

# 东华理工大学长江学院

## 本科生毕业设计论文

论文题目：	跨平台即时通讯软件的设计与实现
姓 名：	张原
学 号：	201230330109
班 级：	1233301 班
年 级：	2012 级
专 业：	软件工程
学 院：	信息工程学院
指导教师：	高小辉（讲师）
完成时间：	2016 年 6 月 18 日

## 作者声明

本人以信誉郑重声明：所呈交的学位毕业设计（论文），是本人在指导教师指导下由本人独立撰写完成的，没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。文中引用他人的文献、数据、图件、资料均已明确标注出，不包含他人成果及为获得东华理工大学长江学院或其他教育机构的学位或证书而使用过的材料。对本设计（论文）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本毕业设计（论文）引起的法律结果完全由本人承担。

本毕业设计（论文）成果归东华理工大学长江学院所有。

特此声明。

毕业设计（论文）作者（签字）：

签字日期：                年    月    日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

                年    月    日

# 跨平台即时通讯软件的设计与实现

张原

Chat Software design and realization By  
Cross-platform Development

ZhangYuan

2016 年 6 月 18 日

## 摘 要

针对目前的 Web2.0 模式及对目前的手机多平台开发的窘境,我准备使用 Web 跨平台技术,实现对多个平台的聊天软件开发。使用 Ionic 来构建软件界面,而 Ionic 在除了构建各色的界面,我用它实现了 web 应用在本地的高性能运转,如超多集合的列表在界面快速显示。使用 AngularJS 来处理服务器和本地数据的同步,还有路由的分发,实现了由一个首页分发到各个页面。此外 AngularJS 实现 MVVM 设计模式,使得大项目在前端开发成为了可能,在本项目中各个地方都有使用。使用 Cordova 来进行本地设备及原生功能的调用,使我们很容易使用 javascript 在本地调用摄像头获取地理位置,而不需要再写各个平台对应的开发语言,就算我们没有学过 Android 和 Ios 开发,依然可以完美写出使用原生的功能,而在本项目中就使用了位置定位和调用相机等功能。使用 Java Web 技术来构建后台,使用 MonogoDB 来实现数据来进行数据持久化储存和地理位置索引,使用 Redis 的集合来实现点赞功能,使用有序集合来实现排行榜等功能。

**关键词:**跨平台; 聊天工具; 移动开发; Hybrid App

## ABSTRACT

In view of the present mode of Web and the current mobile phone platform development more, I'm going to use Web cross-platform technology, realization of multiple chat software development platform. Using Ionic to build software interface, and the Ionic in addition to build all kinds of interface, I use it to realize the web application at the local high performance, such as more than a collection of lists rapidly displayed in the interface. Use AngularJS to handle the server and the local data synchronization, and the distributed routing, realized the distributed to each page consists of a front page. In addition AngularJS realize the MVVM design pattern, make big project on the front end development possible, each place has used in this project. Use Cordova for local equipment and native function calls, we are very easy to use javascript in local call camera to obtain the geographical position, and don't need to write each platform corresponding development language, even if we didn't study for Android and Ios development, still can perfect function that makes use of native, and is used in this project the position location and call the camera, and other functions. Use Java Web technology to build the background, the use of MonogoDB to implement the data for data persistence storage and index, geographical position to use Redis collection thumb up functions, using an ordered set to achieve the function list, etc.

**Keywords:**Cross-platform; chat tools; mobile development; Hybrid App

## 目 录

绪论.....	1
选题背景.....	1
选题目的与意义.....	1
国内外研究现状.....	2
研究内容.....	2
研究方法.....	2
1. 软件需求分析.....	3
1.1 目标需求.....	3
1.2 可行性分析.....	3
1.2.1 技术可行性分析.....	3
1.2.2 操作可行性分析.....	3
1.3 需求分析.....	4
1.3.1 对功能的规定.....	4
1.3.2 其它功能.....	4
2. 软件开发技术基础.....	5
2.1 AngularJS 介绍.....	5
2.2 ionic 介绍.....	5
2.3 Cordova 介绍.....	5
2.4 spring 介绍.....	5
2.5 spring MVC 介绍.....	6
2.6 IDE 介绍.....	6
2.7 开发环境配置.....	6
3. 软件设计.....	8
3.1 软件结构设计.....	8
3.1.1 即时性.....	8
3.1.2 高效性.....	8
3.1.3 数据同步性.....	8
3.2 数据库的设计.....	8
3.2.1 mongoDB 介绍.....	8
3.2.2 mongoDB 的数据表结构.....	9
3.2.3 Redis 介绍.....	11
3.2.4 Redis 的数据结构.....	11
4. 系统的实现.....	12
4.1 前端界面设计原理.....	12

