

项目实训-微移动支付平台

系统设计报告

组 号： G16

组 长： 彭子帆

组 员： 韩汶东 骆晓天
李宇璇 惠铭康

2020 年 7 月 6 日

修改历史

日期	版本	作者	修改内容
2020.7.6	1.0	彭子帆、韩汶东、 骆晓天、李宇璇、 惠铭康	初稿

目录

1. 引言	5
1.1. 编写目的	5
1.2. 项目背景	5
1.3. 项目目的	5
1.4. 参考资料	6
2. 总体设计	6
2.1. 系统设计	6
2.1.1. 总体描述	6
2.1.2. 模块说明	7
2.1.3. 主要技术分析	8
2.2. 需求规定	9
2.2.1. 用户需求	9
2.2.2. 性能需求	10
2.2.3. 权限与安全需求	10
2.2.4. 其它需求	11
2.3. 运行环境	12
2.3.1. 软件层面	12
2.3.2. 硬件层面	12
2.4. 基本设计概念和处理流程	12
2.5. 结构	14
2.5.1. 用户需求分析图	15
2.5.2. 系统模块架构图	15
2.5.3. 系统模块设计	16
2.5.4. 数据上下文	16
2.5.5. 总体 E-R 图	17
2.5.6. 关键 IPO 图	18
2.6. 用例 (用作 Test case)	19
2.6.1. 用户登录/注册	19
2.6.2. 用户注销	20
2.6.3. 二维码支付	21
2.6.4. 二维码验证健康状况	22
2.6.5. 二维码支付和验证健康状况	23
2.6.6. 使用二维码支付	24
2.6.7. NFC 验证健康状况	25
2.6.8. NFC 支付和验证健康状况	26
2.6.9. 查看账单	27
2.6.10. 下载账单	28
2.6.11. 绑定公交卡	29
2.6.12. 绑定银行卡	30
2.6.13. 删除公交卡	31
2.6.14. 删除银行卡	32
2.7. 数据流图	33

2.7.1.	顶层图	33
2.7.2.	层图	33
2.8.	数据字典	34
2.8.1.	数据流定义表	34
2.8.2.	数据元素定义表	35
2.8.3.	数据精度表	35
2.9.	尚未解决的问题	36
3.	接口设计	36
3.1.	用户接口	36
3.2.	外部接口	36
3.3.	内部接口	37
4.	运行设计	37
4.1.	运行模块的组合	37
4.2.	运行控制	37
4.3.	运行时间	37
5.	系统数据接口设计	38
5.1.	逻辑结构设计要点	38
5.2.	物理结构设计要点	39
6.	系统出错设计	40
6.1.	出错信息	40
6.2.	补救措施	40
7.	系统维护设计	40
7.1.	概述	40
7.2.	检测点设计	41
7.3.	系统维护设计	41

1. 引言

1.1. 编写目的

此文档《软件概要设计说明书》旨在使开发的软件产品和项目满足规定的软件规格要求,进而确定软件系统的体系结构、组成部分、数据组织、模块、内外部接口。

工作内容主要有:

- (1) 建立本开发项目的目标系统的总体结构。
- (2) 总体设计
- (3) 接口设计
- (4) 运行设计
- (5) 系统数据结构设计
- (6) 系统出错处理设计

1.2. 项目背景

移动电子支付是指单位,个人直接或授权他人通过电子终端发出支付指令,实现货币支付与资金转移的行为。电子支付是指电子交易的当事人,包括客户,商家和金融机构之间,使用安全电子手段,通过网络进行的货币或资金流转,即把包括电子现金,电子票据,信用卡,借记卡,智能卡等支付手段的支付信息,通过网络安全传到银行或相关的处理机构来实现电子支付。

随着移动支付的普及,线下移动支付的场景越来越频繁,尤其是日常微支付,使用频率逐步升高,微支付方面有取代现金支付的趋势。

自 2020 年初开始,新冠疫情逐渐肆虐,中国采取了很多防疫、管理、检测手段,其中“健康码”的研发和推广使用,在管理防疫方面发挥了巨大的作用。

微移动支付和健康码都是基于移动载体,通过电子互联网发挥作用,其使用形式和使用场景有很多相似和重叠的地方。寻找二者有机的结合方式,尽可能在重合场景保证功能的前提下,尽可能简化使用步骤,达到高效、便捷、安全的用户体验是我们本次项目的初衷。

1.3. 项目目的

本项目是一个供有线下小额移动付款需求的用户,在疫情期间结合健康码和付款功能的移动支付应用。实现以下几个主要的功能模块:

- (1) 二维码支付相关功能
- (2) 健康码(结合支付码)功能

- (3) NFC 支付功能
- (4) NFC 健康识别功能

1.4. 参考资料

- (1)《软件设计文档国家标准》
- (2)《软件工程——面向对象和传统的方法》
- (3)《项目实训开发文档范例》
- (4)《项目计划书》
- (5)《项目需求规格说明书》

2. 总体设计

2.1. 系统设计

2.1.1. 总体描述

在本项目完成并交付后的产品使用者可以分为三种用户，分别为有线下小额支付需求的用户、有在疫情期间证明健康状况的用户、以上两个需求兼备的用户。以下将展开各类用户的常见使用操作：

使用小额支付。针对不同的场景，我们提供了不同的支付方式，分为二维码支付和 NFC 支付，让用户的支付行为更方便。在支付之前，用户需要绑定和授权一些基本信息，主要为银行卡信息和 NFC 识别信息（如公交卡、地铁卡等）。在使用二维码支付时，用户只需要打开二维码即可支付。在使用 NFC 支付时，用户只需要打开 NFC 功能，然后选择支付场景对应的识别信息（如公交卡、地铁卡等）即可支付。

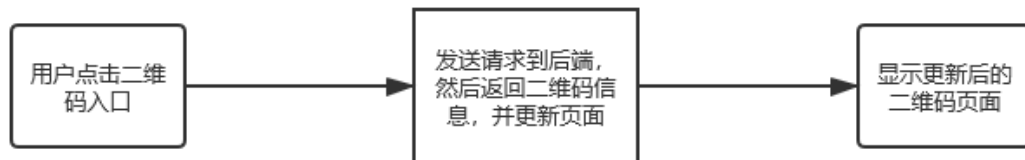
疫情期证明健康状况。在疫情期间，出入各种公共场合时，都需要展示健康码来验证自己的健康身份。如果用户需要验证健康情况，打开二维码，其颜色即可证明健康情况。

小额支付+疫情期证明健康状况。疫情期间按，在很多支付场景中，或支付之前，都需要证明健康身份，用户在两种流程中不断切换，体验较为繁琐。我们将支付和证明健康状况结合，将健康码和支付码结合，同时具备支付和证明的作用，一码多用。在 NFC 支付场景，我们也结合了健康状况的判定，需要判定时我们会先判定健康状况，然后支付（比如在乘坐公交车场景）。

