

# MEMO 软件设计文档

组员：王宝亿、赵凯旋、蔡思婷、廖谊、牛晶晶

交付日期：二〇一七年六月

## 一、 软件概述

Memo 取自 memory，是一款为用户记录生活备忘事项的 APP，它可以追踪用户创建的事项的进度，在事件快完成或刚好结束时提醒用户，并可在不同设备导入用户数据。画面清新可爱，操作简单易上手；是一款实用有效的记录并追踪事项的小工具。

## 二、 软件功能

### a) 主要功能介绍

主要包括用户注册和登录、选择事项分类、新建事项、根据具体内容选择开始时间和结束时间或者当前状态、选择事项的标记图标、保存相关设置、主界面标题栏实时更新当日天气、同步本地和服务器的数据。

### b) 主要页面及页面模块

桌面图标如下：



登录界面（左图），注册界面（右图）：

Memo



请输入用户名

请输入密码

登录

[立即注册>>](#)

Memo

请输入用户名

请输入密码

请再次输入密码

注册

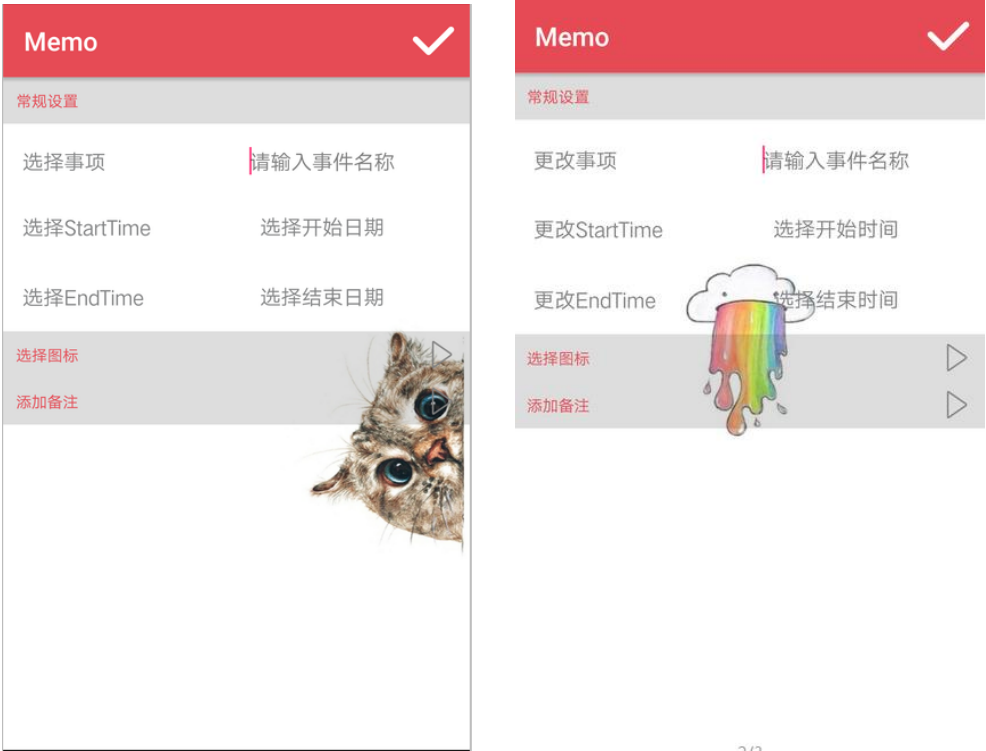
欢迎界面（左图），主界面（右图）：



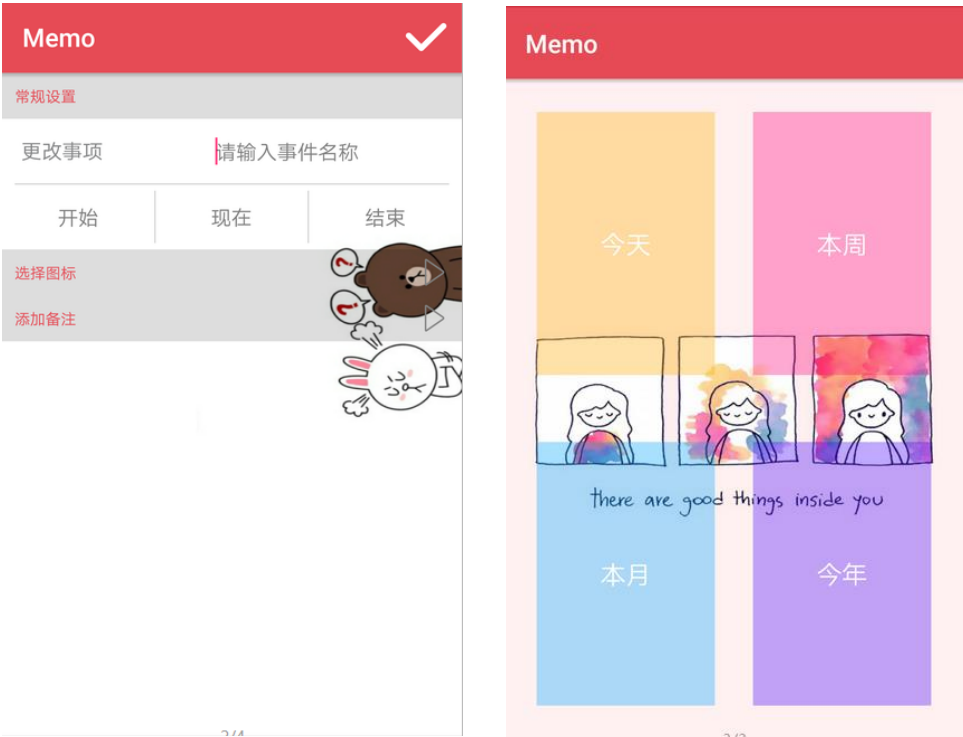
修改头像界面（左图），选择事项分类界面（右图）：



新建“日期”界面（左图），新建“时间”界面（右图）：



新建“私人订制”界面（左图），新建“demo”界面（右图）：



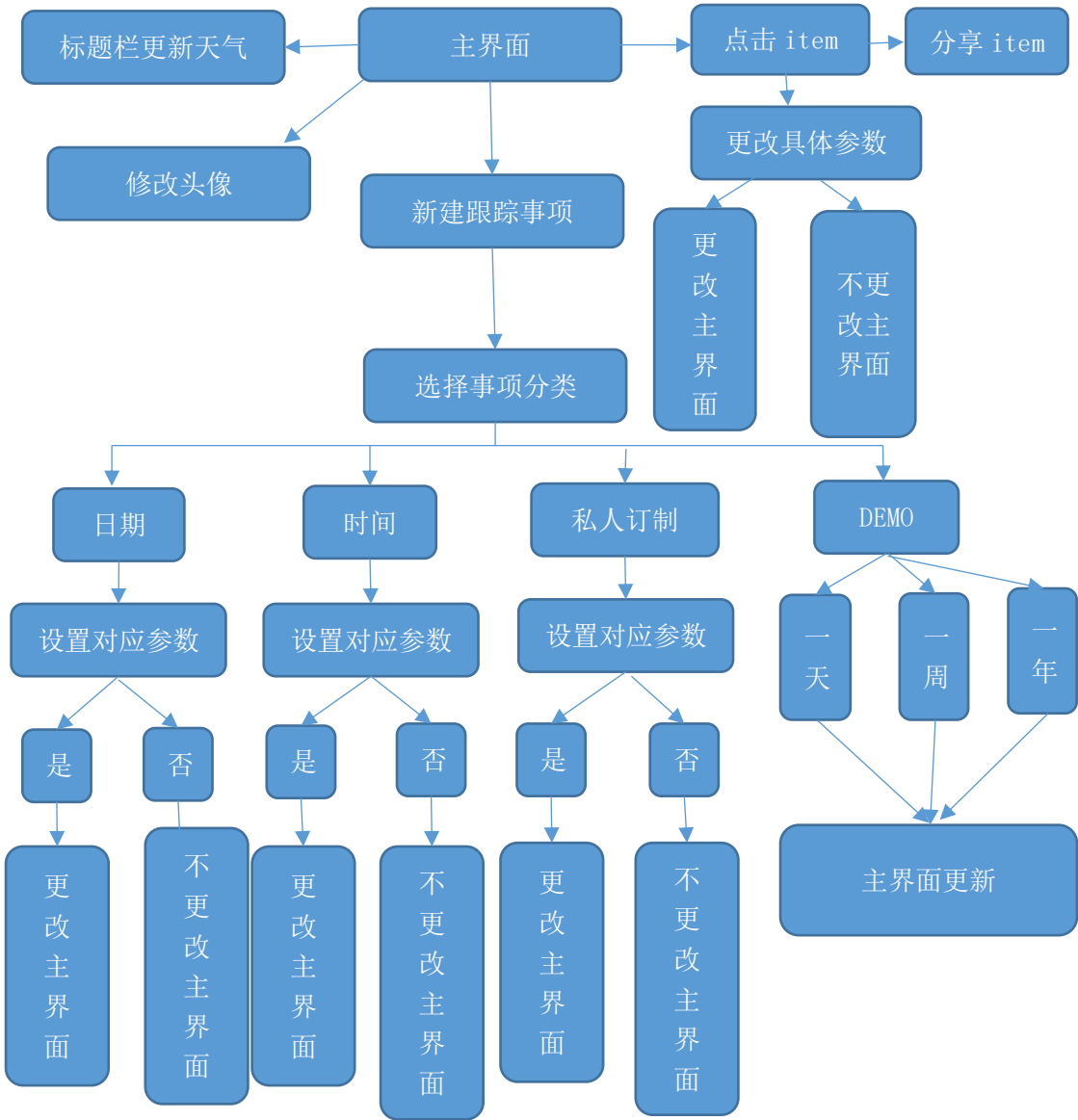
[illegible]