4.1.1 界面设计结构.....	12
4.1.2 欢迎界面.....	12
4.1.3 注册界面.....	13
获取短信代码.....	14
注册代码.....	15
4.1.4 实时聊天技术的实现.....	16
4.2 用户相关操作的实现.....	19
4.2.1 好友管理界面.....	20
4.2.2 附近动态界面.....	20
4.2.3 修改好友资料界面.....	21
5. 软件测试和编译.....	24
5.1 编译客户端.....	24
5.1.1 编译准备.....	24
5.2 编译服务器.....	25
5.2.1 编译准备.....	25
5.2.2 编译.....	25
参考文献.....	26
总    结.....	27
致    谢.....	28

## 绪论

### 选题背景

移动开发的兴起,越来越多人使用手机作为自己的首要交通工具,手机通讯也成为日益成为了人们重要的通讯工具,加之目前的很多通讯软件使用的思想依旧很陈旧,技术来说成本也很大,就比如 QQ 在 Windows 平台,IOS 平台,Android 平台使用效果还不错,可是到了不是很主流的 Windows Phone 就很不好,Windows Phone 商店评分也只有区区 2 分,连谷歌 Play 和 App Store 的一半的平分都不到,由于平台不一致还导致其它平台发最新的功能到 Window Phone 平台,应用程序直接奔溃的情况。可见即时通讯聊天这一块在多平台是多么的重要。当然其它的也不尽如意。如最近阿里巴巴推出的钉钉,在手机端只有 IOS 和 Android 的版本,导致很多其它系统的用户只能望洋兴叹。与此同时,钉钉也损失了大量用户。除了即时通讯这块,还有相对知名的“支付婊”事件。这些种种,让我不得不想要开发一款,使多个平台能真正互相交互的软件。然而尤其是即时通讯在跨平台上十分重要的地位和作用,于是我决定开发一款跨平台的聊天通讯软件。

### 选题目的与意义

随着计算机技术的发展,越来越多人喜欢宅在家,很多人甚至都没有见过自己的邻居。也很少有机会和周边志同道合的朋友一起分享自己的技能和运动。一起登山,打羽毛球也常常找不到一个好的伙伴。那我们为什么不能借助一个通讯软件来实现我们相同的兴趣爱好呢?为我们的爱好集结更多的伙伴,和伙伴们一起呼吸户外的新鲜空气,和周边的陌生人拉近关系,岂不快哉!我希望能通过这样的一个软件能让周边的朋友一起出来运动,让大家的关系能更近一步,同时让我们能一起互相运动让大家有一个更好的身体。让宅在家里吃盒饭,打游戏成为过去式。

对于传统的聊天软件而言,只是能够相互聊天,而不知道对方喜欢什么,别人也不知道自己喜欢什么。在目前传统的聊天软件中,要找个聊得来的人,先要聊才知道是否能和别人聊得来,一起聊是否有乐趣,我觉得这是盲目的,如果能直接根据自己的乐趣去找一个聊的来的人,不就容易多了,那会不会减少盲目性。我想即时通讯的革新或许也将到来。



