

基于 Android 的物流管理系统

Express Sensor

项目报告 V1.0

Moonlight Team @HEBUT

Find more: es.eric Yue.info

● 项目简介

⇒ 团队简介

Moonlight Team,
本着从点滴做起,
让用户满意的态度来完成每一个模块的设计。
使项目具有良好的用户体验。

➤ 团队成员为:

岳斌
男, 20 岁, 本科二年级
河北工业大学, 计算机科学与软件学院, 软件工程专业, 091 班
电话: 15122136075
邮箱: hi.moonlight@gmail.com

⇒ 项目简介

Express Sensor, 是一款基于 Android 的智能物流管理系统。

整个作品采用 JAVA 语言编写, 其中数据部分存在 MySQL 中。手机测试系统为 Android 2.3.4, 其中手机端与服务器端通过 JAVA 服务器上的 Servlet 来交流数据, 完成与数据库的信息传递。

它解决了现有物流方案的一些问题, 简化了投递过程。充分利用手持终端的特性 (如 GPS、GSensor 等) 来优化各个物流环节, 为用户提供更好的体验。

它采用 GPS 自动感应地理位置信息, 自动发送收货通知, 并且采用类 NFC 的感念来实现自动签收, 无需用户出示证件。如果用户满意, 摇一摇手机即可完整整个验货收单的过程。

同时快递员的客户端可以实时的上交自己的 GPS 信息给服务器, 供管理人员分析, 服务器会记录快递员的轨迹流, 但不会记录用户的轨迹信息, 只会记录当前坐标, 并且只在收获的过程中开启记录 GPS 功能, 保障了用户隐私。

快递端支持二维码识别功能, 可以快速掌握包裹的信息 (包裹邮单印二维码)。

⇒ 程序功能分析

✓ 手机快递员客户端:

项目名称 ExpressSensor, 这是本系统的关键部分, 具体功能分析如下:

- 类 NFC 签收。首先本程序大胆的采用“**类 NFC**”的概念(通过“WIFI 辅助投递”(下有介绍)功能, 可以将距离精确到 20m 内)来实现快递的自动感应签收, 改善了物流环节, 优化了签收过程。
- 自动发送通知。快递端可以实时监听用户的当前位置, 确定与自己的距离, 当距离在 2000 米左右的时候, 会自动发送收货通知给顾客, 免去了人为发送的麻烦。
- 记录轨迹到服务器。快递端可以实时的上传自己的 GPS 信息到服务器端, 供管理人员分析使用。
- WIFI 辅助投递。通过快递端监听到与用户距离接近时, 由于 GPS 误差比较大, 所以在近距离是, 通过快递端在后台启动 WIFI 热点, 如果用户进入此范围内, 则说明用户前来签收。
- 排名统计。服务器上的 info. ericyue. web. listener 包中, 包括了服务器启动的时候会注册的定时器。当特定的时间到达时, 会计算每日, 每周, 每月的投递统计。并且计算每日, 每周, 每日的快递员排名, 可以供绩效工资使用。
- 二维码识别。快递员所使用的客户端内置二维码识别功能, 精准快速的识别各种条形码。可以让快递员迅速了解包裹信息。

✓ 手机用户客户端

项目名称 ExpressSensorClient, 这是本系统的辅助部分, 具体功能分析如下:

- ◆ 跟踪包裹。用户可以通过手机端来实时跟踪自己的包裹, 查看包裹当前的位置, 以及距离自己的大概距离。以便安排合适的时间去接收。
- ◆ 类 NFC 感应签收。客户无需出示身份证件, 靠近快递员即可完成身份验证, 免去了复杂的签收步骤。
- ◆ 摇一摇签收。如果客户到场后对自己的货物满意, 可以直接摇一摇手机来完成身份验证的

✓ 服务器端

项目名称 ExpressSensorWeb, 这是本系统的辅助部分, 具体功能分析如下:

- ◆ 数据传递。该部分完成手机端与数据库间的数据传递, 因为 Android 不能直接使用 MySQL 数据库, 所以可以通过 Servlet 来中转。
- ◆ 信息统计。设置了 3 个定时器, 会在每天、每周、每月结束的时候统计每个快递员的投递信息, 存入数据库。
- ◆ 快递排名。根据先前统计的投递信息, 设置监听器计算每个快递员的排名, 提供给管理人员计算绩效工资使用。

● 编程环境

⇒ 手持终端:

