平台（比如iOS或安卓）所特有的应用，使用相应平台支持的开发工具和语言，并直接调用系统提供的SDK API。比如Android原生应用就是指使用Java或Kotlin语言直接调用Android SDK开发的应用程序；而iOS原生应用就是指通过Objective-C或Swift语言直接调用iOS SDK开发的应用程序。原生开发有以下主要优势：可访问平台全部功能（GPS、摄像头）；速度快、性能高、可以实现复杂动画及绘制，整体用户体验好；主要缺点：平台特定，开发成本高；不同平台必须维护不同代码，人力成本随之变大；内容固定，动态化弱，大多数情况下，有新功能更新时只能发版。目前常用的跨平台技术有：（1）H5+原生（Cordova、Ionic、微信小程序）（2）JavaScript开发+原生渲染 （React Native、Weex、快应用）3）自绘UI+原生(QT for mobile、Flutter) 。（杜文，2020）

关键词: Flutter,Android,SpringBoot,跨平台

**Development of campus dating platform**

**based on Flutter**

**Abstract**: In recent years, with the advent of the Internet of Things and the rapid advancement of the mobile Internet, with each passing day, in many business scenarios, traditional pure native development has been unable to meet the growing business needs. Mainly reflected in: the demand for dynamic content increases; when the demand changes, pure native applications need to update the content through version upgrades, but the application shelves and audits require a cycle, which is difficult for the rapidly changing Internet era Accepted, therefore, the need for application dynamics (the application content can be updated without publishing a version) becomes imminent. Business requirements change rapidly, and development costs become larger. Since native development generally maintains two development teams, Android and iOS, when the version is iterated, regardless of labor costs and testing costs, it will become larger. A native application refers to an application unique to a mobile platform (such as iOS or Android), uses the development tools and languages ​​supported by the corresponding platform, and directly calls the SDK API provided by the system. For example, Android native application refers to an application developed by directly calling Android SDK using Java or Kotlin language; iOS native application refers to an application developed by directly calling iOS SDK through Objective-C or Swift language. Native development has the following main advantages: access to all functions of the platform (GPS, camera); fast speed, high performance, complex animation and drawing, and good overall user experience; main disadvantages: platform specific, high development cost; different platforms must be maintained With different codes, the labor cost will increase accordingly; the content is fixed and the dynamics are weak. In most cases, the version can only be issued when there are new feature updates. Currently commonly used cross-platform technologies are: (1) H5 + native (Cordova, Ionic, WeChat applet) (2) JavaScript development + native rendering (React Native, Weex, fast application) 3) self-painting UI + native (QT for mobile, Flutter)

**Key words**: Flutter,Android,SpringBoot,Cross-platform

目 录

[第1章 前 言 1](#_Toc39520180)

[1.1 软件开发背景及目的 1](#_Toc39520181)

[1.2 发展现状 1](#_Toc39520182)

[1.3 本文的主要内容 2](#_Toc39520183)

[1.3.1 需求整理与软件设计 2](#_Toc39520184)

[1.3.2 系统开发与开发成果 3](#_Toc39520185)

# 第1章 前 言

## 1.1 软件开发背景及目的

近年来，随着移动互联网的迅速发展，手机日渐深刻的在改变着人们的生活方式。越来越多的app渗入我们的日常生活，包括网约打车，点外卖，线上购买商品，远程医疗等等。而这一切都需要有专门的公司负责相关app的开发，但是作为主流的手机操作系统，Android和IOS,他们都有自己的开发平台和开发语言，所以在app开发过程中，许多公司使用的解决方案是分别招聘Android和IOS的开发人员，分别负责相关的开发。这无疑给开发过程中增添了许多不必要的麻烦，包括兼容性问题，适配问题。为了使开发者更加专注app的开发，而不是跨平台的学习，业界提出了跨平台的方案，包括Facebook 于2015年四月开源的跨平台移动应用开发框架ReactNative，阿里巴巴公司与2016年6月开源的一种用于构建移动跨平台的 UI 框架Weex, （陈阿票，2016）此外包括Google 开源的 UI 工具包Flutter，帮助开发者通过一套代码库高效构建多平台精美应用，支持移动、Web、桌面和嵌入式平台。（百度，2020）可以说，未来的主流开发应该是不分平台的，这也会大大减少各个公司的开发成本。

本题主要使用了Flutter，这种技术最早于2015年提出，Flutter的第一个版本被称为“Sky”，运行在Android操作系统上。它是在2015年Dart开发者峰会 上亮相的，其目的是能够以每秒120帧的速度持续渲染。Flutter包括许多特性，例如，快速开发，绚丽UI,响应式，访问原生功能。Flutter 组件采用现代响应式框架构建，这是从React中获得的灵感，中心思想是用组件(widget)构建你的UI。 组件描述了在给定其当前配置和状态时他们显示的样子。当组件状态改变，组件会重构它的描述(description)，Flutter 会对比之前的描述， 以确定底层渲染树从当前状态转换到下一个状态所需要的最小更改。（百度，2018）

