客户名字和logo

餐厅订餐系统-客户端详细设计文档

wang linglong

日期：2014-12-08

文档版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修订日期 | 修订人 | 审核人 | 变更内容 |
| V0.1 | 2014/11/30 | Wang LingLong |  | 初始化 |
| V0.2 | 2014/12/8 | Wang LingLong |  | 补充详细内容 |
| V0.3 | 2014/12/8 | Wang LingLong |  | 更新类图和时序图 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 目录

目录 2

1 文档介绍 4

1.1 文档目的 4

1.2 文档范围 4

1.3 缩写词列表 4

1.4 参考内容 4

2 模块设计 5

2.1 菜品浏览 5

2.1.1 设计图 5

2.1.2 功能描述 5

2.1.3 菜品浏览类图 6

2.1.4 菜品浏览时序图 7

2.1.5 函数说明 7

2.2 点菜 10

2.2.1 设计图 10

2.2.2 功能描述 10

2.2.3 类图 11

2.2.4 时序图 11

2.2.5 函数说明 11

2.3 下单 13

2.3.1 设计图 13

2.3.2 功能描述 13

2.3.3 类图 14

2.3.4 时序图 15

2.3.5 函数说明 15

2.4 历史订单查询 17

2.4.1 设计图 17

2.4.2 功能描述 17

2.4.3 类图 18

2.4.4 时序图 18

2.4.5 函数说明 18

3 存储数据设计 20

3.1 应用设置 20

4 数据缓存设计 21

4.1 功能描述 21

4.2 实现方式 21

4.3 函数说明 22

# 文档介绍

## 文档目的

本文描述eOrder餐厅订餐系统架构设计，用于指导开发人员进行系统详细设计；需求人员进行系统需求实现评估；测试人员进行系统测试用例编制。

## 文档范围

本文档主要指导客户端开发人员进行编码工作，并供测试人员等项目其他人员参考。

## 缩写词列表

|  |  |
| --- | --- |
| 缩写词 | 解释 |
| MVC | MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 参考内容

