

# X-Mot

一个基于安卓的法语词典应用

## 概述

本安卓应用有三个主要界面：

1. 查单词
2. 背单词
3. 用户设置





查单词



背单词



用户设置



LTE   2:53

Xmot





极速背词



顺序背词



查单词



背单词



用户设置



2:53

用户名

Password (optional)

**SIGN IN**

**REGISTER**



查单词

LTE   2:54

# Xmot

bon

查找

## 本地释义

bon,bonne :

adj. 1.好的.良好的;2.恰当的,适宜的,正确的,有利的;3.卓越的,优秀的,出色的,精通业务的,有技能的,有经验的;4.令人满意的,得益的;5.令人愉快的,美好的,美味的;6.有益的,适意的,提神的;7.高尚的,殷勤的,亲切的,和蔼的,善良的,仁慈的,乐于助人的;8.品行端正的,合乎道德的;9.大的,满的,足量的,强烈的,厉害的,完全的,十足的;10.[祝愿用语]bon n.m. 好人,善良人;好,善,好东西,好处,优点bon adv. Il fait ~ 天气好,气候宜人bon interj. [表示赞成,结论,确认]bon [bz4] n.m. 票据,凭单,息票

## 网络释义

bon: 1好的, 良好的 être de bon augure 是好兆头 être en bonne condition physique 身体状态良好 être à bonne école 受到良好的教育 être en bonne/en mauvaise posture 处境好/不好 être né sous une bonne étoile 生来命好 être en bons/mauvais termes avec qqn 和某人关系好/不好 vivre en bon voisinage avec qqn 和某人的邻里关系很和睦 entretenir de bons rapports avec ses voisins 与自己的邻居们保持良好的关系 manquer



查单词



背单词



用户设置



LTE 8:01

# Xmot

bien

查找

## 本地释义

bien :

1 下流地 2 适当地 3 好 4 令人满意地 5 顺利地 6 适当地 7 恰当地

adv. 止饬地,适当地;好,令人满意地,有利地,适宜地;许多,很,非常,完全,十分;思想正统的,福利的确,确实,真正地;至少,大概;表示赞同,赞许,认可;表示语气的转折,逻辑的连接  
bien que loc.conj. 尽管,虽然adj.inv. 好多,正直的,正派的,品德好的;令人满意的,正确的;好看的,漂亮的,可爱的,讨人喜欢的;高雅的,潇洒的,有风度的,雍容华贵的n.m. 好,善,正直,善良;好处,益处,利益,福利;财产,产业;资料,物资

## 网络释义

loading...



