|  |  |  |  |
| --- | --- | --- | --- |
| 文件编号 | #0001 | 版 本 号 | 1.3 |
| 编 制 人 |  | 编制日期 |  |

**“家谱树”**

**软件设计规格说明书**

**V1.3**

**北京邮电大学**

**小组：第八组**

评审日期： 2016年6月3日

目录

[1．导言 2](#_Toc452913247)

[1.1 编写目的 2](#_Toc452913248)

[1.2 项目范围 3](#_Toc452913249)

[1.3 参考资料 3](#_Toc452913250)

[1.4 版本更新信息 3](#_Toc452913251)

[2．项目介绍 4](#_Toc452913252)

[2.1 项目来源及背景 4](#_Toc452913253)

[2.2 项目目标 4](#_Toc452913254)

[3．软件层次架构设计 4](#_Toc452913255)

[3.1 设计原则 4](#_Toc452913256)

[3.2 体系结构设计 5](#_Toc452913257)

[4 功能模块设计 6](#_Toc452913258)

[4.1 模块一：登录 7](#_Toc452913259)

[4.2 模块二：注册 7](#_Toc452913260)

[4.3 模块三：查看信息 8](#_Toc452913261)

[4.4 模块四：编辑信息 8](#_Toc452913262)

[4.5 模块五：添加成员 8](#_Toc452913263)

[4.6 模块六：绑定手机 9](#_Toc452913264)

[4.7 模块七：联系成员 9](#_Toc452913265)

[4.8 模块八：发布“家谱树” 9](#_Toc452913266)

[4.9 模块九：发布动态 10](#_Toc452913267)

[4.10 模块十：点 赞 及评论 10](#_Toc452913268)

[4.11 模块十一：查看用户信息 10](#_Toc452913269)

[4.12 模块十二：消除好友关系 11](#_Toc452913270)

[4.13 模块十三：添加好友 11](#_Toc452913271)

[4.14 模块十四：意见反馈 12](#_Toc452913272)

[4.15 模块十五：检查更新 12](#_Toc452913273)

[5 数据库设计 12](#_Toc452913274)

[5.1 数据库选择 12](#_Toc452913275)

[5.2 数据库逻辑结构 12](#_Toc452913276)

[5.3 数据库物理结构 13](#_Toc452913277)

[5.3.1 表1：user 14](#_Toc452913278)

[5.3.2 表2：message 15](#_Toc452913279)

[5.3.3 表3：administrator 15](#_Toc452913280)

[5.3.4 表4：comment 16](#_Toc452913281)

[5.3.5 表5：friend 16](#_Toc452913282)

[5.3.6 表6：member 16](#_Toc452913283)

[5.3.7 表7：suggest 17](#_Toc452913284)

[5.3.8 表8：praise 17](#_Toc452913285)

[5.3.9 表9：story 18](#_Toc452913286)

[6 用户界面设计 18](#_Toc452913287)

[6.1 色彩设计 18](#_Toc452913288)

[6.2 设计线图 19](#_Toc452913289)

[6.3 原型设计 19](#_Toc452913290)

# 1．导言

## 1.1 编写目的

该文档是关于用户对于“家谱树”手机软件的功能和性能的要求，将作为对该项目在概要设计阶段的输入。

本文档的预期读者包括：

* 设计开发人员
* 项目管理人员
* 测试人员
* 用户

## 1.2 项目范围

该文档的目的是解决在项目系统“做什么”的基础之上对每个模块的功能性说明、接口介绍与其后的行为解释。在这里，对于开发技术进行涉及，而主要是通过建立模型与语言描述的方式给开发者一个开发过程中的指导。

## 1.3 参考资料

[1] 《Android 应用开发详解》郭宏志等 电子工业出版社

[2] 《Introduction to the Team Software Process》Watts S.Humphrey Tsinghua University Press

## 1.4 版本更新信息

本文档的更新记录如表格1-1所示。

表格 1－1 版本更新信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2016.6.03 | 0.1 | 全部 | 初始发布版本 |
| 002 | 2016.6.04 | 1.1 | 数据库设计 | 对数据库设计进行修改 |
| 003 | 2016.6.05 | 1.2 | 界面设计 | 对用户界面设计进行修改 |