# 模块设计

## 菜品浏览

### 设计图



### 功能描述

浏览菜单

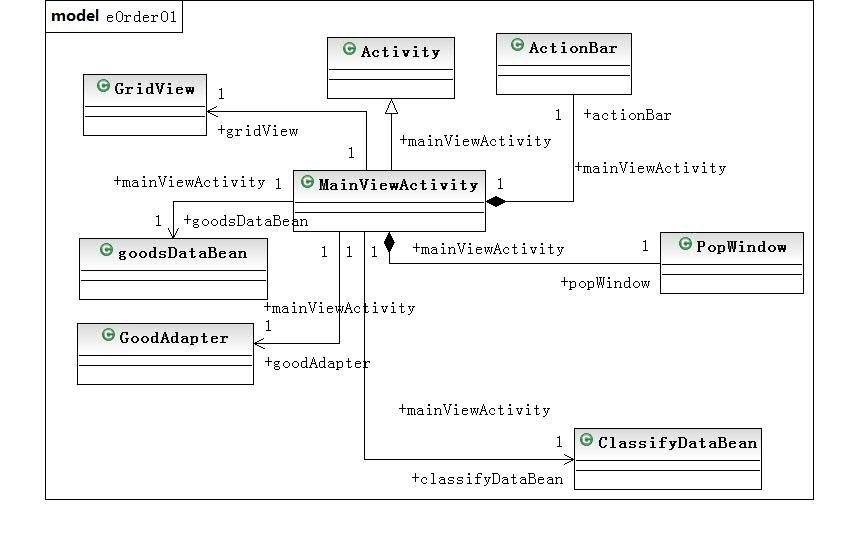
1. 用户打开手机上的电子菜单应用

2. 应用按菜单分类显示第一大类第一分页的菜品信息

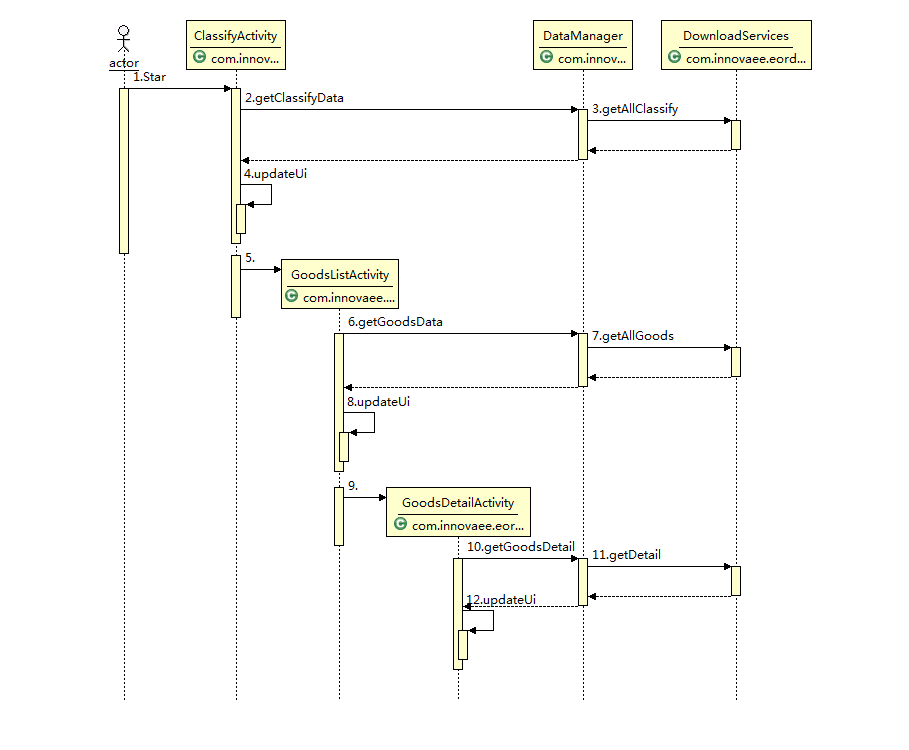
3. 用户点击菜单屏幕的上下滑动翻动菜单页

4. 用户点击菜单左上角的菜单分类筛选器直接跳转到该菜品分类的第一页

### 菜品浏览类图



### 菜品浏览时序图



### 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 包名 | com.innovaee.eorder.mobile.view | | |
| 类名/接口名 | MainViewActivity | | |
| 父类 | Activity | | |
| 实现接口 | N/A | | |
| 成员变量 | | | |
| 变量名 | 数据类型 | 说明 | |
| feedTypeList | List<FeedType> | 所有分类类型列表，记录分类种类状态 | |
| gridView | GridView | 九宫格显示控件 | |
| goodsAdapter | GoodsAdapter | GridView的数据绑定器 | |
| goodsListData | List<GoodsDataBean> | 当前分类下所有菜品的数据列表 | |
| classifyListData | List<ClassifyDataBean> | 服务器下发的分类类型原始数据 | |
| selectOrderGoods | List<GoodsDataBean> | 已经选中的菜品列表 | |
| actionBar | ActionBar | 自定义ActionBar | |
| orderActionView | View | 订单数量显示view | |
| orderCountView | TextView | 订单数量显示view里数量TextView | |
| feedTypePopup | PopupWindow | 分类选择弹出菜单 | |
| feedTypeName | TextView | ActionBar上显示的当前查看分类 | |
| popupMenuList | ListView | 分类弹出菜单中listview | |
| displayMetrics | DisplayMetrics | 显示工具类 | |
| orderHestoryInput | EditText | 订单记录查询会员号输入编辑器 | |