## 国内外研究现状

Hybrid App 的兴起启示是互联网产业的一种偶然。当移动开发兴起的时候,很多互联网公司前赴后继地冲向了移动互联网这块肥肉,但实际开发中发现从事移动互联网的人才并不是很多,导致市场的疯狂追捧移动互联网人才。而当前市场的机制下移动互联网培养的团队又是极其困难的,培养一个有能力进行跨平台开发的团队更是如此。资金,人力对于一个中型公司都难以支撑。而 HTML5 的出现让 Web App 露出了光芒。但在用户体验,用户习惯,用户渠道上,有着难以解决的问题,这也使得 Web App 难以爆发。但益于基于 Html5 的低跨平台成本,也愈发地不断地吸引着开发公司。如何解决这个问题呢?正是这种巧合,基于开发平台优势又兼具 Native App 特质的 Hybrid App 技术进入,使得很快吸引了无数公司的目光。就目前来说主流的跨平台工具来说,PhoneGap 应该是最早出现在公众的眼前,第一个版本的 PhoneGap 代码是在 2008 年 8 月的 iPhoneDevCamp 上写成的,目前由 Adobe 收购,并将源代码贡献给 Apache 基金会成立 Cordova,Adobe 保留在线编译功能。而后的微软在 visual studio 集成了 Cordova,可见互联网巨头对跨平台开发的重视!而更重要的是 facebook 也是一直致力于跨平台开发,而它推出的 Reactjs 就起源于 Facebook 的内部项目,用来架设 Instagram 的网站,并于 2013 年 5 月开源。

## 研究内容

该论文主要研究了在 Intel XDK 中,借助 Cordova 和 HTML5 如何实现跨平台软件的开发和编译,并开发出一款适合 IOS,Android,WindowPhone 和 Win10 应用商店的手机,电脑,平板和 Web 的即时聊天软件,该软件可以在任何设备下与好友进行聊天,添加好友,完善好友信息和查看附近好友,发送附近信息。支持账号登录,注册,取回,修改账号,可以通过短信来确认身份。

## 研究方法

利用跨平台软件编程思想,使用跨平台移动框架 Ionic 结合基本的 UI 知识,打造出简约美观的跨平台 APP 软件前端。使用 Java Web 的 spring 和 spring mvc 框架进行后端模块话层次化,使用 mongodb 和 redis 进行数据储存,利用 NoSql 的特性使得服务器响应速度保持在毫秒级别。并最终将代码上传到服务器进行试调,将前端打包成手机端能识别的格式进行分发安装,并最终全端试调成功。

## 1. 软件需求分析

### 1.1 目标需求

目前聊天软件并不是很多，但也不少，如何在这不多不少的聊天系统中占的一席之地，那只有通过别人没有的不一样的功能才能够占据一席。聊天软件虽然多，但是目前人们的需求正处于不断的变化。很多人都喜欢宅，可是他们是真的都喜欢宅吗？我看未必，我从解放用户活泼天性是一个很好的突破口，我着力从用户需求入手，针对用户的兴趣爱好，设计出一款针对于用户兴趣爱好社交的软件。而目前手机用户越来越多，但手机系统确是多样的，如何在每一个系统中留住全部用户，是所有软件工程师的一个大难题。因此我决定使用跨平台开发技术，通过跨平台技术实现一款在多个平台能够完美运行的即时通讯软件，使用户在多个平台下依然能够通过自己的喜好进行聊天。

### 1.2 可行性分析

#### 1.2.1 技术可行性分析

硬件方面由于使用了 Intel XDK 作为编辑器和编译工具，所以开发只需要任何一台 Windows 电脑或苹果电脑，试调的话也只要任意一台智能手机即可。如要搭建本地编译环境，如果要开发 Windows Phone 则需要一台 windows 电脑，如果要开发 IOS 则需要苹果电脑，手机方面也只要任意一台智能手机即可。如果使用在线编译的话，我想可以和一堆不同的开发和试调设备说再见了，所以对应硬件来说是可行的。

软件方面使用了 Angularjs 和 Ionic 做为前端开发，Ionic 作为知名的国外的移动库，虽然说学习 Angularjs 条件是陡峭的，但是学习完成后，你会发现它非常好用，在 app 的数据处理和双向数据绑定和值得目前的 js 库学习。后端使用 java Web 的 Spring 和 Spring MVC，前者的依赖注入和控制反转使得非常方便的进行数据调用，后者实现了模块视图控制器之间的数据交互。在数据库的选择上选择了速度较快的 mongodb，由于即时通讯对速度的要求，所以选择速度作为第一位。同时使用了 redis，由于它的多种类型，使得它天生就很很高效率的实现数据的使用，如百万级以上的排行榜可以使用有序列表，投票功能可以使用集合等，极大的提升后台反应速度。在实时交互选择了比轮询相比极大节约资源的长链接 WebSocket，使效率得到强有力的提升，所以从软件方面来说是可行的。