## 1.2 发展现状

成理校园交友平台选择了Flutter， Flutter是当今热门的跨平台开发框架，Flutter 构建应用时，可以让开发者对 UI 实现像素级的控制。意味着原本的设计意图可以被完美地执行出来，从而将品牌个性忠实地传达给用户。Flutter 的 UI 渲染性能很好。在生产环境下，Flutter 将代码编译成机器码执行，并充分利用 GPU 的图形加速能力，因此使用 Flutter 开发的移动应用即使在低配手机上也能实现每秒 60 帧的 UI 渲染速度。Flutter 同时做到 “美观” 和 “快速” 的原因，在于其架构本身。Flutter 引擎使用 C++ 编写，包括高效的 Skia 2D 渲染引擎，Dart 运行时和文本渲染库。这个引擎使得 Flutter 框架可以自由、灵活、高效地绘制 UI 组件。而应用开发者则可以用 Flutter 框架来轻松实现各种设计语言和动画效果。Flutter 是开放的，它是一个完全开源的项目。全球的开发者都可以免费使用和拓展 Flutter 的源代码，并为 Flutter 的生态和文档作贡献。 我们已经看到许多中国开发者活跃在社区中，并为 Flutter 做出了坚实的贡献。 Flutter 是开放的，它是一个完全开源的项目。全球的开发者都可以免费使用和拓展 Flutter 的源代码，并为 Flutter 的生态和文档作贡献。 我们已经看到许多中国开发者活跃在社区中，并为 Flutter 做出了坚实的贡献。(董韬,2019)

## 1.3 本文的主要内容

本次毕业设计的基本要求是利用Flutter设计出一款跨平台的校园交友平台app。实现利用Flutter进行快速的开发以及构建富有表现力和灵活的UI，达到和iOS和Android上原生应用一样的性能。实现后端Springboot+Jpa+MySQL的数据存储，实现用户登录，异常处理，密码加密等功能。

### 1.3.1 需求整理与软件设计

通过对“学生社交需求困难”这个问题进行分析，结合用户的期望和实际生活情况以及本APP的宗旨，整理需求如下：

（1）用户可以通过本APP进行用户注册用户名，密码，学院等信息；

（2）用户可以通过本APP进行用户验证授权登录；

（3）在本APP中用户可以发表三种类型信息(表白，吐槽，寻物)；

（4）在本APP中用户可以和其他用户发表的卡片进行点赞，评论；

（5）在本APP中用户可以实现实时查看最新和最热的卡片。

成理表白墙客户端的总体设计风格的就是把最核心功能以最友好的方式展示到应用用户面前，所以应有的操作流程如下：

安装完成后，打开APP，会直接进入用户登录界面，如果用户没有进行注册，则可以跳转到注册页面进行注册，当注册成功后，会直接将注册的账号和密码跳转到登录界面，用户选择登录按钮，用户可以进入主页查看卡片信息。通过TarBarView进行相关标签选择发表卡片信息，通过双击卡片信息可以进行点赞，单击可以进行查看相关卡片信息以及进行评论。

### 1.3.2 系统开发与开发成果

本校园表白墙APP开发主要分为以如下阶段：

（1）APP的设计。按照需求设计相关的原型界面。

（2）Flutter SDK研究和Dart语言的学习。对Flutter SDK的文档和相关常用组件源码进行研究，了解Flutter的底层实现原理和基本掌握Dart语言的相关语法；

（3）APP的逻辑设计，就开始着手APP部分的设计和编码了，首先通过Flutter进行APP的编码，包括UI、交互动画等设计；开始编码APP相关业务逻辑，包括路由管理，状态管理，动画设计。

（4）SpringBoot和MySQL的学习，了解需要用到的数据库范式和整个后端的业务逻辑框架的搭建。；

（5）SpringBoot的搭建，根据Flutter的客户端设计相关的实体模型，搭建好相关的Restful风格的API,并返回正确和错误信息。

（6）后端数据的接收和发送，利用Flutter提供的HTTP库和以及搭建好的Springboot提供的API进行数据的调用。

成理表白墙客户端的功能有：

（1）用户进行填写注册信息；

（2）用户进行填写登录信息；

（3）用户通过填写卡片信息；

成理表白墙服务端的功能有：

（1）用户注册信息保存到数据库；

（2）用户进行验证授权登录；

（3）用户通过删除，更新，查询卡片信息；

（4）实现用户和其他用户卡片信息进行互动；

第2章 开发工具简述

## 2.1 移动端开发工具Android Studio与Flutter的介绍

### 2.1.1开发工具Android Studio的介绍

Android Studio 是谷歌推出的一个Android集成开发工具，基于IntelliJ IDEA. 类似 Eclipse ADT，Android Studio 提供了集成的 Android 开发工具用于开发和调试。Android Studio的架构在IDEA的基础上，Android Studio 提供基于Gradle的构建支持Android 专属的重构和快速修复，提示工具以捕获性能、可用性、版本兼容性等问题，支持ProGuard 和应用签名，基于模板的向导来生成常用的 Android 应用设计和组件，功能强大的布局编辑器，可以让你拖拉 UI 控件并进行效果预览。（百度百科，2017）

