**用框架'yong'kuang'jia﷽﷽﷽﷽﷽﷽﷽﷽﷽'jian﷽﷽﷽﷽﷽﷽﷽﷽'song'tong'zhi﷽﷽﷽﷽﷽﷽﷽﷽**

**软件架构设计说明书**

**【A账 – AA债务记录应用】**

**2015-6**

目录

[一、 简介 4](#_Toc265958296)

[1. 编写目的 4](#_Toc265958297)

[2. 文档范围 4](#_Toc265958298)

[3. 定义 4](#_Toc265958299)

[4. 参考资料 5](#_Toc265958300)

[二、 架构表示方式 5](#_Toc265958301)

[三、 架构设计目标与约束 5](#_Toc265958302)

[1. 关键功能需求 6](#_Toc265958303)

[2. 关键质量需求 8](#_Toc265958304)

[3. 开发策略 8](#_Toc265958305)

[四、 用例视图 9](#_Toc265958306)

[1. 概述 9](#_Toc265958307)

[2. 关键用例 9](#_Toc265958308)

[五、 逻辑视图 16](#_Toc265958309)

[1. 概述 16](#_Toc265958310)

[2. 系统层次模型 16](#_Toc265958311)

[3. 主要的设计包和子系统 17](#_Toc265958312)

[六、 进程视图 17](#_Toc265958313)

[1. 概述 17](#_Toc265958314)

[2. 角色进程视图 17](#_Toc265958315)

[七、 实施视图 21](#_Toc265958316)

[1. 概述 21](#_Toc265958317)

[实施模型视图 22](#_Toc265958318)

[八、 部署视图 23](#_Toc265958319)

[1. 概述 23](#_Toc265958320)

[2. 部署方案视图 23](#_Toc265958321)

# 简介

## 编写目的

本文档全面与系统地表述了A账应用的构架，并通过使用多种视图来从不同角度描述本系统的各个主要方面，明确了该应用的需求。本文档记录并表述了系统架构的设计人员对系统构架方面做出的重要决策。

项目经理将根据构架定义的构件结构制定项目的开发计划；程序设计员将据此进行各构件的详细设计；测试设计员按照构架设计系统的总体测试框架；另外构架文档还用于指导各构件的实施、集成及测试。

本文档的预期阅读人员为项目经理、程序设计人员、测试人员和其他有关的工作人员。

## 文档范围

本软件架构说明文档适合于A帐应用架构。

## 定义

1. **SQLite**: 一个小型关系型数据管理系统，开发者为瑞典Mysql AB公司，属于开源软件。
2. **Java**：开发语言。
3. **eclipse**: 开发工具。
4. **Tornado**：用Python语言编写的Web应用框架。

## 参考资料

1. 《Struts+Hibernate+Spring开发入行真功夫》，电子工业出版社。
2. 《文档写作指南》，清华大学出版社。

# 架构表示方式

本软件架构设计文档以一系列的视图（View）来表示系统的软件构架，主要包括用例视图、逻辑视图、进程视图、部署视图等，每个视图拥有一个或多个模型（Model）。并围绕相关视图来描述系统的基本结构、组成机制与工作原理等。本软件架构设计文档还将系统的构架机制描述也放在了逻辑视图之下。本文档主要使用统一建模语言（UML）来充当相关模型的表达语言。

# 架构设计目标与约束

描述构架设计最主要目标就是满足关键系统功能需求和质量约束，这些功能需求和质量要求对软件构架有重大的影响，并决定了构架的设计。本节同时还列明影响构架的其他相关因素，如开发人员对开发工具的掌握情况等。

## 关键功能需求

按照需求分析文档的规格要求，本A账应用的设计分成了以发起消费功能、偿还债务功能以及查看附近债务关系人功能模块为主的三大模块进行开发，而在此三大模块的的基础上，又细分成了计算AA账务，添加、删除债务以及发送催债消息等子功能模块。系统的开发主要是为了简化聚餐AA账目的计算，以及积累繁琐的小额债务以便一次过还清，给学生之间平日记录债务提供一个平台。

* 发起消费模块

发起消费模块是用户在AA消费时使用的服务，该服务主要包括邀请参加者，输入消费，计算消费金额，更新债务关系四项流程。通过该功能可以简化AA账目的计算，并直接更新债务关系。但对于其他途径产生的债务关系并不能直接在此体现，可通过与其他的模块进行交互。