# 2．项目介绍

## 项目来源及背景

为了实现对家族关系的深刻理解与进一步与家人进行沟通与联系，设计一款基于android的移动手机软件，主要功能是以家谱树的形式将家族关系展示处理，并在“家族圈”这个载体中发布消息，增强亲人之间的交流。

## 2.2 项目目标

本项目设定的目标如下：

1. 软件运用能够提供友好的用户界面与，方便用户进行相关操作。
2. 软件应用将满足“家谱树”类软件的基本要求，同时拓展性地增加了“家族圈”等社交需求。
3. 软件应用具有良好的运行效率，使用户能在短时间内获取交互相应。

# 3．软件层次架构设计

应用环境可以分为网络环境、硬件环境、软件环境。

## 3.1 设计原则

“家谱树”即Family Tree的设计遵循以下的基本原则

（1）实用性

实用性是“家谱树”的主要设计原则，“家谱树”的设计必须最大可能地满足用户的在整理与记录家族关系上的需求，做到操作方便、界面友好、可即时更新，能适应不同层次用户的需求。

（2）稳定性

系统的运行稳定是对“家谱树”的基本要求。

（3）可维护性

在维护方面，主要考虑两个层面，一是对于开发人员来讲，系统编码容易调整，可适应需求的变化和调整；二是对于系统管理维护人员来说，能够对系统进行便捷的维护和管理。

（4）安全性

由于“家谱树”在一定程度上涉及到用户的隐私，因此在安全性方面，要考虑较为全面，比如系统的安全、数据管理的安全、网络安全。尽最大努力保证用户权限、数据安全和系统的稳定性。

（5）单一职责原则

“家谱树”运用在面向对象设计部分采取单一职责原则，其核心思想为：一个类，最好只做一件事，只有一个引起它的变化。单一职责原则可以看做是低耦合、高内聚在面向对象原则上的引申，将职责定义为引起变化的原因，以提高内聚性来减少引起变化的原因。

## 3.2 体系结构设计

为了满足“家谱树”APP响应快速、便于操作、易于维护的要求，在设计实现上，“家谱树”采用MVC的设计模式。

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。其中M层处理数据，业务逻辑等；V层处理界面的显示结果；C层起到桥梁的作用，来控制V层和M层通信以此来达到分离视图显示和业务逻辑层。

“家谱树”的开发过程中采用MVC的架构，采用MVC模式的好处是便于UI界面部分的显示和业务逻辑，数据处理分开。在MVC架构在“家谱树”软件采用如下的运用方式：

M层：适合做一些业务逻辑处理，比如数据库存取操作，网络操作，复杂的算法。较为耗时的任务等都在Model层处理，“家谱树”的主要功能比如存储用户的家庭结构等操作都是在M层进行。 V层：应用层中处理数据显示的部分，XML布局可以视为V层，显示Model层的数据结果，“家谱树”中所有的页面信息都将在这一层中出现。 C层：在Android中，Activity处理用户交互问题，因此可以认为Activity是控制器，Activity读取V视图层的数据（eg.读取当前用户输入的信息，比如EditText控件的数据）并向Model发送数据请求，从而尽心下一步的操作。

# 4 功能模块设计

本系统共有登录注册模块、家族树模块、家族圈模块、通讯录模块和设置模块这5个大模块每个大模块可细分为多个小模块，如下图所示：





## 模块一：登录

* 功能:实现登录功能,输入用户名称和用户密码,正确则进入主界面.不正确则停留在登录界面。
* 性能:为用户提供登录功能。
* 输入项目:用户名称和用户密码。
* 输出项目:正确则进入到欢迎界面,错误则返回留在登录界面。

## 4.2 模块二：注册

* 功能:实现注册功能,输入新的用户名称和用户密码,若成功则注册一个新的用户，否则则停留在注册界面。
* 性能:为用户提供注册功能。
* 输入项目:用户名称和用户密码。
* 输出项目:成功则进入到登录界面,错误则返回留在注册界面。