### 2.1.2 Flutter的介绍

Flutter是谷歌的移动UI框架，可以快速在iOS和Android上构建高质量的原生用户界面。 Flutter可以与现有的代码一起工作。在全世界，Flutter正在被越来越多的开发者和组织使用，并且Flutter是完全免费、开源的。

Flutter可以做到快速开发毫秒级的热重载，修改后，您的应用界面会立即更新。使用丰富的、完全可定制的widget在几分钟内构建原生界面。富有表现力和灵活的UI，快速发布聚焦于原生体验的功能。分层的架构允许您完全自定义，从而实现难以置信的快速渲染和富有表现力、灵活的设计。Flutter具有和Android以及IOS一样的原生性能，Flutter包含了许多核心的widget，如滚动、导航、图标和字体等。统一的应用开发体验，Flutter拥有丰富的工具和库，可以帮助开发人员轻松地同时在iOS和Android系统中实现您的想法和创意。 如果开发人员没有任何移动端开发体验，Flutter是一种轻松快捷的方式来构建漂亮的移动应用程序。 如果开发人员是一位经验丰富的iOS或Android开发人员，则可以使用Flutter作为视图(View)层， 并可以使用已经用Java / ObjC / Swift完成的部分（Flutter支持混合开发）。（Flutter中文网，2019）

## 2.2 服务端开发工具IDEA与SpringBoot+JPA+MySQL的介绍

### 2.2.1开发工具IntelliJ IDEA的介绍

IntelliJ IDEA在业界被公认为最好的java开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。IDEA的特色功能，智能的选取，在很多时候我们要选取某个方法，或某个循环或想一步一步从一个变量到整个类慢慢扩充着选取，IDEA就提供这种基于语法的选择，在默认设置中Ctrl+W，可以实现选取范围的不断扩充，这种方式在重构的时候尤其显得方便。丰富的导航模式，IDEA提供了丰富的导航查看模式，例如Ctrl+E显示最近打开过的文件，Ctrl+N显示你希望显示的类名查找框（该框同样有智能补充功能，当你输入字母后IDEA将显示所有候选类名）。在最基本的project视图中，你还可以选择多种的视图方式。历史记录功能，不用通过版本管理服务器，单纯的IDEA就可以查看任何工程中文件的历史记录，在版本恢复时你可以很容易的将其恢复。JUnit的完美支持，对重构的优越支持，IDEA是所有IDE中最早支持重构的，其优秀的重构能力一直是其主要卖点之一。编码辅助，Java规范中提倡的toString()、hashCode()、equals()以及所有的get/set方法，你可以不用进行任何的输入就可以实现代码的自动生成，从而把你从无聊的基本方法编码中解放出来。灵活的排版功能，本所有的IDE都有重排版功能，但仅有IDEA的是人性的，因为它支持排版模式的定制，你可以根据不同的项目要求采用不同的排版方式。XML的完美支持，xml全提示支持：所有流行框架的xml文件都支持全提示，谁用谁知道。动态语法检测，任何不符合java规范、自己预定义的规范、累赘都将在页面中加亮显示。(百度百科，2012)

### 2.2.2 开发框架SpringBoot的介绍

Spring框架是Java平台上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为EJB（EnterpriseJavaBeans）模型的补充，甚至是替补。Spring框架为开发提供了一系列的解决方案，比如利用控制反转的核心特性，并通过依赖注入实现控制反转来实现管理对象生命周期容器化，利用面向切面编程进行声明式的事务管理，整合多种持久化技术管理数据访问，提供大量优秀的Web框架方便开发等等。Spring框架具有控制反转（IOC）特性，IOC旨在方便项目维护和测试，它提供了一种通过Java的反射机制对Java对象进行统一的配置和管理的方法。Spring框架利用容器管理对象的生命周期，容器可以通过扫描XML文件或类上特定Java注解来配置对象，开发者可以通过依赖查找或依赖注入来获得对象。Spring框架具有面向切面编程（AOP）框架，SpringAOP框架基于代理模式，同时运行时可配置；AOP框架主要针对模块之间的交叉关注点进行模块化。Spring框架的AOP框架仅提供基本的AOP特性，虽无法与AspectJ框架相比，但通过与AspectJ的集成，也可以满足基本需求。Spring框架下的事务管理、远程访问等功能均可以通过使用SpringAOP技术实现。Spring的事务管理框架为Java平台带来了一种抽象机制，使本地和全局事务以及嵌套事务能够与保存点一起工作，并且几乎可以在Java平台的任何环境中工作。Spring集成多种事务模板，系统可以通过事务模板、XML或Java注解进行事务配置，并且事务框架集成了消息传递和缓存等功能。Spring的数据访问框架解决了开发人员在应用程序中使用数据库时遇到的常见困难。它不仅对Java:JDBC、iBATS/MyBATIs、Hibernate、Java数据对象（JDO）、ApacheOJB和ApacheCayne等所有流行的数据访问框架中提供支持，同时还可以与Spring的事务管理一起使用，为数据访问提供了灵活的抽象。Spring框架最初是没有打算构建一个自己的WebMVC框架，其开发人员在开发过程中认为现有的StrutsWeb框架的呈现层和请求处理层之间以及请求处理层和模型之间的分离不够，于是创建了SpringMVC。(百度百科，2016)

