DAYDAYUP的设计文档

17301165 彭崇真

1. 简绍

DAYDAYUP是一款学习软件，主要是通过选择视频观看视频内容，进行学习。

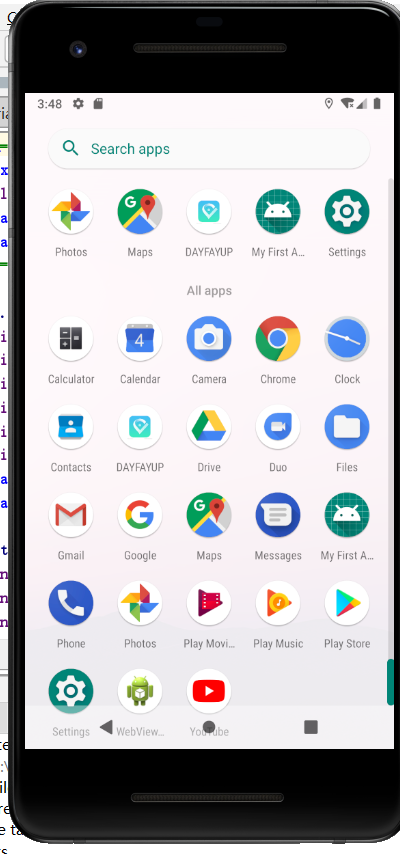
客户端：基于Android所开发的客户端

后端：使用Spring boot搭建的后端