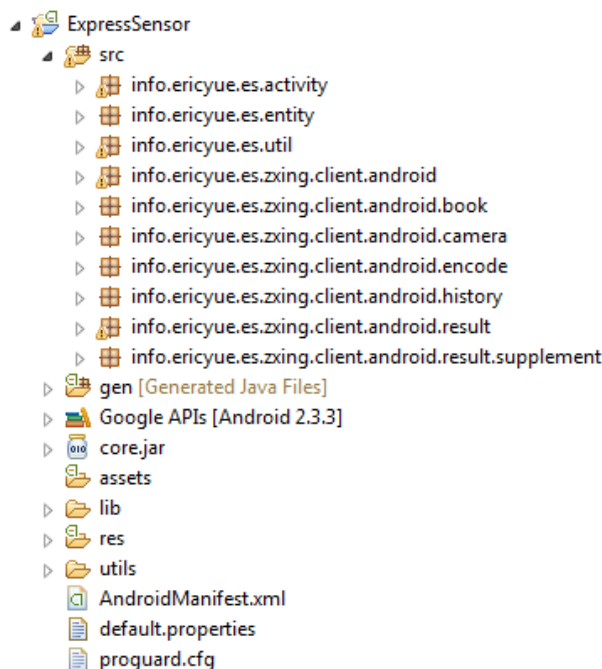
运行平台:	Android 2.3.4
编程语言:	JAVA
编程工具:	Eclipse
有效代码长度:	8000 左右
外调 API:	Google Map API Google Chart API Zxing Lib (QRCode)
真机测试:	HTC G7, Google Nexus One, HTC MyTouch 4G

⇒ 服务器端

运行平台:	Cent OS 6.0 Apache Tomcat 7.0 MySQL Server 5.5
编程语言:	JAVA (Servlet)
编程工具:	Eclipse
有效代码长度:	1000 左右
JAVA 版本:	1.6.0

● 程序结构分析

⇒ 快递端结构分析

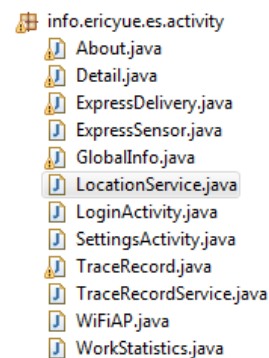


✧ 包 info.ericvue.es.activity

此包包含程序的全部 Activity。

具体结构如右图。

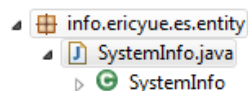
1. About 类。程序的介绍页面。
2. Detail 类。投递详情，显示某个邮单的详细内容。在 ExpressDelivery 中调用此类。
3. ExpressDelivery 类。快递投递类。此类是辅助投递用的，其中显示本次派送の詳細名单以及有邮包状态。
4. ExpressSensor 类。本程序的主界面。
5. GlobalInfo 类。显示全局信息。如当前位置，总共委派，剩余投递数目等信息。
6. LocationService 类。此类为非常重要的后台服务。是程序的核心部分。程序登陆后会激活此后台服务。来实时监听自身以及用户的 GPS 位置信息，以便开启相应功能(如自动通知和 NFC 签收等等)。
7. LoginActivity 类。此类是系统的默认启动类。提供了登陆窗口和身份验证功能。
8. SettingActivity 类。此类为系统设置类。
9. TraceRecord 类。为轨迹记录类，功能为显示快递员开始到现在的行走轨迹。并且可以显示本次派送列表中的用户位置。
10. TraceRecordService 类。此类为开启轨迹监听的服务。



11. WiFiAP 类。此类也是本程序的重点。是 WIFI 辅助签收的实现文件。由于 AndroidSDK 不提供创建无线热点的 API，所以采用 JAVA 反射来调用系统隐藏的 API，来实现创建无线热点的功能。
12. WorkStatistics 类。本类为工作统计类。此类从服务器获得详细的统计信息，并且获得每日每周每月的排名。

➤ 包 info.ericvue.es.entity

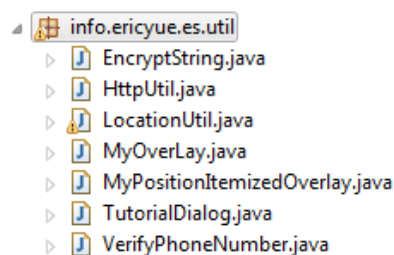
此包为 LoginActivity 类的辅助类。



➤ 包 info.ericvue.es.util

此类提供了一些工具, 如 MD5 加密、HttpRequest 等等

1. EncryptString 类。提供了加密字符串的功能，可以选择 MD5 加密。
2. HttpUtil 此类封装了一系列与 Http 相关的操作。大部分为数据库交互。
3. LocationUtil 类。本程序的核心类。从 LocationService 类中调用。提供了鉴定位置服务以及判断距离等等核心功能。
4. MyOverLay 和 MyPositionItemizedOver lay 类。这两个类提供了在 GoogleMap 上划线或者描点的功能。
5. TutorialDialog 类。提供了第一次打开程序的 Tutorial 的窗口提示。
6. VerifyPhoneNumber 类。提供了验证手机号码是否有效的功能。