### 2.2.3 JPA规范标准的介绍

JPA是Java Persistence API的简称，中文名Java持久层API，是JDK 5.0注解或XML描述对象－关系表的映射关系，并将运行期的实体对象持久化到数据库中。它具有以下优势，标准化JPA 是 JCP 组织发布的 Java EE 标准之一，因此任何声称符合 JPA 标准的框架都遵循同样的架构，提供相同的访问API，这保证了基于JPA开发的企业应用能够经过少量的修改就能够在不同的JPA框架下运行。容器级特性的支持，JPA框架中支持大数据集、事务、并发等容器级事务，这使得 JPA 超越了简单持久化框架的局限，在企业应用发挥更大的作用。

简单方便，JPA的主要目标之一就是提供更加简单的编程模型：在JPA框架下创建实体和创建Java 类一样简单，没有任何的约束和限制，只需要使用 javax.persistence.Entity进行注释，JPA的框架和接口也都非常简单，没有太多特别的规则和设计模式的要求，开发者可以很容易地掌握。JPA基于非侵入式原则设计，因此可以很容易地和其它框架或者容器集成。查询能力，JPA的查询语言是面向对象而非面向数据库的，它以面向对象的自然语法构造查询语句，可以看成是Hibernate HQL的等价物。JPA定义了独特的JPQL（Java Persistence Query Language），JPQL是EJB QL的一种扩展，它是针对实体的一种查询语言，操作对象是实体，而不是关系数据库的表，而且能够支持批量更新和修改、JOIN、GROUP BY、HAVING 等通常只有 SQL 才能够提供的高级查询特性，甚至还能够支持子查询。高级特性，JPA 中能够支持面向对象的高级特性，如类之间的继承、多态和类之间的复杂关系，这样的支持能够让开发者最大限度的使用面向对象的模型设计企业应用，而不需要自行处理这些特性在关系数据库的持久化。(百度百科，2015)

### 2.2.4 MySQL数据库的介绍

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。(百度百科，2015)

## 2.3 Android Studio配置Flutter

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。(百度百科，2015)