* 偿还债务

管理债务是该应用的基本功能之一，其中包括更新债务和清除债务。由于应用只关注债务关系，通过双方确认能够直接更新债务信息，由债务人进行还债，系统自动计算欠债关系。

消除债务功能则是完全清除债务关系，即用户与清除关系的用户不再有任何联系直到有新的债务生成，可以由债务人或债权人任何一方发起。

* 查看附近债务关系人

该模块向用户提供附近其他用户的定位，地图上会注明其他A账用户的坐标，包括欠债人和债主。同时该模块拓展了一个催债功能，通过点击地图上欠债人的图标能够发送催债消息。用户在后台情况下也能收到推送通知。

## 关键质量需求

由于此开发项目针对一般高效学生，使用率相对高，由于涵盖的信息较少，社交关系简单，在实用性和响应速度上要求的较高。为便于使用和推广，软件要求简单实用易上手，稳定便捷。

* 创建消费的邀请过程：不超过20秒；
* 计算过程：不超过5秒；
* 其它所有交互功能反应速度：不超过3秒；
* 可靠性：平均故障间隔时间不低于200小时。

## 开发策略

* **软件复用策略**

系统中重要基础构件应当具备较高的设计与构建质量，可以在产品中复用。

* **使用架构**

本系统借鉴了一些开发Android应用的开放framework。开发出基于MVC(Model-View-Controller)设计模式的应用构架。

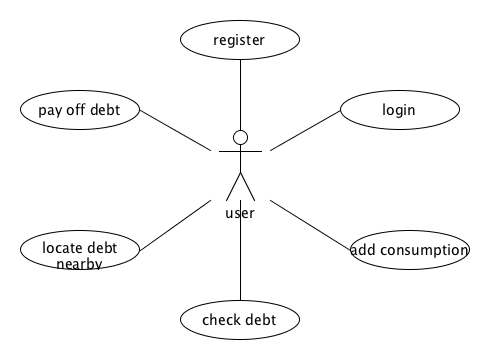
# 用例视图

## 概述

用例视图从用户使用的角度描述系统构架的基本外部行为特性，通常包含业务用例模型与系统用例模型。业务用例模型不适用于本系统，这里只关注系统用例。本应用只面向用户而无管理人员，故，这些用例代表了系统主要的核心功能，决定了系统构架的基本组成元素。有些用例强调或决定了构架的某些具体然而重要的细节，通常也可以列在本节内，总之所列的用例集合应基本覆盖系统构架的主要方面。

## 关键用例

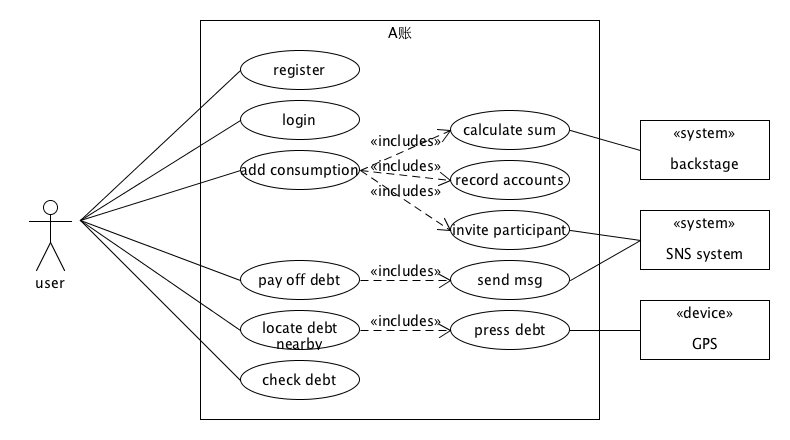
* 关键的系统参与者 ：用户



用户的用例说明

|  |  |
| --- | --- |
| 用例名称 | 简单描述 |
| 注册 | 创建账号 |
| 登录 | 根据账号和密码登录 |
| 发起消费 | 填写消费参与者和消费情况 |
| 查看债务 | 用户能够查看总共欠其他用户的金额和其他用户欠自己的金额 |
| 偿还债务 | 输入偿还金额，系统计算新的债务金额 |
| 查看附近债务关系人 | 游客通过GPS服务能够查看附近的债务关系人，同时也让其他用户看到自己的位置 |

