rBook需求规约

修订历史：

2018年12月6日星期四

版本：第4版

说明：

第1版：需求规约。

第2版：优化整体描述和部分功能需求的描述，合并冗余用例。

第3版：对软件语境和用例进行了 一定的补充。

第4版：优化用例描述和结构，合理化一些非功能性需求。

## 介绍

rBook是一个基于关系的记账工具APP，它提供了一对一关系记账，匿名记账，以及群组记账三大功能。其中一对一关系记账用于解决朋友之间大大小小资金往来不便于记录，不方便理清的问题，同时有系统自动的债务提醒免去朋友间开口讨钱的尴尬。匿名记账类似于一对一记账，但对象不是真实用户而是自定义的虚拟对象，用于单方面记录不方便明面谈论的人情债。群组记账则针对于4到20人的小团体内部的资金往来提供记录、计算和提醒功能。

## 整体描述

本项目是基于用户社会关系的记账工具，特点在于维护用户在传统环境下碍于人情等社会因素不便进行的资金往来相关记录、计算、提醒工作，如朋友间的小额借款，亲戚间的彩礼红包往来，团队出行时产生的垫付行为等，在提供记账的基本功能的基础上，也力求不因此影响用户的相关社会关系。

本项目客户端运行于Android系统。

## 系统与其他系统的接口

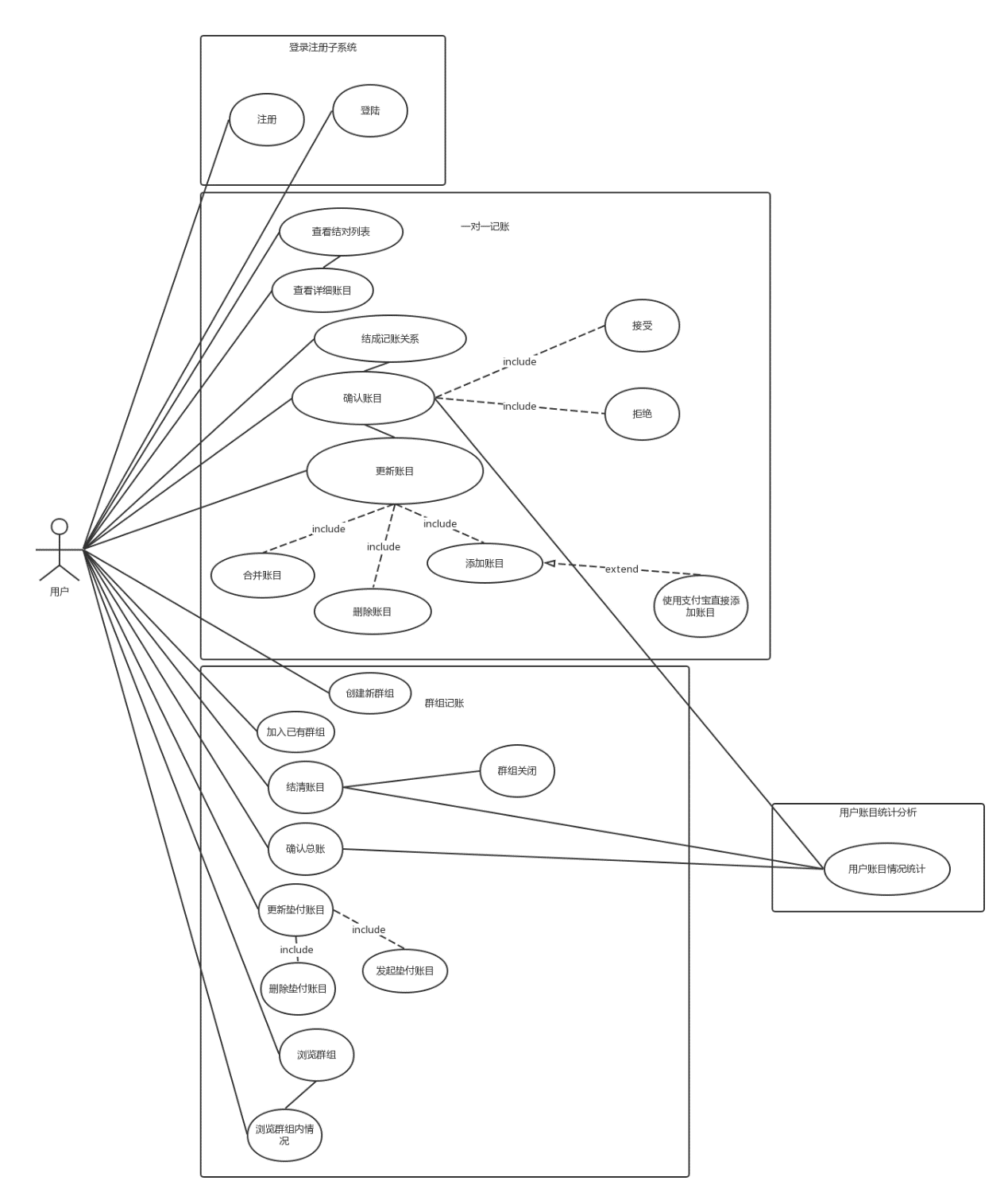
本系统的客户端程序运行于Android系统，会使用到Android系统中相关的推送服务接口进行提醒功能的实现。