（1）使用Git下载Flutter开发包

git clone -b alpha https://github.com/flutter/flutter.git

(2) 配置环境变量

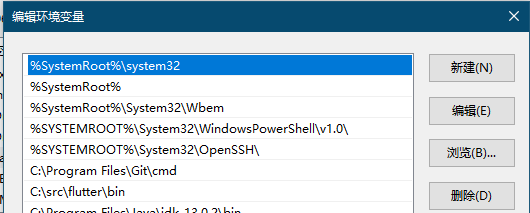


图2-1 配置环境变量

(3) windows命令行PowerShell执行flutter doctor命令

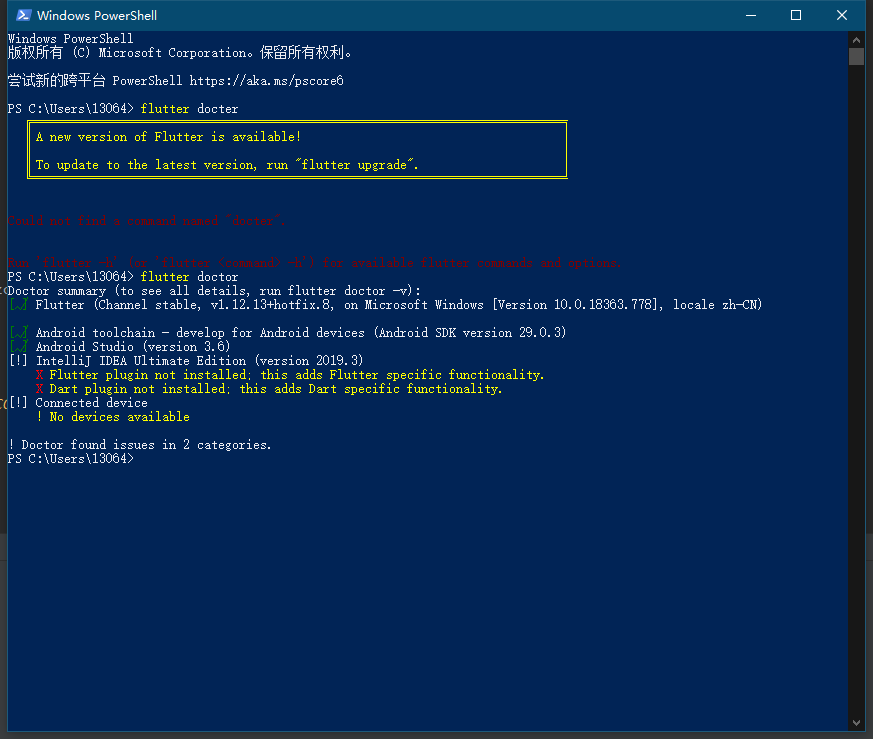


图2-2 运行flutter doctor指令

(4) Android Studio安装Dart和Flutter插件

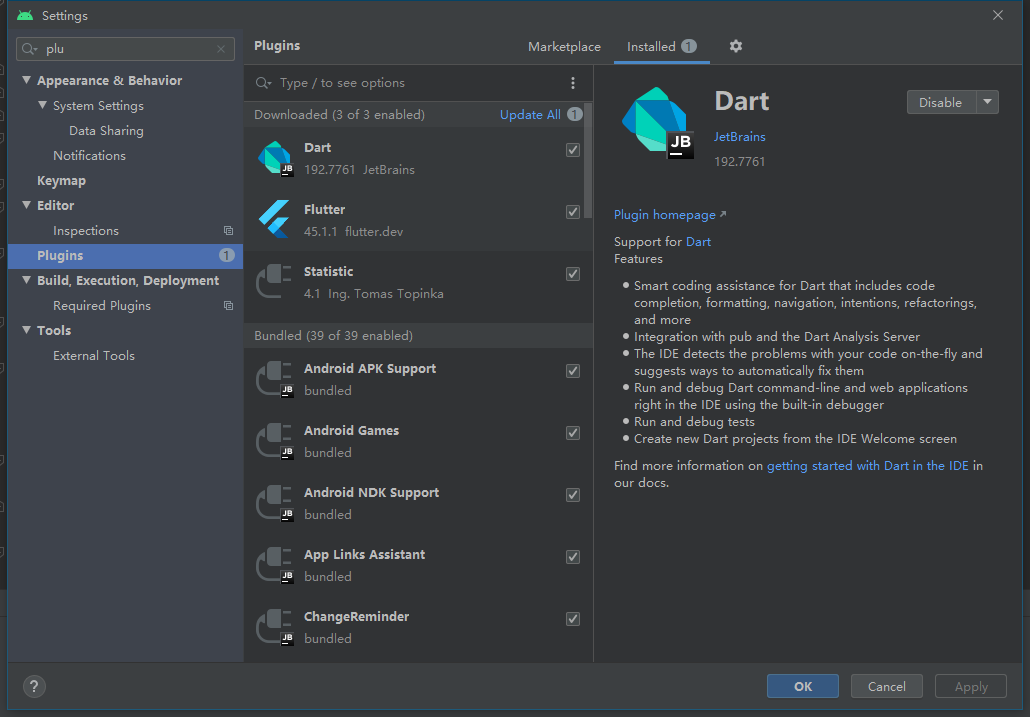


图2-3 安装Dart和Flutter插件

(5) Android Studio创建new Flutter Project

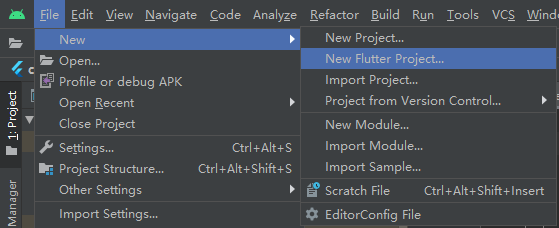


图2-4 创建new Flutter Project

(5) Android Studio打开Flutter Project

## 2.4 IDEA配置SpringBoot+JPA+MySQL

(1) 使用start.spring.io下载一个空项目，不添加相关的依赖。

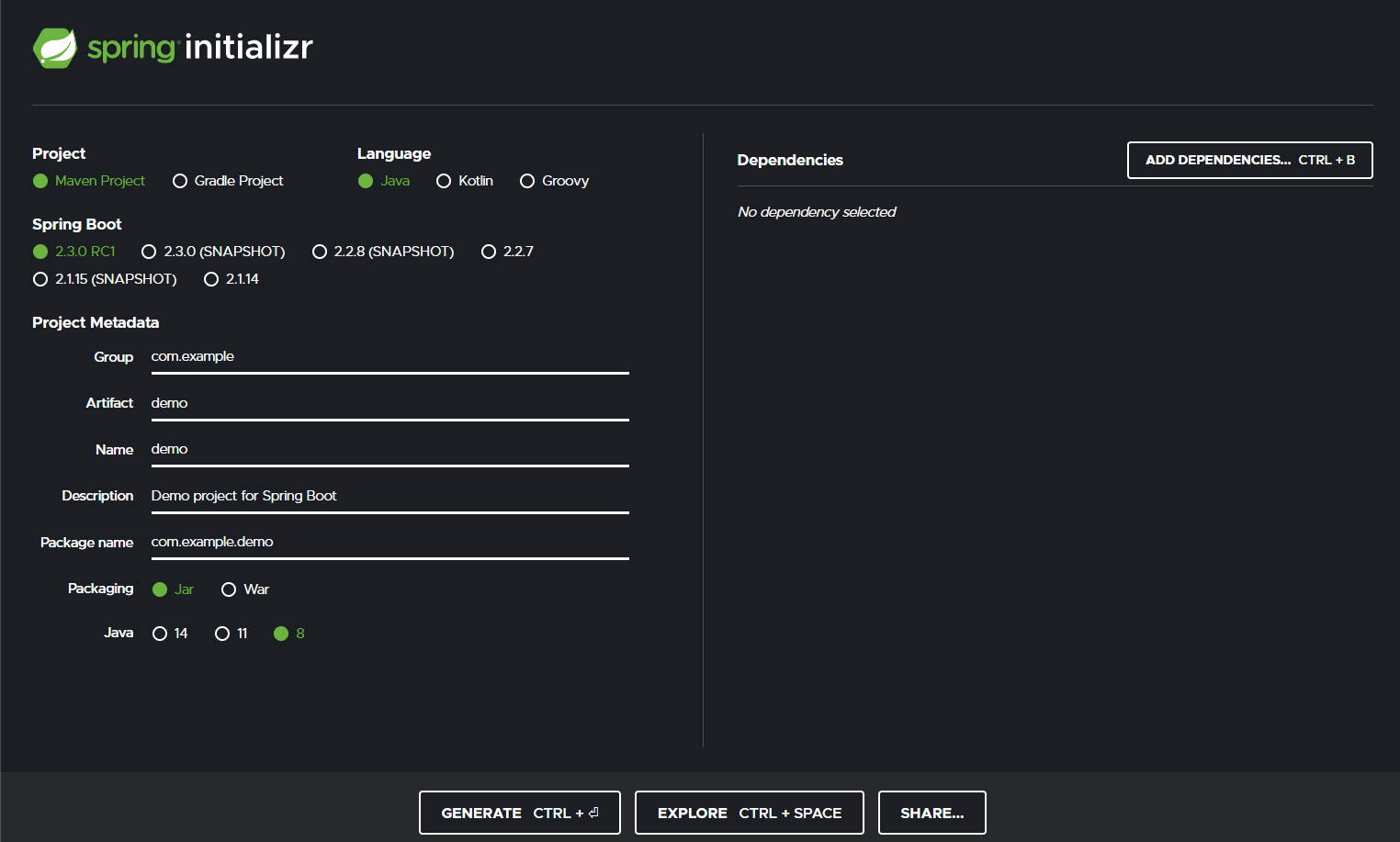


图2-5 创建start.spring.io新项目