查单词



背单词



用户设置



LTE 8:02

Xmot

these

查找

## 本地释义

词库中无此词，请重新输入。

## 网络释义

these: 论文, 论题





查单词

背单词

用户设置

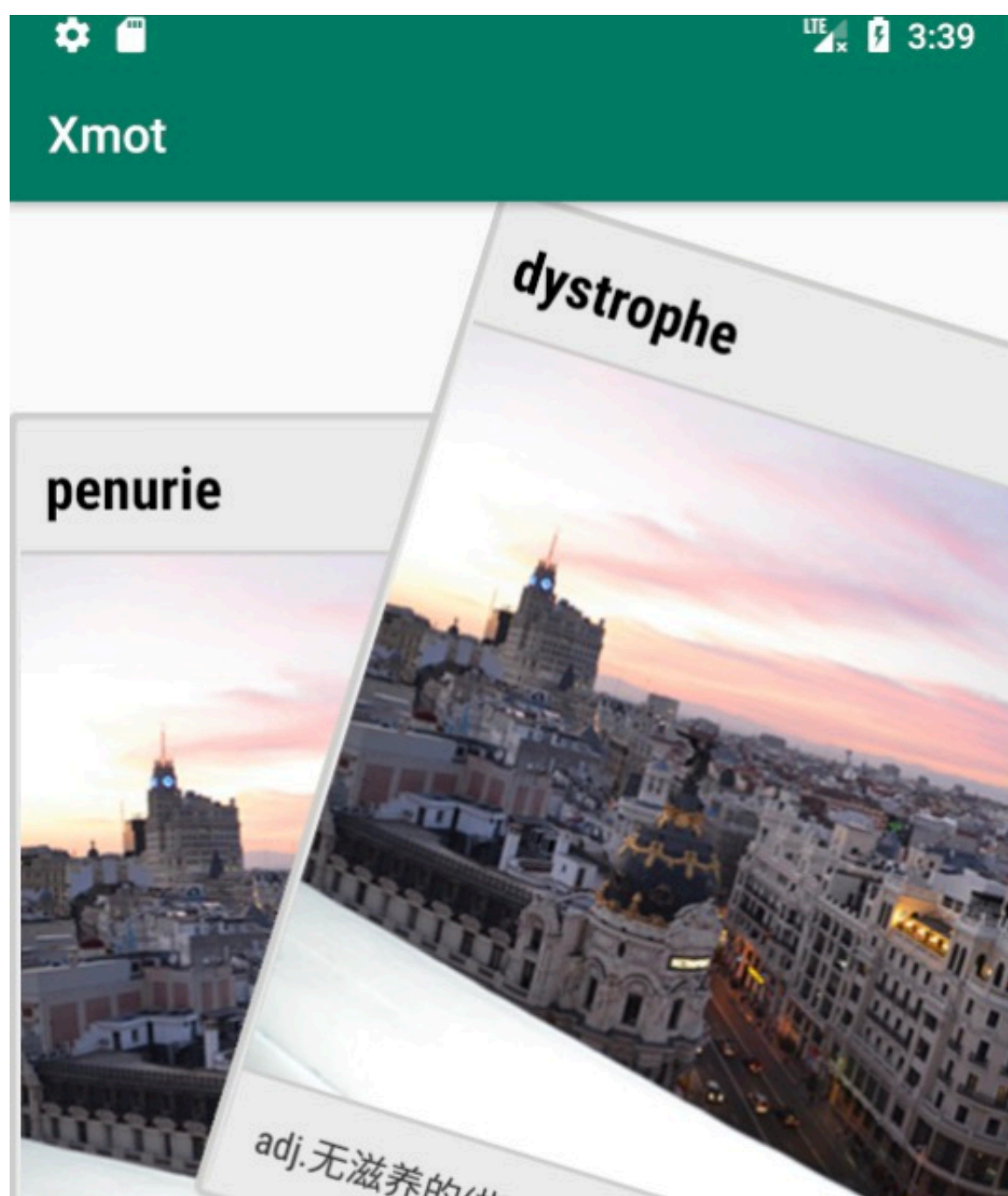


- 当在查询栏输入了想要查的词后，点击查询，会出现本地释义（使用本地数据库）和网络释义（与服务

器通讯获取）。若本地无此词或网络正在连接，则会出现提醒。

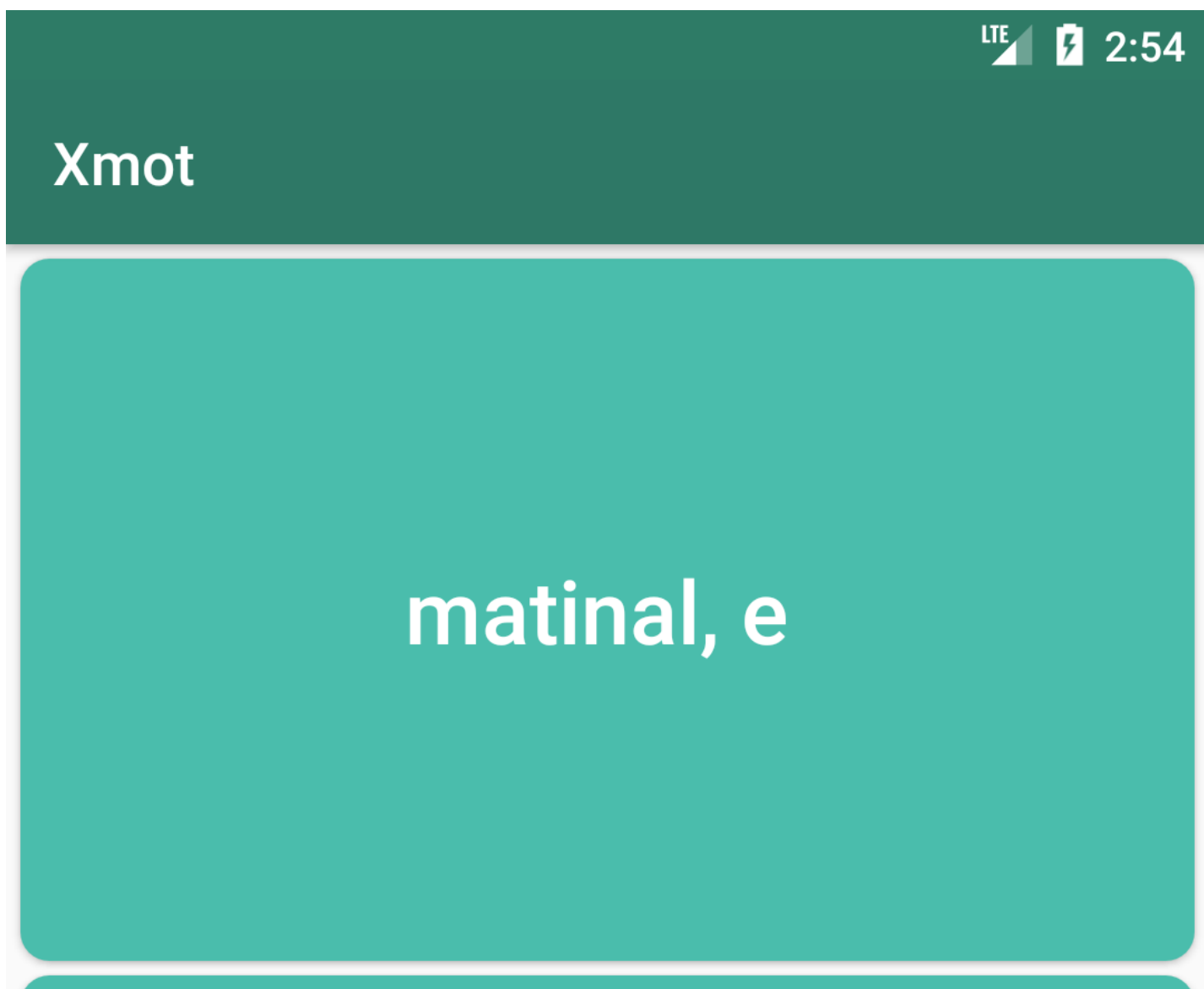
## 背单词

### 极速背词





顺序背词



n. 轨道总称(铁路线全长)

adj. 历时性的, 贯时性的

adj. 早起的, 早晨的

v. 损害(故障, 赔款)



LTE   2:54

Xmot

contre-écrou

n.f. 花边女工

n.m. 防松螺母

v. 链条(电路,一系列,连锁商店,测链)

Wrong Answer!

n.m. 猛撞,[机]冲压,冲制

用户界面

未登录

LTE   2:53

# Xmot

用户名

Password (optional)

**SIGN IN**

**REGISTER**

  
查单词

  
背单词

  
用户设置



LTE   3:06

Xmot

用户名

ldh

Password (optional)

...

提醒

Username or password is wrong. Please try

again!

是



查单词



背单词



用户设置



LTE   3:06

Xmot

用户名

ldh

Password (optional)

提醒

Registration succeed.