#### 1.2.2 操作可行性分析

界面的构建使用了 Ionic，拥有触屏，滑动，下拉等多种功能，极大方便用户使用。再通过 Cordova 使用调用原生功能，使得拍照上传，照片选取变得非常简单，注册之后可以直接和好友聊天，操作也十分便利和简单

## 1.3 需求分析

### 1.3.1 对功能的规定

表 1-1 功能列表

程序功能列表	
注册登陆	可以通过手机登陆注册帐号
好友聊天	与好友进行即时通信
好友管理	可以删除, 添加, 拉黑等操作
附近动态	可以看到附近的朋友的动态
完善好友资料	当好友资料不全时可帮助完善资料

### 1.3.2 其它功能

进入聊天后可以通过右滑弹出菜单, 可通过菜单进行对应的好友查找, 切换账号等功能, 还可以滑动正在聊天的好友, 进行增删改功能。进入好友列表, 可以点击好友的第一个字进行操作, 跳转位置, 可以快速定位好友。进入附近动态, 可以通过下拉操作获取最新的附近动态。这一些些操作主要是为了能快速帮助用户更快更方便实现用户想要功能。

## 2. 软件开发技术基础

### 2.1 AngularJS 介绍

AngularJS 由 Misko Hevery 等人在 2009 年创建，后被谷歌公司收购，它是一款优秀的前端 JS 框架，现出现在谷歌的多款产品中，Angularjs 有着和传统 JS 框架不同的新特新，最为重要的应该是 MVVM，模块化，自动双向数据绑定，依赖注入，语义化标签等等。

而 AngularJS 的创建也是解决了传统 JS 框架或库不足，也是为了 HTML 在构建应用上的不足而设计的。AngularJS 通过使用 directives（指令）的结构，让浏览器识别新的语法，如：使用 `{{}}` 语法进行数据绑定，使用 DOM 控制结构来实现隐藏和迭代 DOM 片段，通过数据绑定直接验证表单，将逻辑代码关联到 DOM 和将 HTML 划分成可重用组件等等。

### 2.2 ionic 介绍

ionic 是一个用于开发 Hybrid App 的开源，免费的代码库，依赖 AngularJS。用于构建高效的应用程序，优化 html, js, css 的性能的 Hybrid App 框架。

ionic 是一个专注于使用 Web 技术来开发高性能的混合应用的的开发框架。同时绑定了 AngularJS 和 Sass，从 web 的角度来开发手机应用，基于 Cordova 的编译平台，可以直接编译成各个平台的手机应用，可实现手机应用的高速开发。

### 2.3 Cordova 介绍

Cordova 通过提供一系列的与设备相关的 API，通过这些 API，使移动应用可以直接使用 JS 来调用原生的设备，如摄像头，话筒，麦克风等等。通过统一的 JavaScript 类库，可以直接调用原生的功能。Cordova 支持如下系统 Android, IOS, Windows Phone, buntu phone os, Blackberry, Palm WebOS, Bada, Symbian 等等。

Cordova 是 Adobe 从 PhoneGap 抽离的核心代码，是驱动 PhoneGap 的核心引擎，现在 Cordova 已经由 Apache 基金会进行管理。同时 Visual Studio 2015 也已经集成了 Cordova，用于开发者开发使用。

### 2.4 spring 介绍

spring 是一个轻量级的开源的 Java 开发框架，现主要用于解决企业应用开发的复杂性，可用于任何的 Java 应用。

框架的特征是轻量，控制反转，面向切面，容器，符合典型的框架模型，具有 MVC 设计思想。拥有方便解耦、简化开发、AOP 的编程支持、声明式的事务支持、通过对 Junit4 的支持，可方便的测试 Spring 程序、支持集成各种优秀的框架，降低了 Java EE API 的使用难度，目前是 java 源代码的学习典范。

## 2.5 spring MVC 介绍

Spring MVC 属于 SpringFrameWork 的后续产品，目前已经融合进入 Spring Web Flow 里面。提供了构建应用程序的全功能 MVC 模块。分离了控制器和模型，使之很好的和其它 View 无缝集成。