## 4.3 模块三：查看信息

* 功能:实现查看个人信息功能，用户通过点击“个人信息“来查看某位成员的详细个人信息。
* 性能:为用户提供查看个人信息功能。
* 输入项目:选中某位成员并点击“个人信息“。
* 输出项目:显示被选中成员相关的个人信息。

## 4.4 模块四：编辑信息

* 功能:实现编辑个人信息功能，用户在“个人信息”页面点击“编辑“来编辑某位成员的详细个人信息。
* 性能:为用户提供编辑个人信息功能。
* 输入项目:在某位成员 “个人信息”页面点击“编辑“。
* 输出项目:对选中成员相关的个人信息进行修改。

## 4.5 模块五：添加成员

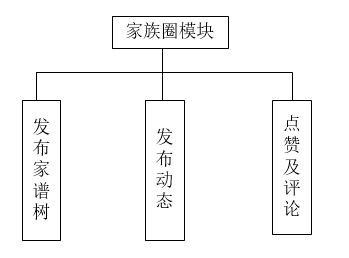
* 功能:实现添加成员功能，用户选中某位成员并点击“添加成员“，便可以为该成员添加一位直系亲属。
* 性能:为用户提供添加成员功能。
* 输入项目: 选中某位成员并点击“添加成员”。
* 输出项目:为选中成员添加一位直系亲属。

## 4.6 模块六：绑定手机

* 功能:实现绑定手机功能，用户选中某位成员并点击“绑定手机“，便可以为该成员绑定通讯录中的对应手机号码。
* 性能:为用户提供绑定成员功能。
* 输入项目: 选中某位成员并点击“绑定手机“。
* 输出项目:为选中成员绑定通讯录中的对应手机号码。

## 4.7 模块七：联系成员

* 功能:实现联系成员功能，用户选中某位成员并点击“联系“，便可以向该成员拨打电话或发送短信。
* 性能:为用户提供联系成员功能。
* 输入项目: 选中某位成员并点击“联系”。
* 输出项目: 向选中成员拨打电话或发送短信。



## 4.8 模块八：发布“家谱树”

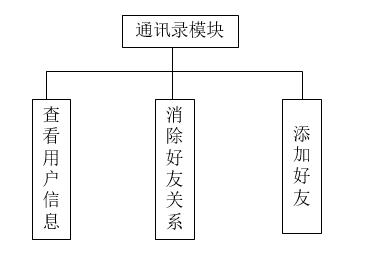
* 功能:实现发布“家谱树”功能，用户点击“发布家谱树”，便可以向“家族圈”发布“家谱树”。
* 性能:为用户提供联系发布“家谱树”功能。
* 输入项目: 点击“发布家谱树”。
* 输出项目: 向“家族圈”发布“家谱树”。

## 4.9 模块九：发布动态

* 功能:实现发布自己的动态信息功能，用户点击“发布动态”，便可以向“家族圈”发布动态信息。
* 性能:为用户提供联系发布动态功能。
* 输入项目: 点击“发布动态”。
* 输出项目: 向“家族圈”发布自己的动态信息功能。

## 4.10 模块十：点 赞 及评论

* 功能:实现对在“家族圈”出现的动态进行点赞和评论的功能，用户选择某条动态点击“点赞”或“评论”，便可以向“家族圈”的动态进行 点 赞或 评论。
* 性能:为用户提供对在“家族圈”出现的动态进行点赞和评论的功能。
* 输入项目: 选择某条动态点击“点赞”或“评论”。
* 输出项目: 向“家族圈”的动态进行 点 赞或 评论。



## 4.11 模块十一：查看用户信息

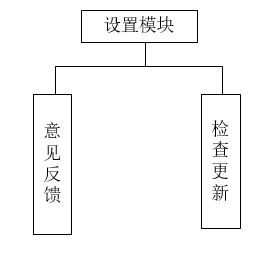
* 功能:实现对在“通讯录”中好友或家人信息查看的功能，用户选择列表中的一个好友或家人点击“查看”，查看他的详细用户信息。
* 性能:为用户提供对在“通讯录”中好友或家人信息查看的功能。
* 输入项目: 选择列表中的一个好友或家人点击“查看”。
* 输出项目: 查看该用户的详细用户信息。

