软件系统分析与设计

Assignment2



**指导老师：张红延**

**组长：王畅（12301106）**

**组员：白亚楠（12301088）**

**赵家进（12301113）**

**许若昕（12301109）**

**游程（12301112）**

目录

[1. Cover page(封面页) 5](#_Toc406495350)

[1.1 Exeuctive summary（摘要） 5](#_Toc406495351)

[1.2 Introduction（介绍） 5](#_Toc406495352)

[2. Context of the Project Launched（项目运行环境） 6](#_Toc406495353)

[2.1Business case statement（业务案例声明） 6](#_Toc406495354)

[2.1.1 Existing problems(系统存在的问题) 6](#_Toc406495355)

[2.1.2 The purpose of the solution(解决方案的目的) 6](#_Toc406495356)

[2.1.3 System Scope(系统范围) 7](#_Toc406495357)

[2.2Systems Development methodology selected for this project （项目系统发展方法论） 8](#_Toc406495358)

[2.3 Key steps undertaken in the design project（项目设计主要步骤） 8](#_Toc406495359)

[2.4 Members responsibilities in this project（项目小组成员） 9](#_Toc406495360)

[2.5 The project and team dynamics （项目和团队分工） 10](#_Toc406495361)

[3.Understanding the current system（现下系统理解） 11](#_Toc406495362)

[3.1 Scope Model（范围模型） 11](#_Toc406495363)

[3.1.1 Context Model（环境模型） 11](#_Toc406495364)

[3.1.2 Conceptual Model（概念模型） 12](#_Toc406495365)

[3.1.3 Objective-Problem Model（目标模型） 13](#_Toc406495366)

[3.2 Bussiness UseCase Statement（系统用例声明） 14](#_Toc406495367)

[3.2.1 Bussiness use case model（系统用例模型） 14](#_Toc406495368)

[3.2.2 Bussiness use case specification（系统用例陈述） 15](#_Toc406495369)

[3.3 Process Modeling(including events)（流程模型） 20](#_Toc406495370)

[3.3.1 Functional decomposition diagrams(功能分解图) 20](#_Toc406495371)

[3.3.2 Functional decomposition with event(带有事件的功能分解) 21](#_Toc406495372)

[3.3.3 The list of event response(事件响应清单) 22](#_Toc406495373)

[3.3.4 Process model(过程模型) 23](#_Toc406495374)

[3.4 Resource Modeling（资源模型） 25](#_Toc406495375)

[3.4.1 Application interface resource(应用界面资源) 25](#_Toc406495376)

[3.4.2 Information resource(信息资源) 25](#_Toc406495377)

[3.5 Rule Definition（规则定义） 26](#_Toc406495378)

[4.SystemRequirementantalysisandSpecification（系统需求分析和陈述） 27](#_Toc406495379)

[4.1 Functionality Documentation（功能陈述） 27](#_Toc406495380)

[4.1.1 Use case Model（用例模型） 27](#_Toc406495381)

[4.1.2 Specification 28](#_Toc406495382)

[4.2 Non\_functional requirements（非功能需求） 39](#_Toc406495383)

[4.2.1 Functionality(功能性) 39](#_Toc406495384)

[4.2.2 Reliability(可靠性) 40](#_Toc406495385)

[4.2.3 Usability易用性 41](#_Toc406495386)

[4.2.4 Efficiency(效率) 41](#_Toc406495387)

[4.2.5 Maintainability(维护性) 42](#_Toc406495388)

[4.2.6 Portability(可移植性) 43](#_Toc406495389)

[4.3 Global Constrain Definition（总体约束定义） 43](#_Toc406495390)

[4.3.1 Schedule(日程) 43](#_Toc406495391)

[4.3.2 Rule of function(性能规定) 43](#_Toc406495392)

[4.3.3 Scale and cost(规模及成本费用) 43](#_Toc406495393)

[5.Communication Design（通讯设计） 44](#_Toc406495394)

[5.1 Layout Design（布局设计） 44](#_Toc406495395)

[5.1.2 Content Frame（目录框架） 46](#_Toc406495396)

[5.1.3 Story Board & Window Design（界面设计） 46](#_Toc406495397)

[5.2 GUI Navigation Design（界面导航设计） 51](#_Toc406495398)

[5.3 Prototype building（原型构建） 51](#_Toc406495399)

[5.3.1 Register(注册界面) 52](#_Toc406495400)

[5.3.2 login(登录界面) 53](#_Toc406495401)

[5.3.3 chat(聊天界面) 54](#_Toc406495402)

[5.3.4 Family circle(家庭圈界面) 54](#_Toc406495403)

[5.3.5 Commont(评论家庭圈界面) 55](#_Toc406495404)

[5.3.6 Memo(备忘录界面) 56](#_Toc406495405)

[5.3.7 Promotion for game(游戏推广界面) 57](#_Toc406495406)

[5.3.8 Promotion for applications(推广界面) 58](#_Toc406495407)

[5.4 Interface Design（接口设计） 59](#_Toc406495408)

[5.4.1 Interface Specification（接口说明） 59](#_Toc406495409)

[6. General Design: Architecture Design（总体设计） 60](#_Toc406495410)

[6.1 General Design 60](#_Toc406495411)

[6.1.1 User information module用户信息模块 60](#_Toc406495412)

[6.1.2 Chat module(聊天模块) 62](#_Toc406495413)

[6.1.3 Family circle module(家庭圈模块) 63](#_Toc406495414)

[6.1.4 Memo module(备忘录界面) 64](#_Toc406495415)

[6.1.5 Game module(游戏模块) 65](#_Toc406495416)

[6.1.6 Promotion module(推广模块) 66](#_Toc406495417)

[7. detail design（详细设计） 67](#_Toc406495418)

[7.1 Presentation layer(表示层) 67](#_Toc406495419)

[7.1.1 User information module(用户信息模块) 67](#_Toc406495420)

[7.1.2 Chat module(聊天模块) 68](#_Toc406495421)

[7.1.3 Family circle module(家庭圈模块) 69](#_Toc406495422)

[7.1.4 Memo module(备忘录模块) 70](#_Toc406495423)

[7.1.5 Game module(游戏模块) 71](#_Toc406495424)

[7.1.6 Promotion module(推广模块) 72](#_Toc406495425)

[7.2 Control layer(控制层) 73](#_Toc406495426)

[7.2.1 User information module(用户信息模块) 73](#_Toc406495427)

[7.2.2 Chat module(聊天模块) 74](#_Toc406495428)

[7.2.3 Family circle module(家庭圈模块) 75](#_Toc406495429)

[7.2.4 Memo module(备忘录模块) 76](#_Toc406495430)

[7.2.5 Game module(游戏模块) 77](#_Toc406495431)

[7.2.6 Promotion module(推广模块) 78](#_Toc406495432)

[7.3 Data layer(数据层) 78](#_Toc406495433)

[7.3.1 User information module(用户信息模块) 79](#_Toc406495434)

[7.3.2 Chat module(聊天模块) 79](#_Toc406495435)

[7.3.3 Family circle module(家庭圈模块) 80](#_Toc406495436)

[7.3.4 Memo module(备忘录模块) 80](#_Toc406495437)

[7.3.5 Game module(游戏模块) 81](#_Toc406495438)

[7.3.6 Promotion module(推广模块) 82](#_Toc406495439)

[8. Data Modeling and Design（数据模型和设计） 82](#_Toc406495440)

[8.1 Conceptual model（概念模型） 83](#_Toc406495441)

[8.2 Logical model（逻辑模型） 84](#_Toc406495442)

[8.3 Physical model（物理模型） 85](#_Toc406495443)

[8.4 Form Introduction（表介绍） 86](#_Toc406495444)

[9. Input and Output Design（输入输出流设计） 87](#_Toc406495445)

[9.1 Normal user Register(普通用户注册) 87](#_Toc406495446)

[9.1.1 Input design(输入设计) 87](#_Toc406495447)

[9.1.2 Output design(输出设计) 87](#_Toc406495448)

[9.2 Householder Register(家庭户主注册家庭账号) 88](#_Toc406495449)

[9.2.1 Input design(输入设计) 88](#_Toc406495450)

[9.2.2 Output design(输出设计) 89](#_Toc406495451)

[9.3 Adding to family group(个人申请加入家庭群) 89](#_Toc406495452)

[9.3.1 Input design(输入设计) 90](#_Toc406495453)

[9.3.2 Output design(输出设计) 90](#_Toc406495454)

[9.4 Adding friends among families(家庭间互加好友) 91](#_Toc406495455)

[9.4.1 Input design(输入设计) 91](#_Toc406495456)

[9.4.2 Output design(输出设计) 91](#_Toc406495457)

[9.5 Chat management(聊天管理) 92](#_Toc406495458)

[9.5.1 Input design(输入设计) 92](#_Toc406495459)

[9.5.2 Output design(输出设计) 93](#_Toc406495460)

[9.6 Memo management(备忘录管理) 93](#_Toc406495461)

[9.6.1 Input design(输入设计) 93](#_Toc406495462)

[9.6.2 Output design(输出设计) 94](#_Toc406495463)

[9.7 Family circle management(家庭圈管理) 95](#_Toc406495464)

[9.7.1 Input design(输入设计) 95](#_Toc406495465)

[9.7.2 Output design(输出设计) 95](#_Toc406495466)

[9.8 Game management(游戏管理) 96](#_Toc406495467)

[9.8.1 Input design(输入设计) 96](#_Toc406495468)

[9.8.2 Output design(输出设计) 97](#_Toc406495469)

[9.9 Promotion for family applications(家庭推广) 97](#_Toc406495470)

[9.1.1 Input design(输入设计) 97](#_Toc406495471)

[9.1.2 Output design(输出设计) 98](#_Toc406495472)

[10. Error handling(错误处理) 99](#_Toc406495473)

[10.1 Target(目标) 99](#_Toc406495474)

[10.2 Steps of handling errors(错误处理步骤) 99](#_Toc406495475)

[10.2.1 Process of handling errors(错误处理过程) 99](#_Toc406495476)

[10.2.2 Flow chart of handling errors(错误处理流程图) 100](#_Toc406495477)

[10.3 Model of handling errors(错误处理模式) 101](#_Toc406495478)

[10.3.1 Relationship because of errors(错误的引起关系) 101](#_Toc406495479)

[10.3.2 Handle errors-frame model(错误处理-框架模式) 102](#_Toc406495480)

[10.4 Architecture （错误树） 102](#_Toc406495481)

[10.5 Error type（错误类型） 104](#_Toc406495482)

[10.6 Error prevention（错误预防） 104](#_Toc406495483)

[11. Poster（海报） 105](#_Toc406495484)

# Cover page(封面页)

## 1.1 Exeuctive summary（摘要）

在上一次的分析中，我们发现现代家庭关系出现**大家族向小家庭**发展的形势。究其原因是随着经济的发展，社会形态的转变导致家庭关系发生了变化，特别是计划生育政策实施以来，子女结婚之后普遍另立门户，使得家庭越来越小。这种转变导致家庭成员之间的**交流减少**，**关系变得疏远**，进而家庭与家庭的联系也趋于平淡甚至消失。

针对这一现象，我们设计了一款帮助增进家庭关系的软件来缓建这种现象，并增强群众的家庭观念的手机软件。其主要的功能类似于当下流行的微信朋友圈，但是在朋友圈的功能上增加了：家庭信息分享，家庭游戏，家庭备忘录，家庭商品推广等与家庭相关的功能，以此来增强用户对于家庭的重视程度，改善家庭间交流贫乏，关系单薄的情况。