| handler | Handler | 消息处理Handler | |
| 成员函数 | | | |
| 函数名 | 返回值 | 参数 | 说明 |
| onCreate | 无 | Bundle savedInstanceState | Activity创建函数，系统调用 |
| onCreateOptionsMenu | boolean是否已经处理 | Menu paramMenu | Menu创建函数，系统调用 |
| onOptionsItemSelected | boolean是否已经处理 | Menu paramMenu | Menu点击处理函数，系统调用 |
| initView | 无 | 无 | 初始化view，控件函数 |
| initData | 无 | 无 | 初始化数据，监听函数 |
| initActionBarCustomView | 无 | 无 | 初始化ActionBar自定义View |
| openSetting | 无 | 无 | 进入设置界面 |
| openAbout | 无 | 无 | 进入关于界面 |
| oepnOrderHestory | 无 | int userId | 进入订单记录界面 |
| openMyOrder | 无 | 无 | 进入我的订单界面 |
| changeFeedType | 无 | FeedType paramFeedType | 切换当前的FeedType，并更新显示 |
| initFeedTypePopup | 无 | 无 | 初始化Popmenu菜单 |
| clickOnFeedType | 无 | View paramView | 处理Actionbar中FeedType自定义View点击事件 |
| clickOnOrderHestory | 无 | View paramView | 处理ActionBar中orderHestory自定义View点击事件 |
| updateMyOrderCount | 无 | int count | 更新我的订单数目 |
| updateUi | 无 | 无 | 更新GridView ui |
| updatePopmenuUi | 无 | 无 | 更新Popmenu ui |
| loadClassifyData | 无 | 无 | 加载某个分类菜品列表数据 |
| loadFeedTypeListData | 无 | 无 | 加载所有分类列表信息 |
| changeClassifyToFeedType | List<FeedType> | List<ClassifyDataBean> classifyListData | 转换分类数据为切换分类所需的数据类型 |

