

中山大学数据科学与计算机学院

软件设计综合实验

题目：《软件设计文档》

| | |
|------|-----------------------|
| 大学专业 | <u>中山大学软件工程</u> |
| 团队成员 | <u>陈喜鹏</u> <u>王颖杰</u> |
| | <u>陈世育</u> <u>曹智惟</u> |
| | <u>黄志镐</u> <u>陈子轩</u> |

2016 年 7 月 12 日

目录

| | |
|---|---|
| 一、 技术选型理由..... | 3 |
| 1. 安卓与 Java | 3 |
| 2. 服务器与 Node.js | 3 |
| 二、 架构设计 | 3 |
| 三、 模块化分 | 3 |
| 四、 模块划分四、使用到的软件设计技术..... | 3 |
| 1. OOP (Object-Oriented Programming), 面向对象编程..... | 3 |
| 2. Asynchronous Programming 异步编程 | 4 |
| 五、 关键模型 | 5 |
| 1. 用例图 | 5 |
| 2. 领域模型 | 5 |
| 3. 架构图 | 5 |

软件设计文档

一、技术选型理由

本项目以 Java 开发的 Android APP 作为前端，以 Node.js (with Express) 开发的后端提供服务。服务器则采用的是非关系型数据库 MongoDB。

1. 安卓与 Java

考虑到本应用的便携性，使用 Web 用户操作起来比较繁琐，于是选择了原生的安卓应用。于是也就才用了官方推荐的 Java 与 XML 进行开发。构建工具使用的是 Android Studio 默认推荐的 Gradle。

Java 具有面向对象、分布式、解释性、健壮、易移植、高性能、多线程等优秀的特性，写起来简单但功完备，可用来开发可靠地、要求严格的应用程序。

XML 语义清楚，结构化，编写简单。

2. 服务器与 Node.js

Node.js 提供了一整套异步编程环境，但能高性能地处理富 I/O 高并发的请求。由于其底层是由 C/C++ 实现的，虽说是单线程，对于对计算能力要求较高的同步请求也能达到很好的性能。

Node.js 让 JavaScript 编写服务器成为可能，可构建快速响应、可扩展的网络应用。同时对于熟悉 JS 的前端人员来说，使用 Node.js 会十分简单。而且，使用 NPM 我们能轻松使用大量优秀的开源项目与开源的包，如 Express 这套服务器框架，非常杰出，同时有大量可插拔的中间件，故在本次采用了该框架。

3. 数据库与 MongoDB

MongoDB 是非关系型数据库。我们本次项目中需要的模型之间关系并不强，使用关系型数据库会加大开发复杂度，MongoDB 则是一个绝佳的选择。作为历史悠久的 NoSQL 数据库的代表，不仅提供了杰出的性能，而且与 Node.js 协同工作能达到很好的效果。（譬如广受推崇的 MEAN 架构中的 M、E、N，正分别对应着 MongoDB、Express、Node.js）

二、架构设计

1. 安卓则采用默认的架构，模块化（页面与对应的控制文件）开发，工具类与公用方法抽离成单独的模块。
2. 后端采用的是 MVC 模式（在此处我们没有 V 的需求，所以准确地说应该是 MC 模式）

三、模块化分

1. 安卓简单地按照不同的 Activity 来划分模块，服务器交互作为一个模块，工具类为一个模块。
2. 服务器则按找 M 与 C 划分。Model 模块负责数据模型构建，而 Controller 模块负责不同路由的控制器，与数据库进行交互。

四、模块划分四、使用到的软件设计技术

1. OOP (Object-Oriented Programming)，面向对象编程

由于 Java 是一个面向对象的语言，所以所有的 Java 代码都使用到了面向对象编程技术。一个简单的例子，Accounts.java，定义了一份账单的具体内容：

```

package com.android.accounts_keeper_android;

/**
 * Accounts metadata
 */
public class Accounts {
    // 食物
    public double food;
    // 水果
    public double fruits;
    // 生活用品
    public double necessities;
    // 学习用品
    public double schoolThings;
    // 备忘
    public String note;
    // 日期 format: yyyy-MM-dd
    public String date;

    public Accounts(String date) throws IllegalArgumentException {
        food = 0;
        fruits = 0;
        necessities = 0;
        schoolThings = 0;
        note = "";

        if (!date.matches("\\d{4}-\\d{2}-\\d{2}"))
            throw new IllegalArgumentException("Wrong date");
        this.date = date;
    }

    public void setFruits(double fruits) {
        this.fruits = fruits;
    }
}