在这一次作业中，我们针对上一次作业分析中选择的方案进行更加详细的业务分析与设计。更深层次的理解了业务需求和整个系统功能，并采用基于面向对象的分析方法分析系统并设计系统。希望能为方案的实现提供坚实而可行的基础。

## 1.2 Introduction（介绍）

在这一次的，我们通过上一次选择的方案和得到的结果，进行更深一步的分析与设计。

首先，我们进行了范围建模，明确了系统的范围、目标。之后进行了业务用例声明，确定了系统的业务范围，类别，关系。之后根据业务用例进行了过程建模，明确地规定了业务流程。确定了系统的业务之后进行了资源建模，分析并明确了系统中永存地不可替代的需要的资源。

明确了系统需求之后，我们确定了系统规格。根据用例模型描述功能需求，之后确定了非功能需求并明确了测试标准以及全局性约束定义。

业务和系统明确之后我们设计了GUI并构建了原型，根据之前的类图进行了整体架构的一般设计。确定架构之后又针对具体类和组建进行了详细设计。框架搭好之后进行了数据库建模与设计，明确了输入输出，异常处理。

最后根据上述分析，做出海报来描述整个的设计过程和思路。

# 2. Context of the Project Launched（项目运行环境）

## 2.1Business case statement（业务案例声明）

### 2.1.1 Existing problems(系统存在的问题)

随着家庭关系模式的转变，家庭成员疏远的问题的日益严重，我们设计的软件旨在加强家庭成员的交流，拉近家人之间的距离。

经过我们的网上调研，现如今朋友之间的关系更要胜过家庭之间的关系，有**46%**的年轻人在闲暇时光喜欢在外面玩、和朋友逛而不是与家人一起度过，即便是过节，也有**30%**以上的人不回家，在这些人中，除去客观原因以外，仍旧不愿意回家的占到了**40%**。朋友之间的共同话题更多，交流频繁，即使不见面也有微信，QQ，人人等社交软件继续线下交流，而针对家庭的社交软件却少之又少，使得家庭成员之间的交流更加少，这也正是家庭观念意识淡化的主要原因。

可想而知，如果任由这种转变长期地发生，今后的社会必然会出现家庭各自为战的情况。每个家庭只在乎自己，人与人之间都缺乏关爱。就像四合院到楼房的转变，一院人变为一屋人。

### 2.1.2 The purpose of the solution(解决方案的目的)

我们的项目旨在增进家庭关系，能够让家庭成员之间的联系，家庭与家庭之间的联系更加紧密，从而培养子女的家庭意识，并为家庭生活带来便利。

根据之前的分析我们拟定了项目需要实现的需求功能：**1.**家庭成员之间的信息分享，譬如家庭成员的生活趣事，以及他们在朋友圈的分享内容；**2.**家庭与家庭之间的交流互动，把家庭看作一个大的单位，相当于一个人，家庭与家庭的交流就相当于好友之间的交流，并且这个好友圈是可扩展的；**3.**家庭成员的互动，通过多人家庭游戏，家庭日常用品的推销，家庭内部的备忘录等将家庭成员的注意力拉向家庭，使得他们更加关注彼此，从而改善家庭间交流贫乏、家庭成员关系淡薄的情况。

### 2.1.3 System Scope(系统范围)

**Scope：**

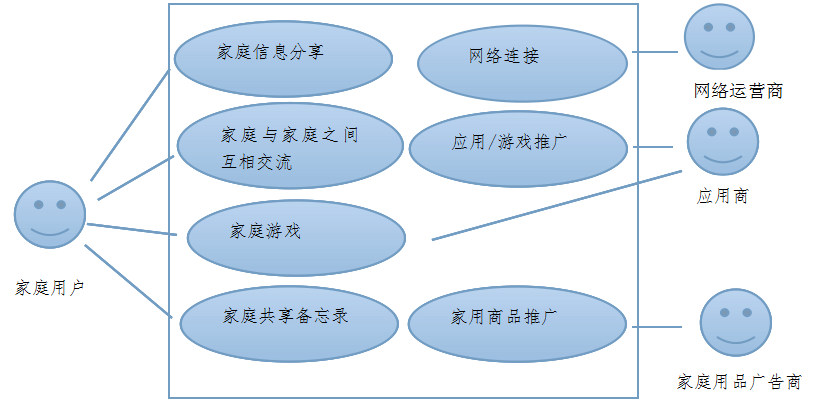


图2.1.3.1 系统范围

**Boundary：**

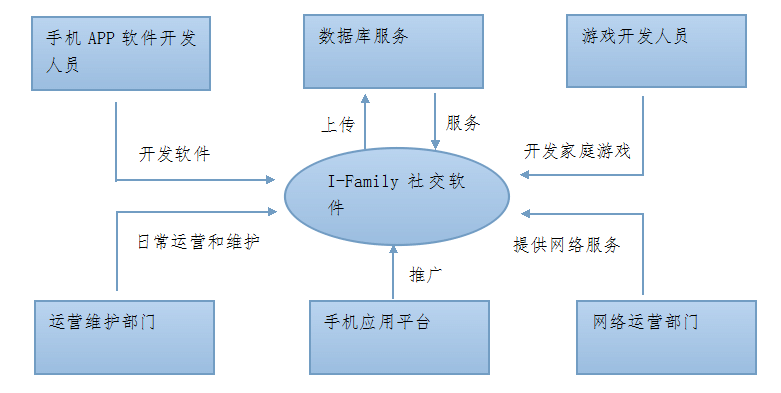


图2.1.3.2 系统边界

## 2.2Systems Development methodology selected for this project （项目系统发展方法论）

在这个项目中，我们选择用面向对象的系统分析和设计完成对上述领域的分析和系统建模。面向对象的方法是基于对象概念的分析与设计方法，认为信息系统是按照不同抽象层次组织的对象的集合。我们会描述系统中的对象、对象的属性和操作、对象的动态特性、对象间的构造关系和通信关系，从而建立系统的静态结构和动态活动模型。

## 2.3 Key steps undertaken in the design project（项目设计主要步骤）

在设计阶段，我们根据需求分析阶段的功能分解图，对主要的系统和复杂的功能分别进行整体架构设计，表⽰示层，控制层，数据层，组建层和部署层的设计。

家用商品推广

家用应用

广告模块

I-Family

信息分享模块

应用模块

个人信息分享

家庭信息分享

家庭成员交流

家庭之间交流

家庭共享备忘录

家庭小游戏

家庭应用推广

家庭游戏推广

图2.3 模块分解图

## 2.4 Members responsibilities in this project（项目小组成员）

|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | **姓名** | **角色** | **职责** |
| 王畅 | 12301106 | 组长 | 组织组员开会讨论，负责整合组员的提交成果，编写文档，记录每次开会成果。 |
| 白亚楠 | 12301088 | 组员 | 主要负责项目主体的设计，包括项目目标制定，模型架构，负责文档排版。 |
| 赵家进 | 12301113 | 组员 | 主要负责项目的框架设计，功能细节设计。 |
| 游程 | 12301112 | 组员 | 主要负责数据库设计，输入输出设计。参与会议修正他人的设计。 |
| 许若昕 | 12301109 | 组员 | 主要负责GUI设计，错误处理和规则定义，参与会议修正他人的设计。 |

## 2.5 The project and team dynamics （项目和团队分工）

|  |  |  |
| --- | --- | --- |
| Week | Task | |
| 第一周 | 王畅 | 整合上一次作业的内容完成文档的cover page以及 context of project |
| 白亚楠 | 根据上一次的分析完成文档的分析现有系统以及系统建模的部分。 |
| 赵家进 | 根据现有系统进行框架设计，再根据具体流程进行细节设计。完成文档的相关部分 |
| 游程 | 根据整体架构的设计以及业务流程设计数据库和输入输出流。 |
| 许若昕 | 根据现有的业务流程和功能需求完成GUI相关部分。并后续完成异常处理。 |
| 第二周 | 所有小组成员完成初步分析、文档编写之后开会讨论，根据每个人负责的部分展开讨论，由其他小组成员提出质疑并逐步完善文档和分析。 | |
| 第三周 | 所有小组成员进行文档完善之后，修缮细节部分并开始设计海报部分。 | |

# 3.Understanding the current system（现下系统理解）

## 3.1 Scope Model（范围模型）

### 3.1.1 Context Model（环境模型）

上下文图主要是识别系统内与系统外的关系，根据对I-Family项目的分析，与项目系统相关的外部参与者主要有：家庭用户、网络运营商、家庭用品广告商以及家庭应用、游戏推广商。

其中家庭用户可以使用系统进行家庭活动与互动，家庭用品广告商向系统推送合适的家庭用、其他应用与游戏商为系统推送适合家庭使用的应用与游戏、而网络运营商为系统提供了网络支持。