此外，本系统的服务端使用Apache Tomcat为容器进行相关部署，以SpringBoot作为基本框架进行开发。

数据持久使用Neo4j图数据库。

## 主要的功能需求描述

Use Case Diagram用例图



1. **注册**

参与者：用户

前置条件：无

后置条件：若用例成功，则在用户数据中新增参与者的用户信息，并使其直接进入登录状态。反之，系统不做变化

基本事件流：

当参与者打开注册功能时，用例启动

1. 系统提示参与者输入新的用户名和密码及身份认证信息
2. 参与者输入欲注册账号之用户名和密码及身份认证信息
3. 服务端验证用户名，密码及身份信息的合法性，若成功则注册成功，同时使新用户登陆

异常事件流：

用户输入的用户名或密码不合法，或客户端与服务端间的交互受阻，用例失败，用户回到本用例开头

1. **登陆**

参与者：用户

前置条件：无

后置条件：若用例成功，则用户可以进入本程序系统进行之后的用例操作。反之，系统不做变化

基本事件流：

当用户打开登陆时，用例启动

1. 客户端系统提示用户输入欲登陆账号之用户名与密码
2. 用户输入用户和密码
3. 服务端接受登陆信息并验证用户输入用户名与密码之有效性，若成功则使对应用户登陆到系统之中并返回成功信息

异常事件流：

用户给出的用户名与密码无效或不存在，或客户端与服务端之间的通信受阻，用例失败，客户端显示失败信息并使用户回到本用例开头

1. **一对一记账：查看结对列表**
2. **一对一记账：查看详细账目**
3. **一对一记账：结成记账关系**

参与者：用户（双方，以下记主动方为用户A，被动方为用户B）

前置条件：双方用户均已注册且登陆

后置条件：若用例成功，则用户A与B间新增待确认的关系。反之，系统不做变化

基本事件流：

用户A打开新增记账关系开始用例：

1. 客户端系统提示用户输入对方用户名
2. 用户A输入用户B之用户名搜索并发送首次添加账目请求
3. 系统新增该条待确认的关系请求

异常事件流：

第2步中，用户A输入的用户名不存在，则回到1,；第2步中，用户A客户端与服务端的通信受阻，回到1

1. **一对一记账：更新账目**

参与者：用户（双方）

此用例包含的用例：

7-一对一记账：添加账目

9-一对一记账：合并账目

10-一对一记账：删除账目

1. **一对一记账：添加账目**

参与者：用户

前置条件：双方用户已经结成记账关系

后置条件：若用例成功，则用户A与用户B间新增一笔待确认的添加账目，账目内容包括资金数目，方向，说明与时间。反之，系统不做变化

基本事件流：

用户A打开新增账目功能，开始用例

1. 客户端系统提示用户输入新账目的资金数目、方向（出/进）、账目说明
2. 用户输入要求的信息并提交
3. 服务端收到请求后新增该条待确认添加账目

异常事件流：

第2步中，用户客户端与服务端的通信受阻，回到1；第3步中，用户提交请求后服务端判断该对关系下已有账目达到上限（100条），则关闭本次用例并提示达到上限，提示用户合并账目；第2步中，用户客户端与服务端的通信受阻，该步失败，需要重新进行

1. **一对一记账：使用支付宝直接转账添加账目**

此用例为5. 一对一记账：添加账目的扩展用例

参与者：用户（双方，以下记主动方为用户A，被动方为用户B）

前置条件：双方用户已经结成记账关系

后置条件：若用例成功，则用户A与用户B间新增一笔账目，账目内容包括资金数目，方向，说明与时间。反之，系统不做变化

基本事件流：

用户A打开新增账目功能，开始用例

1. 客户端系统提示用户A输入新账目的资金数目、方向（出/进）、账目说明
2. 用户A输入要求的信息并提交，选择支付宝直接转账
3. 用户A跳转至支付宝并完成转账
4. 系统新增记录至该结对关系下，无需对方确认