The image displays two screenshots from the mobile game 'Angry Birds'.

The left screenshot shows the '本周' (This Week) progress screen. At the top, there is a profile icon and the text 'DT'. The progress bar is at 71%, with a cake icon and a speech bubble indicating '71%'. Below this, the '时间' (Time) section shows a progress bar at 77%. A white confirmation dialog box is overlaid in the center, asking '是否删除' (Do you want to delete?). The dialog has two buttons: '否' (No) in green and '是' (Yes) in blue. At the bottom, the '私人订制' (Personalized) section shows a progress bar at 20%, with a pig icon and a speech bubble indicating '20%'.

The right screenshot shows a grid of app icons. The icons are arranged in a 3x3 grid. The icons are: 电子邮件 (Email), 蓝牙 (Bluetooth), 信息 (Messages), 百度网盘 (Baidu Netdisk), 闪电互传 (Lightning Transfer), 保存到记事本 (Save to Notepad), 保存到QQ收藏 (Save to QQ Favorites), 保存为私密文章 (Save as Private Article), 发送到去哪儿好友 (Send to Qunar Friends), and 取消 (Cancel) at the bottom.

三、 设计流程



#### 四、 采用的技术，方法，工具

##### 1. 技术和方法

###### 1) UI 设计

主界面标题栏分别为头像，天气，‘+’，点击头像可选择本地图片更换头像，天气通过网络访问 6h 更新一次，以图标形式显示，‘+’ 点击可以创建新跟踪事项；以 listview 显示所有记录事项，每个 item 分别显示事项名称，progressbar 记录事项的进度，并且进度条的以图标+气泡显示具体进度值，兼顾了实用性和趣味性。进度条两端显示开始时间和结束时间。

事项分类界面采用 listview 表示具体事项分类，每个 item 显示事项分类以及简单描述。点击某个 item 可以创建对应类型追踪事项。

每种事项涉及了符合其特点的参数设定界面，具体见界面部分，分别可以设置事项名称，开始结束时间，以及根据自己喜好选择标记事项的图标。

图标选择部分使用了一个 ScrollView 嵌套 LinearLayout，LinearLayout 中再嵌套一个 TableLayout，其中有数个 TableRow，每个都包含四个 ImageView。为了防止 ImageView 中图标图片变形，将图标图片设为 ImageView 的 src。使用 ScrollView 的目的是为了能够使其上下滚动，而 TableLayout 不能放入 ScrollView 里面，需要在其外面套一个 LinearLayout。为了显示某个图标被选定，当某个 ImageView 被选中时，它的 background 被设为一个更深的颜色。