2.1.2. 模块说明

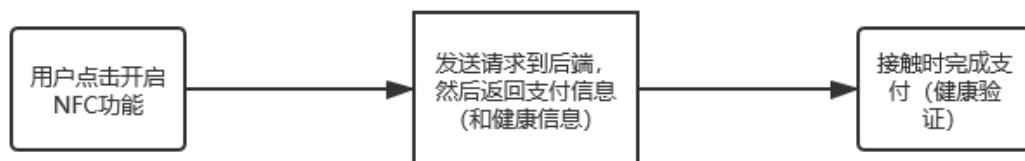
(1) 二维码（支付+健康码）

用户点击二维码入口，显示出每个用户对用的二维码，该二维码有支付功能，同时也具有健康码功能（颜色显示健康状况）。



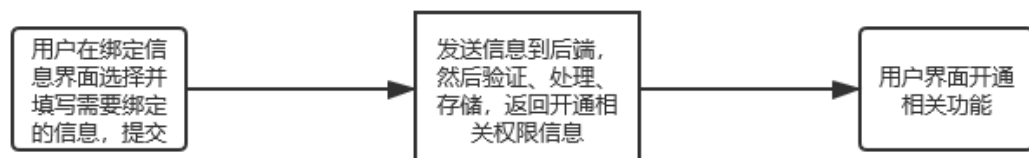
(2) NFC（支付+健康识别）

用户选择 NFC 功能，提供给用户 NFC 接触付款功能，在某些需要验证健康状况的支付场景（如乘公交），先验证健康状况，在健康的前提下进行支付，全程用户不感知。



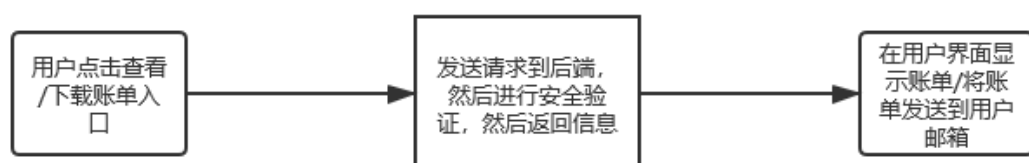
(3) 绑定信息（银行卡、公交卡等）

在使用二维码和 NFC 支付+健康码功能之前，需要绑定使用者的相关信息（身份证、银行卡、公交卡等）。



(4) 查看/下载账单信息

开通支付功能的用户，可以查看/下载自己在本平台的资金流水记录（账单）。



2.1.3. 主要技术分析

2.1.3.1. 前端技术

(1) Android

安卓是一种基于 Linux 内核（不包含 GNU 组件）的自由及开放源代码的操作系统。主要使用于移动设备，如智能手机和平板电脑，由美国 Google 公司和开放手机联盟领导及开发。Android 操作系统最初由 Andy Rubin 开发，主要支持手机。

(2) C/S 结构

服务器-客户机，即 Client-Server (C/S) 结构。C/S 结构通常采取两层结构。服务器负责数据的管理，客户机负责完成与用户的交互任务。

客户机通过局域网与服务器相连，接受用户的请求，并通过网络向服务器提出请求，对数据库进行操作。服务器接受客户机的请求，将数据提交给客户机，客户机将数据进行计算并将结果呈现给用户。服务器还要提供完善安全保护及对数据完整性的处理等操作，并允许多个客户机同时访问服务器，这就对服务器的硬件处理数据能力提出了很高的要求。

在 C/S 结构中，应用程序分为两部分：服务器部分和客户机部分。服务器部分是多个用户共享的信息与功能，执行后台服务，如控制共享数据库的操作等；客户机部分为用户所专有，负责执行前台功能，在出错提示、在线帮助等方面都有强大的功能，并且可以在子程序间自由切换。

(3) Java

Java 是一门面向对象编程语言，不仅吸收了 C++ 语言的各种优点，还摒弃了 C++ 里难以理解的多继承、指针等概念，因此 Java 语言具有功能强大和简单易用两个特征。Java 语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

(4) XML

可扩展标记语言与 Access, Oracle 和 SQL Server 等数据库不同，数据库提供了更强大有力的数据存储和分析能力，例如：数据索引、排序、查找、相关一致性等，可扩展标记语言仅仅是存储数据。事实上它与其他数据表现形式最大的不同是：可扩展标记语言极其简单，这是一个看上去有点琐细的优点，但正是这点使它与众不同。

2.1.3.2. 后端技术

(1) Spring Boot 框架

Spring Boot 是由 Pivotal 团队提供的全新框架,其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置,从而使开发人员不再需要定义样板化的配置。通过这种方式, Spring Boot 致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

(2) MySQL 数据库

MySQL 是最流行的关系型数据库管理系统,在 WEB 应用方面 MySQL 是最好的关系数据库管理系统应用软件之一。

采用相同技术栈的项目非常多,运营也十分稳定。在技术上,本项目没有仍待解决的技术难题。因此,本项目在技术上具有可行性。

(3) MyBatis 框架

MyBatis 是一款优秀的持久层框架,它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息,将接口和 Java 的 POJOs(Plain Ordinary Java Object,普通的 Java 对象)映射成数据库中的记录。

(4) Docker

Docker 是一个开源的应用容器引擎,让开发者可以打包他们的应用以及依赖包到一个可移植的容器中,然后发布到任何流行的 Linux 机器或 Windows 机器上,也可以实现虚拟化,容器是完全使用沙箱机制,相互之间不会有任何接口。

(5) Redis

Redis (Remote Dictionary Server),即远程字典服务,是一个开源的使用 ANSI C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库,并提供多种语言的 API。从 2010 年 3 月 15 日起,Redis 的开发工作由 VMware 主持。从 2013 年 5 月开始,Redis 的开发由 Pivotal 赞助。

2.2. 需求规定

2.2.1. 用户需求

(1) 支付需求

- 可以绑定各种支付源(银行卡、信用卡等);

- 可以二维码形式支付；
- 可以 NFC 触碰支付；

(2) 健康验证需求

- 实时跟新用户健康状态；
- 二维码形式（健康码）反应用户健康状态；
- NFC 后台验证健康状况；

(3) 需要验证健康状况的场景下支付需求

- 结合支付码和二维码，在这种场景下便捷用户行为；
- NFC 功能合并健康码和支付行为，先验证健康状况，然后支付。

2.2.2. 性能需求

- 系统应保证运行稳定，避免出现崩溃。
- 主流浏览器均能正常访问本系统。
- 系统应能保证至少 500 人的并发访问。
- 当用户登录以及进行任何操作时，系统应该能及时进行反应，反应的时间在 3s 以内。
- 系统应该能及时检测出各种非正常情况，如与设备的通信终端，无法连接数据库服务器等，避免长时间等待。
- 每个页面一般情况下应在 2s 内加载完毕，高峰期应在 6s 内加载完毕。
- 系统保证在一周内不超过一次的维护与重启。

2.2.3. 权限与安全需求

对于任何一个系统来说，安全是保证其正常运行的关键因素之一。因此在我们的系统中，对于安全与权限进行了如下设计：

- 所有涉及功能信息或个人信息的网络事务，都应进行加密操作。
- 除浏览菜单外，用户必须登录后才能完成其他操作。
- 用户修改密码时必须进行验证。
- 用户密码设置具有如下的强度要求：必须 8 位以上，且为数字与英文的组合。

- 用户无法非法修改后端数据。
- 本系统对重要的数据应进行加密，如用户口令、重要参数等。
- 本系统应该能够记录系统运行时所发生的所有错误，包括本机错误和网络错误、这些错误记录便于查找错误的原因。日志同时记录用户的关键性操作信息。
- 系统应具备加密登录、数据加密传输、数据存储等安全方面的保障，以确保系统的安全性。
- 系统时基于开放的操作系统平台和数据库上的，因此，要求建立操作系统和数据库的安全保障体系，保证操作系统和数据库的安全。
- 对可能发生严重后果的操作要有补救措施。通过补救措施用户可以回到原来的正确状态。对可能造成等待时间较长的操作应该提供取消功能。
- 对一些特殊符号和计算机代码的输入，与系统使用的符号相冲突的字符等进行判断并阻止用户输入该字符；
- 对错误操作支持可逆性处理，如取消系列操作。在输入有效性字符之前应该阻止用户进行只有输入之后才可进行的操作。

2.2.4. 其它需求

- 软件必须严格按照设定的安全权限机制运行，并有效防止非授权用户进入本系统。
- 软件必须提供对系统中各种码表的维护、补充操作。
- 软件必须按照需求规定记录各种日志。
- 软件对用户的所有误操作或不合法操作进行检查，并给出提示信息。