* **关键的系统用例**

****

# 逻辑视图

## 概述

逻辑视图从系统内在逻辑结构的角度描述系统的基本结构与动态行为，通常包括分析模型（Analysis Model）、设计模型（Design Model）以及数据模型（Data Model）等。设计模型说明了系统的组成元素、组织架构和关系，并描述了各组成元素的协作以及状态转换关系等（通过用例实现Use Case Realization予以表达）。本节将分别在系统层次结构模型中描述系统的层次组织结构；在主要的包和子系统中说明系统的具体组成。

在本软件中，我们主要将逻辑分为UI层，Application层和Domain层。通过这三层体现软件模块之间的联系。

## 系统层次模型

UI层：主要分为注册登录、发起消费、偿还债务和附近的债务关系人几个子系统。

Application层：主要包含登录控制器、消费控制器，债务控制器和附近债务控制器。

Domain层：主要包括用户信息、债务情况、债务参与人以及位置信息。

## 主要的设计包和子系统

## Macintosh HD:Users:user:Documents:Study:【SunYat-Sen】:大三下:系统分析与设计:Project:2.pic.jpg

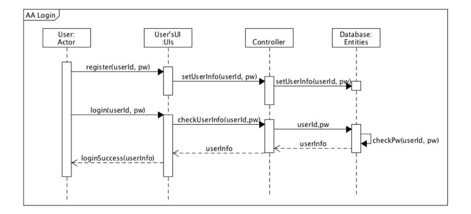
# 进程视图

## 概述

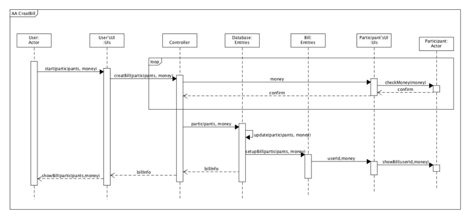
进程视图从系统运行时刻的角度，描述系统划分为进程、线程的结构，及其动态关系。模型主要说明不同系统角色之间的创建、交互和消息通讯关系等。

## 角色进程视图

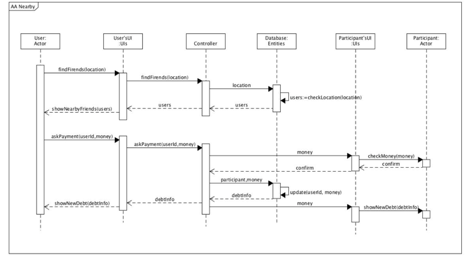
* 注册登录：



* 创建消费



* 查看附近的债，催债，更新债务：

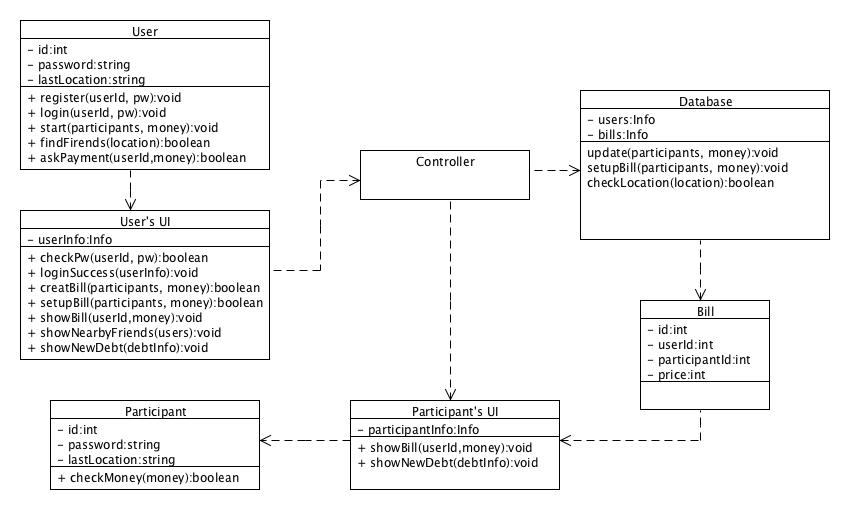


# 实施视图

## 概述

本部分从编译与构建的角度，描述系统实施构件的组织结构与依赖关系（主要是编译依赖）。模型包括实施子系统和构件结构，及其依赖关系。同时还表达了逻辑视图中各个包和类分配到实施视图中的子系统和构件的映射关系。

## 实施模型视图



# 部署视图

## 概述

从系统软硬件物理配置的角度，描述系统的网络逻辑拓扑结构。模型包括各个物理节点的硬件与软件配置，网络的逻辑拓扑结构，节点间的交互和讯关系等。同时还表达了进程视图中的各个进程具体分配到物理节点的映射关系。

## 部署方案视图