是



查单词



背单词



用户设置

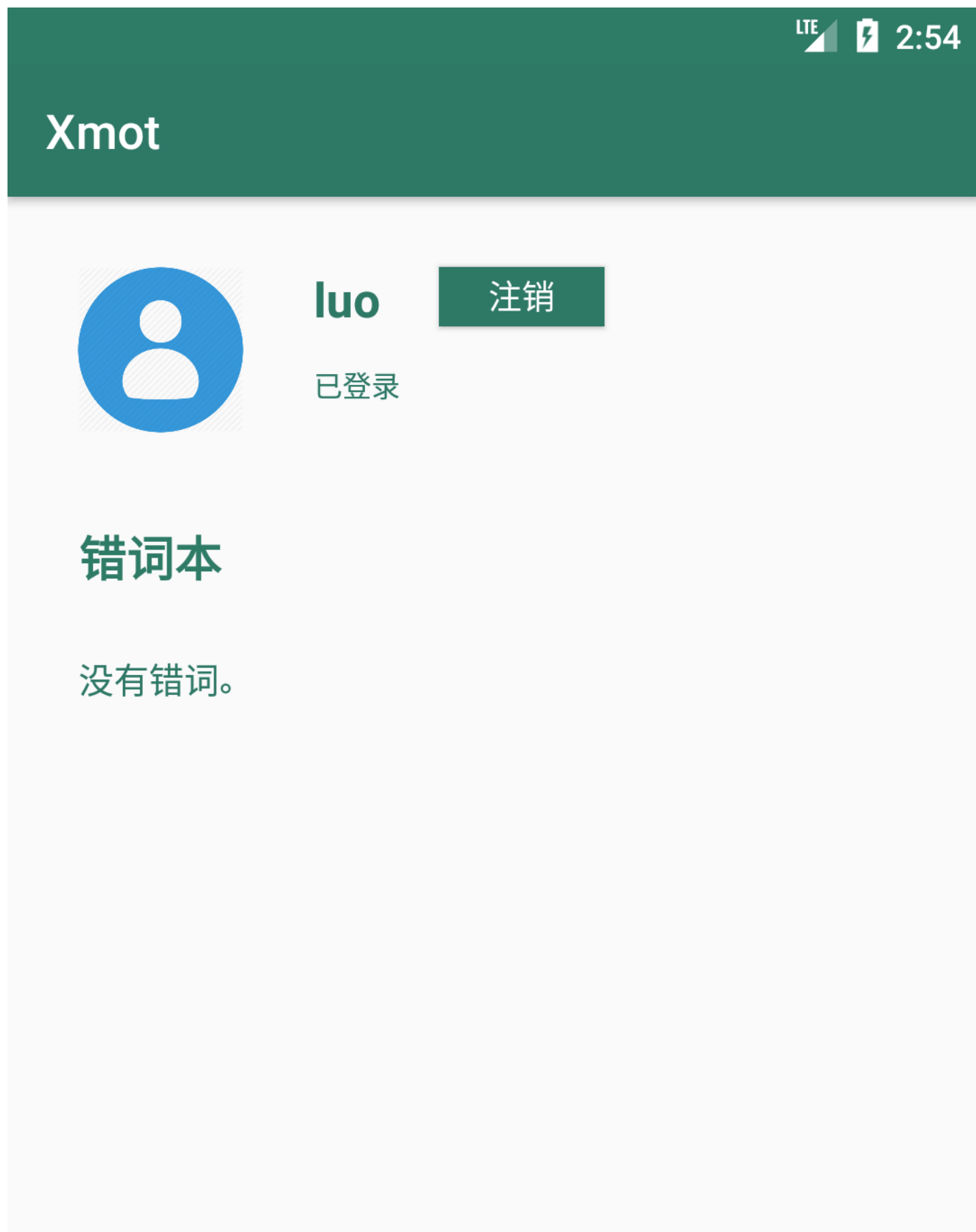


- 输入用户名和密码后，点击登录，与服务器通信，若正确则成功进入登录后界面，若错误则会提示“用户



名或密码错误”。点击注册，提示“注册成功”，之后便可登录。

登录后





查单词



背单词



用户设置