2.3. 运行环境

2.3.1. 软件层面

本移动支付平台的客户端可使用基于 Android(安卓)系统的移动设备(手机、平板等)。

平台的服务端则要求运行在一单独服务器。平台开发主要使用 Java、Node.js、JavaScript 等开发,并使用数据库 MySQL 提供平台系统的数据存储功能。

2.3.2. 硬件层面

硬件层面对于所使用的服务器有以下要求:

- CPU: 主频率>2.0GHz。
- 内存: >=2GB。
- 硬盘: 硬盘容量>200GB、转速>=5400 转/分钟。
- 网卡: 百兆网卡。
- 移动设备: 正常连接使用的设备。

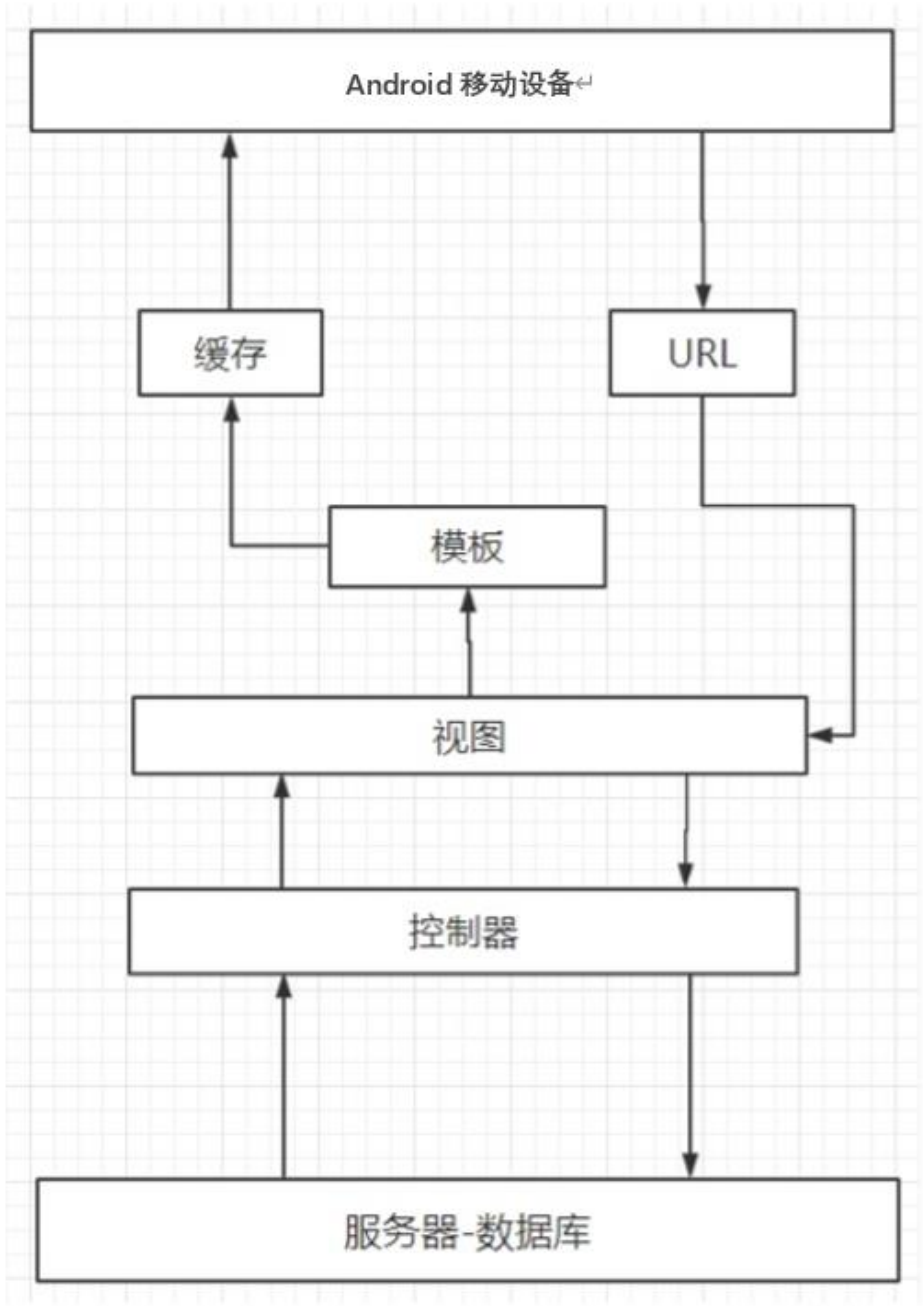
2.4. 基本设计概念和处理流程

服务器

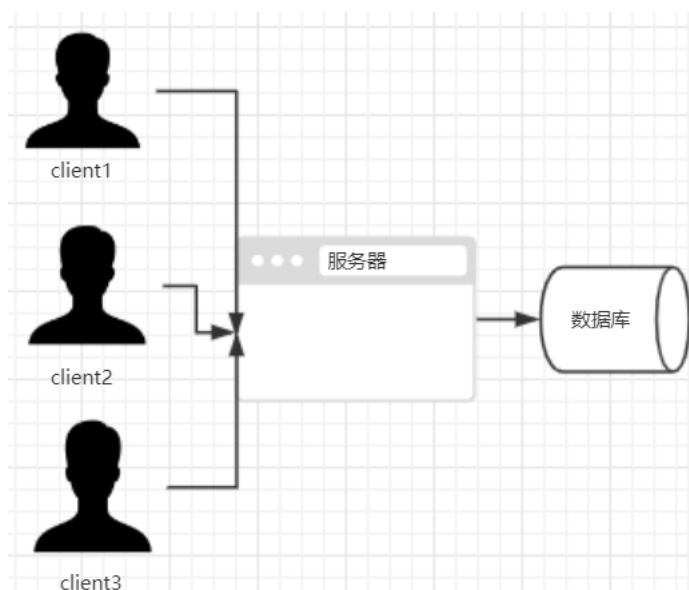
以 Apache2 为服务器, Java 语言编写后端代码,数据库采用 Mysql 和 MongoDB,搜索引擎使用 Elasticsearch,当用户通过浏览器使用网站系统时,浏览器接收用户的请求,并传送到服务器,执行相应的程序,通过搜索引擎 ES 查询相关内容,并且从数据库接口函数向数据库发送 SQL 查询语句,数据库接收 SQL 查询语句后执行,返回查询结果,处理查询结果后返回给前端,并显示在 App 页面上。

客户端

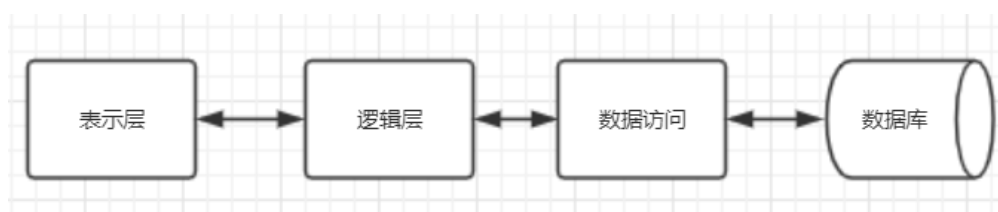
用户移动设备需要时 Android 系统。客户端在不频繁的操作页面时完成操作后断开与数据库的连接以减轻服务器负荷;在操作频繁时保持连接以增加访问速度。客户端动态页面: 嵌入 Vue,动态网页以数据库技术为基础,能降低网站维护的工作量。Vue :页面的各种文本框与按键操作能够完成,同时能实现无刷新页面的一些动画效果,包括下拉菜单等。