###### 2) 事件处理（点击事件，activity 跳转）



主界面标题栏 ‘+’ 点击跳转到事项分类界面，点击具体项进入该项参数设置界面，依次设定具体参数，设定后点击标题栏右侧 ‘√’ 保存相关属性，并跳转到主界面更新主界面信息。

点击主界面 item 项，跳转到该事项参数设置界面，可以改动相应参数（时间和图标），保存返回主界面并更新主界面相关信息；不保存返回主界面则相应信息不改变。长按主界面 item 项，弹出对话框 ‘是否删除该事项’，选择确定则删除该项，返回主界面更新相关信息，否则保留该项，用户若在联网的情况下完成这些操作后会自动同步到服务器端。

特别的，DEMO 选项中设定了 ‘天’ ‘周’ ‘月’ ‘年’ 几个选项，点击具体选项，选择图标可以在主界面生成对应项，不用设置相应的时间参数。


### 3) SQLite 数据存储结构

每一个事项在数据库中存储了 6 个内容，分别是事项的名称，事项的开始时间、结束时间，该事项对应的小图标、事项的进度和时间戳。以事项的名称作为主键，在插入一个事项时首先检查是否有重复事项，有重复则弹出 Toast 提示 “事件名称重复啦，请核查”，没有重复则在数据库插入该事项。在主界面点击每个事项，可更改事项的开始时间、结束时间和对应小图标，若长按一个事件，可进行删除操作。

### 4) Broadcast & Notificaion

针对 “日期” 和 “时间” 做了提醒设置，对应 “日期” 的提醒在截止日期前 1 天晚上 8 点提醒，对应 “时间” 的提醒在截止时间提醒，发出广播以 Notification 的形式在通知栏中显示将要到期的事项。

## 5) 网络访问

主界面标题栏右侧通过百度 API 获取对应的天气数据，并以图标的形式进行显示。天气类型主要分为晴，雨，风，雪；默认显示图标 。

PS：由于采用 menu 实现标题栏，长按图标时会出现白色矩形（微信标题栏长按也会出现白色矩形框）；避免用户操作不当出现白色矩形，影响整体美观，依次做出以下纠正：长按天气图标时矩形框内显示天气类型；点击‘+’时，矩形框内显示文字提醒‘增加’；点击‘√’，矩形框内显示文字题型‘保存’。

## 6) 服务器端与本地数据库的交互

用户首次使用必须登录，输入用户名和密码在服务器数据库进行匹配，若匹配成功则进入主界面，此时软件会对比本地数据库和服务器端传回信息，同步本地数据库和服务器端数据库，实现用户在不同设备上可以导入自己的信息的功能。之后用户无需再登录便可直接使用软件。而用户对数据库的操作会在联网时自动同步至服务器端

## 2. 所采用的工具：

- 1) 采用 AndroidStudio 进行开发
- 2) 采用了 GitHub 上的开源控件 **BubbleTextView**，即带箭头的 Android 气泡 TextView，支持在布局中通过自定义属性或代码进行丰富的定制，比如：
  - ① 自定义箭头朝向：上、下、左、右、无箭头
  - ② 自动指向目标气泡

③ 填充颜色、填充 Padding、边框、边框颜色、圆角等等

链接：<https://github.com/cpiz/BubbleView>

3) 采用 GitHub 上的开源控件 CircleImageView 显示用户头像：

链接：<https://github.com/hdodenhof/CircleImageView>

## 五、 参考资料

BubbleView - 弹出视图 (Popup View)