LTE   2:55

# Xmot



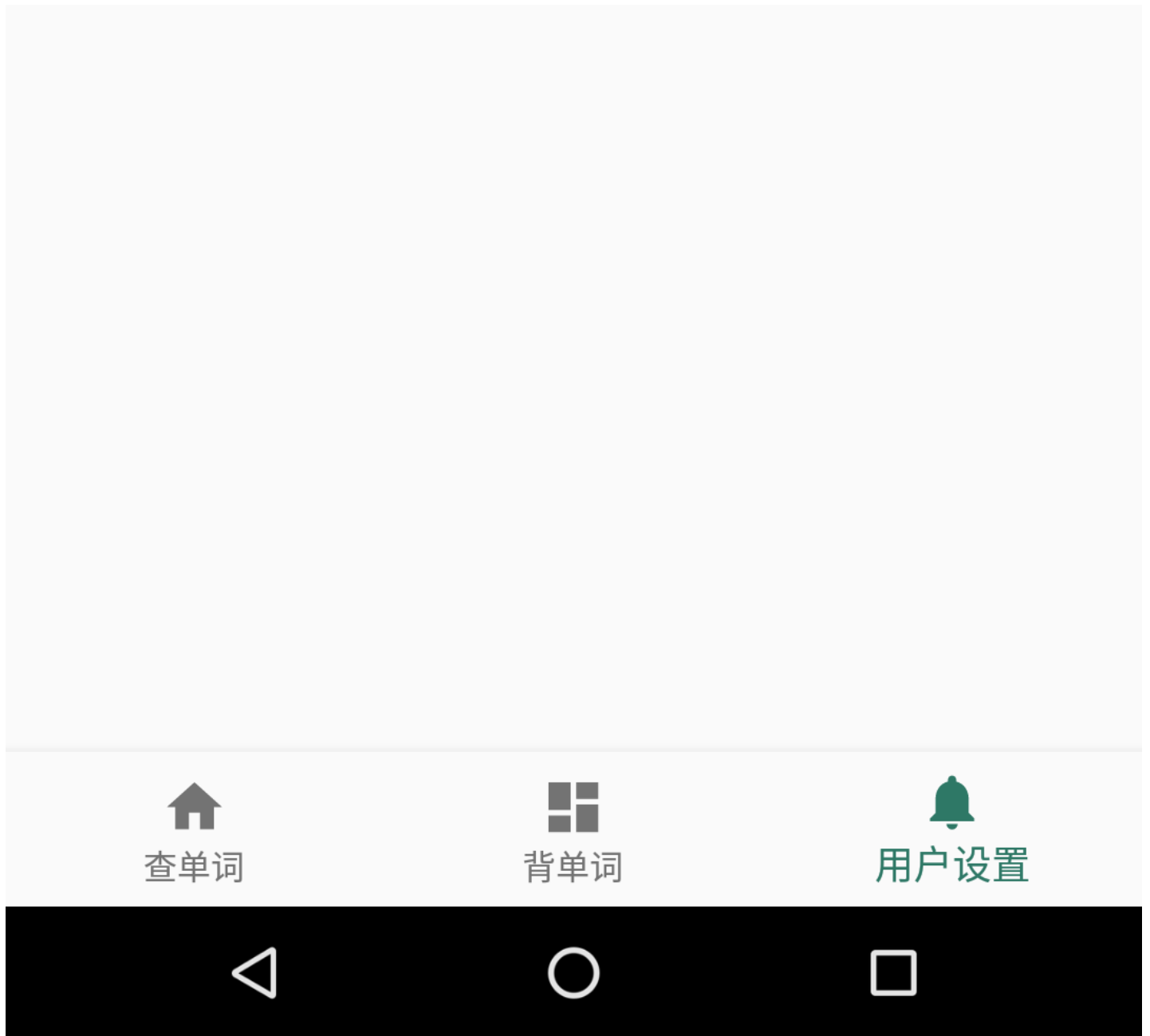
luo

注销

已登录

## 错词本

vermisseaux : n.m.小蚯蚓,小虫; [转]小小的可怜虫  
electrique : adj. 电的;发电的,用电的  
agent de voyage : n.m.旅行代理人