2.5. 结构



C/S 体系结构示意图



C/S 三层架构模型图

服务器-客户机，即 Client-Server (C/S) 结构。C/S 结构通常采取两层结构。服务器负责数据的管理，客户机负责完成与用户的交互任务。

客户机通过局域网与服务器相连，接受用户的请求，并通过网络向服务器提出请求，对数据库进行操作。服务器接受客户机的请求，将数据提交给客户机，客户机将数据进行计算并将结果呈现给用户。服务器还要提供完善安全保护及对数据完整性的处理等操作，并允许多个客户机同时访问服务器，这就对服务器的硬件处理数据能力提出了很高的要求。

在 C/S 结构中，应用程序分为两部分：服务器部分和客户机部分。服务器部分是多个用户共享的信息与功能，执行后台服务，如控制共享数据库的操作等；客户机部分为用户所专有，负责执行前台功能，在出错提示、在线帮助等方面都有强大的功能，并且可以在子程序间自由切换。

C/S 结构在技术上已经很成熟，它的主要特点是交互性强、具有安全的存取模式、响应

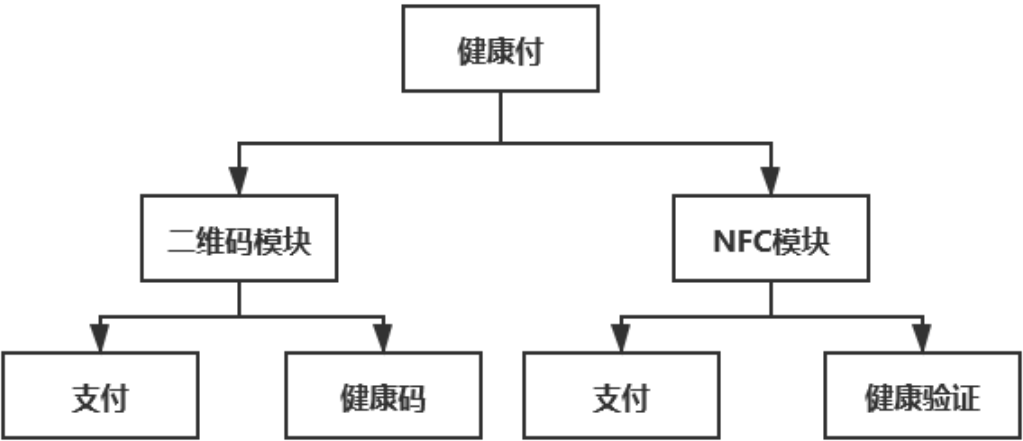
速度快、利于处理大量数据。但是 C/S 结构缺少通用性，系统维护、升级需要重新设计和开发，增加了维护 and 管理的难度，进一步的数据拓展困难较多，所以 C/S 结构只限于小型的局域网。服务器安装 SQL Server、Oracle、MYSQL 等数据库。App 通过 Web Server 同数据库进行数据交互。