## 点菜

### 设计图



### 功能描述

点菜下单

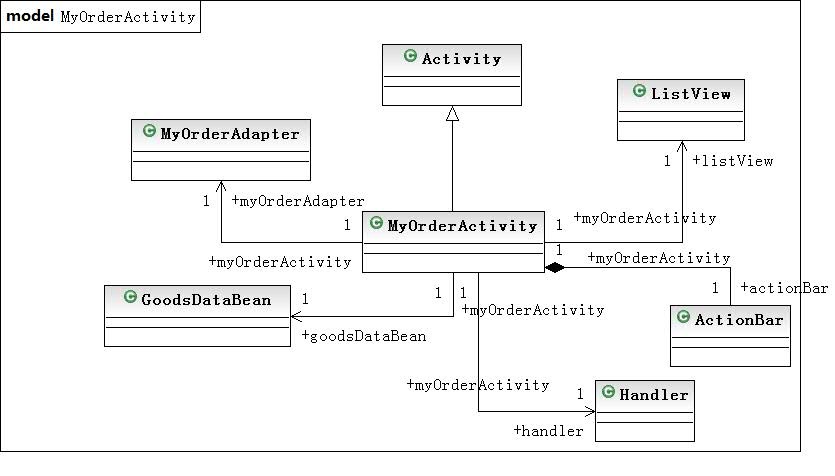
1. 用户单击菜品图片，菜品加入页面下方的购物车图标

2. 用户点击页面上方的购物车图标，进入订单详情页面

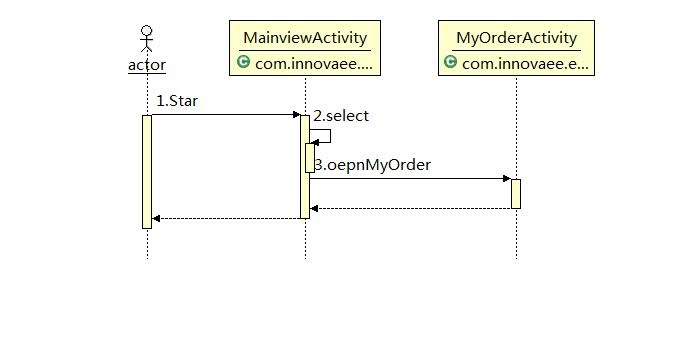
3. 用户上下滑动查看订单详情

4. 用户点击返回按钮回到菜单首页

### 类图



### 时序图



### 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 包名 | com.innovaee.eorder.mobile.view | | |
| 类名/接口名 | MyOrderActivity | | |
| 父类 | Activity | | |
| 实现接口 | N/A  N/A | | |
| 成员变量 | | | |
| 变量名 | 数据类型 | 说明 | |
| selectOrderGoods | List<GoodsDataBean> | 已经选择商品列表 | |
| listView | ListView | 列表显示控件，显示订单项 | |
| myOrderAdapter | MyOrderAdapter | 订单数据绑定器 | |
| actionBar | ActionBar | ActionBar显示器 | |
| allPirce | TextView | 总价格显示框 | |
| delAllBtn | Button | 全部删除按钮 | |
| cancelBtn | Button | 取消返回按钮 | |
| mHandler | Handler | 消息接收Handler | |
| 成员函数 | | | |
| 函数名 | 返回值 | 参数 | 说明 |
| onCreate | 无 | Bundle savedInstanceState | Activity创建函数，系统调用 |
| onOptionsItemSelected | boolean是否处理 | MenuItem item | ActionBar上的按键消息处理函数，系统调用 |
| initView | 无 | 无 | 初始化控件View |
| initData | 无 | 无 | 初始化控件数据，监听器 |
| getAllPrice | Double总价 | 无 | 得到当前订单的总价 |