异常事件流：

第2步中，用户A客户端与服务端的通信受阻，回到1；第2步中，用户A提交请求后服务端判断该对关系下已有账目达到上限（100条），则关闭本次用例并提示达到上限，提示用户合并账目；第3步中，转账失败或中断，用例失败，退出用例

1. **一对一记账：合并账目**

参与者：用户

前置条件：双方用户已经结成记账关系且关系下存在至少两条账目

后置条件：若用例成功，双方的关系下被选择的账目将被标记待合并，取而代之一条新的账目会被新增为待合并结果，其资金数目为所有合并账目之和，说明为合并发起方指定的说明，账目时间为合并的时间。反之，系统不做变化

基本事件流：

用户打开合并账目功能，开始用例：

1. 客户端系统要求用户选择需要合并的账目并输入新的说明信息
2. 用户复选需要合并的账目并输入新的说明信息提交
3. 系统计算出新账目的资金数目，添加新待确认账目（其中包括用户指定的新说明信息）至关系并标记待合并的账目

异常事件流：

第2步中，用户客户端与服务端的通信受阻，回到1；第3步中，系统检测到该结对关系下已经存在一个待处理的合并请求，返回失败信息，用例失败

1. **一对一记账：删除账目**

参与者：用户

前置条件：双方用户已经结成记账关系

后置条件：若用例成功，则用户A与用户B间指定账目标记为待确认删除。反之，系统不做变化

基本事件流：

用户A打开删除账目功能，开始用例

1. 客户端系统提示用户A选择欲删除的账目
2. 用户A选中欲删除的账目并提交
3. 服务端确认该请求并标记其为待确认删除

异常事件流：

第2步中，用户A客户端与服务端的通信受阻，回到1

1. **一对一记账：确认账目**
2. **一对一记账：接受账目**
3. **一对一记账：拒绝账目**
4. **群组记账：创建新群组**

参与者：用户

前置条件：用户已经注册并登录系统

后置条件：若用例成功，则创建一个新的群组和唯一的邀请码，且发起用户将在群组内。反之，系统不做变化

基本事件流：

用户打开创建新群组功能，用例开始：

1. 客户端系统提示用户输入新的群组名称
2. 用户输入新的群组名称并提交
3. 服务端系统受到用户的新群组请求并处理，在系统下创建新的群组并生成唯一群组码
4. 服务端将成功信息，群组信息以及群组码返回客户端显示

异常事件流：

第2步中，客户端与服务端之间的通讯受阻，回到第1步；第4步中，客户端与服务端之间的通讯受阻，用例关闭并显示网络错误信息，但用户之后在网络状况良好的情况下查看自己所处群组时可以获取群组信息

1. **群组记账：加入已有群组**

参与者：用户

前置条件：用户已经注册并登录系统，且已经通过其他方式得知群组发起人的群组邀请码

后置条件：若用例成功，则用户加入目标群组，且在群组中有一额外的备注名。反之，系统不做变化

基本事件流：

用户打开加入群组功能，开始用例：

1. 客户端系统提示用户输入邀请码
2. 用户输入邀请码和备注名后并提交
3. 服务端系统根据唯一邀请码找到目标群组，并将用户加入其中
4. 服务端向客户端返回成功信息并显示

异常事件流：

在第2步中，客户端与服务端之间的通信受阻，回到第1步；在第3步中，服务端判断用户的邀请码不合法或者没有对应群组，直接返回失败信息；第3步中，服务端根据邀请码找到群组后，发现该群组无法加入新成员（群组已满20人或者群组已经进行了确认总账），返回失败信息；第4步中，服务端与客户端的通信受阻，客户端系统显示网络错误信息，但用户下次访问自身相关群组时可以看到本次用例的结果

1. **群组记账：更新垫付账目**

参与者：用户（垫付人）

此用例包含的用例：

13-群组记账：发起垫付账目

14-群组记账：删除垫付账目

1. **群组记账：发起垫付账目**

参与者：用户（垫付人）

前置条件：用户已经登录且加入了群组，且该群组尚未进行确认总账

后置条件：若用例成功，则在该群组下新增一笔属于该用户的垫付账目，包括发起人（该用户），目标用户（群组中指定的其他需要来结清该笔账目的用户，可能有多个），资金数目

基本事件流：

用户在群组内打开发起账目功能，用例开始：

1. 客户端系统提示用户输入资金数目与目标结清用户
2. 发起用户输入资金数目并选择目标用户后提交
3. 服务端接受到用户发起的账目，在群组下新增该笔账目

异常事件流：

在第2步中，客户端与服务端之间的通信受阻，回到第1步；在第3步中，服务端收到请求后，发现该群组无法加入新的账目（账目数量到达上限），则关闭用例并向用户返回错误或失败信息