```

2. Asynchronous Programming 异步编程

Node.js 最为著称的就是其异步特性。异步编程能快速响应请求...在服务器的控制器代码中进行数据库查询基本都采用了异步编程模式。其中一个例子：

```

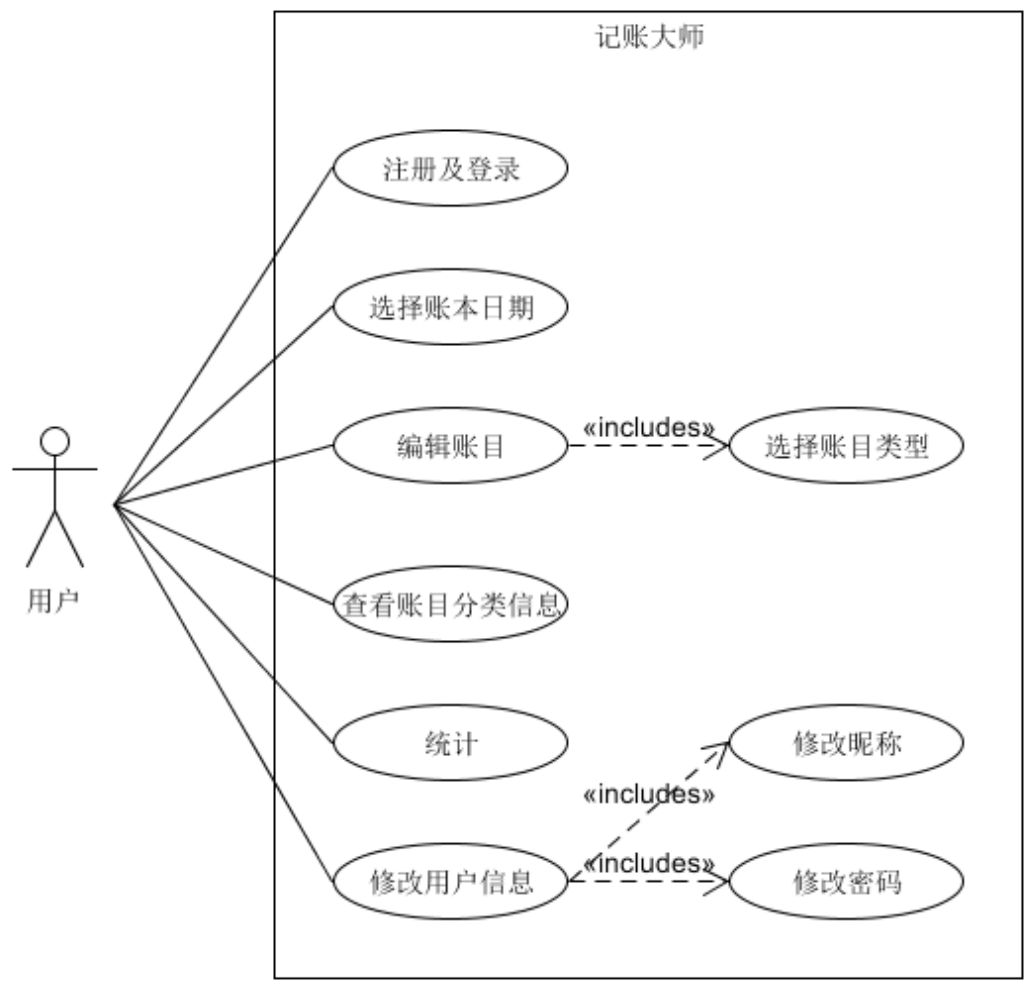
user.js
var User = require('../models/user');

exports.get = function(app) {
  function get(req, res) {
    User.findOne({ username: req.params.username }, function(err, user) {
      if (err)
        res.json({ success: 0, err: '服务器出了点小故障，请稍后再试！' });
      else if (!user)
        res.json({ success: 0, err: '用户不存在！' });
      else
        res.json({ success: 1, username: user.username, nickname: user.nickname, email: user.email });
    });
  }
  return get;
}

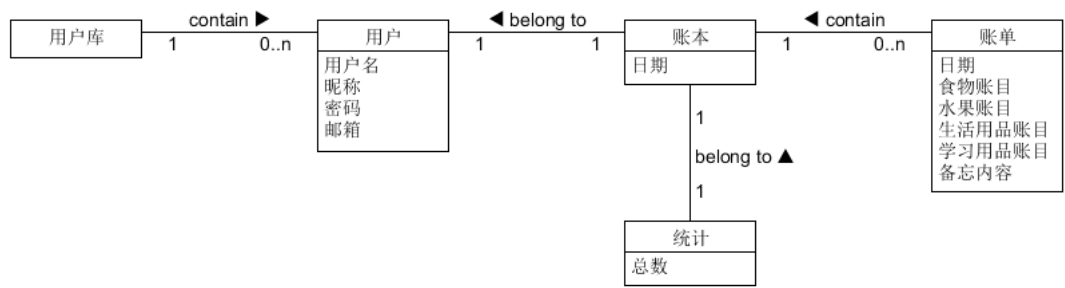
```

五、关键模型

1. 用例图



2. 领域模型



3. 架构图