通过接口，Spring MVC 框架是高可配置度的，而且包含多种配置技术，如 JSP，Velocity 的模版技术。同时分离控制器，模型对象，分派器和处理程序对象。这样的分离即给予了超高的自由度，又使得模块的可维护度增加了。

## 2.6 IDE 介绍

Spring tool suite，是一个基于 Eclipse 开发的一款专门用于 spring 开发的一款编辑器，自带了服务器和集成了 Maven 等有用的功能

Intel XDK，是 Intel 公司推出的一款制作跨平台软件的免费的编辑器，集成了 Cordova 和在线编译，无需搭建跨平台环境，即可实现跨平台编辑和编译一键式开发。

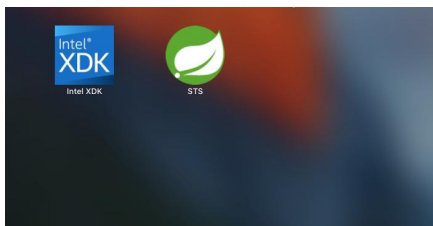


图 2-1 工具需求

## 2.7 开发环境配置

首先在 Intel@XDK 的官网进行 XDK 的下载和安装，之后打开 XDK，新建 HTML+Cordova 项目，建立完成后选择完成，如图 2-2 所示，即可简单完成前端程序的建立。后端程序的建立需要使用 STS 这个工具，也是需要在 Spring 的官网中下载，下载完成后建立 MVN 项目，建立完成后，目录结构如 2-3 所示，在 pom 文件中配置依赖关系图如 2-4 所示，建立完成依赖后，右击工程点击 MVN 选择 Update Project，即可完成后端程序的配置。这样我们的基本环境都建好了，可以直接进入开发阶段了。