1. **群组记账：删除垫付账目**

参与者：用户（垫付人）

前置条件：用户在群组中，用户已发起某特定垫付账目，该群组尚未进行确认总账

后置条件：若用例成功，则在该群组下删除一笔属于该用户的垫付账目

基本事件流：

用户在群组内打开删除垫付账目功能，开始用例

1. 客户端系统提示用户选择欲删除该群组内该用户发起的垫付账目
2. 用户选中欲删除的垫付账目并提交，发送请求至服务端
3. 服务端收到来自用户的删除请求，在该群组内删除此垫付账目

异常事件流：

在第2步中，客户端与服务端之间的通信受阻，回到第1步；第3步中，服务端与客户端的通信受阻，客户端系统显示网络错误信息，但用户下次访问自身相关群组时可以看到本次用例的结果

1. **群组记账：确认总账**

参与者：用户

前置条件：用户已经加入群组，且群组内所有垫付人都已发起了自己的账目（本项不由本系统保证）

后置条件：若用例成功，则该群组将不可进行添加或删除账目操作，由系统计算群组内所有用户账目收支信息，群组内所有用户的客户端上显示信息，包括计算后该用户和群组内其他用户（若有资金往来）的资金流动信息，以及一个总体收支信息。

基本事件流：

任意一个群组内用户在群组内打开确认总账功能，开始用例

1. 客户端向服务器发送该群组确认总账请求，服务器向群组内所有其他用户发送确认通知
2. 在群组内所有用户都点击确认之后，服务器将该群组标记为已确认总账状态，统计该群组内所有账目信息，计算该群组内所有用户的账目收支信息
3. 服务器向该群组内所有用户发送账目收支信息，在用户客户端上显示最终账目信息。

异常事件流：

在第1步，发送请求时通信受阻，客户端显示失败信息，回到用例开始；在第3步中，服务器向客户端发送账目信息时通信受阻，用例关闭并显示网络错误信息，但用户之后在网络状况良好的情况下可查看到最终和群组内其他用户的账目信息

1. **群组记账：结清账目**

参与者：用户（垫付账目中被指定结算的用户）

前置条件：用户在群组中，被指定参与垫付账目结算，且该群组已经进行了确认总账

后置条件：若用例成功，用户将其在群组内和其他一个用户（有资金往来）的账目结清

基本事件流：

用户在收到确认总账后发来的账目信息后打开结清账目功能，用例开始

1. 用户选择和群组内另一用户的账目进行结清，点击结清按钮，发送请求至服务端（真实资金往来不通过系统）
2. 服务端接收请求，将该群组内这一账目标记为已结清，并检查是否该群组内所有账目已结清
3. 服务端向客户端发送成功信息，客户端对应账目状态显示为已结清

异常事件流：

在第1步中，发送请求时通信受阻，客户端显示失败信息，回到用例开始；在第3步中，服务器向客户端发送成功信息时通信受阻，用例关闭并显示网络错误信息，但用户之后在网络状况良好的情况下可以看到该账目已结清

1. **群组记账：群组关闭**

参与者：无外部角色参与

前置条件：群组进行确认总账且以完成结清所有内部账目

后置条件：群组关闭，系统将其删除

基本事件流：

在一个结清账目用例中，再将其对应账目标记为已结清后，服务端判断该群组内所有账目已结清，用例开始

1. 服务端将此群组从数据库中删除
2. 服务端向该群组内所有用户发送群组关闭信息，所有用户退出该群组。

异常事件流：

无

1. **群组记账：浏览群组**
2. **群组记账：浏览群组内情况**
3. **用户账目情况统计**

参与者：无外部角色参与

前置条件：用户的群组账目或一对一账目发生了变化

后置条件：用户的统计数据发生改变，并在客户端界面有所体现，在用户与其他用户发出结对请求时也会包含此信息

基本事件流：

在用户的群组账目或一对一账目发生了变化后，用例开始

1. 系统获取该用户在所有记账关系中的出度和入度，计算总账情况
2. 系统根据用户的总账情况计算其信用情况
3. 系统更新用户总账账目及信用情况

异常事件流：

无

## 非功能需求描述

1. 数据持久化中的需求

在一对一记账关系下，至多维护100条账目记录，如用户在上限后仍想要继续添加，则需要先合并原有账目

一个群组至多可加入成员20名，至多维护100条垫付账目（单个垫付人发起一次请求计算一次，无论目标结算用户有几人）

1. 性能需求

在服务端处理本系统中会出现的各种请求时，若无特殊情况，从服务端接收到请求开始计算，至处理完毕并回报信息，用时不超过2秒

1. 安全性需求

在涉及跳转第三方转账的敏感操作时，需要满足与第三方转账接口之间连接幂等性要求

用户登录密码在数据库中不使用明文保存