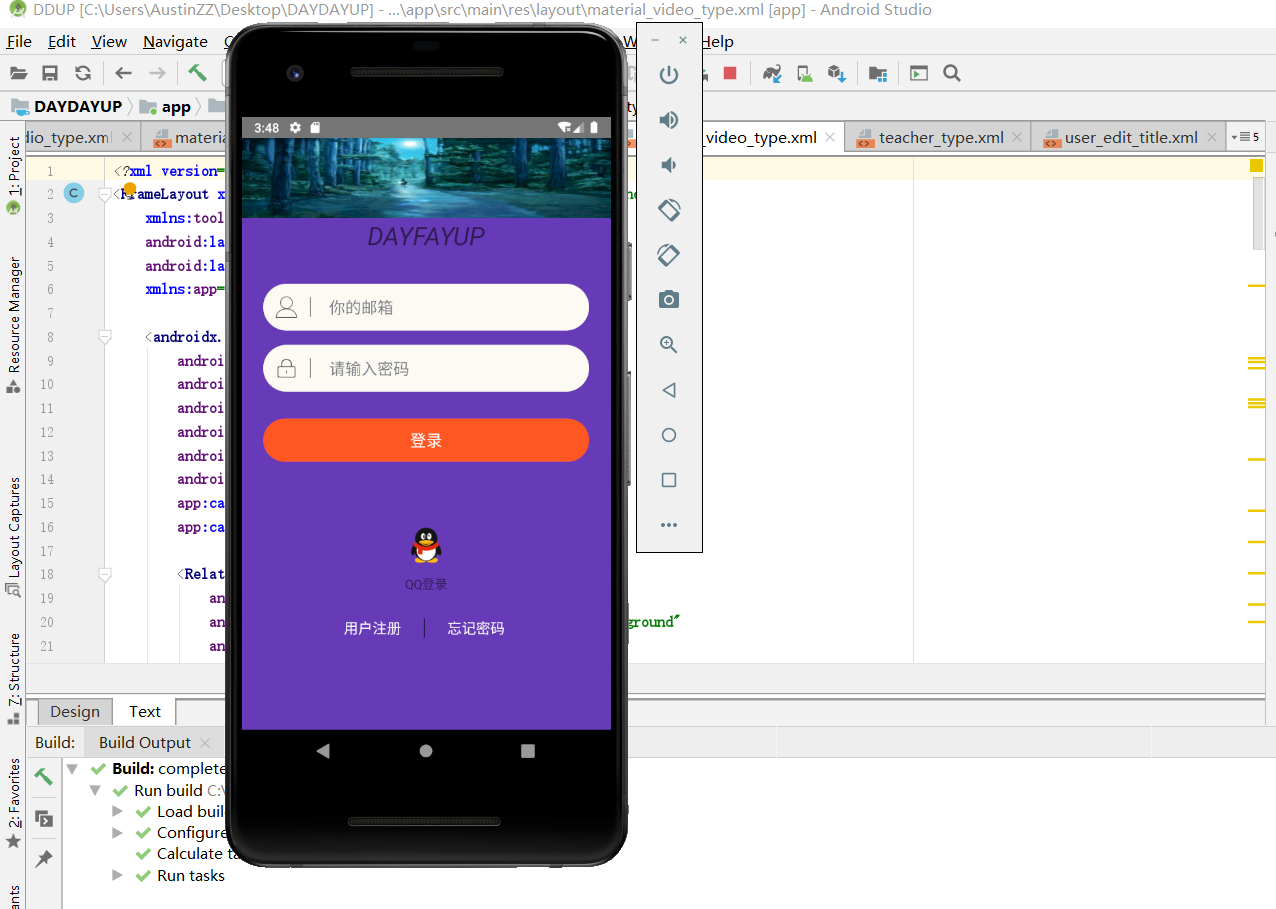
前后端:通过接口来分析传入的JSON

1. 图标

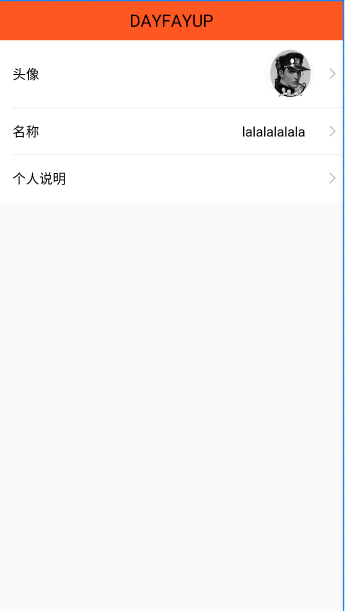


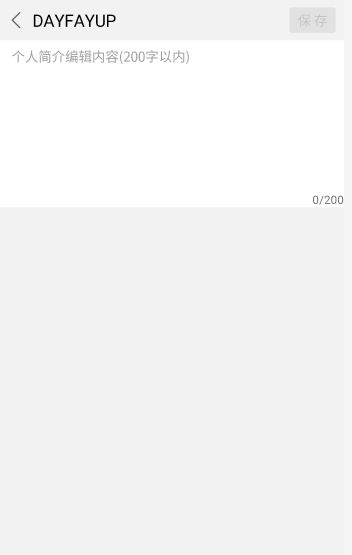
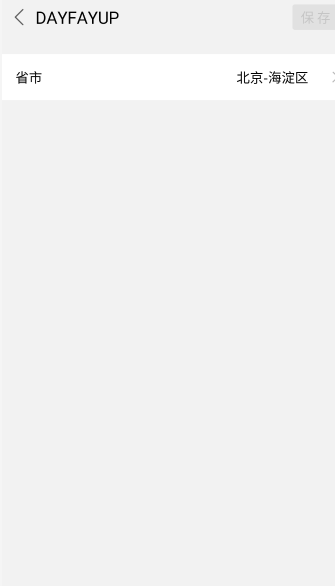