## 下单

### 设计图



### 功能描述

1. 用户点击订单详情页面的下单按钮，系统显示订单附加信息页面

2. 用户点击订单附加信息中桌号对应的下拉框选择桌号

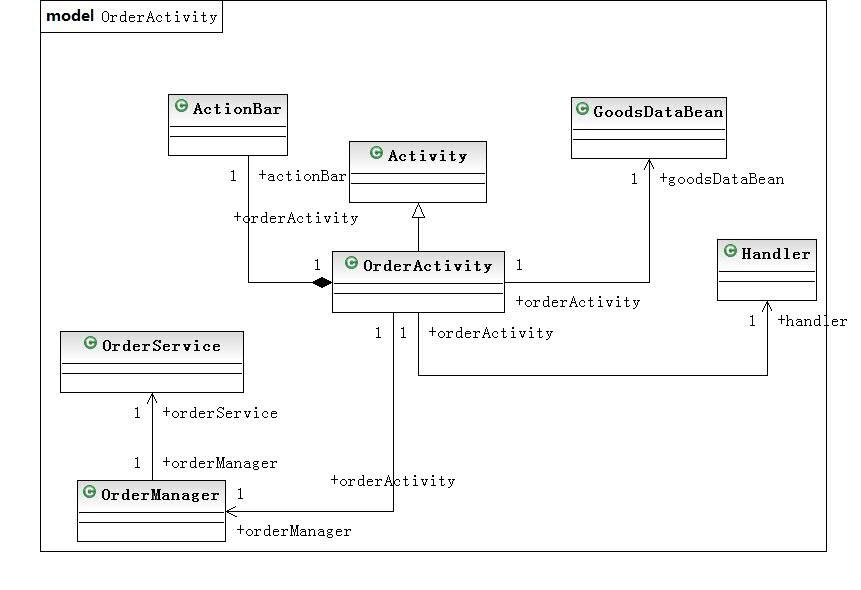
3. 用户点击输入员工号对应的的照相机图标进行的员工号对应的二维码扫描到订单

4. 用户在订单信息页面会员号输入框内输入顾客会员号，查看会员等级对应的折扣信息，系统显示会员信息和打折信息

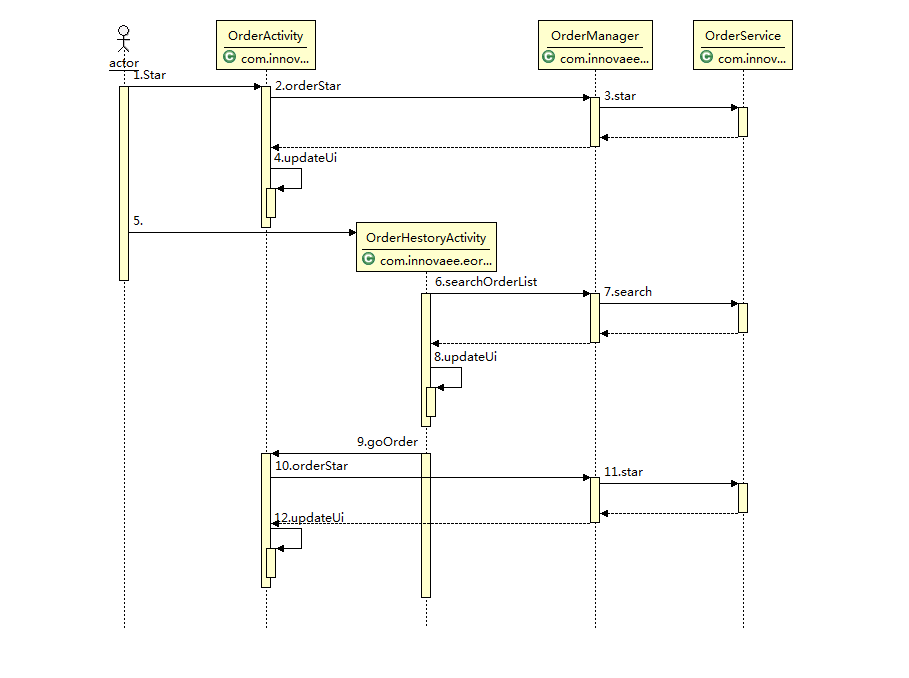
5. 点击下单按钮，系统生成订单

6. 系统显示Msg401

### 类图



### 时序图



### 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 包名 | com.innovaee.eorder.mobile.view | | |
| 类名/接口名 | N/A | | |
| 父类 | Activity | | |
| 实现接口 | N/A | | |
| 成员变量 | | | |
| 变量名 | 数据类型 | 说明 | |
| emplyeeId | TextView | 显示员工工号 | |
| userId | TextView | 显示顾客会员号 | |
| originalPrice | TextView | 显示订单原价 | |
| discount | TextView | 显示折扣 | |
| actualPrice | TextView | 真实价格 | |
| listView | ListView | 订单列表显示ListView | |
| orderAdapter | MyOrderAdapter | 订单数据绑定器 | |
| tableNumSpinner | Spinner | 桌号选择器 | |
| selectOrderGoods | List<GoodsDataBean> | 已经选择商品列表 | |
| 成员函数 | | | |
| 函数名 | 返回值 | 参数 | 说明 |
| onCreate | 无 | Bundle savedInstanceState | Activity创建函数，系统调用 |
| onOptionsItemSelected | boolean是否处理 | MenuItem item | ActionBar上的按键消息处理函数，系统调用 |
| initView | 无 | 无 | 初始化控件View |
| initData | 无 | 无 | 初始化控件数据，监听器 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 历史订单查询

