<blackhole>

设计文档

小　　组：深黑洞穴吃锅组
完成日期：2017.6.24

目录

1.技术选型理由

作为一款匿名聊天软件，从便携性角度考虑，显然选择移动端作为产品的客户端。从用户基数来说，Android 用户基数大于 IOS、windows 等其他 OS 系统；从开发条件来看，团队不具备开发 IOS APP 的基础。所以，客户端最终选择 Android 作为客户端的环境。

MVP 设计模式：

降低耦合度；模块职责划分明显；利于测试驱动开发；代码复用；隐藏数据

代码灵活性

RxJava2

简化了异步操作，结构清晰

Retrofit

使用注解来描述 http 请求

1.URL 参数的替换和 query 参数的支持

2.对象转化为请求体（如：JSON，protocol buffers 等）

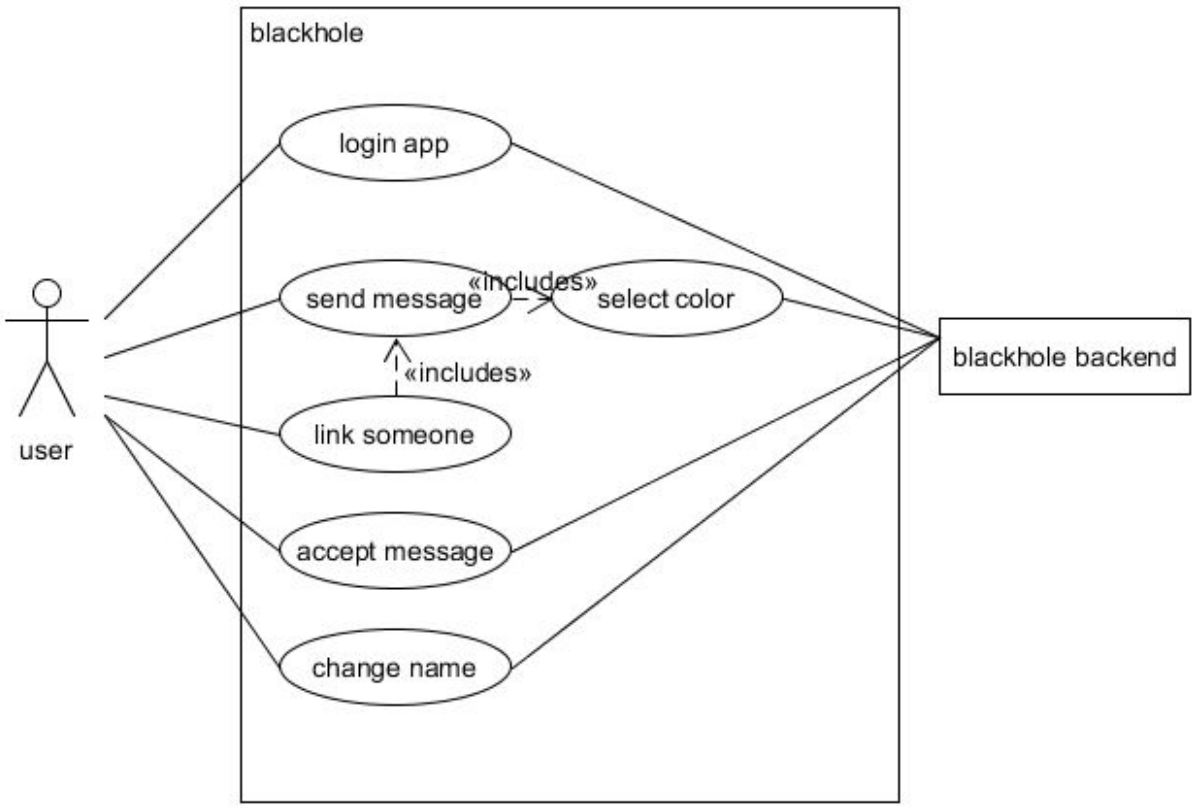
3.多重请求体和文件上传

服务器方面，作为一款聊天软件，服务器请求高并发程度较高，选择 nodejs 作为后台开发语言有效地处理了高并发问题，并降低了软件开发难度。框架方面，选择了著名的 Express 框架。Express 是一个基于 Node.js 平台的极简、灵活的 web 应用开发框架，它提供一系列的特性，帮助你创建各种 Web 和移动设备应用。Express 的 API 提供了丰富的 HTTP 快捷方法和任意排列组合的 Connect 中间间，让我们创建健壮、友好的 API 变得即快速又简单。同时 Express 不对 Node.js 已有的特性进行二次抽象，只是在 Node.js 之上扩展了 Web 应用所需的基本功能。