本系统将采用 C/S 架构，用户只需要将计算机连接至互联网，下载 App 后，用户便可以直接通过安卓 App 使用。

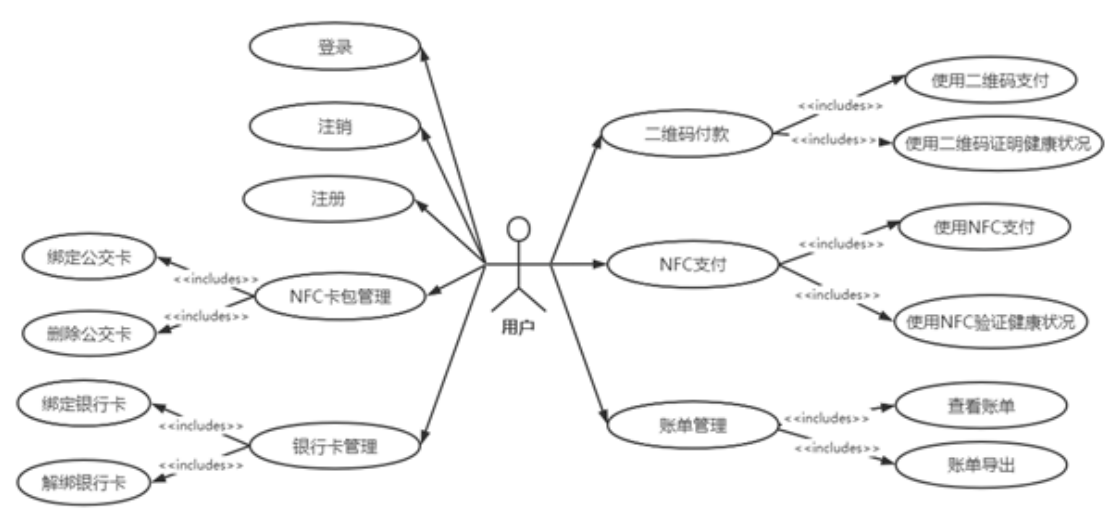
2.5.1. 用户需求分析图



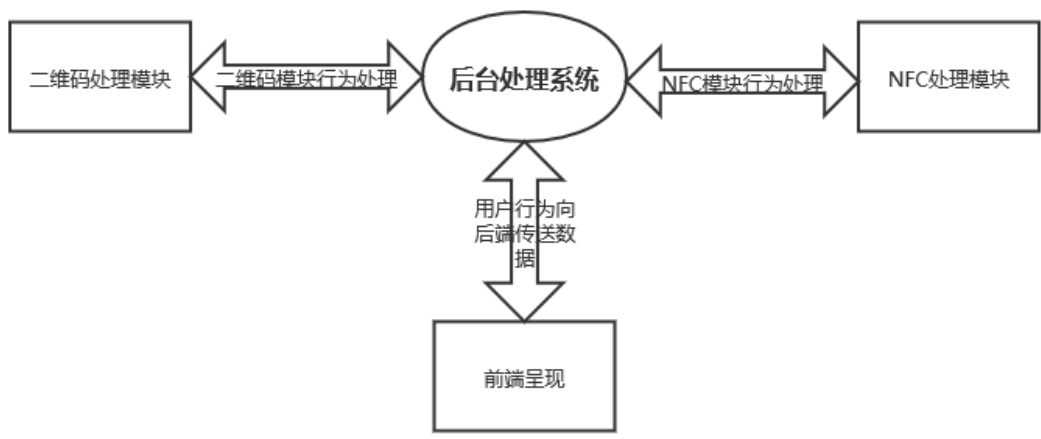
2.5.2. 系统模块架构图



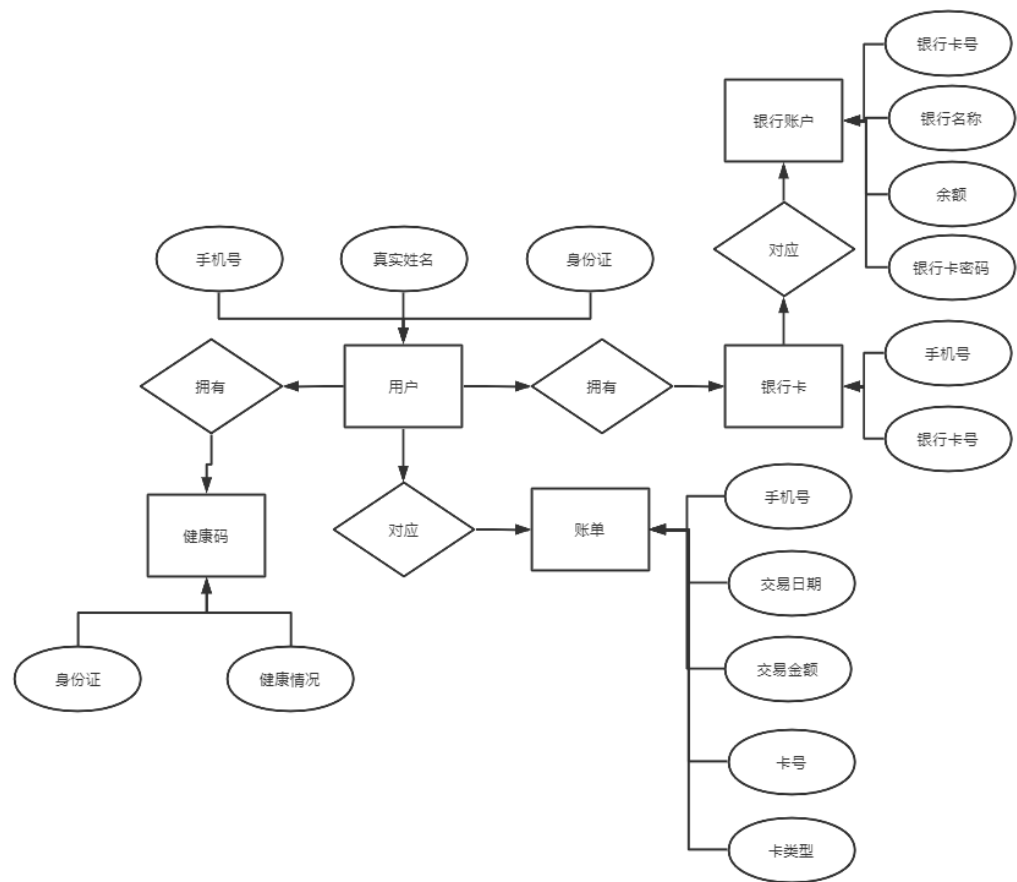
2.5.3. 系统模块设计



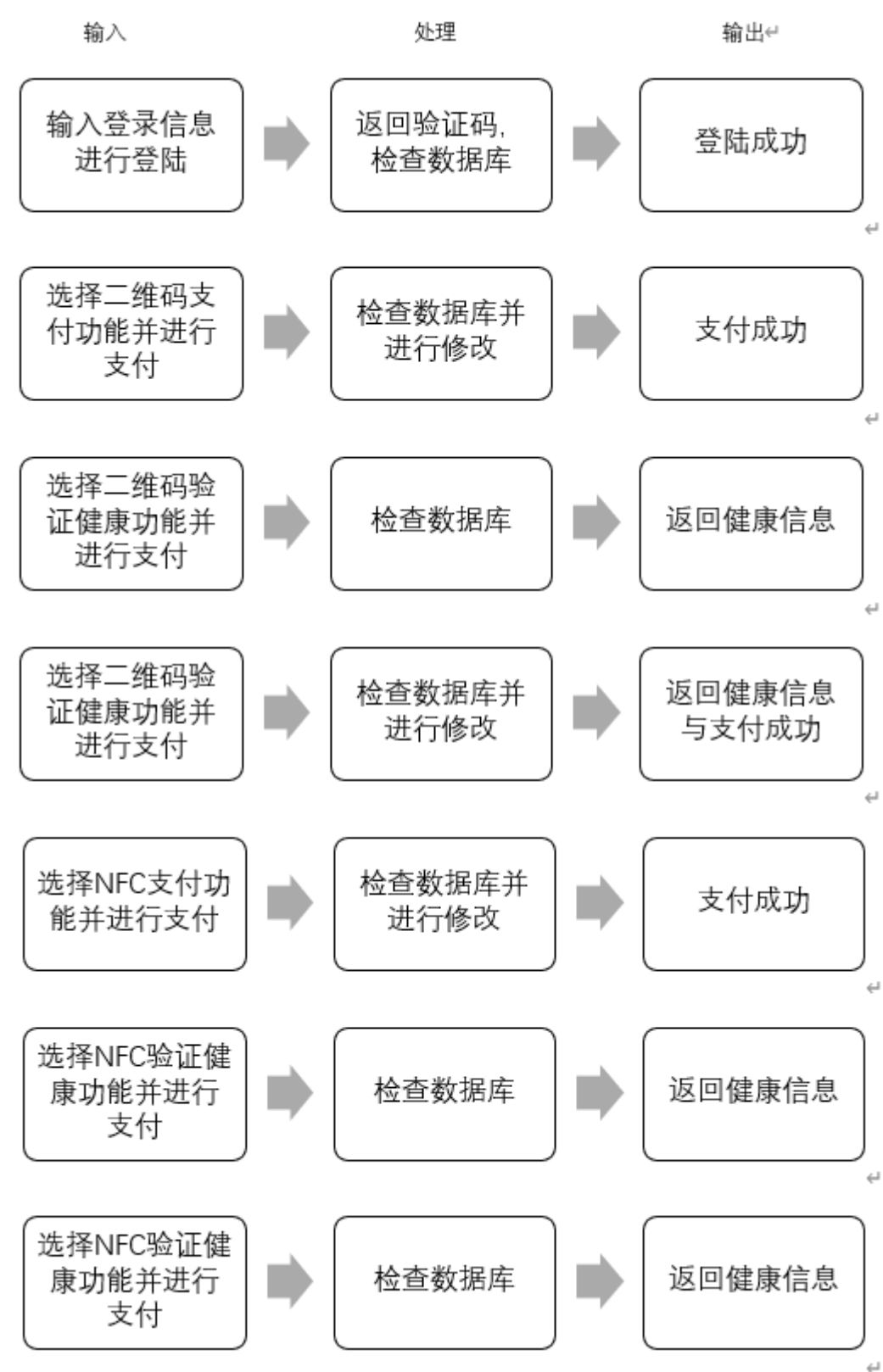
2.5.4. 数据上下文



2.5.5. 总体 E-R 图



2.5.6. 关键 IPO 图





2.6. 用例（用作 Test case）

2.6.1. 用户登录/注册

用例编号	UC-TC-01	用例名称	用户登录/注册
创建人	彭子帆	最后修改人	彭子帆
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户在登录/注册界面输入手机号，发送验证码后即可登录/注册		
前置条件	1. 系统正常运行 2. 用户可以正常打开 App		
触发器	用户打开 App 且处于未登录状态		
后置条件	1. 系统运行正常 2. 前后端交互正常		

输入信息	手机号、验证码
主干过程	1. 用户进入 App 2. 用户在输入手机栏输入手机号 3. 用户点击发送验证码 4. 用户输入手机收到的验证码 5. 用户点击登录按钮
分支过程	无
异常	1. 用户无法正常进入 App 2. 点击登录后系统无响应 处理：用户重新打开 App 后若仍然出现已成则联系系统管理员
假设	无异常发生
输入	手机号码:phone 验证码:captcha
输出	App 主页面
包括用例	无
优先级	1
使用频率	低
备注与问题	无

2.6.2. 用户注销

用例编号	UC-TC-02	用例名称	用户注销
创建人	彭子帆	最后修改人	彭子帆
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户点击注销按钮，而后 App 返回至登录页面		

前置条件	1. 系统正常运行 2. 用户可以正常打开 App
触发器	用户点击注销按钮
后置条件	1. 系统运行正常 2. 前后端交互正常
输入信息	无
主干过程	1. 用户进入个人中心 2. 用户点击注销按钮
分支过程	无
异常	1. 用户无法进入个人中心 2. 用户点击注销按钮后无响应 处理：用户重新打开 App 后若仍然出现已成则联系系统管理员
假设	无异常发生
输入	无
输出	登录/注册页面
包括用例	无
优先级	1
使用频率	低
备注与问题	无