- 初始状态下，错词本没有生词。若在登录状态下，顺序背词中有选错词，则会自动上传至服务器记录，并在用户界面显示。
- 点击注销，则会回到登录前界面。

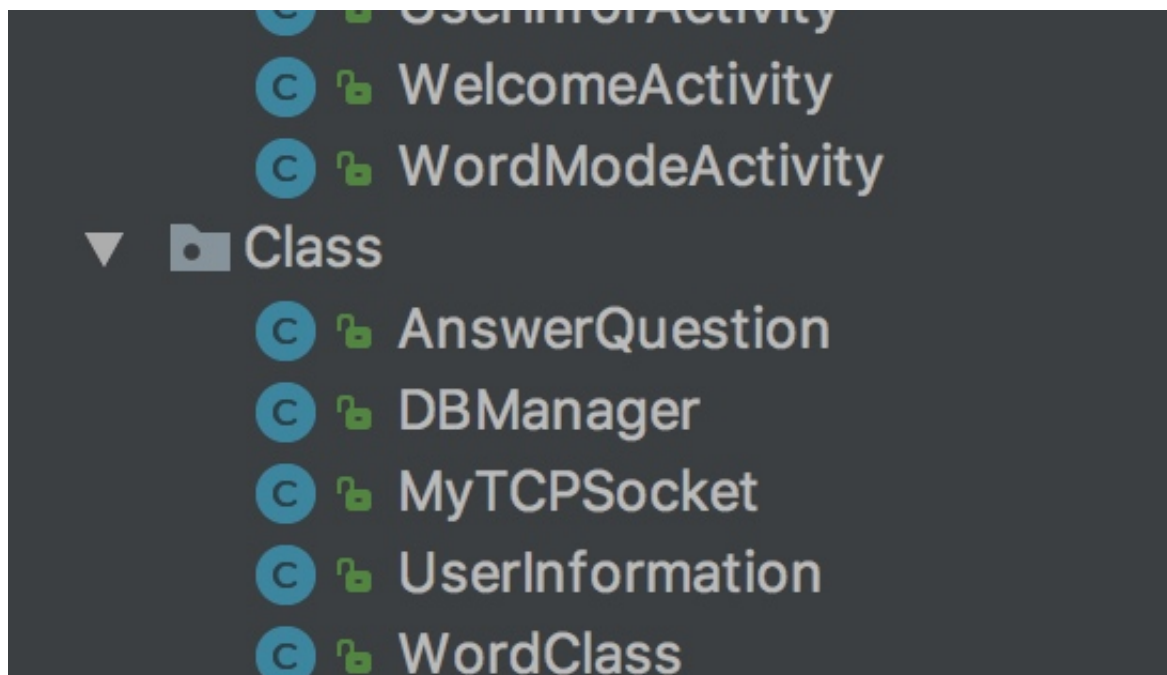
## 网络连接

---

### 服务器端

### 本地端

- 多个 activity 之间共享同一个 socket：使用单例模式



```
public class MyTCPSocket extends Socket {
    private static final String host = "192.168.1.100";
    private static final int port = 6789;
    private static BufferedReader br = null;
    private static PrintWriter pw = null;
    private static MyTCPSocket socket = null;
    private static DataOutputStream outToServer = null;
    private static BufferedReader inFromServer = null;

    private MyTCPSocket(String host, int port) throws UnknownHostException, IOException {
        super(host, port);
    }

    public static MyTCPSocket getsocket() throws IOException {
        if(socket==null){
            socket= new MyTCPSocket(host,port);
        }
        return socket;
    }
}
```

- 发送与接收时使用 utf-8 编码
- 使用 Socket, 不能在主线程中, 需要创建一个新线程

```
class Connect_Thread extends Thread implements Runnable//继承Thread
{
    public void run()
    {
        try {
            socket_helper.getsocket();
            text_InternetMeaning.setText(socket_helper.sendMessage("/demand "
+ text.getText()));
        }
        catch (Exception ex){
            Log.d("IndexActivity", "run: " + ex.getMessage());
        }
    }
}
```