I-Family 系统

网络运营商

家庭用品广告商

家庭应用、游戏

推广商

家庭用户

为广告商提供潜在目标客户群

为广告商提供推广平台

为系统用户提供便捷的服务

为推广商提供推广平台

为系统提供多人家庭游戏

创建家庭群：聊天、朋友圈

创建、共享备忘录

共同加入家庭游戏

与其他家庭结为好友

实现家庭成员间、家庭间互动

提供网络支持

图3.1.1.1 上下文分析图

### 3.1.2 Conceptual Model（概念模型）

概念模型能够反映信息需求的概念结果，基于需求分析，通过抽象概念，我们用如下概念模型来表示我们的结果。

家庭成员使用同一备忘录

发布家庭圈，更新备忘录

快速注册加入家庭群

I-Family系统

家庭群

家庭成员

普通注册加入家庭群

QQ、新浪API

通过家庭号加入系统

网络运营商

网络服务

网络服务

家庭生活用品

推广商

多人应用、游戏推广商

推广商

应用商店

提供平台

提供平台

提供生活用品服务

提供娱乐服务

备忘录

同步朋友圈、备忘录给家庭群

浏览

浏览

同步其他家庭成员对备忘录的修改

图3.1.2 概念模型

### 3.1.3 Objective-Problem Model（目标模型）

目标模型能够帮助我们更深入的了解要解决的问题、问题对应的目标客户以及对于不同的目标客户，什么解决方案能够达到客户的要求。

I-Family

家庭成员较少、关系疏远已成为一种社会现象

类似社交产品

竞争激励

客户不易接受应用中的各种推广

应用只能提供适 于家庭使用的

推广

问题

目标客户

家庭用户

广告商

解决方案

在应用宣传时主要体现亲情的温暖来唤醒用户

应用宣传时主要突出亲情元素、I-Family的功能全面性

推广中的产品相较其他平台的同类产品价格较低

相较降低广告商的广告费

图3.1.3 目标模型

## 3.2 Bussiness UseCase Statement（系统用例声明）

### 3.2.1 Bussiness use case model（系统用例模型）

游戏、应用推广商

家庭用品推广商

I-Family系统

家庭用户

网络运营商

图 3.2.1 系统用例模型

### 3.2.2 Bussiness use case specification（系统用例陈述）

|  |  |  |
| --- | --- | --- |
| 用例名 | 用户基本信息管理 | 用例类型  业务需求 |
| 用例ID | BC001 |
| 主要业务参与者 | I-Family应用系统 家庭用户 | |
| 其他参与者 |  | |
| 项目相关人员兴趣 | **Family应用系统：** 收集用户信息与家庭信息，为系统中各项功能（家庭备忘录、家庭圈等）的实现提供数据基础。  **家庭用户：**用户注册登录之后，将信息提交给系统得到验证之后才可以正式获得使用应用中各项系统功能的权限，例如家庭备忘录、家庭圈等功能。 | |
| 描述 | 该用例主要存储家庭个人及家庭群的基本信息，用来登录验证以及系统中其他功能实现（家庭备忘录、家庭圈等）提供基本的数据基础。 | |
| 前置条件 | 应用下载者同意注册项中的各个条例并进行注册。 | |
| 后置条件 | 系统管理员确认用户信息，将用户信息更新到数据库中。 | |
| 触发条件 | 用户注册使用该应用或者加入某个家庭群。 | |
| 基本流程 | 1.用户下载应用，认真填写注册信息，同意注册条例。  2.用户登录之后，通过家庭群编号或者家庭群名称加入家庭群并且家庭户主同意加入。  3.系统将用户信息及家庭群信息更新到数据库中。 | |
| 拓展流程 | 1.可是通过使用QQ或者微博的API帮助用户实现快速注册。 | |
| 结束 | 当用户注册结束或者申请加入家庭组并经户主审核以后结束该用例 | |
| 业务规则 | 必须提供合理的注册信息才可以通过审核 | |
| 实现约束和说明 | 用户必须填写合理的注册信息才能通过审核 | |
| 假设 | 每次注册信息都准确无误，没有欺骗行为与故意填写虚假信息，管理员每次审核客观并且准确 | |
| 待解决的问题 | 注册信息的有效性、合法性 | |

|  |  |  |
| --- | --- | --- |
| 用例名 | 聊天管理 | 用例类型  业务需求 |
| 用例ID | BC002 |
| 主要业务参与者 | I-Family应用系统 家庭用户 | |
| 其他参与者 | 网络运营商 | |
| 项目相关人员兴趣 | **Family应用系统：** 系统实时将用户聊天信息存入数据库并进行备份，为查询聊天记录功能提供数据基础  **家庭用户：**用户选择聊天功能以后，选择聊天对象（家庭内成员，家庭群组以及家庭与家庭间的群组），可以进行实时通信。  **网络运营商：**获取用户在聊天的时候所需要网络费用。 | |
| 描述 | 该用例主要描述了个人家庭用户加入家庭群之后，可以与家庭内成员、家庭群组以及家庭与家庭间群组发起回话，实现实时通信。 | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | |
| 后置条件 | 系统将每个人的聊天内容实时写入数据库，并实时从数据库读取出来并显示在家庭群内。 | |
| 触发条件 | 家庭群内的任意一名家庭成员在聊天室内发起会话。 | |
| 基本流程 | 1. 用户加入家庭群并获得户主的审核通过之后可以选择聊天功能。  2. 选择聊天对象：家庭成员个人、家庭群组或者家庭间群组。  3.输入聊天内容（文字或表情等）。 | |
| 拓展流程 | 3.在聊天室内可以发布语音，方便交流。 | |
| 结束 | 任意用户在退出登录之后，功能随之结束（用户切换到其他功能时，只要有其他用户对本用户发起会话，改用户会收到提醒）。 | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和接收消息成功 | |
| 实现约束和说明 | 在双方用户网络畅通的情况下，该用例才可以正常使用 | |
| 假设 |  | |
| 待解决的问题 | 对于聊天内容的甄别，要求合法 | |

|  |  |  |
| --- | --- | --- |
| 用例名 | 家庭圈管理 | 用例类型  业务需求 |
| 用例ID | BC003 |
| 主要业务参与者 | 家庭用户 | |
| 其他参与者 | 网络运营商 | |
| 项目相关人员兴趣 | **家庭用户：**每一个家庭用户可以在家庭圈发布文字、图片，获取其他家庭用户对发布内容的评论。同时可以实时看到其他家庭成员的发布共享内容，并进行评论。  **网络运营商：**获取用户在访问家庭圈的时候所需要网络费用。 | |
| 描述 | 该用例主要描述了个人家庭用户加入家庭群之后在家庭圈发布共享文字、图片以及相互评论 | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | |
| 后置条件 | 系统将每个人的发布共享内容以及评论信息实时写入数据库，并实时读出来显示在家庭群内。 | |
| 触发条件 | 家庭群内的任意一名家庭成员点击进入家庭圈功能 | |
| 基本流程 | 1. 用户加入家庭群并获得户主的审核通过之后可以选择家庭圈功能。  2. 用户在家庭圈内发布共享信息，选择发布者（个人身份或者家庭身份）以及指定可看人群。  3.查看其它家庭成员的共享信息并进行评论 | |
| 拓展流程 | 在家庭圈可以发布简短视频 | |
| 结束 | 用户在切换到应用其它功能时，此用例结束 | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和查看家庭圈 | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | |
| 假设 |  | |
| 待解决的问题 | 对于家庭圈发布内容及评论内容的甄别，要求合法 | |

|  |  |  |
| --- | --- | --- |
| 用例名 | 备忘录管理 | 用例类型  业务需求 |
| 用例ID | BC004 |
| 主要业务参与者 | 家庭用户 | |
| 其他参与者 | 网络运营商 | |
| 项目相关人员兴趣 | **家庭用户：**家庭个人成员在家庭备忘录中新建备忘事项，并且可以完成家庭备忘录中的事项并且实时修改事项状态。  **网络运营商：**获取用户在同步和更新以及查看备忘录时所需要的网络费用 | |
| 描述 | 该用例主要用于家庭备忘录操作。任意家庭用户可以在家庭备忘录中新建备忘录，新建时，可以选择特别提醒某个家庭成员，也可以设置闹钟来进行提醒。用户也可以完成备忘录中由其他用户添加的备忘事项，完成之后实时修改事项状态。 | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | |
| 后置条件 | 系统将新建的备忘录信息以及备忘录状态实时写入数据库并显示给每一个家庭用户 | |
| 触发条件 | 家庭群内的任意一名家庭成员点击进入备忘录功能或者被@的家庭成员收到的闹钟提醒 | |
| 基本流程 | 1. 用户加入家庭群并获得户主的审核通过之后可以选择家庭备忘录功能。  2. 用户在家庭备忘录添加备忘事项（选择特别提醒的人，也可以设置事项闹钟）。  3.完成备忘事项之后及时修改备忘事项状态。 | |
| 拓展流程 | 可以在备忘录中添加图片、视频不仅仅是文字 | |
| 结束 | 用户在切换到应用其它功能时，此用例结束 | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和查看家庭备忘录 | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | |
| 假设 |  | |
| 待解决的问题 | 对于家庭备忘录发布内容的甄别，要求合法 | |

|  |  |  |
| --- | --- | --- |
| 用例名 | 家庭游戏管理 | 用例类型  业务需求 |
| 用例ID | BC005 |
| 主要业务参与者 | I-Family应用系统 家庭用户 | |
| 其他参与者 | 网络运营商 | |
| 项目相关人员兴趣 | **Family应用系统：** 通过游戏功能来吸引更多的使用者  **家庭用户：**在游戏中可以和家人多一些互动，拉近家人之间的关系。  **网络运营商：**获取用户在玩游戏时所需要的网络费用。 | |
| 描述 | 该用例主要用于家庭多人游戏，可以家庭成员内玩，也可以家庭间进行比拼。在家庭比拼中，可以邀请好友一起来对战。在游戏中拉近家人的距离，同时也为家人间的相处提供了乐趣。 | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | |
| 后置条件 | 系统将家庭成员游戏的具体详细信息写入数据库 | |
| 触发条件 | 家庭群内的任意一名家庭成员点击进入游戏功能或者某一家庭成员接收到其他家人送的体力 | |
| 基本流程 | 1. 用户加入家庭群并获得户主的审核通过之后可以选择家庭游戏。  2. 用户在家庭游戏中开始游戏（也可邀请好友一起来对战）。 | |
| 拓展流程 |  | |
| 结束 | 用户在切换到应用其它功能时，此用例结束 | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以顺利开始游戏 | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | |
| 假设 |  | |
| 待解决的问题 | 游戏形式不涉及暴力等不合法内容 | |

|  |  |  |
| --- | --- | --- |
| 用例名 | 推广管理 | 用例类型  业务需求 |
| 用例ID | BC006 |
| 主要业务参与者 | I-Family应用系统 家庭用户 家庭应用、游戏、商品推广商 | |
| 其他参与者 | 网络运营商 | |
| 项目相关人员兴趣 | **Family应用系统：**争取更多的广告商更多的广告投资。  **家庭用户：**在推送中看到更多适合家庭的、物美价廉的商品，可以找到更适合家庭一起使用的应用和一起玩的游戏。  **广告商：**广告商的广告更新能够及时准确的显示给用户。  **网络运营商：**获取用户在浏览及下载过程中所产生的流量费用 | |
| 描述 | 该用例主要是系统将广告商的广告审核之后推送给家庭用户 | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | |
| 后置条件 | 推送广告内容显示改变 | |
| 触发条件 | 家庭群内的任意一名家庭成员登录之后即可看到推送广告 | |
| 基本流程 | 1. 用户加入家庭群并获得户主的审核通过之后登录，可以看到系统的推送广告 | |
| 拓展流程 |  | |
| 结束 | 用户退出登录之后，推送广告结束 | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和查看推送广告 | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | |
| 假设 |  | |
| 待解决的问题 | 对于家庭广告内容，要求合法 | |