2.6.3. 二维码支付

用例编号	UC-TC-3	用例名称	使用二维码支付
创建人	李宇璇	最后修改人	李宇璇
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		

描述	用户在绑定银行卡后，可以使用二维码进行支付
前置条件	系统正常运行 用户已经登录 银行卡已经绑定
触发器	用户点击支付按钮
后置条件	跳转至二维码界面
输入信息	无
主干过程	用户在主页上点击支付按钮，显示支付二维码
分支过程	无
异常	余额不足，支付失败 系统网络故障
假设	无其他异常
输入	无
输出	支付成功或支付失败，并提示失败原因
包括用例	无
优先级	高
使用频率	高
备注与问题	无

2.6.4. 二维码验证健康状况

用例编号	UC-TC-4	用例名称	使用二维码验证健康状况
创建人	李宇璇	最后修改人	李宇璇
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		

描述	用户在注册登录后，可以使用二维码验证健康状况
前置条件	系统正常运行 用户已经登录
触发器	用户点击支付按钮
后置条件	跳转至二维码界面
输入信息	无
主干过程	用户在主页上点击支付按钮，显示支付二维码
分支过程	无
异常	无相关健康信息 系统网络故障
假设	无其他异常
输入	无
输出	显示健康码状态
包括用例	无
优先级	高
使用频率	高
备注与问题	无

2.6.5. 二维码支付和验证健康状况

用例编号	UC-TC-5	用例名称	使用二维码支付且同时验证健康状况
创建人	李宇璇	最后修改人	李宇璇
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户在绑定银行卡后，可以使用二维码进行支付，并同时验证健康		

	状况。健康码与支付码合一，方便用户使用
前置条件	系统正常运行 用户已经登录系统 银行卡已经绑定
触发器	用户点击支付按钮
后置条件	跳转至二维码界面
输入信息	无
主干过程	用户在主页上点击支付按钮，显示支付二维码
分支过程	无
异常	余额不足，支付失败 系统网络故障
假设	无其他异常
输入	无
输出	支付成功或支付失败，并提示失败原因
包括用例	无
优先级	高
使用频率	高
备注与问题	无

2.6.6. 使用二维码支付

用例编号	UC-TC-5	用例名称	使用二维码支付
创建人	惠铭康	最后修改人	惠铭康
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户在绑定公交，可以使用 NFC 进行支付		

前置条件	系统正常运行 用户已经登录 公交卡已经绑定
触发器	用户点击支付按钮
后置条件	跳转至 NFC 界面
输入信息	无
主干过程	用户在主页上点击支付按钮，显示使用 NFC 支付
分支过程	无
异常	余额不足，支付失败 系统网络故障
假设	无其他异常
输入	无
输出	支付成功或支付失败，并提示失败原因
包括用例	无
优先级	高
使用频率	高
备注与问题	无

2.6.7. NFC 验证健康状况

用例编号	UC-TC-6	用例名称	使用二维码验证健康状况
创建人	惠铭康	最后修改人	惠铭康
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户在注册登录后，可以使用 NFC 验证健康状况		
前置条件	系统正常运行		

	用户已经登录
触发器	用户点击支付按钮
后置条件	跳转至 NFC
输入信息	无
主干过程	用户在主页上点击支付按钮，显示 NFC 验证健康
分支过程	无
异常	无相关健康信息 系统网络故障
假设	无其他异常
输入	无
输出	显示健康码状态
包括用例	无
优先级	高
使用频率	高
备注与问题	无

2.6.8. NFC 支付和验证健康状况

用例编号	UC-TC-7	用例名称	使用二维码支付且同时验证健康状况
创建人	惠铭康	最后修改人	惠铭康
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户在绑定公交卡后，可以使用 NFC 付，并同时验证健康状况。健康与支付合一，方便用户使用		
前置条件	系统正常运行		

	用户已经登录系统 银行卡已经绑定
触发器	用户点击支付按钮
后置条件	跳转至 NFC 界面
输入信息	无
主干过程	用户在主页上点击支付按钮，显示支付 NFC
分支过程	无
异常	余额不足，支付失败 系统网络故障
假设	无其他异常
输入	无
输出	支付成功或支付失败，并提示失败原因
包括用例	无
优先级	高
使用频率	高

2.6.9. 查看账单

用例编号	UC-TC-12	用例名称	查看账单
创建人	韩汶东	最后修改人	韩汶东
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	所有用户都可以查看，不允许更改和删除操作。登陆是前置条件		
前置条件	用户已登录		
触发器	用户点击查看账单入口		
后置条件	进入账单详情界面		

输入信息	无
主干过程	1. 用户登录 2. 用户点击查看账单入口 3. 界面显示账单详情
分支过程	无
异常	后台账单信息记录有误
假设	无异常发生
输入	无
输出	账单详情
包括用例	无
优先级	1
使用频率	中
备注与问题	无

2.6.10. 下载账单

用例编号	UC-TC-13	用例名称	下载账单
创建人	韩汶东	最后修改人	韩汶东
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	所有用户都可以下载，导出文件为 PDF。登陆是前置条件		
前置条件	用户已登录		
触发器	用户点击下载账单入口		
后置条件	用户邮箱收到账单 PDF		
输入信息	无		
主干过程	1. 用户登录 2. 用户点击下载账单入口		

	3. 用户邮箱收到账单 PDF
分支过程	无
异常	后台账单信息记录有误
假设	无异常发生
输入	无
输出	账单 PDF
包括用例	无
优先级	1
使用频率	中
备注与问题	无

2.6.11. 绑定公交卡

用例编号	UC-TC-14	用例名称	绑定公交卡
创建人	骆晓天	最后修改人	骆晓天
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户绑定公交卡是使用 NFC 功能的前提。登陆是前置条件		
前置条件	用户已登录		
触发器	用户点击绑定公交卡入口		
后置条件	NFC 功能开启		
输入信息	公交卡信息		
主干过程	1. 用户登录 2. 用户点击绑定公交卡入口 3. NFC 功能开启		
分支过程	无		
异常	用户输入公交卡信息格式错误, 数据插入数据库出错		

假设	无异常发生
输入	公交卡信息
输出	数据操作结果
包括用例	无
优先级	1
使用频率	高
备注与问题	无

2.6.12. 绑定银行卡

用例编号	UC-TC-15	用例名称	绑定银行卡
创建人	骆晓天	最后修改人	骆晓天
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	用户绑定银行卡是使用二维码支付功能的前提。登陆是前置条件		
前置条件	用户已登录		
触发器	用户点击绑定银行卡入口		
后置条件	二维码支付功能开启		
输入信息	银行卡信息		
主干过程	1. 用户登录 2. 用户点击绑定银行卡入口 3. 二维码支付功能开启		
分支过程	无		
异常	用户输入银行卡信息格式错误, 数据插入数据库出错		
假设	无异常发生		
输入	银行卡信息		
输出	数据操作结果		

包括用例	无
优先级	1
使用频率	高
备注与问题	无

2.6.13. 删除公交卡

用例编号	UC-TC-16	用例名称	删除公交卡
创建人	骆晓天	最后修改人	骆晓天
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	删除公交卡是用户管理 NFC 卡包的必要功能。登陆是前置条件		
前置条件	用户已登录		
触发器	用户选中已绑定公交卡, 点击删除公交卡入口		
后置条件	删除公交卡信息		
输入信息	无		
主干过程	1. 用户登录 2. 用户选中已绑定公交卡, 点击删除公交卡入口 3. 删除公交卡信息		
分支过程	无		
异常	数据库操作出错		
假设	无异常发生		
输入	选中公交卡 ID		
输出	数据操作结果		
包括用例	无		
优先级	1		
使用频率	中		