<http://www.jcodecraeer.com/a/opensource/2016/0604/4335.html>

【android】ImageView 的 src 和 background 的区别以及两者的妙用

<http://www.tuicool.com/articles/M3Ujmq>

Android 实现 APP 欢迎页面简单制作思路

<http://www.jb51.net/article/91040.htm>

Android 中 GridView、ListView 的 getChildAt 方法返回 null

<http://xie2010.blog.163.com/blog/static/211317365201402395944633/>

安卓开发学习之 Menu

<http://www.cnblogs.com/zhuangshq/p/5684287.html>

Android menu 按键的监听

<http://www.open-open.com/lib/view/open1348151768818.html>

Android 入门开发之 Handler 使用

<http://blog.csdn.net/newjerryj/article/details/8740457>

关于手机软键盘弹出挤压界面的问题

<http://blog.csdn.net/aiynmimi/article/details/50971283>

## 六、 遇到的问题及解决方法

1. 每个 item 的 progressBar 都有用户自己选择的图标和显示进度的气泡悬停在当前进度处。

查阅了很多资料如何写出一个带有文字的气泡，最后在 GitHub 上找到了一个开源的 BubbleTextView 控件，兼具气泡样式和 TextView 的属性。设置 BubbleTextView 的长度、宽度、字体大小、字体颜色、文字内容则调用同 item 内的 progressBar 的 **getProgress()** 获取其进度，指向下面显示图标的 ImageView，箭头方向向下，即完成 BubbleTextView 的设置。

采用图标的大小并不统一，因此要将它们缩放到同样的尺寸，设置其 minHeight 和 maxHeight 相同即可。而 imageView 的 background 并不会保持图片比例，因此设置其 src 为我们提供的图标图像。

为了使 ImageView 和 BubbleTextView 能够跟随当前进度进行移动，在处理 listView 的 Adapter 中调用它们的 **setX()** 函数，而传入其中的参数为：

$$\text{process\_bar.getX()} + \text{process\_bar.getProgress()} * \text{process\_bar.getWidth()} / 100 - \text{img.getWidth()} / 2$$

这样可以使 ImageView 和 BubbleTextView 的中点的 x 坐标都在 progressBar 当前进度处。

2. 主界面 item 背景色渐变。

为了保证当某个 item 删除时，其他 item 还能保持整体渐变而不是缺少一块颜色。我们没有选择将颜色信息存入 item 对应的数据库中，而是创建了一个保存了渐变颜色的 int 数组，命名为 color。再在 Adapter 中分别对每个 item

调用 setBackgroundColor()函数，传入的参数为：

**i (该 item 在整个 list 中的序号) % color.length**

由于删除 item 后，其他 item 的序号会随之发生改变，填补删除 item 产生的空缺。因此对应的背景颜色也会随之发生改变，填补空缺的那一块。

### 3. 创建 item 界面下用户自己选择该 item 对应的图标。

图标选择部分使用了 ScrollView 嵌套 LinearLayout，LinearLayout 中再嵌套一个 TableLayout 的方法实现。TableLayout 中有数个 TableRow，每个都包含四个 ImageView

为了使 ImageView 中图标保持相同大小，并且保持比例不变形，将其设为 ImageView 的 src 而不是 background。



使用 ScrollView 的目的是为了能够使该图标选择界面可以上下滚动，而 TableLayout 不能放入 ScrollView 里面，需要在其外面套一个 LinearLayout。

当某个图标被选定时，我们在代码部分记录了它的序号，但是用户并不能知道是否已经完成选定，为了使界面对用户更加友好，我们作了如下修改：

因此当某个 ImageView 被选定时，我们实时修改了界面。将它的 background 设为 #22000000（带透明度的黑色），使其比周围的底色更深一点以突出它被选中。

### 4. 图标选择界面默认隐藏(visibility=gone)，点击才会出现(模拟下拉菜单效果)。

图标选择界面分为两部分，标题的图标选择部分（RelativeLayout）和一个装有图标 ScrollView 部分。图标选择部分在右侧添加了一个 ImageView，

其 src 默认为 ，表示图标部分被隐藏。单击后 ScrollView 的 visibility 被设为 visible，并且 ImageView 的 src 会更改为 ，表示图标部分被显示在下面。

最开始有了这个模拟下拉菜单的想法，将 ImageView 的 clickable 设为了 true，再添加了 ImageView 的点击函数。后来发现 ImageView 太小了，有时候并不能点到它，要多点几下。

随后将 ImageView 的点击事件改为了 RelativeLayout 的点击事件，整个 RelativeLayout 被点击时，ScrollView 出现，并且修改 ImageView 的 src。

然后发现点击到 ImageView 上时，无响应。而点击 ImageView 以外的部分，功能正常。发现要将 ImageView 的 clickable 参数设为 false，否则点击它会触发 ImageView 的点击事件（空），而不是触发 RelativeLayout 的点击事件。