## 3.3 Process Modeling(including events)（流程模型）

首先，我们根据context model对整个系统进行功能分解，形成功能分解图。

接下来，就编译了一张事件响应清单，来确定并证实系统必须提供响应的业务事件。并且这张清单里描述了对每个事件来说需要或可能的响应。

最终，我们根据这些事件以及系统中出现的过程绘制了整个系统的过程模型。

### 3.3.1 Functional decomposition diagrams(功能分解图)

I-Family

信息分享模块

娱乐模块

广告模块

个人信息分享

家庭信息分享

家庭成员交流

家庭之间交流

家庭共享备忘录

家庭小游戏

家庭用品推广

家庭游戏推广

家庭应用推广

图 3.3.1 功能分解图

### 3.3.2 Functional decomposition with event(带有事件的功能分解)

I-Family

信息分享模块

应用模块

广告模块

用户信息

信息分享

个人用户注册

家庭用户注册

个人用户注销

个人信息修改

家庭信息修改

个人发布家庭圈

家庭发布家庭圈

个人查看家庭圈

家庭成员间聊天

家庭之间群聊

个人查看备忘录

个人更新备忘录

家庭小游戏

邀请好友玩游戏

家庭间游戏比分

家庭用品推广

家庭游戏推广

家庭应用推广

个人加入家庭

家庭互加好友

评论家庭圈

删除发布家庭圈

删除评论家庭圈

备忘录提醒

图 3.2.2 带有事件的功能分解图

### 3.3.3 The list of event response(事件响应清单)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 参与者 | 事件 | 触发器 | 响应 |
| 1 | 家庭用户 | 个人注册 | 注册请求 | 生成“用户注册确认”并在数据库中生成用户注册信息 |
| 2 | 家庭用户 | 个人注销 | 注销请求 | 生成“用户注销确认” |
| 3 | 家庭用户 | 个人信息修改 | 信息修改请求 | 生成“用户修改请求”并在数据库中即时更新数据 |
| 4 | 家庭用户 | 发布个人家庭圈 | 发布信息请求 | 跳转到发布信息界面，选择个人发布信息，将发布信息即时更新到数据库并显示到家庭圈界面上 |
| 5 | 家庭用户 | 评论家庭圈内容 | 评论请求 | 生成“评论确认”，将评论内容实时写入数据库 |
| 6 | 家庭用户 | 删除在家庭圈的评论 | 删除评论请求 | 生成“删除评论确认”，将评论信息即时更新到数据库中 |
| 7 | 家庭用户 | 删除已发表的家庭圈状态 | 删除状态请求 | 生成“删除状态确认”，将状态信息即时更新到数据库中 |
| 8 | 家庭用户 | 备忘录提醒 | 计时备忘录提醒 | 从数据库读取的备忘录计时时间与系统时间相等，弹出界面及音乐提醒备忘录事件 |
| 9 | 家庭用户 | 发布家庭的家庭圈 | 发布信息请求 | 跳转到发布信息界面，选择家庭发布信息，将发布信息即时更新到数据库并显示到家庭圈界面上 |
| 10 | 家庭用户 | 个人加入家庭 | 申请加入请求 | 跳转到申请界面，通过家庭号或家庭名寻找申请加入，等待家庭户主同意 |
| 11 | 家庭用户 | 查看家庭圈 | 查看家庭圈请求 | 跳转到家庭圈界面，并且从数据库读取信息显示到界面 |
| 12 | 家庭用户 | 家庭群聊、家庭间聊天 | 聊天请求 | 跳转到群聊界面，输入聊天信息，将信息更新到数据库并实时显示到界面上 |
| 13 | 家庭用户 | 查看备忘录 | 查看备忘录请求 | 跳转到家庭备忘录界面，从数据库读取备忘录信息实时显示到界面上 |
| 14 | 家庭用户 | 更新备忘录 | 更新备忘录请求 | 跳转到编辑备忘录界面，并将更新信息实时保存到数据库 |
| 15 | 家庭用户 | 开始家庭小游戏 | 游戏请求 | 游戏自动使用当前应用用户登录，将游戏记录保存到数据库 |
| 16 | 家庭用户 | 查看推广 | 查看推广请求 | 跳转到推广界面，从数据库将相应信息读取并显示到界面上 |
| 17 | 家庭户主 | 家庭注册 | 家庭注册请求 | 生成“家庭注册确认”并在数据库中生成家庭注册信息 |
| 18 | 家庭户主 | 家庭信息修改 | 家庭修改请求 | 生成“家庭修改请求”并在数据库中即时更新数据 |
| 19 | 家庭户主 | 同意加入家庭 | 同意申请请求 | 同意个人申请加入家庭群，并且将家庭新关系以及个人新关系保存到数据库 |
| 20 | 家庭户主 | 申请关注家庭 | 申请关注请求 | 跳转到申请界面，通过家庭号或家庭名寻找申请关注，等待其家庭户主同意 |
| 21 | 家庭户主 | 同意申请关注 | 同意关注请求 | 跳转到同意申请关注界面，并将两个家庭中新的家庭关系同步更新到数据库中， |
| 22 | 家庭户主 | 邀请好友共同玩游戏 | 邀请好友请求 | 生成“邀请好友确认”，系统向被邀请好友发出邀请请求 |
| 23 | 家庭户主 | 查看游戏比分 | 查看比分请求 | 跳转到查看比分界面，将比分排名从数据库计算显示到界面上 |
| 24 | 系统管理员 | 推广设置 | 推广设置请求 | 生成“推广设置确认”，将管理员更新的广告内容实时写入到数据库中 |

### 3.3.4 Process model(过程模型)

个人用户

注册用户

注册过程

注册成功

家庭户主

注册家庭用户

注册过程

注册成功

个人用户

申请加入家庭群

申请过程

申请成功

家庭户主

申请关注家庭

申请过程

关注成功

同意其他家庭申请

同意过程

个人用户

选择聊天对象

聊天过程

通信成功

个人用户

查看备忘录

更新备忘录过程

创建新备忘录过程

更新备忘录成功

个人用户

查看家庭圈

查看家庭圈过程

发布家庭圈过程

家庭圈更新成功

个人用户

选择游戏功能

游戏过程

游戏结束

个人用户

查看推广信息

查询过程

查询结束

图 3.3.4 过程模型

## 3.4 Resource Modeling（资源模型）

### 3.4.1 Application interface resource(应用界面资源)

I-Family系统

首页

聊天页面

家庭圈页面

备忘录页面

游戏页面

推广页面

个人页面

发布页面

修改页面

新建页面

修改页面

图 3.4.1 应用界面资源

### 3.4.2 Information resource(信息资源)

用户信息

User：String

Password：String

家庭圈信息

User： String

Time：Date

Authority：Int

Content：String

游戏记录信息

User: String

Time：Date

Record：String

聊天信息

User： String

Recive：String

Time：Date

Content：String

备忘录信息

User : String

Event : String

State: Int

Circle: String

1

\*

1

1

1

\*

\*

\*

图 3.4.2 信息资源

## 3.5 Rule Definition（规则定义）

|  |  |
| --- | --- |
| 编号 | 项 |
| Rule1 | 所有用户数据应在数据库中存有备份 |
| Rule2 | 要求备忘录功能的数据一致性、同步性以及并发性控制。 |

# 4.SystemRequirementantalysisandSpecification（系统需求分析和陈述）

## 4.1 Functionality Documentation（功能陈述）

### 4.1.1 Use case Model（用例模型）

I-Family系统

信息分享模块

家庭用户

家庭户主

系统管理员

个人用户注册

家庭用户注册

个人用户注销

个人信息修改

家庭信息修改

个人加入家庭

家庭互加好友

个人发布家庭圈

家庭发布家庭圈

家庭成员群聊

家庭间群聊

个人查看备忘录

个人更新备忘录

备忘录提醒

评论家庭圈

删除发布家庭圈

删除评论家庭圈

邀请好友玩游戏

家庭间游戏比分

家庭用品推广

家庭游戏推广

娱乐模块

应用模块

家庭应用推广

图 4.4.1 用例模型

### 4.1.2 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名 | 用户基本信息管理 | 用例类型  系统用例 | |
| 用例ID | SC001 |
| 主要业务参与者 | I-Family应用系统 家庭用户 | | |
| 其他参与者 |  | | |
| 项目相关人员兴趣 | **Family应用系统：** 收集用户信息与家庭信息，为系统中各项功能（家庭备忘录、家庭圈等）的实现提供数据基础。  **家庭用户：**用户注册登录之后，将信息提交给系统得到验证之后才可以正式获得使用应用中各项系统功能的权限，例如家庭备忘录、家庭圈等功能。 | | |
| 描述 | 该用例主要存储家庭个人及家庭群的基本信息，用来登录验证以及系统中其他功能实现（家庭备忘录、家庭圈等）提供基本的数据基础。 | | |
| 前置条件 | 应用下载者同意注册项中的各个条例并进行注册。 | | |
| 后置条件 | 系统管理员确认用户信息，将用户信息更新到数据库中。 | | |
| 触发条件 | 用户注册使用该应用或者加入某个家庭群。 | | |
| 基本流程 | **参与者:** | | **系统响应:** |
| **第一步：**用户下载应用，认真填写注册信息，同意注册条例。  **第三步：**用户登录之后，通过家庭群编号或者家庭群名称加入家庭群并且家庭户主同意加入。 | | **第二步：**系统审核用户提交的注册信息，同意注册。  **第四步：**系统将用户信息及家庭群的信息更新到数据库中。 |
| 拓展流程 | 1.可以通过使用QQ或者新浪微博的API帮助用户实现快速注册。 | | |
| 结束 | 当用户注册结束或者申请加入家庭组并经户主审核以后结束该用例 | | |
| 业务规则 | 必须提供合理的注册信息才可以通过审核 | | |
| 实现约束和说明 | 用户必须填写合理的注册信息才能通过审核 | | |
| 假设 | 每次注册信息都准确无误，没有欺骗行为与故意填写虚假信息，管理员每次审核客观并且准确 | | |
| 待解决的问题 | 注册信息的有效性、合法性 | | |

注册

失败

系统通过认证

申请加入家庭圈

失败

成为家庭圈成员

状态图-用户基本信息管理图

个人用户

系统

家庭户主

提交注册信息

注册成功

申请加入家庭群

审核结果

审核结果

顺序图-用户基本信息管理图

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名 | 聊天管理 | 用例类型  系统用例 | |
| 用例ID | SC002 |
| 主要业务参与者 | I-Family应用系统 家庭用户 | | |
| 其他参与者 | 网络运营商 | | |
| 项目相关人员兴趣 | **Family应用系统：** 系统实时将用户聊天信息存入数据库并进行备份，为查询聊天记录功能提供数据基础  **家庭用户：**用户选择聊天功能以后，选择聊天对象（家庭内成员，家庭群组以及家庭与家庭间的群组），可以进行实时通信。  **网络运营商：**获取用户在聊天的时候所需要网络费用。 | | |
| 描述 | 该用例主要描述了个人家庭用户加入家庭群之后，可以与家庭内成员、家庭群组以及家庭与家庭间群组发起回话，实现实时通信。 | | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | | |
| 后置条件 | 系统将每个人的聊天内容实时写入数据库，并实时从数据库读取出来并显示在家庭群内。 | | |
| 触发条件 | 家庭群内的任意一名家庭成员在聊天室内发起会话。 | | |
| 基本流程 | **参与者：** | | **系统响应：** |
| **第一步：**用户选择聊天功能。  **第二步：**用户选择聊天对象： 家庭成员个人、家庭群组或者家庭间群组。  **第三步**：.输入聊天内容（文字或表情等）。  **第五步：**消息接受者接受到发送者的消息记录。 | | **第四步：**系统将用户发送的内容实时存入数据库，并向消息的接受者动态显示在聊天内容中。 |
| 拓展流程 | 3.在聊天室内可以发布语音，方便交流。 | | |
| 结束 | 任意用户在退出登录之后，功能随之结束（用户切换到其他功能时，只要有其他用户对本用户发起会话，改用户会收到提醒）。 | | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和接收消息成功 | | |
| 实现约束和说明 | 在双方用户网络畅通的情况下，该用例才可以正常使用 | | |
| 假设 |  | | |
| 待解决的问题 | 对于聊天内容的甄别，要求合法 | | |