## 4.12 模块十二：消除好友关系

* 功能:实现对在“通讯录”中好友消除好友关系的功能，用户选择列表中的一个好友点击“取消关注”，消除与该用户的好友关系。
* 性能:为用户提供对在“通讯录”中好友消除好友关系的功能。
* 输入项目: 选择列表中的一个好友点击“取消关注”。
* 输出项目: 消除与该用户的好友关系。

## 4.13 模块十三：添加好友

* 功能:实现对非好友账号申请添加好友的功能，用户点击“新的朋友”并输入另一非好友用户的账号来发出好友申请。
* 性能:为用户提供对非好友账号申请添加好友的功能。
* 输入项目: 用户点击“新的朋友”并输入另一非好友用户的账号。
* 输出项目: 向非好友账号发出好友申请。



## 4.14 模块十四：意见反馈

* 功能:实现对制作方反馈自己的使用体验和意见的功能，用户点击“意见反馈”即可向制作方反馈自己的使用体验和意见。
* 性能:为用户提供对制作方反馈自己的使用体验和意见的功能。
* 输入项目: 用户点击“意见反馈”。
* 输出项目: 向制作方反馈自己的使用体验和意见。

## 4.15 模块十五：检查更新

* 功能:实现检查当前软件版本的功能，用户点击“检查更新”即可检查当前软件的版本，若不是最新版本会提示用户是否进行升级。
* 性能:为用户提供检查当前软件版本的功能。
* 输入项目: 用户点击“检查更新”。
* 输出项目: 检查当前软件的版本，若不是最新版本会提示用户是否进行升级。

# 5 数据库设计

## 5.1 数据库选择

本项目采用MySQL数据库。

## 5.2 数据库逻辑结构

本项目的数据库逻辑结构中，有用户、家族消息、管理员、事迹、好友、成员、意见、赞、评论等9个实体。其中各实体的属性为：

* 用户：ID，用户名，密码，手机号，出生地，first name，last name，头像，昵称，性别，排行，生日。
* 家族消息：ID，文本，图片。
* 管理员：ID，用户名，密码。
* 事迹：ID，事迹人ID，内容，发布人ID。
* 好友：ID，用户1ID，用户2ID。
* 成员：ID，出生地，first name，last name，性别，排行，关系，用户ID，生日。
* 意见：ID，文本，用户ID。
* 赞：ID，用户ID，消息ID。
* 评论：ID，用户ID，消息ID，内容。