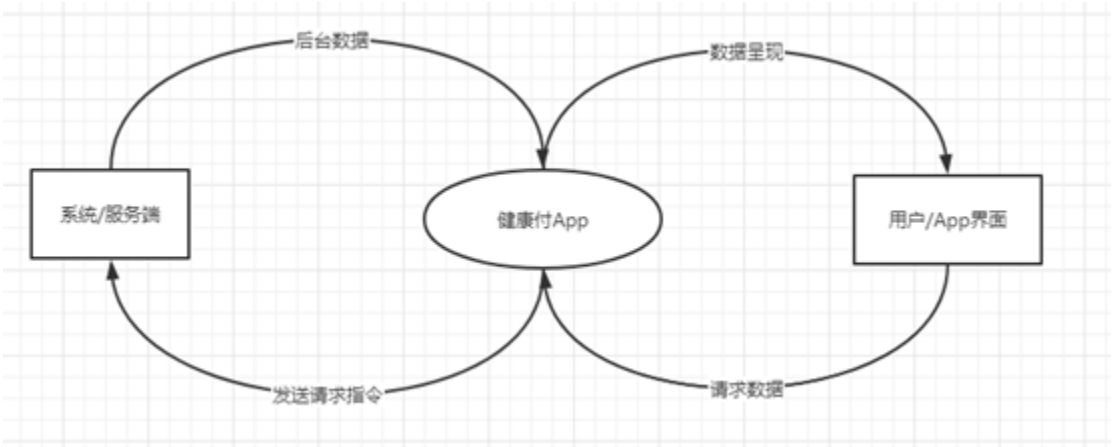
备注与问题	无
-------	---

2.6.14. 删除银行卡

用例编号	UC-TC-17	用例名称	删除银行卡
创建人	骆晓天	最后修改人	骆晓天
创建日期	2020/07/05	最后修改日期	2020/07/06
角色	用户	需求来源	用户代表
主要参与者	用户		
描述	删除银行卡是用户管理银行卡包的必要功能。登陆是前置条件		
前置条件	用户已登录		
触发器	用户选中已绑定银行卡, 点击删除银行卡入口		
后置条件	删除银行卡信息		
输入信息	无		
主干过程	1. 用户登录 2. 用户选中已绑定银行卡, 点击删除银行卡入口 3. 删除银行卡信息		
分支过程	无		
异常	数据库操作出错		
假设	无异常发生		
输入	选中银行卡 ID		
输出	数据操作结果		
包括用例	无		
优先级	1		
使用频率	中		
备注与问题	无		

2.7. 数据流图

2.7.1. 顶层图



2.7.2. 层图

图 1:

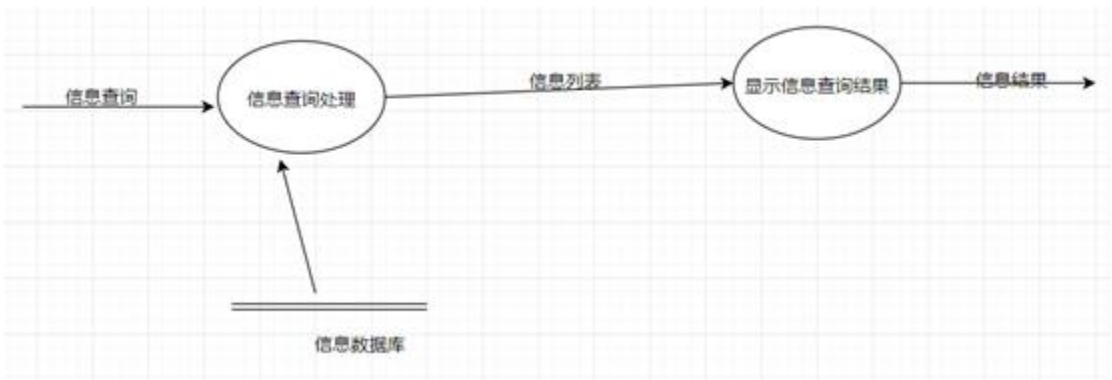


图 2:

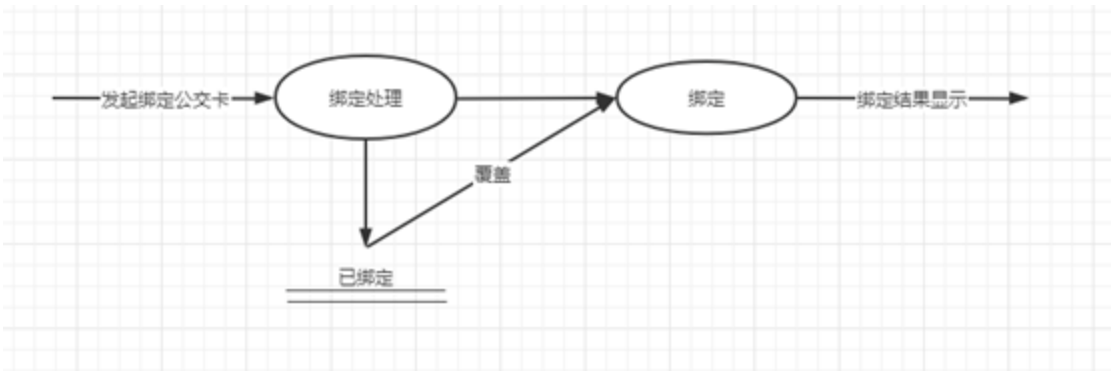


图 3:

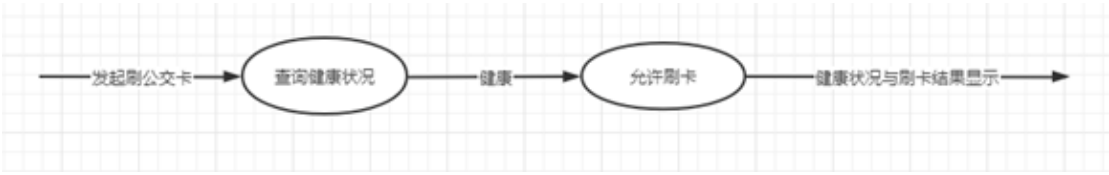


图 4:



2.8. 数据字典

2.8.1. 数据流定义表

编号	数据流名	来源	去向	组成	说明
L1	用户注册	用户	系统	E1+E2+E3+E10+E11	用户注册
L2	用户登录	系统	用户	E1+E10	用户开始使用系统
L3	使用二维码支付	用户	系统	E1+E5+E6	通过二维码进行支付
L4	使用二维码验证健康	系统	用户	E1+E7	通过二维码返回用户健康信息
L5	使用 NFC 支付	用户	系统	E1+E5	通过 NFC 进行支付
L6	使用 NFC 后台验证健康	系统	用户	E1+E7	通过 NFC 返回用户健康信息
L7	绑定公交与银行卡	用户	用户	E1+E10+E12	和各种卡绑定
L8	查询账单	用户	用户	E10+E13	查询用户账单

2.8.2. 数据元素定义表

编号	数据元素名	类型	值域	说明
E1	用户编号	字符	"U"+"0000000".."9999999"	
E2	用户姓名	字符		1-10 位汉字
E3	交易代码	字符	"C"+"0000000".."9999999"	
E4	交易金额	字符		
E5	交易类型	字符		2-6 位汉字
E6	健康信息	字符		2-6 位汉字
E7	交易时间	时间		
E8	交易备注	字符		1-50 位汉字
E9	用户手机号	字符		手机号
E10	用户密码	字符		12 位字符(是否使用还未定)
E11	绑定卡信息	字符		绑定公交卡与银行卡所需信息
E12	用户账单	字符		基于账单设计