聊天界面

选择聊天对象并发送消息

发送失败

系统将消息同步到接受者

状态图-聊天管理图

发送成功

发送消息者

系统

接收消息者

发送聊天内容

发送成功

顺序图-聊天管理图

提供聊天内容

接收成功

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名 | 家庭圈管理 | 用例类型  系统用例 | |
| 用例ID | SC003 |
| 主要业务参与者 | 家庭用户 | | |
| 其他参与者 | 网络运营商 | | |
| 项目相关人员兴趣 | **家庭用户：**每一个家庭用户可以在家庭圈发布文字、图片，获取其他家庭用户对发布内容的评论。同时可以实时看到其他家庭成员的发布共享内容，并进行评论。  **网络运营商：**获取用户在访问家庭圈的时候所需要网络费用。 | | |
| 描述 | 该用例主要描述了个人家庭用户加入家庭群之后在家庭圈发布共享文字、图片以及相互评论 | | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | | |
| 后置条件 | 系统将每个人的发布共享内容以及评论信息实时写入数据库，并实时读出来显示在家庭群内。 | | |
| 触发条件 | 家庭群内的任意一名家庭成员点击进入家庭圈功能 | | |
| 基本流程 | **参与者：** | | **系统响应：** |
| 第一步：用户进入家庭圈界面  第三步： 用户在家庭圈内发布共享信息，选择发布者（个人身份或者家庭身份）以及指定可看人群。  第六步：其他用户对共享内容进行评论 | | 第二步：系统将家庭圈共享内容显示到用户界面。  第四步：系统将用户发布的共享内容实时更新到数据库  第五步：系统将更新后的分享信息显示到其他用户界面  第六步：系统将用户评论实时更新到数据库  第七步：系统将更新后的评论信息显示到用户界面 |
| 拓展流程 | 在家庭圈可以发布简短视频 | | |
| 结束 | 用户在切换到应用其它功能时，此用例结束 | | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和查看家庭圈 | | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | | |
| 假设 |  | | |
| 待解决的问题 | 对于家庭圈发布内容及评论内容的甄别，要求合法 | | |

家庭圈界面

向系统发送查询家庭圈请求

请求失败

发布家庭圈

状态图-家庭圈管理图

请求成功

发布失败

发布成功

评论家庭圈状态

评论成功

评论失败

发布用户

系统

评论用户

发布共享内容

发布成功

顺序图-家庭圈管理图

更新共享内容

接收成功

发送查询请求

查询成功

发送评论内容

更新共享内容

评论成功

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名 | 备忘录管理 | 用例类型  系统用例 | |
| 用例ID | SC004 |
| 主要业务参与者 | 家庭用户 | | |
| 其他参与者 | 网络运营商 | | |
| 项目相关人员兴趣 | **家庭用户：**家庭个人成员在家庭备忘录中新建备忘事项，并且可以完成家庭备忘录中的事项并且实时修改事项状态。  **网络运营商：**获取用户在同步和更新以及查看备忘录时所需要的网络费用 | | |
| 描述 | 该用例主要用于家庭备忘录操作。任意家庭用户可以在家庭备忘录中新建备忘录，新建时，可以选择特别提醒某个家庭成员，也可以设置闹钟来进行提醒。用户也可以完成备忘录中由其他用户添加的备忘事项，完成之后实时修改事项状态。 | | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | | |
| 后置条件 | 系统将新建的备忘录信息以及备忘录状态实时写入数据库并显示给每一个家庭用户 | | |
| 触发条件 | 家庭群内的任意一名家庭成员点击进入备忘录功能或者被@的家庭成员收到的闹钟提醒 | | |
| 基本流程 | **参与者：** | | **系统响应：** |
| **第一步：**用户进入家庭备忘录界面。  **第三步：**用户新建家庭备忘录（设置备忘录时间、特别提醒某个家庭成员）  **第五步：**任意用户修改备忘录状态内容 | | **第二步：**系统将备忘录信息从数据库返回到用户界面上。  **第四步：**系统将新建的备忘录信息实时存入到数据库中。  **第六步：**系统将备忘录的更新内容实时更新到数据库 |
| 拓展流程 | 可以在备忘录中添加图片、视频不仅仅是文字 | | |
| 结束 | 用户在切换到应用其它功能时，此用例结束 | | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和查看家庭备忘录 | | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | | |
| 假设 |  | | |
| 待解决的问题 | 对于家庭备忘录发布内容的甄别，要求合法 | | |

备忘录界面

向系统发送查看备忘录请求

新建备忘录

状态图-备忘录管理图

请求成功

创建成功成功

创建失败

修改备忘录状态

修改成功成功

修改失败

请求失败

顺序图-备忘录管理图

新建用户

系统

新建备忘录

创建成功

发送查看备忘录请求

查询成功

修改备忘录状态

修改成功

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名 | 家庭游戏管理 | 用例类型  系统用例 | |
| 用例ID | SC005 |
| 主要业务参与者 | I-Family应用系统 家庭用户 | | |
| 其他参与者 | 网络运营商 | | |
| 项目相关人员兴趣 | **Family应用系统：** 通过游戏功能来吸引更多的使用者  **家庭用户：**在游戏中可以和家人多一些互动，拉近家人之间的关系。  **网络运营商：**获取用户在玩游戏时所需要的网络费用。 | | |
| 描述 | 该用例主要用于家庭多人游戏，可以家庭成员内玩，也可以家庭间进行比拼。在家庭比拼中，可以邀请好友一起来对战。在游戏中拉近家人的距离，同时也为家人间的相处提供了乐趣。 | | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | | |
| 后置条件 | 系统将家庭成员游戏的具体详细信息写入数据库 | | |
| 触发条件 | 家庭群内的任意一名家庭成员点击进入游戏功能或者某一家庭成员接收到其他家人送的体力 | | |
| 基本流程 | **参与者：** | | **系统响应：** |
| **第一步：**用户进入家庭游戏界面  **第二步：**用户选择游戏  **第四步：**用户可选择自己游戏或者邀请好友进行家庭对战 | | **第三步：**系统根据用户选择游戏启动游戏程序 |
| 拓展流程 |  | | |
| 结束 | 用户在切换到应用其它功能时，此用例结束 | | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以顺利开始游戏 | | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | | |
| 假设 |  | | |
| 待解决的问题 | 游戏形式不涉及暴力等不合法内容 | | |

游戏界面

选择游戏进入

是否对战模式

游戏开启

否

开始游戏

状态图-游戏管理图

游戏异常

邀请好友

是

用户一

系统

用户二

选择游戏模式

顺序图-游戏管理图

发送邀请信息

接收邀请

选择游戏开始

进入游戏成功

邀请好友

接收成功

开始游戏

开始游戏

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名 | 推广管理 | 用例类型  系统用例 | |
| 用例ID | SC006 |
| 主要业务参与者 | I-Family应用系统 家庭用户 家庭应用、游戏、商品推广商 | | |
| 其他参与者 | 网络运营商 | | |
| 项目相关人员兴趣 | **Family应用系统：**争取更多的广告商更多的广告投资。  **家庭用户：**在推送中看到更多适合家庭的、物美价廉的商品，可以找到更适合家庭一起使用的应用和一起玩的游戏。  **广告商：**广告商的广告更新能够及时准确的显示给用户。  **网络运营商：**获取用户在浏览及下载过程中所产生的流量费用 | | |
| 描述 | 该用例主要是系统将广告商的广告审核之后推送给家庭用户 | | |
| 前置条件 | 家庭个人用户必须加入家庭群并获得家庭户主的审核同意之后才可以使用此功能。 | | |
| 后置条件 | 推送广告内容显示改变 | | |
| 触发条件 | 家庭群内的任意一名家庭成员登录之后即可看到推送广告 | | |
| 基本流程 | **参与者：** | | **系统响应：** |
| **第一步：**用户进入推广页面  **第三步：**用户点击具体一项推广 | | **第二步：**系统将推广信息显示到用户界面上  **第四步：**系统将用户请求的具体信息显示到用户界面上 |
| 拓展流程 |  | | |
| 结束 | 用户退出登录之后，推送广告结束 | | |
| 业务规则 | 在2G/3G/4G或者WiFi稳定的情况下才可以发布和查看推送广告 | | |
| 实现约束和说明 | 在用户网络畅通的情况下，该用例才可以正常使用 | | |
| 假设 |  | | |
| 待解决的问题 | 对于家庭广告内容，要求合法 | | |

推广界面

向系统发送查询请求

发送失败

状态图-推广管理图

发送成功

顺序图-推广管理图

用户

系统

发送查看推广请求

查询成功

## 4.2 Non\_functional requirements（非功能需求）

### 4.2.1 Functionality(功能性)

准确性：

|  |  |
| --- | --- |
| 编号 | 项 |
| 1 | 系统各功能运⾏结果与预期一致 |

互操作性：

|  |  |
| --- | --- |
| 编号 | 项 |
| 1 | 因为这款APP我们会开发Android和IOS两个系统，所以在任何智能手机上都可以使用 |

安全性：

|  |  |
| --- | --- |
| 编号 | 项 |
| 1 | 系统的数据库字段进行加密保证用户信息安全性 |
| 2 | 当受到病毒攻击时，系统应该能够检测到攻击操作并立刻停止服务器运⾏行。 |
| 3 | 应用程序必须扫描所有进入的或更改的数据及软件，以发现所有被公布的知名计算机病毒，蠕虫以及木马。 |
| 4 | 至少**99.9%**以上的时间，系统能够保护用户之间的消息不被非授权增加，删改。 |

### 4.2.2 Reliability(可靠性)

成熟性：

|  |  |  |
| --- | --- | --- |
| 编号 | 项 | 值 |
| 1 | 在任意时刻服务器运⾏的可能性 | **99.99%** |
| 2 | 服务器平均正常运⾏时间 | **360** |
| 3 | 服务器平均故障恢复时间 | **1小时** |

易恢复性：

|  |  |
| --- | --- |
| 编号 | 项 |
| 1 | 系统可以进行数据备份，最近一个月的业务数据、数据库数据全备份**(30**份，每日⼀一份，保留两个月**)**，每周五进行数据完全备份一次**(**保留两个月**)**，每月末进行一次数据完全备份**(**保留一年**)**，每一小时业务数据、数据库数据增量备份一次。 |
| 2 | 重大故障需要在**4~8**小时恢复服务的可用性，并在**5~10**小时内恢复历史数据。 |

### 4.2.3 [Usability](javascript:void(0);)易用性

易理解性：

|  |  |  |
| --- | --- | --- |
| 编号 | 项 | 值 |
| 1 | 专业⼈人员理解系统所需时间 | 1天 |
| 2 | 有软件基础的人员理解系统所需时间 | **4**天 |
| 3 | ⽆无软件基础理解系统所需时间 | **1周** |

易学习性：

|  |  |  |
| --- | --- | --- |
| 编号 | 项 | 值 |
| 1 | 专业人员学习该软件所需时间 | **10min** |
| 2 | 有软件基础的人员学习该软件所需时间 | **15min** |
| 3 | 无软件基础学习软件所需时间 | **30min** |

易操作性：

|  |  |  |
| --- | --- | --- |
| 编号 | 项 | 值 |
| 1 | 专业人员透彻操作系统所需时间 | **3h** |
| 2 | 管理员操作熟练时间 | **1h** |
| 3 | 外部⽤用户操作熟练时间 | **1h** |

### 4.2.4 Efficiency(效率)

吞吐量：

|  |  |  |
| --- | --- | --- |
| 编号 | 项 | 吞吐量 |
| 1 | 游戏对战间隙 | **30min一次** |
| 2 | 个人用户申请加入家庭 | **1天一次** |
| 3 | 用户备忘录更新 | **5分钟一次** |

容量：

|  |  |  |
| --- | --- | --- |
| 编号 | 项 | 吞吐量 |
| 1 | 一个家庭人数 | **<=100** |
| 2 | 备忘录数 | **<=1000** |
| 3 | 聊天记录数 | **<=1000,000,000** |
| 4 | 分享信息数 | **<=5000** |