### 设计图



### 功能描述

历史订单查询

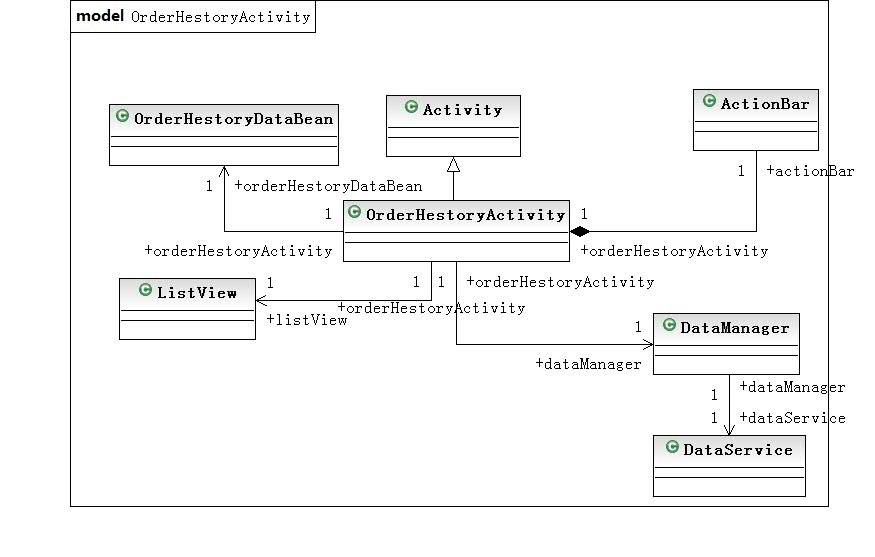
1. 用户在菜单首页的上方的会员号对应的文本框内输入会员号

2. 点击历史订单按钮

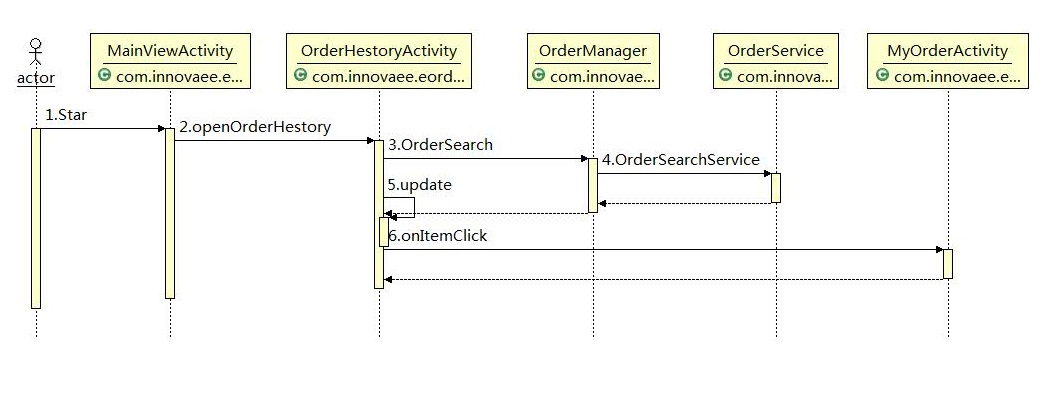
3. 系统显示用户点菜历史订单

4. 用户从上往下滑动，查看订单内菜品信息

### 类图



### 时序图



### 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 包名 | com.innovaee.eorder.mobile.view | | |
| 类名/接口名 | OrderHestory | | |
| 父类 | Activity | | |
| 实现接口 | N/A | | |
| 成员变量 | | | |
| 变量名 | 数据类型 | 说明 | |
| listView | ListView | 历史订单列表ListView | |
| orderHestoryAdapter | OrderHestroyAdapter | 订单数据绑定器 | |
| orderHestoryDataBean | OrderHestoryDataBean | 历史订单记录数据结构 | |
| orderHestoryList | List<OrderHestoryDataBean> | 历史记录订单列表 | |
| 成员函数 | | | |
| 函数名 | 返回值 | 参数 | 说明 |
| onCreate | 无 | Bundle savedInstanceState | Activity创建函数，系统调用 |
| onOptionsItemSelected | boolean是否处理 | MenuItem item | ActionBar上的按键消息处理函数，系统调用 |
| initView | 无 | 无 | 初始化控件View |
| initData | 无 | 无 | 初始化控件数据，监听器 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 存储数据设计

## 应用设置

本程序客户端只需要存储简单数据到手机端，比如手机的设置项等等简单参数，所以采用Android中应用比较广泛，轻量级的存储类SharedPreferences．

在Android系统中，这些信息以XML文件的形式保存在 /data/data/　PACKAGE\_NAME /shared\_prefs 目录下。

存储实现方式：

1. 打开Preferences，名称为setting，如果存在则打开它，否则创建新的Preferences

SharedPreferences settings = getSharedPreferences(“setting”, 0);

2. 让setting处于编辑状态

SharedPreferences.Editor editor = settings.edit();

3. 存放数据

editor.putString(“name”, ”xxxxx”);

editor.putString(“URL”, ”xxxxx”);

4. 完成提交

editor.commit();

存储实现方式：

1. 获取Preferences

SharedPreferences settings = getSharedPreferences(“setting”, 0);

2. 取出数据

String name = settings.getString(“name”, ”默认值”);

String url = setting.getString(“URL”, ”default”);