综上分析，“家谱树”的实体关系图（概念数据模型）如下图5.2-1所示。

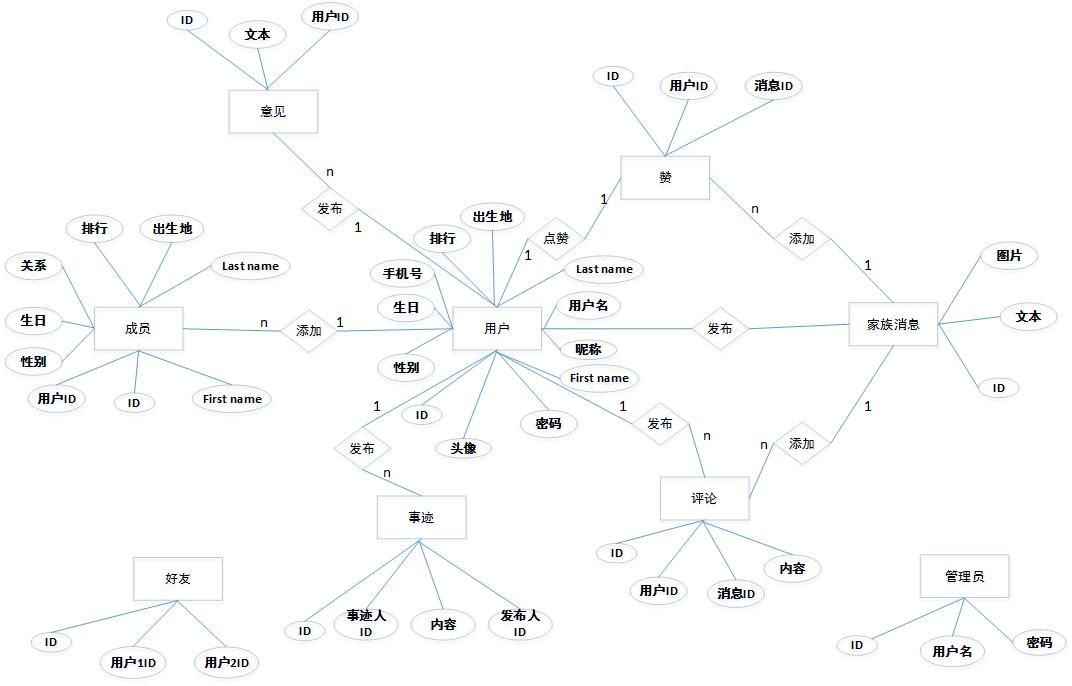


图5.2-1 实体关系图

## 5.3 数据库物理结构

由下图5.3-1物理数据模型图可知，“家谱树”项目数据库中共有user、message、administrator、comment、friend、member、suggest、praise、story等9个表。下面是关于数据库表的详细说明。

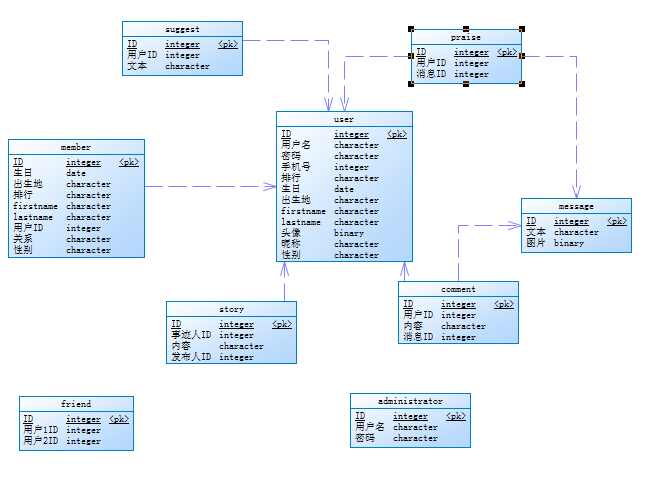


图5.3-1 物理数据模型图

### 5.3.1 表1：user

user表的表结构如表5.3.1-1所示，这个表描述了用户信息，包括用户名、密码、手机号、排行、生日、出生地、firstname、lastname、头像、昵称、性别等。

表5.3.1-1 user表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 用户名 | character (45) | N |  |
| 密码 | character (45) | N |  |
| 手机号 | character (45) | N |  |
| 排行 | character (45) | Y |  |
| 生日 | Date | Y |  |
| 出生地 | character (45) | Y |  |
| Firstname | character (45) | Y |  |
| Lastname | character (45) | Y |  |
| 头像 | binary | Y |  |
| 昵称 | character (45) | Y |  |
| 性别 | character (45) | Y |  |

### 5.3.2 表2：message

message表的表结构如表5.3.2-1所示，这个表描述了用户发布的家族消息，包括文本，图片等。

表5.3.2-1 message表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 文本 | character (45) | Y |  |
| 图片 | binary | Y |  |

### 5.3.3 表3：administrator

administrator表的表结构如表5.3.3-1所示，这个表描述了管理员的基本信息，包括用户名，密码等。

表5.3.3-1 administrator表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 用户名 | character (45) | N |  |
| 密码 | character (45) | N |  |

### 5.3.4 表4：comment

comment表的表结构如表5.3.4-1所示，这个表描述了用户发布的评论消息，包括用户ID，评论内容，所评消息ID等。

表5.3.4-1 comment表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 用户ID | integer | N |  |
| 内容 | character (45) | N |  |
| 消息ID | integer | N |  |

### 5.3.5 表5：friend

friend表的表结构如表5.3.5-1所示，这个表描述了用户的好友信息，包括双方用户的ID等。

表5.3.5-1 friend表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 用户1ID | integer | N |  |
| 用户2ID | integer | N |  |