延迟：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 项 | 响应时间 | 优先级 |
| 1 | 点击 | **<2**秒 | **高** |
| 2 | 注册 | **<5**秒 | **中** |
| 3 | 查询 | <5秒 | **低** |

### 4.2.5 Maintainability(维护性)

易分析性：

|  |  |
| --- | --- |
| 编号 | 项 |
| 1 | |  | | --- | | 分析普通系统故障的平均工作量小于1组天 | |
| 2 | 分析系统瘫痪或由于过多繁琐故障导致的系统破坏问题小于1组周 |
| 3 | 分析升级添加新功能需要更改的模块时间小于1组月 |

易改变性：

|  |  |
| --- | --- |
| 编号 | 项 |
| 1 | 修复问题**(**包括文档更新**)**的平均工作量小于一人周 |
| 2 | 完成一次小版本升级的平均工作量小于**1**人周 |
| 3 | 完成一次重大版本升级的工作量小于**1**人月 |
| 4 | 完成一次数据恢复的平均工作量小于**1**人天 |

稳定性：

|  |  |
| --- | --- |
| 编号 | 项 |
| 1 | 系统可添加新的功能，改进已有的功能或者修复系统中的缺陷。 |
| 2 | 系统可删除不再想要的功能。即优化或者简化现有系统的功能。 |

### 4.2.6 Portability(可移植性)

适应性：全球的智能手机设备

已安装性：全球可安装的只能手机设备

## 4.3 Global Constrain Definition（总体约束定义）

### 4.3.1 Schedule(日程)

项目需在3个月内完成。包括平台开发、测试和上线等。

### 4.3.2 Rule of function(性能规定)

1）用户信息确保安全，不会泄露

2）用户界面友好，适合家庭中各个年龄段的家庭成员使用

3）服务器系统能够应对各种病毒入侵，杜绝系统崩溃，能够对各种不可预测的突发情况给出相应的解决方案。

4）备忘录的并发性和同步性性能保证备忘录的正常工作

5）保证推广的内容是与家庭相关的内容

### 4.3.3 Scale and cost(规模及成本费用)

1）配备至少10个熟练的开发人员

2）开发资金少于20万

# 5.Communication Design（通讯设计）

## 5.1 Layout Design（布局设计）

#### 5.1.1 Menu Tree（菜单树）



图 5.1.1 菜单树

### 5.1.2 Content Frame（目录框架）

图 5.1.2 目录框架

### 5.1.3 Story Board & Window Design（界面设计）

Story-Board 普通用户



备注：由于显示限制，将各个界面的详细信息在下文展示

图 5.1.3.1 普通用户界面

**个人信息管理界面：**



图 5.1.3.2 个人信息界面

**聊天界面：**



图 5.1.3.3 聊天界面

**家庭圈界面：**



图 5.1.3.4 家庭圈界面

**游戏界面：**



图 5.1.3.5 游戏界面

**备忘录界面：**



图 5.1.3.6 备忘录界面

**推广界面：**



图 5.1.3.7 推广界面

Story-board 家庭户主



图 5.1.3.8 家庭户主界面

## 5.2 GUI Navigation Design（界面导航设计）

**普通用户登录方式：**

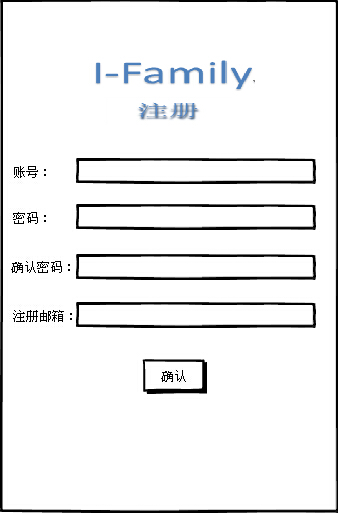
普通用户 → 登录成功 → 主界面 → 选择功能 →更新信息→将更新后的信息写入数据库



图 5.2 界面导航设计

## 5.3 Prototype building（原型构建）

### 5.3.1 Register(注册界面)



### 5.3.2 login(登录界面)

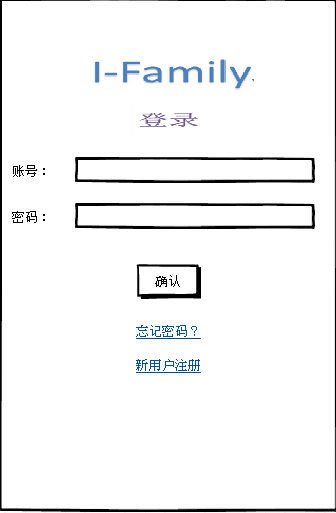


图 5.3.2 登录界面

### 5.3.3 chat(聊天界面)

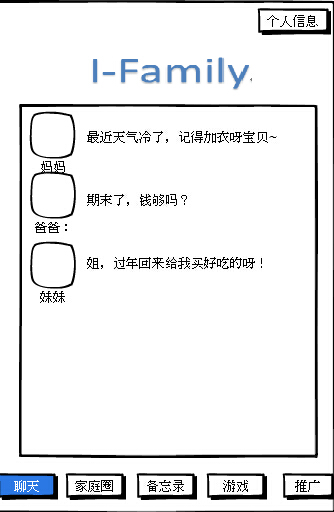


图 5.3.3 聊天界面

### 5.3.4 Family circle(家庭圈界面)

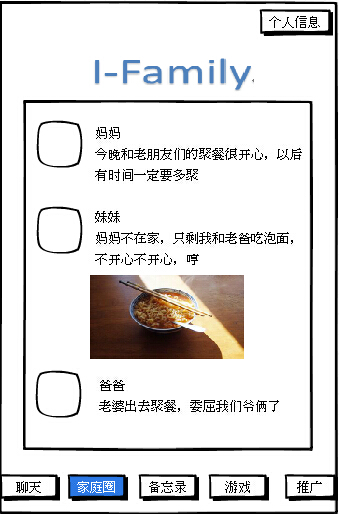


图 5.3.4 家庭圈界面

### 5.3.5 Commont(评论家庭圈界面)

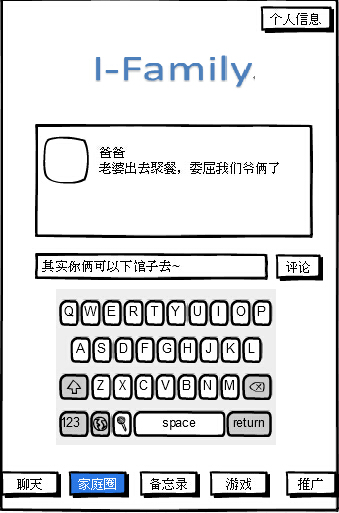


图 5.3.5 评论家庭圈界面

### 5.3.6 Memo(备忘录界面)

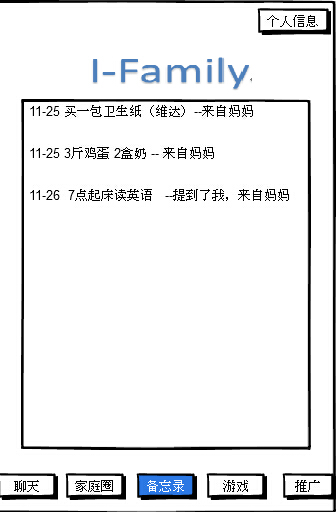


图 5.3.6 备忘录界面

### 5.3.7 Promotion for game(游戏推广界面)



图 5.3.7 游戏推广界面

### 5.3.8 Promotion for applications(推广界面)



图 5.3.8 推广界面

## 5.4 Interface Design（接口设计）

### 5.4.1 Interface Specification（接口说明）

我们的系统分为前端和后台。前端向后台发布数据请求，后台将数据返回给前端。下面根据一个结构图来更加直观的从数据的角度看到我们的接口设计。

调用数据

发送请求

系统

后台

用户

调用应用

传送信息

反馈

发送数据

图 5.4.1 接口说明

# 6. General Design: Architecture Design（总体设计）

## 6.1 General Design

### 6.1.1 User information module用户信息模块

User

顾客

View

注册界面

登陆界面

用户信息界面

Controller

账号注册

账号登陆

用户信息修改

Data

用户信息

家庭信息

群组信息

用户信息模块架构设计

登陆界面

注册界面

**注册：**

**登录：**

登陆界面

首页

用户信息界面

**个人信息：**

### 6.1.2 Chat module(聊天模块)

User

顾客

View

聊天界面

好友界面

Controller

好友聊天

查看好友

Data

聊天信息

好友信息

聊天模块架构设计

首页

聊天界面

**聊天：**

### 6.1.3 Family circle module(家庭圈模块)

User

顾客

View

家庭圈界面

评论界面

Controller

查看家庭圈

查看评论

发表评论

Data

家庭圈信息

家庭圈模块架构设计

评论信息

发布家庭圈

首页

朋友圈界面

**查看家庭圈：**

首页

朋友圈界面

评论界面

**评论家庭圈：**

### 6.1.4 Memo module(备忘录界面)

User

顾客

View

查看备忘录界面

编辑备忘录界面

Controller

查看备忘录

编辑备忘录

Data

备忘录模块架构设计

备忘录信息

首页

朋友圈界面

评论界面

**查看备忘录：**

### 6.1.5 Game module(游戏模块)

User

顾客

View

游戏界面

得分界面

Controller

进行游戏

查看得分

Data

游戏信息

得分信息

游戏模块架构设计

首页

游戏界面

**进行游戏:**

首页

游戏界面

得分界面

**查看得分：**

### 6.1.6 Promotion module(推广模块)

User

顾客

View

推广界面

Controller

查看推广

Data

推广信息

推广模块架构设计

首页

推广界面

**查看推广：**

# 7. detail design（详细设计）

## 7.1 Presentation layer(表示层)

### 7.1.1 User information module(用户信息模块)

主页

用户登录界面

用户注册界面

用户信息界面

填写用户信息

确认注册

填写用户名\密码

登录成功

登录失败

注册信息重复或不完整

图 7.1.1 表示层-用户信息

### 7.1.2 Chat module(聊天模块)

提示

选择好友

主页

用户登录

好友列表

发送消息

接收消息

聊天界面

重选好友

失败

成功

图 7.1.2 表示层-聊天模块

### 7.1.3 Family circle module(家庭圈模块)

更新

查看评论

发表评论

删除评论

查看状态

发表状态

删除状态

主页

用户登录

家庭圈界面

评论界面

更新

图 7.1.3 表示层-家庭圈模块

### 7.1.4 Memo module(备忘录模块)

更新

主页

用户登录

备忘录界面

查看备忘录界面

编辑备忘录界面

编写备忘录内容

设定提醒时间和人物

图 7.1.4 表示层-备忘录模块

### 7.1.5 Game module(游戏模块)

游戏界面

选择游戏

开始游戏

结束游戏

计算得分

得分界面

主页

用户登录

更新

图 7.1.5 表示层-游戏模块

### 7.1.6 Promotion module(推广模块)

主页

用户登录

推广界面

游戏推广

应用推广

下载

更新

图 7.1.6 表示层-推广模块

## 7.2 Control layer(控制层)

### 7.2.1 User information module(用户信息模块)

更新

查看

用户

个人信息界面

登陆界面

注册界面

用户信息数据

图 7.2.1 控制层-用户信息

### 7.2.2 Chat module(聊天模块)

查看

更新

用户

聊天界面

好友界面

聊天数据

好友数据

图 7.2.2 控制层-聊天模块

### 7.2.3 Family circle module(家庭圈模块)

更新

查看

家庭圈状态数据

查看

评论数据

用户

评论界面

家庭圈界面

更新

图 7.2.3 控制层-家庭圈模块

### 7.2.4 Memo module(备忘录模块)

更新

查看

用户

编辑备忘录界面

查看备忘录界面

备忘录数据

图 7.2.4 控制层-备忘录模块

### 7.2.5 Game module(游戏模块)

调用

更新

用户

得分界面

游戏界面

得分数据

后台游戏应用

查看

图 7.2.5 控制层-游戏模块

### 7.2.6 Promotion module(推广模块)

查看

用户

推广界面

推广APP数据

图 7.2.6 控制层-推广模块

## 7.3 Data layer(数据层)

### 7.3.1 User information module(用户信息模块)

|  |
| --- |
| **用户** |
| -用户编号 int |
| +获取用户信息() String  +获取修改信息权限() String  +修改用户信息() void |