(2) 在IDE中打开项目，然后在DemoApplication.java文件src/main/java/com/example/demo夹中找到文件，查看相关的文件目录。src/main/java：主程序入口 Application，可以通过直接运行该类来启动 pring Boot应用。src/main/resources：配置目录，该目录用来存放应用的一些配置信息，比如应用名、服务端口、数据库配置等。由于我们应用了Web模块，因此产生了 static目录与templates目录，前者用于存放静态资源，如图片、CSS、JavaScript等；后者用于存放Web页面的模板文件。src/test：单元测试目录，生成的 ApplicationTests 通过 JUnit4实现，可以直接用运行 Spring Boot应用的测试。application.properties/application.yml 用于存放程序的各种依赖模块的配置信息。

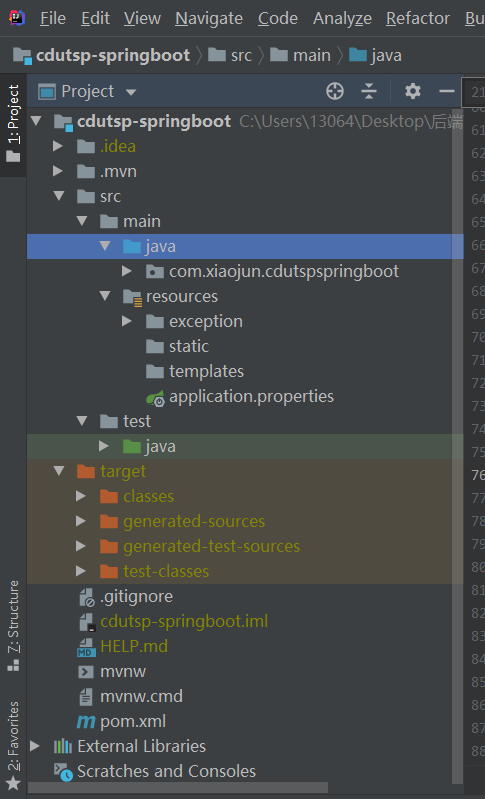


图2-6 创建SpringBoot项目目录

(3)在pom.xml添加相关的依赖，包括JPA,Springboot,MySQL等依赖。

<dependency>

<groupId>com.github.ulisesbocchio</groupId>

<artifactId>jasypt-spring-boot-starter</artifactId>

<version>3.0.2</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-mail</artifactId>