1. 登录注册等界面



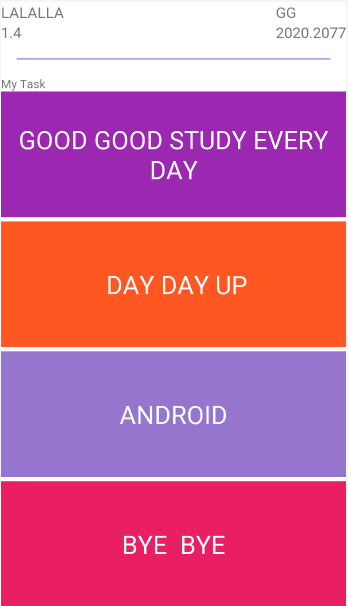
个人信息界面



可以选择所在地



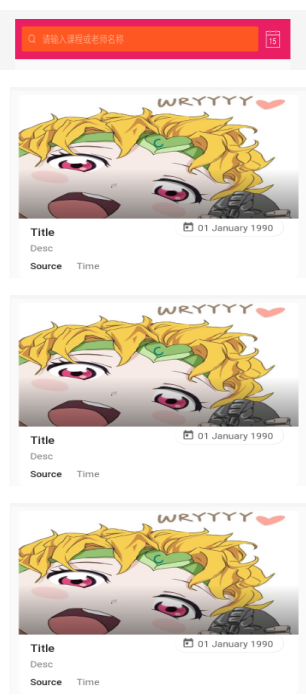
我的任务

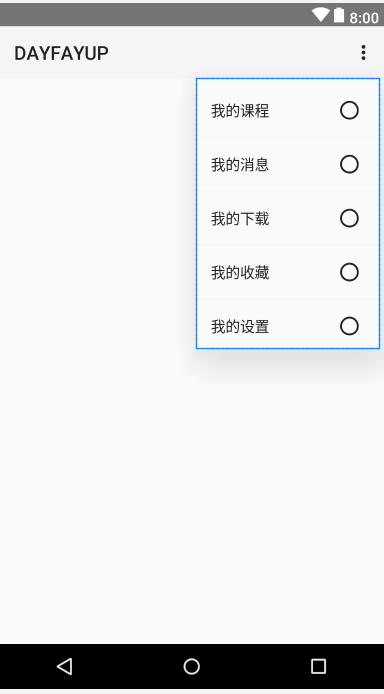


主界面

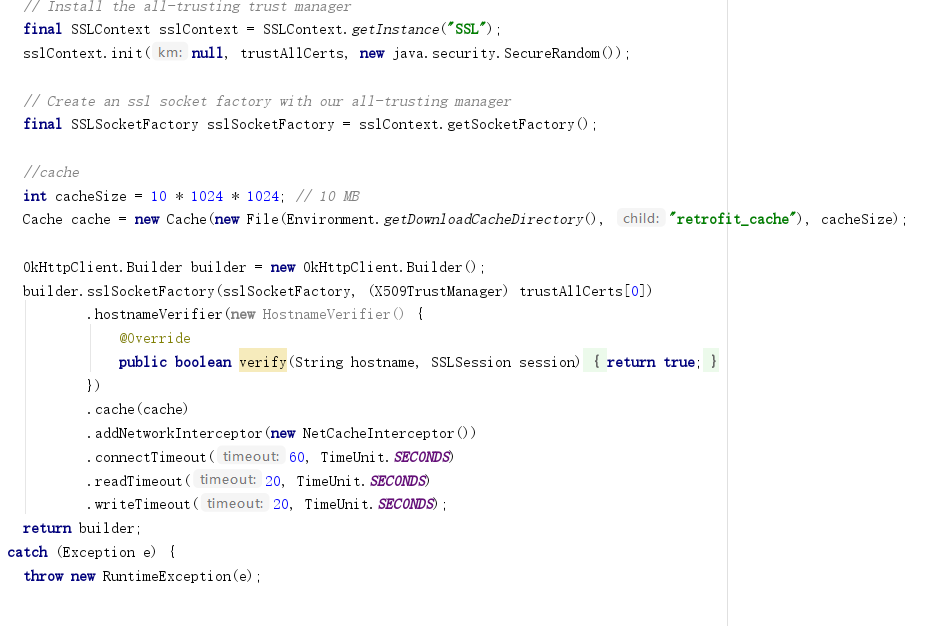








1. 代码





**package** com.star.e\_learning.ui.activity;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.core.app.ActivityCompat;  
  
**import** android.app.ProgressDialog;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.gson.Gson;  
**import** com.google.gson.JsonObject;  
**import** com.star.e\_learning.R;  
  
**import** com.basgeekball.awesomevalidation.AwesomeValidation;  
**import** com.basgeekball.awesomevalidation.ValidationStyle;  
**import** com.star.e\_learning.api.ApiClient;  
**import** com.star.e\_learning.api.ApiInterface;  
**import** com.star.e\_learning.util.AppConfig;  
**import** com.star.e\_learning.util.Utils;  
**import** com.star.e\_learning.bean.User;  
**import** com.star.e\_learning.repository.AppRepository;  
  
**import** java.util.HashMap;  
**import** java.util.Iterator;  
**import** java.util.Map;  
  
**import** cn.sharesdk.framework.Platform;  
**import** cn.sharesdk.framework.PlatformActionListener;  
**import** cn.sharesdk.framework.PlatformDb;  
**import** cn.sharesdk.framework.ShareSDK;  
**import** cn.sharesdk.tencent.qq.QQ;  
**import** retrofit2.Call;  
**import** retrofit2.Callback;  
**import** retrofit2.Response;  
  
**public class** LoginActivity **extends** AppCompatActivity **implements** View.OnClickListener {  
  
 **private** ProgressDialog **progressDialog**;  
 **private** Button **login**;  
 **private** TextView **forget**;  
 **private** TextView **register**;  
 *//actually it's the email* **private** EditText **username**;  
 **private** EditText **password**;  
 **private** ImageView **qqLogin**;  
  