图 7.3.1 数据层-用户信息

### 7.3.2 Chat module(聊天模块)

|  |
| --- |
| **用户** |
| -用户编号 int |
| +获取用户信息() String  +获取用户好友() String  +获取聊天权限() String |

|  |
| --- |
| **聊天信息** |
| -用户编号 int  -好友编号 int |
| +发送消息() String  +接收消息() String  +删除消息() String |

图 7.3.2 数据层-聊天模块

### 7.3.3 Family circle module(家庭圈模块)

|  |
| --- |
| **家庭圈** |
| -用户编号 int |
| +获取家庭圈信息() String  +获取用户好友() String  +获取评论权限() String |

|  |
| --- |
| **评论** |
| -用户编号 int  -好友编号 int  -状态编号 int  -评论编号 int |
| +获取评论信息() String  +获取用户好友() String  +获取评论权限() String  +删除评论() void |

|  |
| --- |
| **状态** |
| -用户编号 int  -状态编号 int |
| +获取状态信息() String  +获取用户好友() String |

图 7.3.3 数据层-家庭圈模块

### 7.3.4 Memo module(备忘录模块)

|  |
| --- |
| **备忘录** |
| -用户编号 int  -备忘录编号 int |
| +获取备忘录信息() String  +获取家庭好友() String  +获取修改备忘录权限() String  +修改备忘录() void |

图 7.3.4 数据层-备忘录模块

### 7.3.5 Game module(游戏模块)

|  |
| --- |
| **得分** |
| -用户编号 int  -游戏编号 int |
| +获取游戏信息() String  +获取用户好友() String  +获取得分信息 () String  +修改得分() void  +排序得分() void |

|  |
| --- |
| **游戏** |
| -游戏编号 int  -游戏厂商 String |
| +获取游戏信息() String  +获取应用权限() String  +开始游戏() void  +结束游戏() void |

图 7.3.5 数据层-游戏模块

### 7.3.6 Promotion module(推广模块)

|  |
| --- |
| **推广** |
| -推广编号 int  -厂商 String |
| +获取应用信息() String  +获取推广权限() String  +下载() void  +获取厂商信息() String |

图 7.3.6 数据层-推广模块

# 8. Data Modeling and Design（数据模型和设计）

## 8.1 Conceptual model（概念模型）

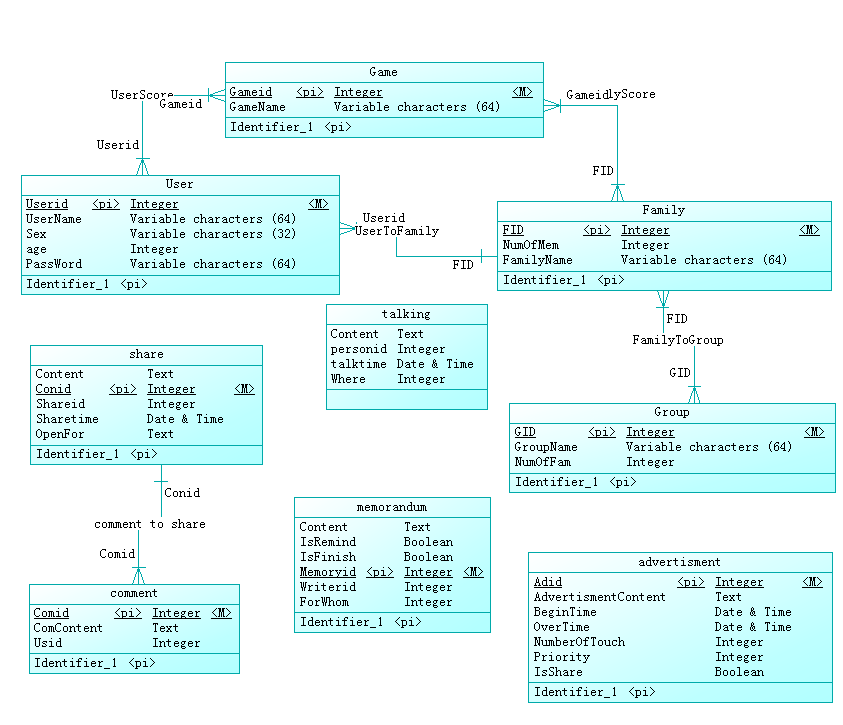


图 8.1 概念模型

## 8.2 Logical model（逻辑模型）

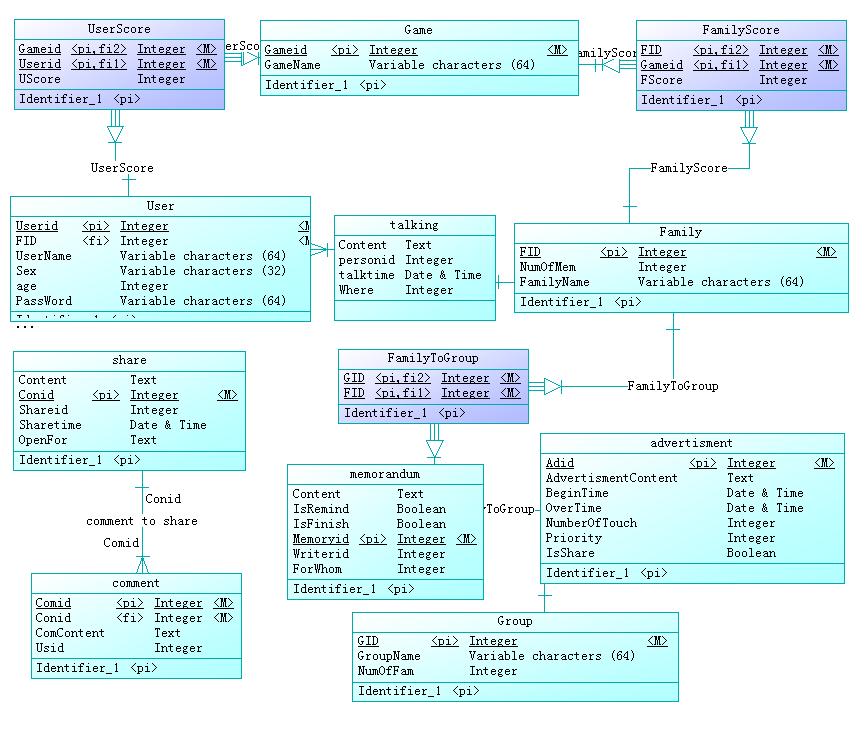


图 8.2 逻辑模型

## 8.3 Physical model（物理模型）

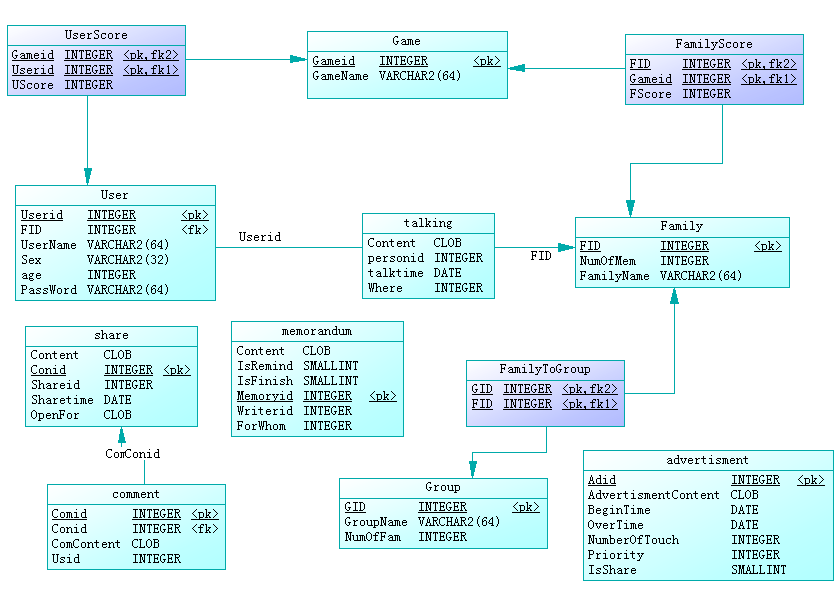


图 8.3 物理模型

## 8.4 Form Introduction（表介绍）

|  |  |
| --- | --- |
| **表** | **说明** |
| User | 存储用户信息 |
| Family | 存储家庭信息 |
| Group | 存储群组信息 |
| Share | 存储家庭圈信息 |
| Comment | 存储评论信息 |
| Advertisement | 存储推送广告信息 |
| Memorandum | 存储备忘录信息 |
| Game | 存储游戏信息 |
| Talking | 存储聊天信息 |
| UserScore | 存储用户单人得分信息 |
| FScore | 存储家庭得分信息 |

**备注：**

（1）“talking”表中的“Where”属性是聊天对象的id

（2）“share”表中的“OpenFor”属性是朋友圈的可见对象的id

（3）“memorandum”表中的“ForWhom”属性是备忘录提醒的对象

# 9. Input and Output Design（输入输出流设计）

## 9.1 Normal user Register(普通用户注册)

个人用户

注册用户

注册过程

注册成功

### 9.1.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 普通用户-用户基本信息 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 用户输入数据 | 用·户id | Int | 非空且唯一 | 单个输入 |
| 用户密码 | String | 非空 | 单个输入 |
| 用户名称 | String | 非空且唯一 | 单个输入 |
| 用户年龄 | Int | 非空 | 单个输入 |
| 用户性别 | String | 男或女 | 单个输入 |

### 9.1.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 普通用户-用户基本信息 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出数据 | 用户id | Int | 客户端输出 | 单个输出 |
| 用户密码 | String | 客户端输出 | 单个输出 |
| 用户名称 | String | 客户端输出 | 单个输出 |
| 用户年龄 | Int | 客户端输出 | 单个输出 |
| 用户性别 | String | 客户端输出 | 单个输出 |

## 9.2 Householder Register(家庭户主注册家庭账号)

家庭户主

注册家庭用户

注册过程

注册成功

### 9.2.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 家庭用户-用户基本信息 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 用户输入数据 | 家庭id | Int | 非空且唯一 | 单个输入 |
| 家庭名称 | String | 非空且唯一 | 单个输入 |
| 内部  输入 | 家庭成  员数量 | Int | 非空 | 单个输入 |