<version>2.2.6.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

图2-8 pom.xml添加依赖

(4) 先将之前的依赖通过Maven点击clean去除掉，再点击install获取新依赖。

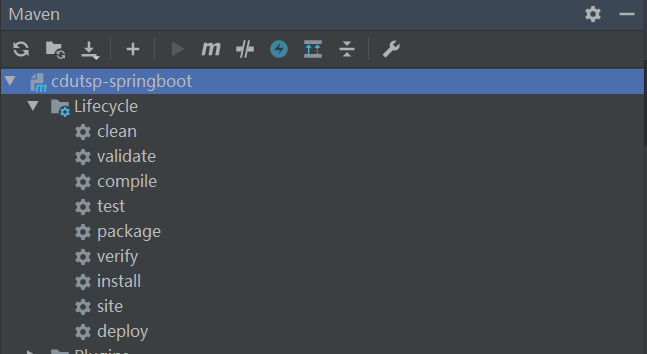


图2-7 Maven获取依赖

## 2.5 Flutter以及SpringBoot的研究

### 2.5.1 Flutter的工作原理

Flutter构建的用户界面含有大量 widget, 为了支撑这些负载，Flutter 使用了次线性算法来布局和构建 widget，这些数据结构使树形结构优化更加高效，并且具有很多常量因子优化。

组合性是 Flutter 最为出众的一个特性。widget 通过组合其他 widget 的方式进行构建，并且这些 widget 自身由更基础的 widget 构建。widget 递归构建的底层是 RenderObjectwidget，它将在渲染树的底部创建子节点。渲染树是一种存储用户界面几何信息的数据结构，该几何信息在布局期间计算并在绘制及命中测试期间使用。

简述下RenderObjectwidget的工作原理。

以最基本的部件Opacity来说，接受一个孩子部件，改变其透明度。

const Opacity({

Key key,

@required this.opacity,

this.alwaysIncludeSemantics = false,

Widget child,

}) : assert(opacity != null && opacity >= 0.0 && opacity <= 1.0),

assert(alwaysIncludeSemantics != null),

super(key: key, child: child);

在Android Studio打开Opacity的源代码，Opacity的父类是SingleChildRenderObjectWidget，再通过Crtl+右键进入SingleChildRenderObjectWidget的源代码，可以看到SingleChildRenderObjectWidget是继承于RenderObjectWidget，而RenderObjectWidget又继承于RenderObjectWidget。

Opacity部件的渲染通过创建和更新RenderObject，如下方法。

@override

RenderOpacity createRenderObject(BuildContext context) => new RenderOpacity(opacity: opacity);

@override

void updateRenderObject(BuildContext context, RenderOpacity renderObject) {

renderObject.opacity = opacity;

}

Opacity部件将子部件设置大小一样，但改变其透明度则是通过RenderOpacity。

RenderOpacity需要实现所有方法（例如执行布局/命中测试/计算大小），并要求其子级执行实际工作。

RenderOpacity({

double opacity = 1.0,

bool alwaysIncludeSemantics = false,

RenderBox child,

}) : assert(opacity != null),

assert(opacity >= 0.0 && opacity <= 1.0),

assert(alwaysIncludeSemantics != null),

\_opacity = opacity,

\_alwaysIncludeSemantics = alwaysIncludeSemantics,

\_alpha = ui.Color.getAlphaFromOpacity(opacity),

super(child);

设置其透明度，通过markNeedsPaint()以及markNeedsLayout()方法告诉Flutter，重新进行绘制。

@override

void paint(PaintingContext context, Offset offset) {

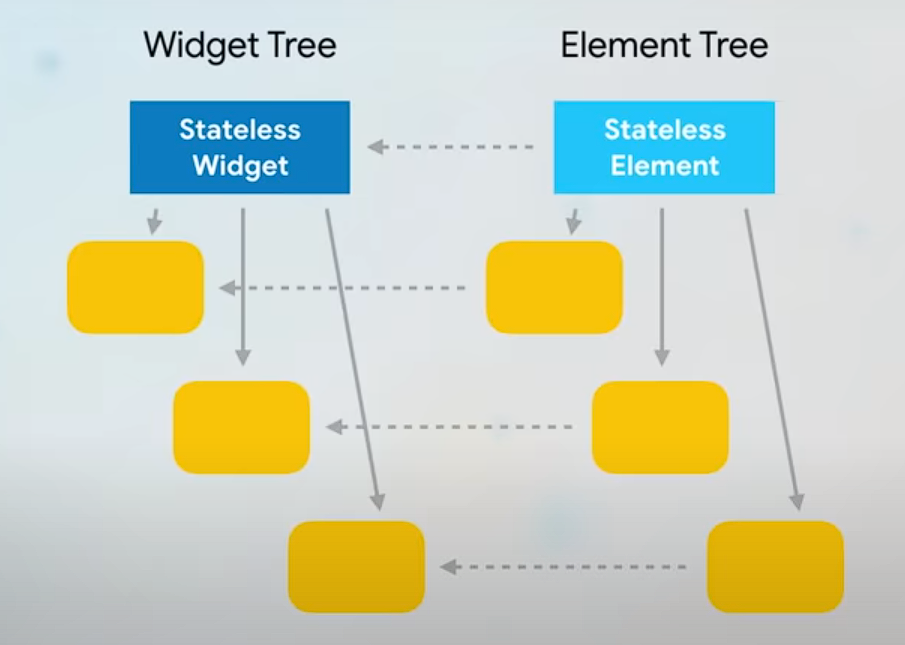
context.pushOpacity(offset, \_alpha, super.paint);