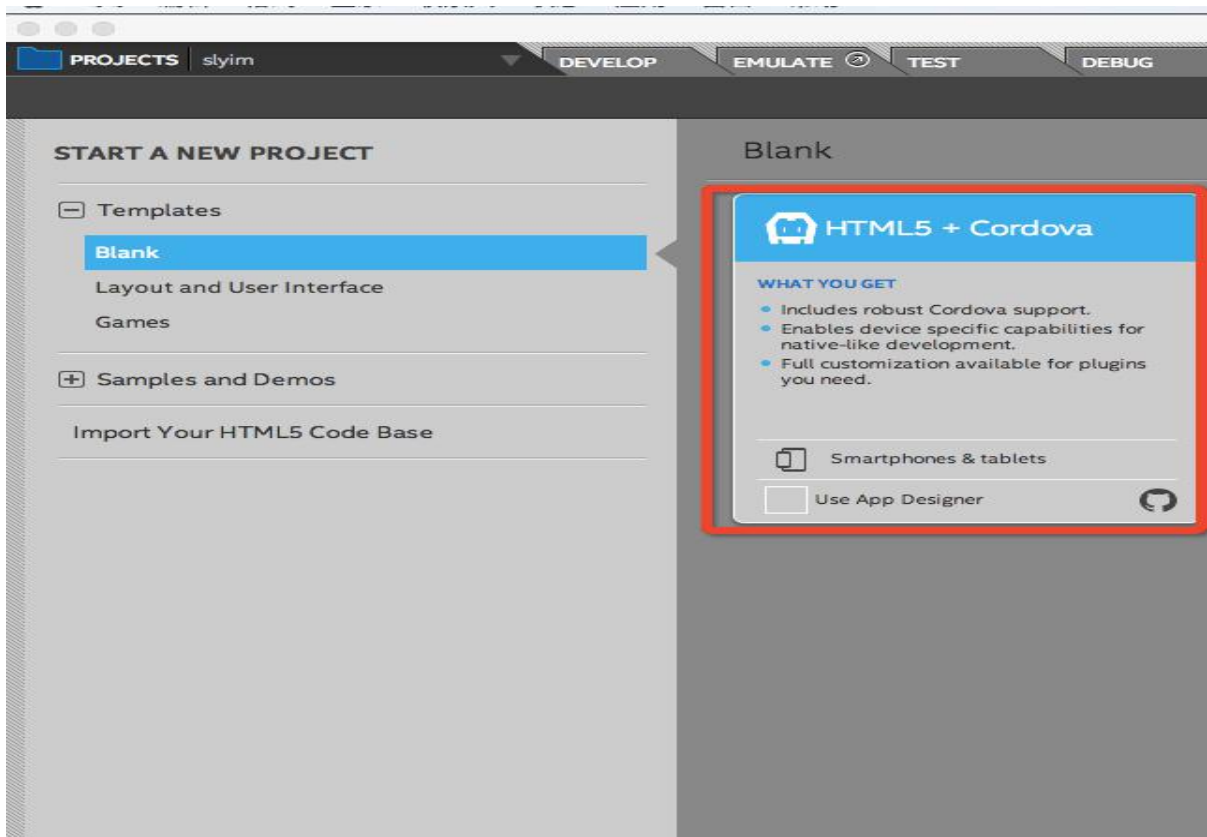


图 2-2 创建项目

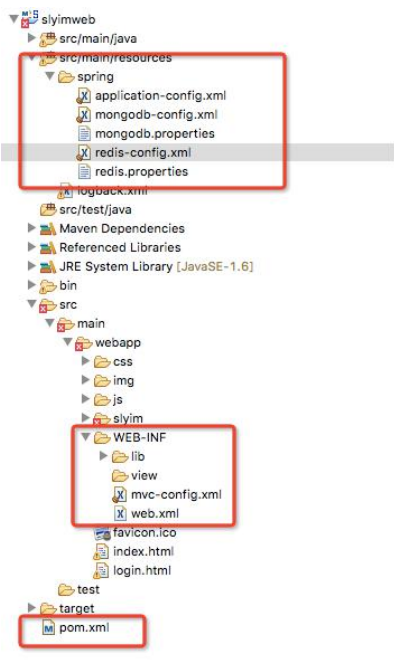


图 2-4 后端目录结构（左）

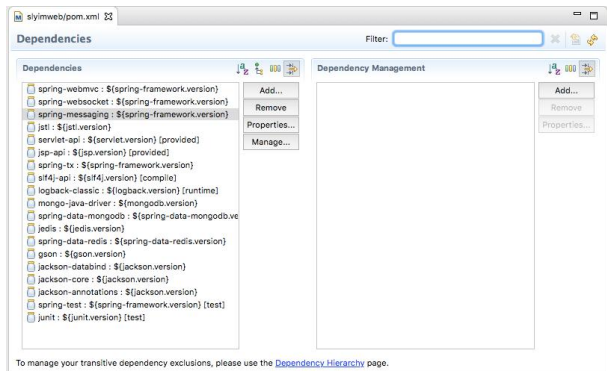


图 2-5 文件依赖（右）

### 3. 软件设计

#### 3.1 软件结构设计

##### 3.1.1 即时性

聊天通讯软件，即时通讯那就是要保证即时，那跨平台软件如何保证即时性呢？由于我们是使用 html 构建程序核心的，所以只能 html 的方法来实现即时，原有的方法是使用轮询的方法来实现即时通讯，这种方法要不断地发送 html 请求，如果量大的话十分消耗服务器性能。但我们使用 html5 的新特性 WebSocket, WebSocket 是长连接，只要一次请求就会一直连接服务器，不需要不断发送 html 请求，减轻服务器压力的同时，也保证了即时性。

##### 3.1.2 高效性

做过 Android 或 IOS 的同学都应该知道，如果是一个较大的列表都有一个回收机制，就是只显示能看到的条数，不能看到的不加载，已经看过去的回收回来继续给新的内容使用。这样使一个大的列表具有高效性。而 HTML 并没有原本就要解析，需要时间。如果做到高效呢？其实 ionic 其中有一条主旨就是就是为了开发高效的应用。那居然如此 ionic 是如何使我们的跨平台程序变的高效呢？比如在 ionic 有一个指令是 collection-repeat，这个就是用于高效循环显示列表，原理则是使用了 html5 的新特性 canvas，使用这个可以做到不断销毁不需要的项，保证了跨平台应用的高效性。