2.8.3. 数据精度表

数据	类型	精度要求	说明	示例
用户编号	字符		8 位	U0000000
用户姓名	字符	1-10 位汉字		张三
交易代码	字符	1-20 位字符串	8 位	C0000000
交易金额	文件	两位浮点数	.ppt;.word	10.00
交易类型	字符	2-6 位汉字		“支付”
健康类型	字符	2-6 位汉字		“绿码”
交易时间	整数	精确到秒	指学年开始年份	2030.7.2 12.33

交易备注	时间	1-50 位汉字		“钱”
用户手机号	字符	11 位数字		“关于题目一的描述，如何进行...”
用户密码	字符	8 位数字		123456
绑定卡信息	字符		基于银行卡	
用户账单	字符		基于基本账单	

2.9. 尚未解决的问题

系统的负载均衡配置、最大并发量较低，容易在大量用户同时访问时崩溃，后续需要进一步完善。

3. 接口设计

3.1. 用户接口

本系统作为移动平台层面的应用，用户的主要设备为 Android 系统智能手机。用户通过智能手机的屏幕触摸或虚拟键盘输入完成与系统的输入输出交互。用户主要通过按钮、输入框等可视化元素与服务器后端进行交互。本系统的主要接口分布如下：

一级链接	二级链接	三级链接
登录	注册	
二维码支付	验证健康信息	支付成功
绑定银行卡	绑定成功	
开通 NFC	开通成功	
NFC 支付	验证健康信息	支付成功
查看账单	导出账单	

上述接口均属于本系统的用户接口。

3.2. 外部接口

本系统的用户数据存储在服务器及数据库中。资源文件及不适宜数据库表项存储的超长文本存储在文件中。APP 前端获取用户输入后，由 APP 后端完成与服务器及数据库的交互。

利用 Java 和 MySQL 之间的接口完成应用外部接口设计。

本系统的初始数据依靠人工导入存储。

3.3. 内部接口

本系统只设计了用户模块，消除了模块间的接口。用户模块内各功能之间主要依靠数据库进行数据交互，因此重点是保证数据库的一致性。

4. 运行设计

4.1. 运行模块的组合

本系统按照用户类型划分模块，每个模块不共享界面，相对独立。每个模块按照流程划分客户端界面，客户端脚本和后台服务器程序。

各个模块之间不会共享界面，但共享数据库数据和缓存数据库数据，后台程序只共享数据库连接。

4.2. 运行控制

（1）界面

界面是用户直接与系统交互的部分，界面力求简洁而不简陋，能引导用户进行无碍操作。设计时，以在提供用户便捷操作的基础上增加美观度为基准。

（2）运行控制的条件与限制

本项目的开发要求小组成员足够的参与度，能及时保质保量完成任务。且项目开发过程中可能会有技术上的难点和设备方面的欠缺，需要开发小组合理利用现有设备和资源，积极查找资料解决问题，在完成项目开发的基础上，同时保证项目的可用性、安全性、可维护性等。

（3）前台与后台的关系

前台主要展示搜索词条信息、词条内容等显示信息，后台主要负责业务流程，控制前台显示信息，负责与数据库交互

4.3. 运行时间

用户在前端使用各种功能时会不断地向 后端发送请求，而后后端向数据库请求数据，会频繁与数据库交互以获取信息，会占用较多的数据库资源。

5. 系统数据接口设计

5.1. 逻辑结构设计要点

MySQL 数据库表设计:

表 user:

Column	Type	Default Value	Nullable	Key	Extra	Comments
phone	varchar(13)		NO	PRI		手机号码
ID	varchar(20)		YES			身份证号
name	varchar(16)		YES			姓名

表 bank_card:

Column	Type	Default Value	Nullable	Key	Extra	Comments
phone	varchar(13)		NO			手机号码
card_number	varchar(20)		NO			银行卡号

表 NFC_card:

Column	Type	Default Value	Nullable	Key	Extra	Comments
phone	varchar(13)		NO			手机号码
tag	varchar(300)		NO			NFC 卡标签

表 NFC_bill:

Column	Type	Default Value	Nullable	Key	Extra	Comments
phone	varchar(13)		NO			手机号码
tag	varchar(300)		NO			NFC 卡标签
amount	int	0	NO			交易金额
time	datetime	CURRENT_TIMESTAMP	NO		DEFAULT_GENERATED	交易时间

表 bank_bill:

Column	Type	Default Value	Nullable	Key	Extra	Comments
phone	varchar(13)		NO			手机号码
card_number	varchar(30)		NO			银行卡号
amount	int	0	NO			交易金额
time	datetime	CURRENT_TIMESTAMP	NO		DEFAULT_GENERATED	交易时间

健康码和银行卡模拟接口的表:

表 bank_card_api:

Column	Type	Default Value	Nullable	Key	Extra	Comments
card_number	varchar(30)		NO		PRI	银行卡号
bank_name	varchar(30)		NO			银行名称
balance	int	0	NO			余额
password	char(6)		NO			密码

表 health_api:

Column	Type	Default Value	Nullable	Key	Extra	Comments
ID	varchar(13)		NO		PRI	手机号码
health_state	tinyint	0	NO			健康状态 (0 绿 1 黄 3 红)

5.2. 物理结构设计要点

1. 持久层数据表项存储在放到服务的 MySQL 数据库中，前端 App 通过调用接口向服务器后端发送请求，而后后端通过 MyBatis 持久层框架进行 SQL 语句的数据库查询，访问数据库获取，最后再传回前端 App。

2. 手机验证码等信息在服务器中的 Redis 缓存数据库中，Redis 具有定时删除信息的功

能，可以实现验证码过期的功能。后端会发送验证码到手机，同时存储验证码到 Redis 中，设置过期时间。当用户输入并请求的时候，后端便会到 Redis 中查询验证码，并返回 App 结果。

6. 系统出错设计

6.1. 出错信息

输出信息形式	含义	处理方法
数据库连接失败	由于并发操作的用户数量很大,导致 ES 访问读写率降低; 或者 ES 的节点配置不对,导致 ES 连接失败	修改 ES 节点配置,尝试重连
磁盘损坏	由于物理因素等,导致数据库中的数据丢失	定期对数据库中的数据进行备份
数据库读取乱码或汉字输出为?	客户端页面,数据库,数据库读取过程编码不一致	统一各处的编码方式
不可识别的二维码	生成二维码的过程中,某个色块位置错误	重新申请生成二维码

6.2. 补救措施

系统备份

定期备份系统数据，当系统数据因不可抗力丢失时，可以启用备份数据

分布式部署

将系统部署到不同计算机上，减小硬件损坏造成的数据丢失的影响

7. 系统维护设计

7.1. 概述

1. 连接数据库时，需要在创建数据库连接时使用 try catch 语句捕获异常，对不同的错误信息尽量区分输出

2. 系统维护人员每次维护后要留下完备可读的系统维护日志便于管理员和其他维护人员查看

7.2. 检测点设计

NFC 卡包管理功能

- a)进入 NFC 卡包
- b)新增公交卡
- c)查看公交卡余额
- d)删除公交卡

银行卡包管理功能

- a)进入银行卡包
- b)新增银行卡
- c)查看银行卡余额
- d)删除银行卡

二维码+健康码功能

- a)打开二维码
- b)验证健康情况
- c)使用二维码付款

NFC 付款功能

- a)打开 NFC 功能
- b)贴近收款设施付款

注册登录功能

- a)注册新用户
- b)登录系统
- c)注销用户

7.3. 系统维护设计

硬件资源维护：定期清理服务器硬盘垃圾，可根据网站实际需求选择升级服务器性能

数据库维护：定期备份数据库文件

系统功能升级：根据用户实际访问平台的需求。对于系统功能进行合理的更新。