### 9.2.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 家庭用户-用户基本信息 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出  数据 | 家庭id | Int | 终端输出 | 单个输出 |
| 家庭名称 | String | 终端输出 | 单个输出 |
| 家庭成  员数量 | Int | 终端输出 | 单个输出 |

## 9.3 Adding to family group(个人申请加入家庭群)

个人用户

申请加入家庭群

申请过程

申请成功

### 9.3.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 普通用户-个人申请加入家庭群 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 用户输入数据 | 家庭id | Int | 非空且唯一 | 单个输入 |
| 内部  输入 | 家庭成  员数量 | Int | 非空 | 单个输入 |
| 用户id | Int | 非空且唯一 | 单个输入 |

### 9.3.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 普通用户-个人申请加入家庭群 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出  数据 | 家庭id | Int | 终端输出 | 单个输出 |
| 家庭名称 | String | 终端输出 | 单个输出 |
| 家庭成  员数量 | Int | 终端输出 | 单个输出 |
| 用户id | Int | 终端输入 | 单个输出 |

## 9.4 Adding friends among families(家庭间互加好友)

家庭户主

申请关注家庭

申请过程

关注成功

同意其他家庭申请

同意过程

### 9.4.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 家庭管理员-家庭互加好友 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 用户输入数据 | 群组id | Int | 非空且唯一 | 单个输入 |
| 内部  输入 | 群组成  员数量 | Int | 非空 | 单个输入 |
| 家庭id | Int | 非空且唯一 | 单个输入 |

### 9.4.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 家庭管理员-家庭互加好友 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出  数据 | 群组id | Int | 终端输出 | 单个输出 |
| 群组名称 | String | 终端输出 | 单个输出 |
| 群组成  员数量 | Int | 终端输出 | 单个输出 |
| 家庭id | Int | 终端输入 | 单个输出 |

## 9.5 Chat management(聊天管理)

个人用户

选择聊天对象

聊天过程

通信成功

### 9.5.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-聊天 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 用户输入数据 | 聊天内容 | TXT | 非空 | 单个输入 |
| 内部  输入 | 聊天时间 | TIME | 非空 | 单个输入 |
| 聊天对象 | Int | 非空且唯一 | 单个输入 |
| 用户id | Int | 非空且唯一 | 单个输入 |

### 9.5.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-聊天 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出  数据 | 聊天内容 | TXT | 目标终端 | 批量输出 |
| 聊天时间 | TIME | 目标终端 | 批量输出 |
| 用户id | Int | 目标终端 | 批量输出 |

## 9.6 Memo management(备忘录管理)

个人用户

查看备忘录

更新备忘录过程

创建新备忘录过程

更新备忘录成功

### 9.6.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-备忘录管理 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 用户  输入  数据 | 是否需要提醒 | Bool | 非空 | 单个输入 |
| 提醒人 | Int | 非空且唯一 | 单个输入 |
| 备忘录  内容 | TXT | 非空 | 单个输入 |
| 内部  输入 | 是否已完成 | Bool | 非空 | 单个输入 |
| 备忘录id | Int | 非空且唯一 | 单个输入 |
| 撰写人id | Int | 非空且唯一 | 单个输入 |

### 9.6.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-备忘录管理 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出内容  输出  内容 | 备忘录  内容 | TXT | 所有家庭成员终端 | 批量输出 |
| 备忘录id | Int | 所有家庭成员终端 | 批量输出 |
| 撰写人id | Int | 所有家庭成员终端 | 批量输出 |

## 9.7 Family circle management(家庭圈管理)

个人用户

查看家庭圈

查看家庭圈过程

发布家庭圈过程

家庭圈更新成功

### 9.7.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-家庭圈管理 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 用户输入数据 | 分享内容 | TXT | 非空 | 单个输入 |
| 内部  输入 | 分享时间 | TIME | 非空 | 单个输入 |
| 对谁公开 | Int | 非空且唯一 | 单个输入 |
| 分享人id | Int | 非空且唯一 | 单个输入 |

### 9.7.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-家庭圈管理 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出  数据 | 分享内容 | TXT | 目标终端 | 批量输出 |
| 分享时间 | TIME | 目标终端 | 批量输出 |
| 分享人id | Int | 目标终端 | 批量输出 |

## 9.8 Game management(游戏管理)

个人用户

选择游戏功能

游戏过程

游戏结束

### 9.8.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-游戏管理 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 内部  输入 | 游戏id | Int | 非空且唯一 | 单个输入 |
| 游戏名称 | String | 非空且唯一 | 单个输入 |
| 个人得分 | Int | 非空 | 批量输入 |
| 家庭得分 | Int | 非空 | 批量输入 |

### 9.8.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-游戏管理 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出  内容 | 游戏id | Int | 终端输出 | 批量输出 |
| 游戏名称 | String | 终端输出 | 批量输出 |
| 个人得分 | Int | 终端输出 | 批量输出 |
| 家庭得分 | Int | 终端输出 | 批量输出 |

## 9.9 Promotion for family applications(家庭推广)

个人用户

查看推广信息

查询过程

查询结束

### 9.1.1 Input design(输入设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-家庭推广 | | | | |
|  | 数据名称 | 数据类型 | 约束条件 | 输入方式 |
| 内部  输入 | 广告id | Int | 非空且唯一 | 单个输入 |
| 点击数量 | Int | 非空 | 单个输入 |
| 是否正在被推广 | BOOL | 非空 | 单个输入 |
| 优先级 | Int | 非空 | 单个输入 |

### 9.1.2 Output design(输出设计)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 个人用户-家庭推广 | | | | |
|  | 数据名称 | 数据类型 | 输出媒介 | 输出方式 |
| 输出  内容 | 广告内容 | TXT | 终端输出 | 单个输出 |
| 开始时间 | TIME | 终端输出 | 单个输出 |
| 结束时间 | Time | 终端输出 | 单个输出 |
| 点击量 | Int | 终端输出 | 单个输出 |

# 10. Error handling(错误处理)

## 10.1 Target(目标)

（1） 建立错误完善的错误处理机制。

（2） 错误侦查能够及时发现错误，并提交系统处理。

（3） 错误处理系统在获取错误提交后，能够自动处理错误，并尽量不中断用户使用程序，比如频繁崩溃等问题，而是多线程解决错误时让错误界面自动跳转至正常界面，比如返回自动上一级等。

（4） 预加提醒，在用户操作前，提示用户，防止不正当输入。即使用户输入不合法字符，也要能够及时提醒用户，并不影响用户。

（5） 拥有一套较为完善的异常处理机制

## 10.2 Steps of handling errors(错误处理步骤)

### 10.2.1 Process of handling errors(错误处理过程)

1. 错误检测
2. 抛出异常
3. 错误类型判断

（4）错误提交

（5）错误处理

（6）错误报告并上传

（7）显示错误处理情况

### 10.2.2 Flow chart of handling errors(错误处理流程图)

检测错误

错误类型判断

提交错误

错误处理

错误报告

上传错误

显示错误处理情况

图 10.2 错误处理流程图

## 10.3 Model of handling errors(错误处理模式)

### 10.3.1 Relationship because of errors(错误的引起关系)

**Raises**

Fault

Error

Failure

**Causes**

**Causes**

**detects**

**detects**

Failure

Exception

图 10.3.1 错误的引起关系图

**Fault:** 错误操作。用户的错误操作将导致程序错误。

**Error：**程序错误。 程序错误是由用户错误操作导致的，并有可能导致用户此次的操作失败。能够被错误检测器检测。

**Failure：**操作失败。 由程序错误导致的操作失败。同时也能够被检测。和程序错误一样被检测到后由错误处理机制判断错误类型处理并抛出异常。

**Detector:** 错误检测。 检测整个程序生命周期出现的程序错误和操作失败。并抛出异常

**Exception：** 异常。 较为完整的异常处理机制，为处理错误提供方便。

### 10.3.2 Handle errors-frame model(错误处理-框架模式)

Error Dialog

Error Protocol

Error handler

**……**

Client

Error

图 10.3.2 错误处理-框架模式

## 10.4 Architecture （错误树）

Error Handling

Framework

Error Object

Error Object

Error Types and Structure

Exception Abstraction

Error Wrapper

Error Integration

Multithreading

Exception

Handling

Multithreading

Error Traps

Assertion Checking

Object

Error Detection

Centralized Error **Logging**

Backtrace

Error Logging

Error Handler

Error Dialog

Default Error

Handling

Resource

Preallocation

Checkpoint Restart

Error Logging

图 10.4 错误树

## 10.5 Error type（错误类型）

|  |  |  |
| --- | --- | --- |
| **错误操作** | **处理机制** | **错误编号** |
| **输入非法字符** | **自动略过，提示用户输入非法字符并要求用户重新输入** | **1001** |
| **无法登录** | **核对用户输入信息与数据库信息，若是正常账号密码错误，提示用户，若是数据库与客户端信息不匹配，提示用户系统自动恢复中。** | **1002** |
| **无权限操作** | **用户无权限操作家庭信息，给出提示，拒绝用户操作，并不做其他操作。** | **1003** |
| **网络连接错误** | **自动检测网络连接，并在客户端显示当前网络连接状况，当网络信号不好，或是断网时及时提醒用户。** | **1004** |
| **程序崩溃** | **程序因不明异常错误崩溃后，提示用户，回滚未完成操作，并及时存储更新用户之前已修改的数据** | **1005** |
| **服务器瘫痪** | **尽量保证不出现服务器瘫痪的情.做好备份处理机制启用应急况.** | **1006** |

## 10.6 Error prevention（错误预防）

(1) 对客户端定期更新维护。

（2）对用户上传错误及时处理，修改程序漏洞

(3) 定期对数据库备份，维护。

(4) 做好应急方案

# 11. Poster（海报）



、

# 12. Conclusion

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 贡献 |
| 王畅 | 12301106 | 0.2 |
| 白亚楠 | 12301088 | 0.21 |
| 游程 | 12301112 | 0.22 |
| 赵家进 | 12301113 | 0.16 |
| 许若昕 | 12301109 | 0.21 |

第二次作业也告一段落，相比于第一次作业这一次的任务还是非常多的。这也导致了我们的任务计划发生了改变。我们调整了写文档流程，从小组一起讨论完成改为每个人领取任务，完成之后再一起讨论修改，这样才能在有限的时间内更好地完成任务。事实证明这样也的确是比较有效率。

但相比于第一次作业这一次的调整也带来了新的问题。因为每个部分都是又不同人完成的，其他人对这个部分并不是很了解，所以讨论的时候意见分歧也是比较大。每一个部分都需要讨论很长时间，完善起来也比较麻烦，因为有些部分是承上启下的所以要和其他的内容相吻合才可以。这样也带来了另一个问题，因为每一个同学的时间都是不一样的，这样如果后一个部分需要前一个部分的内容，那么就得商量完成时间才行。这主要还是一开始对文档内容不理解分工不好导致的。

相比较于教训，这一次更多的收获。因为这一次作业的内容详细，我们也从方方面面了解了老师上课所讲的，以及真正的系统分析需要做的东西有哪些。在不断的查阅资料，比较例子中我们也充实了自己对于系统分析的知识。到了最后，我们感觉不仅仅是这个作业，对于其他方面也会或多或少去系统地分析事务。这在别的课的作业中都有体现。因此，这次作业不仅锻炼了我们的专业知识，还培养了我们的思维方式。这就是我们最大的收获。