## 本地数据库

---

从灵格斯词霸官网 (<http://www.lingoes.cn/zh/dictionary>) 下载开源法汉词典。使用SQLite搭建本地词库，由于词库较大，我们对数据库进行了一定的删减，去除了动词的不同变位形式以及生僻词。将数据库wordlib.db导入到文件路径\main\src\res\raw中。

在项目中新增类DBManager，同时保存手机存储数据库的目录。由以下代码导入数据库。

```

public void openDatabase() {
    this.database = this.openDatabase(DBPATH + "/" + DBNAME);
}
private SQLiteDatabase openDatabase(String dbfile) {
    try {
        InputStream is = this.context.getResources().openRawResource(
            R.raw.word); //欲导入的数据库
        FileOutputStream fos = new FileOutputStream(dbfile);
        byte[] buffer = new byte[BUFFER_SIZE];
        int count = 0;
        while ((count = is.read(buffer)) > 0) {
            fos.write(buffer, 0, count);
        }
        fos.close();
        is.close();
        SQLiteDatabase db = SQLiteDatabase.openOrCreateDatabase(dbfile,
            null);
        return db;
    } catch (FileNotFoundException e) {
        Log.e("Database", "File not found");
        e.printStackTrace();
    } catch (IOException e) {
        Log.e("Database", "IO exception");
        e.printStackTrace();
    }
    return null;
}
}

```

在初始活动IndexActivity中打开数据库。

## 具体实现

---

### 查单词

- 本地释义：使用本地的数据库。

```

private String searchForMeaning(){
    Log.d("IndexActivity", text.getText().toString());
    Cursor cur = database.rawQuery("SELECT word,meaning FROM words WHERE word
Like '\" + text.getText().toString() + \"%'\", null);
    if (cur != null) {
        if(cur.moveToFirst()) {
            Log.d("IndexActivity", cur.getString(cur.getColumnIndex("meaning")
));
            return cur.getString(cur.getColumnIndex("word")) + ":\n" + cur.get
String(cur.getColumnIndex("meaning"));
        }
        return "词库中无此词，请重新输入。";
    }
    else {
        Log.d("IndexActivity", "NO meaning");
        return "No find. Please try again.";
    }
}
}

```

- 网络释义：socket 获取后 TextView.setText(String)

```

class Connect_Thread extends Thread implements Runnable//继承Thread
{
    public void run()
    {
        try {
            socket_helper.getsocket();
            text_InternetMeaning.setText(socket_helper.sendMessage("/demand "
+ text.getText()));
        }
        catch (Exception ex){
            Log.d("IndexActivity", "run: " + ex.getMessage());
        }
    }
}
}

```

- 滑动：ScrollView 里再加 LinearLayout

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="50dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="15dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/Local"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="0dp"
                android:text="本地释义\n"
                android:textColor="@color/colorPrimaryDark"
                android:textSize="20dp"
                android:textStyle="bold"
                android:visibility="visible" />

```

## 背单词

### 极速背词

- 使用github开源的Swipeable-Cards (<https://github.com/kikoso/Swipeable-Cards>).
- 在rocketmodeactivitylayout中添加CardContainer.

```

com.andtinder.view.CardContainer
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/layoutview"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

```

- 在RocketActivity中声明类SimpleCardStackAdapter变量



```
CardContainer mCardContainer = (CardContainer) findViewById(R.id.layoutview);
mCardContainer.setOrientation(Orientations.Orientation.Ordered);
SimpleCardStackAdapter adapter = new SimpleCardStackAdapter(this);
```

- 往adapter容器中添加Card, 并赋给CardContainer

```
for (WordClass word:Words) {
    final CardModel card = new CardModel(word.word, word.meaning,
    this.getResources().getDrawable(R.drawable.picture3));
    card.setOnClickListener(new CardModel.OnClickListener() {
        @Override
        public void OnClickListener() {
            Log.i("Swipeable Cards","I am pressing the card");
        }
    });
    adapter.add(card);
}
mCardContainer.setAdapter(adapter);
```

## 顺序背词

- 自定义button样式。使用Cursor随机从数据库中取10个单词，选择错误选项将以Toast方法提示用户。限制单词释义长度以防止释义挤压按钮。
- 新建类MyClickListener来监听按钮，每次点击错误答案将向服务器发送信息。