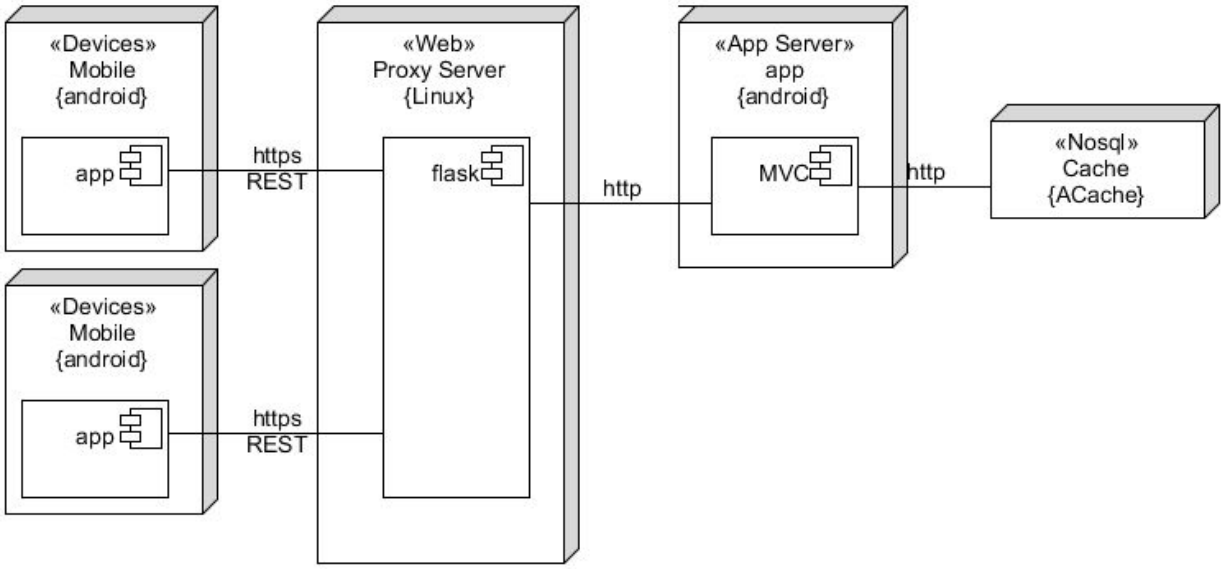
后台测试框架使用了 Node.js 应用中最流行的 Mocha。使用 Mocha，开发者只需要专注于编写单元测试本身，让 Mocha 自动运行所有测试，并生成高效、准确的测试报告。同时，基于 BDD 的 单元测试编写模式，使测试用例更像一份说明书，详细描述软件的每一个功能。最后，使用对 Mocha 友好的 Istanbul，高效快捷地统计测试覆盖率，并自动化生成测试报告。

2.架构设计

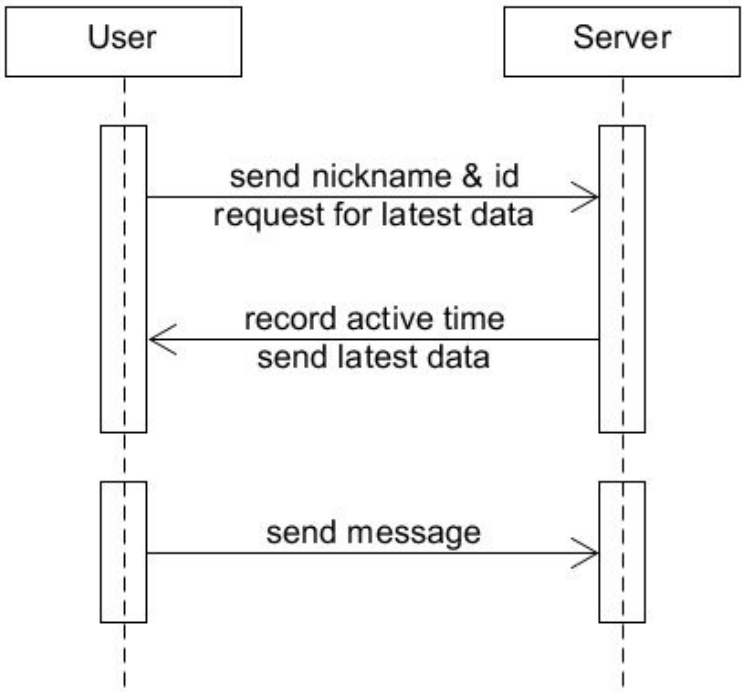
2.1 用例图



2.2 部署图


«Devices»
Mobile
{android}
app
https
REST
«Web»
Proxy Server
{Linux}
flask
http
«App Server»
app
{android}
MVC
http
«Nosql»
Cache
{ACache}
«Devices»
Mobile
{android}
app
https
REST

2.3 泳道图


User
Server
send nickname & id
request for latest data
record active time
send latest data
send message

3. 软件设计技术

采用**结构化编程技术**(Structure Programming)，实现各模块文件分离，各文件单独编写便于分工，同时减少 debug 的修改难度。各模块相互独立不会受到其他模块的牵连降低耦合。结构化编程的基本思想是“自顶向下”的编程方法，经过逐步细化将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图。

安卓端采用 MVP 模式将 model 与 view 层分离，通过 presenter 充当中间件传递消息。View 层负责处理用户事件和视图部分的展示。在 Android 中，它可能是 Activity 或者 Fragment 类。Model 层负责访问数据。数据可以是远端的 Server API，本地数据库或者 SharedPreference 等。Presenter 层是连接（或适配）View 和 Model 的桥梁。

服务端采用 MVC 模式，有助于管理复杂的应用系统，同时让测试更加容易。其中 Model 是应用程序中用于处理应用程序数据逻辑的部分。View 是应用程序中处理数据显示的部分。Controller 是应用程序中处理用户交互的部分。