## 5. 添加 item 界面设置了背景后，点击 EditText 出现的软键盘会挤压界面导致背景变形

最初版的背景是纯色，后来感觉太单调了，于是找了一些图片作为背景，并且每个添加界面的背景都不同，让应用看起来更加美观。

然后发现在点击了 EditText，出现了软键盘后，它会挤压整个界面，导致背景图片变形。背景图片是在 LinearLayout 上设置的 background，不会保持尺寸。查阅了资料以后，得到了如下的结论：

输入框偏上时，弹出软键盘一般情况下不会挤压界面。但是如果界面纵向使用的是 layout\_weight 即比重来分布页面的话，软键盘会压缩整个窗口。

解决方法：

在 Manifest 中对 activity 设置：

```
android:windowSoftInputMode="adjustPan"
```

这样键盘不会压缩原窗口，只会遮盖下面一部分内容。这样背景就不会被压缩导致变形了。

## 6. 设置每个事件的进度到 75%的时候发送广播。

为了使每个事件进度到达 75%时发送广播，需要遍历所有事件，获取进度条的进度，一开始使用 getChildAt 方法，发现这个方法只能返回当前屏幕可视范围内的 view。比如屏幕只能容下 6 个，则 getChildAt (0)到 getChildAt (5)都是可用的，大于 5 就会返回空值。

上网查找资料发现很多都说 setAdapter 后马上 getChildAt 返回子 View，UI 还没更新完，要用 handler.postDelayed 方法延迟一下。但是我试了一下发现还是返回 null。

后面在网上终于找到了答案，其实是对 getChildAt 这个方法理解错了，一开始以为传入一个绝对的 position（就是 adapter 的第几个 item）就可以返回该 position 的 View。但是 GridView 和 ListView 对 View 采用回收机制，即如果屏幕最多可以显示 n 个子 View，那么内存中其实只有 n 个 View，当我们在滚动时，第(n+1)个 View 复用第 1 个 View，依次类推。

因此最后采用 AffairList.get(i)，获取到该事件后再获取 progress，成功解决问题。

## 7. 前端与后端的数据交互

实验过程中基本做到了前后端分离开发，但是数据格式的一致性非常重要，采用 post 方法提交数据的过程中，因为后台数据格式的问题卡了很久，数据格式就是简单的参数组装，并不是常用的 JSON 格式，一开始直接多次传数据发现行不通，最后在一博客中找到了解决方法，将每个参数用&连接起来组装成字符串发送即可

#### 8. 长按标题栏的“天气”等图标会出现小框

一开始致力于使小框不出现，冥思苦想了很久，也找了很多资料，发现解决方法都非常复杂，需要重写标题栏的 layout 文件，而使用 menu 来给标题栏布局似乎没有办法使小框不出现。

最后玩微信随意长按了一下图标，发现也有这个问题！于是学习了微信的解决方案，在小框中添加了对该图标的文字说明，解决了这个细节问题。

#### 9. 为适应 4 类事件不断修改数据存储结构

一开始数据库中针对每个事件只存储了该事件的名称，开始时间，结束时间和进度，后来发现在更新进度时需要对每一类事件进行不同的处理，并且需要保存每个事件的小图标。

因此又重新修改了数据库的内容，增加了一个标志位表示该事件属于四类中的哪一类，并为图标设置一个标记，再在自定义 Adapter 中进行不同处理。

#### 10. 多用户问题

软件默认条件是单一用户不退出登录的情况，因为是从原有项目 DT 中移



植过来的产物，原有项目是不考虑用户的，因而本地数据库中直接采用事项名称做主键。而考虑用户后，本地数据库修改起来很麻烦，因为需要通过用户名和事件名才能定位到一条记录，因此没有修改本地数据库。而服务器端是可以考虑多用户情况的。本地数据库和服务器端数据库存在的矛盾是：本地若登录多个用户，会自动导入多个用户的事件，造成本地数据库的事项紊乱；此外若用户退出登录，还是可以继续使用 APP 添加或删除事件的，而这个事件未绑定执行者也会造成本地数据库的紊乱，这个问题的解决方法是完全依赖网络实现导入数据，或者彻底修改本地数据库