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，是传统记账本的一种替代。相对于已有的记账工具，其特点在于它基于关系，且需要账目关系的参与方一同来维护。它提供了一对一关系记账，以及群组记账两大功能，同时一对一关系记账的数据会参与系统的数据统计分析。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；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、计算结果的功能。

## 整体描述

本项目是基于用户关系的记账工具，特点在于账目参与方的共同维护。提供账目资金往来相关记录、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。

本项目预期用户与应用场景：

朋友之间，亲戚之间，同学之间的小额账目记录清算；

小规模个体商户之间的小额商务账目记录清算；

不大于20人的小团体内的群体账目记录清算等。

基本记账功能包括一对一间记账以及群组内记账及相关计算功能，此外有部分统计分析用户账目信息的功能。

本项目客户端运行于Android系统。

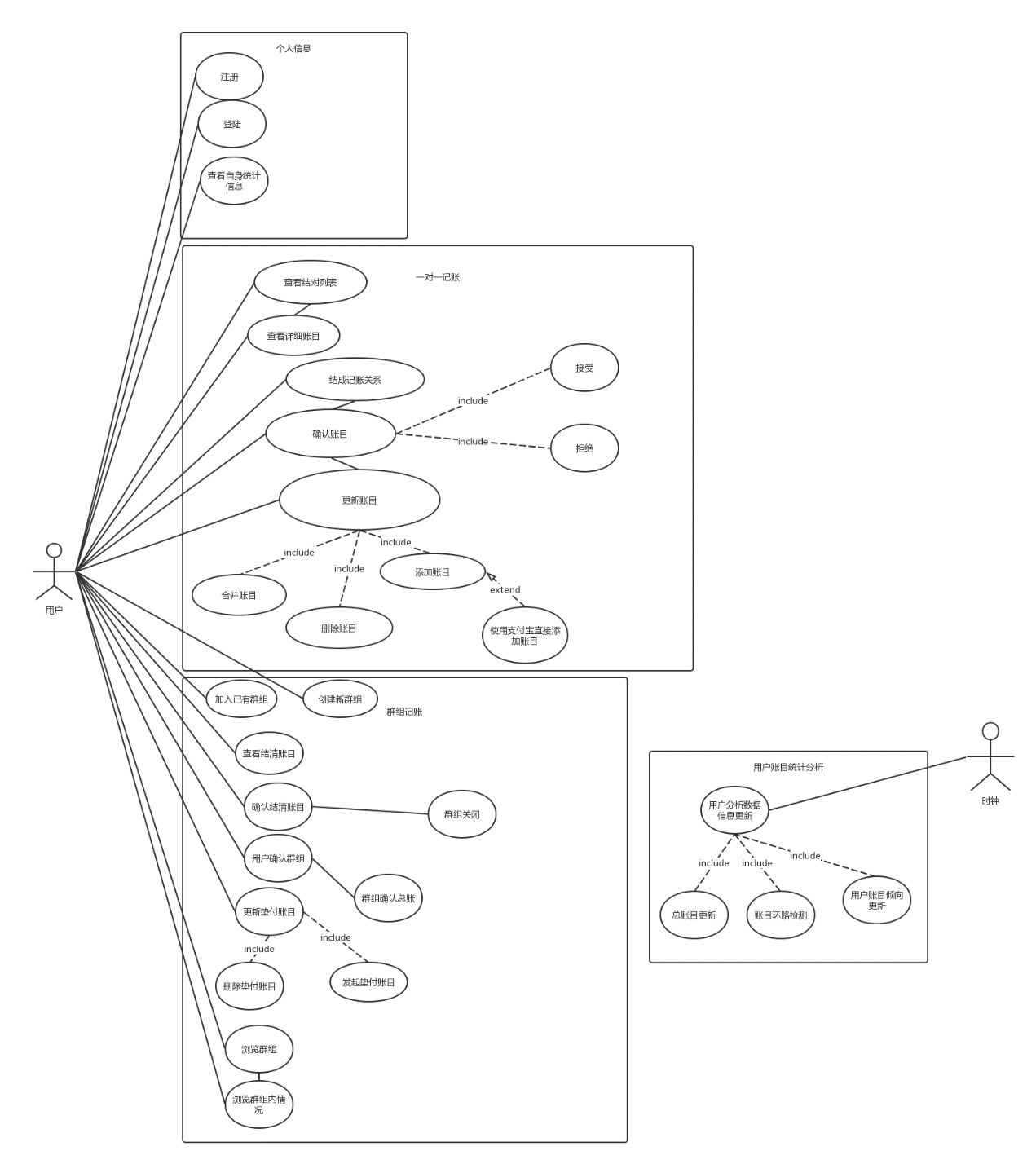
## 系统与其他系统的接口

本系统的客户端程序运行于Android系统，应有与运行的操作系统相对应的接口。

本系统需求中包含了通过第三方支付平台转账直接添加账目的需求，故应当有与第三方支付平台如支付宝相关功能的接口。

## 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：

第3步中，用户输入的用户名或密码不合法；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：

第3步中，用户给出的用户名与密码无效或不存在；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**

参与者：用户

前置条件：用户已登入

后置条件：若用例成功，则返回显示用户已经结对记账关系

基本事件流：

1. 用户发送浏览结对列表请求
2. 系统查询用户所有的一对一记账账目记录
3. 系统整合出用户参与的一对一记账结对关系并返回显示（包括总资金额，是否有新请求，对方用户信息，最后更新时间）