```
Answer1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent();
        intent.putExtra("data_return","Correct");
        setResult(RESULT_OK, intent);
        finish();
    }
});
Answer2.setText(Questions.get(randomnums.get(0)).GetAnswer());
Answer2.setOnClickListener(new MyClickListener(i));
```

- 类MyClickListener实现OnClickListener接口

```
class MyClickListener implements View.OnClickListener {
    private Integer pos;
    public MyClickListener(Integer Pos){
        this.pos = Pos;
    }
    public void onClick(View v) {
        Toast.makeText(NormalModeActivity.this, "Wrong Answer!", Toast.LENGTH_SHORT).show();
        SendWrongWord(this.pos);
    }
}
```

- 创建新线程，向服务器发送信息，格式为 **/store #Username #Question #Answer**

```
class Connect_Thread extends Thread implements Runnable//继承Thread
{
    private String Question;
    private String Answer;

    public Connect_Thread(int i){
        this.Question = Questions.get(i).GetQuestion();
        this.Answer = Questions.get(i).GetAnswer();
    }

    public void run()
    {
        try {
            Log.d("NormalModeActivity", "/store " + "#" + User.get_Username() +
                "#" + Question + "#" + Answer);
            socket_helper.getsocket();
            socket_helper.sendMessage("/store " + "#" + User.get_Username()+ "
            #" + Question + "#" + Answer);
        }
        catch (Exception ex){
            Log.d("NormalModeActivity", "run: " + ex.getMessage());
        }
    }
}
```

## 用户界面

### 未登录时界面

- 使用 socket 发送和收取服务器的消息
- 使用单例模式在不同 activity 之间共享 username 和 password

```

public class UserInformation {
private static String user_name = null;
private static String password = null;
public UserInformation(){}
public String get_Username(){
    return user_name;
}
public String get_Password(){
    return password;
}
public boolean getLoginState(){
    if (user_name != null && password != null){
        return true;
    }
    else return false;
}
public boolean set_Username(String str){
    if (user_name != null){
        return false;
    }
    else {
        user_name = str;
        return true;
    }
}
public boolean set_Password(String str){
    if (password != null){
        return false;
    }
    else {
        password = str;
        return true;
    }
}
public boolean reset_all(){
    user_name = null;
    password = null;
    return true;
}
}

```

- 在启动此页面时，首先判断是否已登录。

```

if (user_infor.getLoginState()){
    Intent intent_index = new Intent(UserActivity.this, UserInforActivity.
class);
    startActivity(intent_index);
    finish();
}

```

- 需先等待服务器确认是否成功登录，再提示用户/切换 activity，但 alert 中有 loop 不能使用 Thread.join()，于是分两个线程写。

```

class MyListener_signin implements View.OnClickListener {
    public void onClick(View v){
        try {
            Connect_Thread_signin connect_Thread = new Connect_Thread_signin()
;

            connect_Thread.start();
            try {
                connect_Thread.join();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            Log.d("UserActivity", "onClick: " + login_state);
            if (login_state){
                Log.d("UserActivity", "onClick: " + email.getText() + " " + pa
ssword.getText().toString());
                user_infor.set_Username(email.getText().toString());
                user_infor.set_Password(password.getText().toString());
                Log.d("UserActivity", "run: loading to information activity...
");

                Intent intent_index = new Intent(UserActivity.this, UserInforA
ctivity.class);
                startActivity(intent_index);
                finish();
            }
            else {
                alert_thread alert = new alert_thread();
                alert.start();
            }
        } catch (Exception e) {
        }
    }
}

```

## 已登录后界面

- 注销功能
  - 重置 UserInformation Class

- 向服务器发送 /quit
- 错词本
  - 向服务器请求 /require # username