以上就是Android中SharedPreferences的使用方法，其中创建的Preferences文件存放位置可以在Eclipse中查看：

DDMS->File Explorer /<package name>/shared\_prefs/setting.xml

# 数据缓存设计

## 功能描述

网络应用通常会采用缓存技术，可以进一步大大缓解数据交互的压力，又能提供一定的离线浏览，缓存一般适合下面这样的应用．

1. 提供网络服务的应用

2. 数据更新不需要实时更新，哪怕是3-5分钟的延迟也是可以采用缓存机制。

3. 缓存的过期时间是可以接受的

这样所带来的好处：

1. 减小服务器的压力

2. 提高客户端的响应速度

3. 一定程度上支持离线浏览．

## 实现方式

本系统也采用了缓存技术，因为本客户端主要是对网络图片的读取比较频繁，需要不断的刷新和加载图片，所以保存网络下载的图片比较重要．客户端采用了对图片进行缓存的技术，将从网络端下载的图片保存缓存和本地sdcard上，使用的时候会先去缓存获取图片，如果获取不到图片再去sdcard获取相应的图片．

实现类如下：

private static final int DEFAULT\_SIZE = 5 \* 1024 \* 1024;

/\*\*

\* SDcard storage path

\*/

private final static String STORAGE\_PATH = Environment

.getExternalStorageDirectory().getPath()

+ "/eOrderMobile";

/\*\*

\* Soft cache for bitmaps kicked out of hard cache

\*/

// private static LruCache<String, SoftReference<Bitmap>> softBitmapCache =

// new LruCache<String, SoftReference<Bitmap>>(

// 5);

private static LruImageCache softBitmapCache = new LruImageCache(

DEFAULT\_SIZE);

private static ImageDataManager instance = null;

＊＊＊＊＊＊

｝

## 函数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 包名 | com.innovaee.eorder.mobile.util | | |
| 类名/接口名 | ImageDataManager | | |
| 父类 | N/A | | |
| 实现接口 | OnImageLoaderListener | | |
| 成员变量 | | | |
| 变量名 | 数据类型 | 说明 | |
| softBitmapCache | LruImageCache | 图片缓存 | |
| imageThreadPool | ThreadPoolExecutor | 下载Image的线程池 | |
| activeQueue | HashMap<String, ImageDownloadTask> | 活跃任务队列 | |
| waitTasksQueue | Queue<Runnable> | 被reject的任务队列 | |
| rejectedExecutionHandler | RejectedExecutionHandler | 任务被拒绝执行的处理器 | |
| scheduledRunnable | ScheduledRunnable | 调度Runnable | |
| scheduledExecutorService | ScheduledExecutorService | 调度线程池 | |
| 成员函数 | | | |
| 函数名 | 返回值 | 参数 | 说明 |
| getThreadPool | ThreadPoolExecutor | 无 | 获取线程池的方法，因为涉及到并发的问题，我们加上同步锁 |
| downloadImage | 无 | String url, int hashCode,  OnImageLoaderListener listener | 先从内存缓存中获取Bitmap,如果没有就从SD卡或者手机缓存中获取，SD卡或者手机缓存 没有就去下载 |
| cancelTask | 无 | 无 | 取消正在下载的任务 |
| getBitmapFromCache | Bitmap | String url | 从缓存获取图片 |
| getBitmapFromSdCard | Bitmap | String url | 从sdcard获取图片 |
| saveBmpToSd | 无 | String url, Bitmap bitmap | 保存图片到sdcard |
| hideMedia | 无 | String folder | 在媒体库中隐藏文件夹内的媒体文件 |
| autoCleanCache | 无 | String dirPath | 清理缓存 |
| isSdExits | boolean | 无 | 判断sdcard是否存在 |
| getSdAvailable | int | 无 | 得到sdcard可用空间 |
| initRejectedExecutionHandler | 无 | 无 | 初始化任务被拒绝执行的处理器的方法 |
| hasMoreWaitTask | boolean | 无 | 是否还有等待任务的方法 |
| removeTask | boolean | int hashCode, String url | 删除任务 |
| destory() | 无 | 无 | 释放函数 |
|  |  |  |  |