异常事件流：

用户端与服务端通信受阻，用例失败，显示错误信息

1. **一对一记账：查看详细账目**

参与者：用户

前置条件：用户已登入，并和目标对象结成记账关系或拥有结对请求

后置条件：若用例成功，则返回与目标对象间的所有一对一张目记录

基本事件流：

用户在3-查看结对列表中从返回的结对信息中选择一项，用例开始

1. 用户发送查看详细账目的请求
2. 服务端接受请求，查询用户和目标对象间的所有账目记录
3. 系统将记录返回显示（包括资金，方向，说明，时间，账目状态）

异常事件流：

用户端与服务端通信受阻，用例失败，显示错误信息

1. **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；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

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. 服务端收到请求后判断添加是否合法
4. 新增该条待确认添加账目

异常事件流：

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；用户客户端与服务端的通信受阻，用例失败

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：

用户A客户端与服务端的通信受阻，用例失败；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. 系统验证该合并是否合法
4. 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：

用户客户端与服务端的通信受阻，用例失败；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. 系统验证该删除请求是否合法
4. 服务端确认该请求并标记其为待确认删除

异常事件流：

用户客户端与服务端的通信受阻，用例失败；第3步中，系统验证删除请求不合法，用例失败，退出用例

1. **一对一记账：确认账目**

参与者：用户

此用例包含的用例：

12-一对一记账：接受账目

13-一对一记账：拒绝账目

1. **一对一记账：接受账目**

参与者：用户

前置条件：用户在与其他用户的记账关系中存在对方请求的未确认账目（状态为：待添加，待删除，待被合并后添加的合并账目）

后置条件：若用例成功，视目标未确认账目的未确认状态而定

基本事件流：

用户选择一条未确认账目，用例开始

1. 用户发送确认该条账目的请求
2. 服务端收到请求，验证该确认操作合法性
3. 服务端确认该账目，若其状态为待添加，则4；若其状态为待删除，则5；若其状态为待合并后添加的合并账目，则6
4. 系统将该账目置为确认状态
5. 系统删除该账目
6. 系统将该账目置为确认状态，并删除所有待合并账目

异常事件流：

用户端与服务端通信受阻，用例失败，显示错误信息；第2步中，系统验证该操作请求不合法(账目状态不符合等)用例失败，返回错误信息

1. **一对一记账：拒绝账目**

参与者：用户

前置条件：用户在与其他用户的记账关系中存在对方请求的未确认账目（状态为：待添加，待删除，待被合并后添加的合并账目）

后置条件：若用例成功，视目标未确认账目的未确认状态而定

基本事件流：

用户选择一条未确认账目，用例开始

1. 用户发送拒绝该条账目的请求
2. 服务端收到请求，验证该确认操作合法性
3. 服务端确认该账目，若其状态为待添加，则4；若其状态为待删除，则5；若其状态为待合并后添加的合并账目，则6
4. 系统删除该账目
5. 系统将该账目置为确认状态
6. 系统删除该账目，并将所有待合并账目置为确认状态

异常事件流：

用户端与服务端通信受阻，用例失败，显示错误信息；第2步中，系统验证该操作请求不合法(账目状态不符合等)用例失败，返回错误信息

1. **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：

客户端与服务端之间的通讯受阻，用例失败，退出用例

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，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群组码
2. 用户输入群组码后提交
3. 服务端系统根据唯一邀请码找到目标群组
4. 系统判断用户是否能加入群组
5. 服务端向客户端返回成功信息并显示

异常事件流：

客户端与服务端之间的通信受阻，用例失败，退出用例；第3步中，系统未找到目标群组，用例失败，回到1 ；第3步中，因群组满员或群组已是确认总账状态而无法加入新成员，用例失败，回到1

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：

17-群组记账：发起垫付账目

18-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，且对群组未进行确认

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，判断发起账目合法性
4. 系统在该群组下添加该笔新账目

异常事件流：

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（账目数量到达上限或用户对该群组已经为确认状态），则关闭用例并向用户返回错误或失败信息；客户端与服务端通信受阻，用例失败