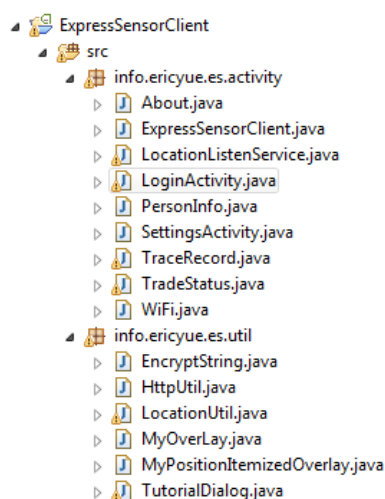


➤ 包 info.ericvue.es.zxing.*

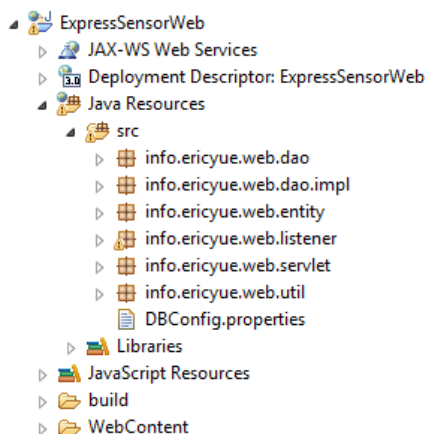
这一系列包为二维码识别的相关文件。

⇒ 用户客户端模块

大部分与 ExpressSensor 类似，不介绍了。



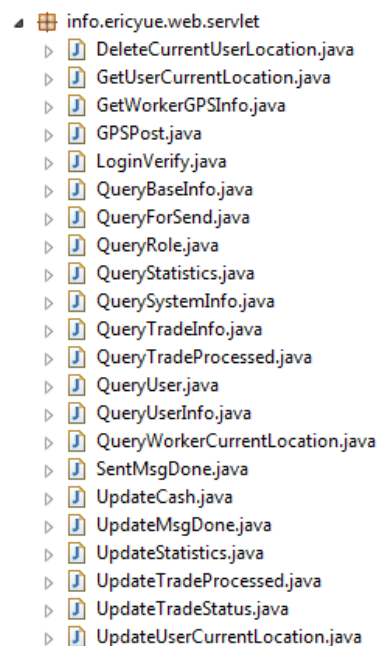
⇒ 服务器模块



- 包 `info.ericvue.web.dao`、`info.ericvue.web.dao.impl`、`info.ericvue.web.entity` 这三个提供了验证登陆信息的相关操作。
- 包 `info.ericvue.web.listener`。此包中包含服务器的三个定时器，以及相关的处理函数。主要功能是在服务器启动后，在每天、每周、每月计算排名。
- 包 `info.ericvue.web.servlet`。

此包中包含数据库相关的操作。

1. `DeleteCurrentUserLocation` 删除数据库 `current_location` 中特定 `id` 的数据。
2. `GetCurrentLocation` 取得 `current_location` 中用户当前位置
3. `GetWorkerGPSInfo` 取得快递员的最新的 GPS 信息。
4. `GPSPost` 向数据库提交快递员的 GPS 信息
5. `LoginVerify` 登陆验证功能
6. `QueryBaseInfo` 查询 `user` 表中的数据
7. `QueryForSend` 查询 `tradeinfo` 表中的信息
8. `QueryRole` 查询用户的角色
9. `QueryStatistics` 查询某个 `id` 的投递统计信息
10. `QuerySystemInfo` 查询系统信息
11. `QueryTradeInfo` 查询 `tradeinfo` 中特定的条目。
12. `QueryTradeProcessed` 取得当前 `trade` 条目的处理状态。
13. `QueryUser` 通过 `username` 查询 `id`
14. `QueryUserInfo` 查询 `user` 表的全部信息
15. `QueryWorkerCurrentLocation` 查询某个快递员的最近 GPS 信息。
16. `SentMsgDone` 查询某单是否发送过短信通知
17. `UpdateCash` 更新 `tradeinfo` 中的现金项
18. `UpdateMsgDone` 更新 `tradeinfo` 中发送短信的状态
19. `UpdateStatistics` 更新数据库 `statistics` 中的信息
20. `UpdateTradeProcessed` 更新货单的处理状态



- 21. UpdateTradeStatus 更新货单状态
- 22. UpdateUserCurrentLocation 更新用户当前位置

● 项目展望

很遗憾的说，由于消息闭塞，得知本比赛的时间较晚，本项目从开始计划到编码完毕只有短短的一个月时间。曾经想的很多很多的创意点还有好多又没加入到其中。只能说我尽力了，一个人完成初步的设计。

我对这个项目的未来抱有很大的信心。因为信息化的物流时代必然到来。

我希望在未来的改进中，可以加入物联网的概念。我初步设想是这样的，由于物流公司在全国个地方都有快递人员，遍布全国各地，那么，完全有可能将这一资源形成一个巨大的自组网，不用现有的因特网连接，采用内部自组网来实现全部功能。而且我们可以让用户资源的加入创建自组网的行列，为物流行业作出一份贡献，自组网的终端越多，网络的范围越大，越稳定。当然有很多事情还有待考虑，智能手机的自组网现在还有没先例。

其次，有可能的话未来加入云计算的元素，毕竟物流公司的用户群的巨大的，必须保重其速度与稳定性。

● 参考

- Google Android Docs
- 《Android 应用开发揭秘》
- 《Android SDK 开发范例》

感谢@MathiasLin 的无私帮助.