 **private** AwesomeValidation **awesomeValidation**;  
 **private** AppRepository **appRepository**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_login***);  
 **login** = findViewById(R.id.***bt\_login\_submit***);  
 **forget** = findViewById(R.id.***tv\_login\_forget\_pwd***);  
 **register** = findViewById(R.id.***tv\_login\_register***);  
 **username** = findViewById(R.id.***et\_login\_username***);  
 **password** = findViewById(R.id.***et\_login\_pwd***);  
 **qqLogin** = findViewById(R.id.***iv\_login\_qq***);  
 **login**.setOnClickListener(**this**);  
 **forget**.setOnClickListener(**this**);  
 **register**.setOnClickListener(**this**);  
 **qqLogin**.setOnClickListener(**this**);  
 **awesomeValidation** = **new** AwesomeValidation(ValidationStyle.***BASIC***);  
 initValidation();  
 verifyStoragePermissions();  
 **appRepository** = **new** AppRepository(LoginActivity.**this**);  
 }  
  
 **public void** initValidation(){  
 **awesomeValidation**.addValidation(**username**, **"^[a-zA-Z0-9\_-]+@[a-zA-Z0-9\_-]+(\\.[a-zA-Z0-9\_-]+)+$"**, **"请输入正确的邮箱"**);  
 **awesomeValidation**.addValidation(**password**, **"^[\\w]{6,12}$"**, **"请输入6-12位密码"**);  
 }  
  
 @Override  
 **public void** onClick(View view) {  
 **switch**(view.getId()){  
 **case** R.id.***bt\_login\_submit***:  
 **final** Intent intent = **new** Intent();  
 **if**(**awesomeValidation**.validate()){  
 **final** String email = **username**.getText().toString().trim();  
 **final** ApiInterface apiInterface = ApiClient.*getApiClient*().create(ApiInterface.**class**);  
 Call<JsonObject> call;  
 call = apiInterface.login(email, **password**.getText().toString().trim());  
 showProgressDialog(**this**, **"正在登陆..."**);  
 call.enqueue(**new** Callback<JsonObject>() {  
 @Override  
 **public void** onResponse(Call<JsonObject> call, Response<JsonObject> response) {  
 JsonObject object = response.body();  
 dismissProgressDialog();  
 **if** (response.isSuccessful() && response.body() != **null**) {  
 System.***out***.println(**"bbb"**);  
 **if** (object.get(**"result"**).getAsString().equals(**"success"**)) {  
 User user = **new** Gson().fromJson(object.get(**"user"**), User.**class**);  
 Utils.*putBooleanValue*(LoginActivity.**this**, AppConfig.***LOGIN\_STATE***, **true**);  
 Utils.*putValue*(LoginActivity.**this**, AppConfig.***CURRENT\_EMAIL***, email);  
 **appRepository**.insertUser(user);  
 AppConfig.*CURRENT\_USER* = user;  
 intent.setClass(LoginActivity.**this**, HomeActivity.**class**);  
 startActivity(intent);  
 overridePendingTransition(R.anim.***push\_up\_in***, R.anim.***push\_up\_out***);  
 finish();  
 }**else**{  
 Toast.*makeText*(LoginActivity.**this**,  
 **"登录失败, "** + object.get(**"msg"**).getAsString() + **"!"**,  
 Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 }  
  
 @Override  
 **public void** onFailure(Call<JsonObject> call, Throwable t) {  
 dismissProgressDialog();  
 Toast.*makeText*(LoginActivity.**this**,  
 **"Network failure, Please Try Again"** + t.toString(),  
 Toast.***LENGTH\_LONG***).show();  
 }  
 });  
 }  
 **break**;  
 **case** R.id.***tv\_login\_register***:  
 Intent intent1 = **new** Intent();  
 intent1.setClass(LoginActivity.**this**, RegisterActivity.**class**);  
 startActivity(intent1);  
 overridePendingTransition(R.anim.***push\_right\_in***, R.anim.***push\_right\_out***);  
 **break**;  
 **case** R.id.***tv\_login\_forget\_pwd***:  
 Intent intent2 = **new** Intent();  
 intent2.setClass(LoginActivity.**this**, ForgetPwdActivity.**class**);  
 startActivity(intent2);  
 overridePendingTransition(R.anim.***push\_right\_in***, R.anim.***push\_right\_out***);  
 **break**;  
 **case** R.id.***iv\_login\_qq***:  
 **final** Intent intentqq = **new** Intent();  
 Utils.*showLongToast*(LoginActivity.**this**, **"You are using qq login!"**);  
 **final** Platform plat = ShareSDK.*getPlatform*(QQ.***NAME***);  
 ShareSDK.*setActivity*(**this**);*//抖音登录适配安卓9.0  
//回调信息，可以在这里获取基本的授权返回的信息，但是注意如果做提示和UI操作要传到主线程handler里去执行* plat.setPlatformActionListener(**new** PlatformActionListener() {  
 @Override  
 **public void** onError(Platform arg0, **int** arg1, Throwable arg2) {  
 *//* ***TODO Auto-generated method stub*** arg2.printStackTrace();  
 }  
  