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，且对群组未进行确认

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，判断该删除请求的合法性
4. 系统删除该笔账目

异常事件流：

客户端与服务端之间的通信受阻，用例失败；在第3步中，系统验证该删除请求不合法（确认状态不符合要求）用例失败，返回失败信息

1. **群组记账：用户确认群组**

参与者：用户

前置条件：用户已经加入群组，且已发起了自己应当发起的账目（本项不由本系统保证）

后置条件：若用例成功，用户对于该群组的状态置为确认状态，用户在该群组中将无法进行16-更新垫付账目的操作

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
2. 服务器收到请求并使用户进入确认状态

异常事件流：

发送请求时通信受阻，客户端显示失败信息，回到用例开始

1. **群组记账：群组确认总账**

参与者：无外部参与者

前置条件：有用户对该群组进行了19-用户确认群组操作

后置条件：若群组所有成员均处于确认状态，则群组转为确认状态，并计算群组内的账目结果；否则不做变化

基本事件流：

群组内有用户完成确认群组操作之后，开始本用例

1. 系统查询该群组所有用户状态
2. 若群组内所有用户均处于确认状态，进入下一步；否则，用例结束
3. 系统将群组置为确认状态
4. 系统针对每一笔群组内的账目及其参与者计算最后用户之间一对一的结算金额并记录

异常事件流：

无

1. **群组记账：查看结清账目**

参与者：用户

前置条件：用户处于群组中，且该群组已为确认总账状态

后置条件：若用例成功，则返回显示与该用户相关的所有群组内结算结果

基本事件流：

1. 用户进入群组页面，发送查看结清账目请求
2. 系统收到请求，验证查看请求合法性
3. 系统返回与用户相关的结算账目，内容包括金额，方向（用户应当付钱还是收钱，对方确认结算状态，自己确认结算状态）

异常事件流：

第2步中，系统验证该请求不合法（群组状态不符合要求），用例失败；用户端和服务端通信受阻，用例失败

1. **群组记账：确认结清账目**

参与者：用户

前置条件：用户在群组中，且群组已经处于确认状态

后置条件：若用例成功，用户将自身对于在群组内和其他一个用户（有资金往来）的结算账目置为确认结清，若对方也已确认结清则该笔结算账目置为结清状态

基本事件流：

用户在收到21-查看结清账目中返回的结算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，将用户对于该笔结算账目的状态置为确认结清
3. 服务端检查该笔结算账目另一方用户对于该账目的状态，若也已确认结清则将该账目置为结清状态

异常事件流：

客户端与服务端通信受阻，用例失败

1. **群组记账：群组关闭**

参与者：无外部角色参与

前置条件：群组内用户进行了22-确认结清账目操作

后置条件：若群组内所有结算账目已是结清状态（即双方确认结清），群组变成关闭状态，相关内容不再接受任何非只读操作

基本事件流：

在22-确认结清账目操作后，用例开始

1. 系统查询该群组下所有结算账目的状态
2. 若所有结算账目状态为结清，则进入下一步；否则用例结束
3. 系统将群组置为关闭状态

异常事件流：

无

1. **群组记账：浏览群组**

参与者：用户

前置条件：用户已登入

后置条件：若用例成功，则返回显示用户所有已加入的群组信息列表

基本事件流：

用户打开浏览群组功能，用例开始：

1. 用户发送浏览群组请求
2. 服务端系统受到请求，查询用户所有已加入群组
3. 系统整合相关信息，将群组信息返回显示（包括群组名称，人数，最后更新时间，群组状态，用户对群组确认状态）

异常事件流：

用户端与服务端通信受阻，用例失败

1. **群组记账：浏览群组内情况**

参与者：用户

前置条件：用户已加入该群组

后置条件：若用例成功，则返回显示该群组内的详细信息，包括所有群组内账目，群组内成员，群组状态

基本事件流：

用户在24-浏览群组中的返回结果中选择一项，用例开始

1. 用户发送浏览群组内情况的请求
2. 服务端收到请求，验证浏览请求的合法性
3. 系统整合相关信息，将群组内成员、账目信息返回显示

异常事件流：

用户端与服务端通信受阻，用例失败；第2步中，验证合法性失败，用例失败，返回错误信息

1. **用户分析数据信息更新**

参与者：时钟

此用例包含的用例：

27-用户总账目更新

28-用户账目环路检测

29-用户账目倾向更新

1. **用户总账目更新**
2. **用户账目环路检测**
3. **用户账目倾向更新**
4. **查看自身统计信息**

## 非功能需求描述

1. 数据需求

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）

单笔账目数目最大不超过10000元

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. 健壮性需求

服务端需要对所有接受的请求拥有良好的异常检查和判断能力，包括但不限于请求并非发自于rBook客户端、伪造请求、相关用户同时操作时的一致性等

服务端所有非只读业务需要满足事务性要求

1. 安全性需求

用户登录密码在数据持久化中不使用明文保存

1. 其他需求

服务端对于HTTP请求均使用REST接口，以便日后的多客户端支持及再扩展需求