### 5.3.6 表6：member

member表的表结构如表5.3.6-1所示，这个表描述了用户的家庭成员信息，包括关系、排行、生日、出生地、firstname、lastname、性别、用户ID等。

表5.3.6-1 member表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 排行 | character (45) | Y |  |
| 生日 | Date | Y |  |
| 出生地 | character (45) | Y |  |
| Firstname | character (45) | Y |  |
| Lastname | character (45) | Y |  |
| 关系 | character (45) | Y |  |
| 用户ID | integer | Y |  |
| 性别 | character (45) | Y |  |

### 5.3.7 表7：suggest

suggest表的表结构如表5.3.7-1所示，这个表描述了用户上传的意见，包括用户的ID，文本等。

表5.3.7-1 suggest表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 文本 | character (45) | N |  |
| 用户ID | integer | N |  |

### 5.3.8 表8：praise

praise表的表结构如表5.3.8-1所示，这个表描述了用户点赞的信息，包括用户的ID，所赞的消息ID等。

表5.3.8-1 praise表

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段类型** | **可否为空** | **备注** |
| ID | integer | N | 主键 |
| 消息ID | integer | N |  |
| 用户ID | integer | N |  |

### 5.3.9 表9：story

story表的表结构如表5.3.9-1所示，这个表描述了用户上传的事迹信息，包括上传人的ID，事迹人的ID，事迹内容等。

表5.3.9-1 story表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 可否为空 | 备注 |
| ID | integer | N | 主键 |
| 事迹人ID | integer | N |  |
| 用户ID | integer | N |  |
| 内容 | character (45) | N |  |

# 6 用户界面设计

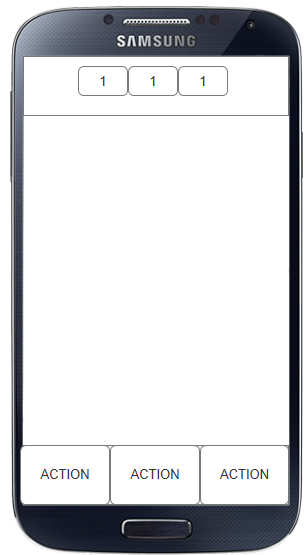
## 6.1 色彩设计

本产品以简约的色彩搭配带给用户醒目的效果。本产品的主背景色为白色，下方功能栏以蓝色勾勒出图形形状显示在白色的背景下，而上方辅助功能选择栏为了给人以明显视觉效果，选择以黑色为底，白色文字和按钮作为视觉反差为主要提示对象。

而家谱树上的人物显示，男士以蓝色为主色调显示在白色背景的底面上，女士则以粉色为主色调显示在白色背景的底面上。

总体上，本产品为了给人以简洁大方、清晰直观的感受，运用黑白分明的配色，以及上下层次清楚的布局，进行产品的设计。

## 6.2 设计线图



主功能选择栏

主要内容显示部分

辅助功能选择栏

图 6-1 设计线图

## 6.3 原型设计

1.家族树页面：

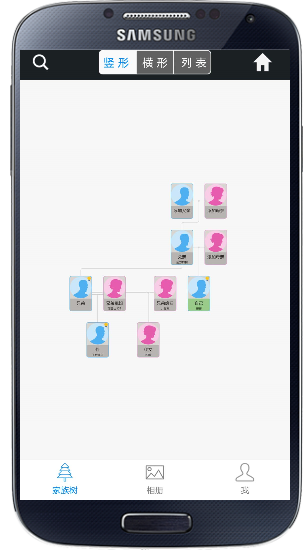


图6-2 家族树-竖形

下方主功能按择为“家族树”，上方辅助功能选择为“竖形”。进入图3-1的界面，在中间可进行人物的创建、删除和修改等功能



图6-3 家族树-列表

下方主功能按择为“家族树”，上方辅助功能选择为“列表”。进入图3-2的界面，在中间可以首字母排序的列表的形式查询想要查找的人物。



图6-4 相册

下方主功能按择为“相册”，上方辅助功能选择为“我家相册”。进入图3-3的界面，在中间可以创建相册分享给家谱中的成员。



图6-5 我

下方主功能按择为“我”。进入图3-4的界面，在中间可以对相关内容进行设置。