}

完成整个部件的渲染。

Flutter 使用类似于布局的次线性算法来构建widget。widget 构建完成后，它们将被保留了用户页面逻辑结构的element 树保存。 Element树是非常有必要的，这是因为 widget 自身是不可变的，这意味着（其他情况除外），它们无法记住父（或子）节点与其他 widget 的关系。 Element 还保存了与Stateful widget 相关联的state对象。

\



Widget可以重复使用多次，但是在树中有唯一的Elements。

不同于传统做法，Flutter 没有使用树差异比较算法。相反，框架通过使用 O(N) 算法独立地检查每个 element 的子节点来决定是否重用该 element。子列表协调算法针对以下情况进行了优化：

1. 旧的子列表为空。
2. 两个列表完全相同。
3. 在列表的某个位置插入或删除一个或多个 widget。
4. 如果新旧列表都包含相同 key 5 的 widget，那么这两个 widget 就会被认为是相同的。

通常的做法是从新旧子列表的头部和尾部开始对每一个 widget 的运行时类型和 key 进行匹配，这样就可能找到在两个列表中间所有不匹配子节点的（非空）范围。然后框架将旧子列表中该范围内的子项根据它的 key 放入一个哈希表中。接下来，框架将会遍历新的子列表以寻找该范围内能够匹配哈希表中的 key的子项。无法匹配的子项将会被丢弃并从头开始重建，匹配到的子项则使用它们新的 widget 进行重建。

参考文献

Apache.Spring Quickstart Guide[EB/OL].https://spring.io/quickstart,2020-2.

（美）Abraham Silberschatz ,（美）Henry F.Korth ,（美）S.Sudarshan.数据库系统概念[M].机械工业出版社:北京,2012:20-.

baeldung.Learn Spring Boot[EB/OL].https://www.baeldung.com/spring-boot,2019-6.

baeldung.Error Handling for REST with [EB/OL].https://www.baeldung.com/exception-handling-for-rest-with-spring,2019-3.

Flutter开发团队.Flutter 开发文档[EB/OL].https://flutter.cn/docs,2019-12-11.

Flutter.flutter project[EB/OL].https://github.com/flutter,2020-3.

Iam-pawan.FlutterExampleApps[EB/OL].https://github.com/iampawan/FlutterExampleApps,2019-3.

ityouknow.Spring Boot 精选课程[EB/OL].https://github.com/ityouknow/spring-boot-leaning,2019-3.

Salisu Wada.Building a Spring Boot REST API — Part 1: Setting Up the Application[EB/OL].https://medium.com/better-programming/building-a-spring-boot-rest-api-a-php-developers-view-part-i-6add2e794646,2017-9.

Salisu Wada.“Bottom Overflowed” error caused by the keyboard[EB/OL].https://medium.com/zipper-studios/the-keyboard-causes-the-bottom-overflowed-error-5da150a1c660,2019-7.

百度.Flutter[EB/OL].https://baike.baidu.com/item/Flutter/22498985?fr=aladdin,2018-4-17.

陈阿票.微信小程序、Facebook-RN、阿里-Weex 三者分析比较[EB/OL].https://www.jianshu.com/p/fb28b8e14bc5,2016-6-11.

菜鸟教程.MySQL 教程[EB/OL].https://www.runoob.com/mysql/mysql-tutorial.html,2017-6.

董韬.Flutter 最新进展与未来展望[EB/OL].https://www.sohu.com/a/321869129\_670669,2019-6-20.

杜文.Flutter实战[M].机械工业出版社:北京,2020-03-01:40.

郭霖.第一行代码：Android[M].人民邮电出版社:北京,2016:4.

小嵩.Flutter 基础之 Theme 主题（样式篇）[EB/OL].https://blog.csdn.net/qq\_22393017/ article/details/98505307,2019-8.

[日]竹下隆史,[日]村山公保 ,[日]荒井透 ,等.图解TCP/IP（第5版）[M].人民邮电出版社:北京,2013:10-.

百度.Android Studio[EB/OL].https://baike.baidu.com/item/Android%20Studio/4352922?fr=aladdin,2017.

Flutter中文网.Flutter[EB/OL].https://flutterchina.club/,2019.