 @Override  
 **public void** onComplete(Platform platform, **int** action, HashMap<String, Object> res) {  
 *//* ***TODO Auto-generated method stub*** *//输出所有授权信息* platform.getDb().exportData();  
 *//遍历Map* Iterator ite =res.entrySet().iterator();  
 **while** (ite.hasNext()) {  
 Map.Entry entry = (Map.Entry)ite.next();  
 Object key = entry.getKey();  
 Object value = entry.getValue();  
 System.***out***.println(key+**"： "**+value);  
 }  
 **if** (action == Platform.***ACTION\_USER\_INFOR***) {  
 PlatformDb platDB = platform.getDb();*//获取数平台数据DB  
 //通过DB获取各种数据* System.***out***.println(**"token: "** + platDB.getToken());  
 System.***out***.println(**"userId: "** + platDB.getUserId());  
 platDB.getUserGender();  
 platDB.getUserIcon();  
 platDB.getUserId();  
 platDB.getUserName();  
 }  
 Utils.*putBooleanValue*(LoginActivity.**this**, AppConfig.***LOGIN\_STATE***, **true**);  
 AppConfig.*CURRENT\_USER* = **null**;  
 intentqq.setClass(LoginActivity.**this**, HomeActivity.**class**);  
 startActivity(intentqq);  
 overridePendingTransition(R.anim.***push\_up\_in***, R.anim.***push\_up\_out***);  
 finish();  
 }  
  
 @Override  
 **public void** onCancel(Platform arg0, **int** arg1) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
 plat.SSOSetting(**false**);  
 plat.showUser(**null**);  
 plat.removeAccount(**true**);  
 **break**;  
 }  
 }  
  
 **public void** showProgressDialog(Context mContext, String text) {  
 **if** (**progressDialog** == **null**) {  
 **progressDialog** = **new** ProgressDialog(mContext);  
 **progressDialog**.setProgressStyle(ProgressDialog.***STYLE\_SPINNER***);  
 }  
 **progressDialog**.setMessage(text); *//设置内容* **progressDialog**.setCancelable(**false**);*//点击屏幕和按返回键都不能取消加载框* **progressDialog**.show();  
  
 *//设置超时自动消失* **new** Handler().postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 *//取消加载框* **if**(dismissProgressDialog()){  
  
 }  
 }  
 }, 30000);*//超时时间30秒* }  
  
 **public** Boolean dismissProgressDialog() {  
 **if** (**progressDialog** != **null**){  
 **if** (**progressDialog**.isShowing()) {  
 **progressDialog**.dismiss();  
 **return true**;*//取消成功* }  
 }  
 **return false**;*//已经取消过了，不需要取消* }  
  
 **private static final int *REQUEST\_EXTERNAL\_STORAGE*** = 1;  
 **private static** String[] *PERMISSIONS\_STORAGE* = {  
 **"android.permission.READ\_EXTERNAL\_STORAGE"**,  
 **"android.permission.WRITE\_EXTERNAL\_STORAGE"**,  
 **"android.permission.WRITE\_SETTINGS"**};  
  
 **public void** verifyStoragePermissions() {  
 **try** {  
  
 ActivityCompat.*requestPermissions*(LoginActivity.**this**, *PERMISSIONS\_STORAGE*, ***REQUEST\_EXTERNAL\_STORAGE***);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
}

由于代码过多不一一展示

1. 包

Adapter**存放控件的适配器**

**activity具体活动**

**bean实例对象**

**api接口网络请求**

1. 功能基本实现

后台服务器:使用 Spring boot 框架，严格遵守接口分离原则。Dao 层向Services层提供数据库操作相关接口，由Controller层调用 Services 层接口完成业务逻辑操作，并将数据以 Json 格式返回给客户端。

前后端交互:客户端通过 HTTP 请求服务器接口，服务器接口根据对应参数，返回 JSON 格式数据。客户端端解析 JSON 数据，进行数据显示。

目前具备的功能：查看课程、浏览课程、更改信息、注册账户、修改密码、用户登录、查看课程老师、观看课程材料（包含视频）、第三方登录（QQ）