##### 3.1.3 数据同步性

做跨平台应用，怎么样将较大的数据进行存储呢？众所周知 Android 和 IOS 在本地都有一个 SQLite，那我们如何将较大的数据储存起来呢？很多人或许会说使用 cookie，但这个方法是肯定不行的，因为 cookie 在每次发送 html 请求的时候都会发给服务器，这不但会浪费服务器性能也会浪费本地应用的性能，更何况是大量数据呢？这个也是使用 html 的新特性 localStorage，这不仅仅可以存放较大的数据，也不会再在发送请求的时候上传到服务器。但这与数据同步有什么关系呢？我们可以利用它存储 json 格式数据，应用开始时解压数据，而应用数据变化时再压缩回 localStorage 中，这样就实现了应用数据的同步，还很好地将数据长时间地保留了下来。

#### 3.2 数据库的设计

##### 3.2.1 MongoDB 介绍

MongoDB 是一个机遇分布式文件存储的数据库，由 C++ 进行编写。旨在提供高可扩展性和高性能的数据存储解决方案。

MongoDB 是介于关系型数据库和非关系型数据库之间的产品，目前是非关系型数据库功能最丰富的，最像关系型数据库的数据库。但它支持的数据结构却是十分松散的，是类似于 Json 的 bson 格式，因此可以存储相对复杂的数据类型。并且它支持的查询语言非常强大，类似于面向对象查询，且支持关系型数据库查询的绝大部分功能，

并且支持建立索引。由于 mongodb 不是结构型数据库，表的字段是可变的，所以以结构出现。

表 3-1 数据库表

MongoDB 数据表	
userMongo	用户信息表
addressMongo	联系人信息表
nearMongo	附近的人信息表
chatMongo	聊天记录表
calendarMongo	日程表

### 3.2.2 mongoDB 的数据表结构

数据表使用类型结构：

UserBase 结构：

```
{
float latitude=0;
float longitude=0;
}
```

Localtion 结构：

```
{
String id;
String name;
String avatar;
}
```

以下为数据表结构：

用户信息表, 用于存储用户信息

表 3-2 用户表

用户表		
字段	类型	作用
id	String	唯一信息标识
name	String	用户昵称
avatar	String	用户头像地址
phone	String	用户手机号码
password	String	用户密码
sign	String	用户签名
strong	List<String>	用户爱好
sex	int	用户性别
rent	float	用户租金
localtion	Localtion	用户位置
modifyTime	long	修改资料时间
vipScore	Int	VIP 分数
vipDeadLine	long	VIP 过期时间
vipRefreshTime	long	VIP 上次更新分数时间



表 3-3 联系人表

联系人列表		
字段	类型	作用
id	String	唯一信息标识
friendName	String	好友昵称
friendAvatar	String	好友头像地址
friendSign	String	好友签名
friendId	String	好友唯一用户标识
myId	String	我的唯一用户标识
commit	String	申请留言信息

表 3-4 附近动态表

联系人列表		
字段	类型	作用
id	String	唯一信息标识
sendTime	Long	发起时间
content	String	发起内容
type	String	内容类型
favorNum	Long	点赞数
sender	UserBase	发送者
localtion	Localtion	地理位置
commentId	String	评论他人唯一动态标识
forwardId	String	被评论唯一动态标识

表 3-5 聊天记录表

联系人列表		
字段	类型	作用
id	String	唯一信息标识
sendTime	Long	发送时间
acceptTime	Long	接受时间
content	String	内容
type	String	内容类型
myId	String	我自己的用户唯一标识
friendId	String	好友的用户唯一标识

表 3-6 日历表

联系人列表		
字段	类型	作用
id	String	唯一信息标识
startTime	Long	开始时间
endTime	Long	结束时间
event	String	事件名字
address	String	地址名字
sponsor	UserBase	发起人
acceptor	UserBase	接受人
rent	float	租金
state	int	状态
star	int	评分

### 3.2.3 Redis 介绍

Redis 是一个使用 ANSI C 语言编写的，支持网络和数据持久层可日志的键值数据库，并提供多种语言的 API 提供操作。于 2010 年 3 月 15 日由 VMware 主持，现从 2013 年 5 月起由 Pivotal 赞助。

Redis 是一个键值存储系统，是一个 NoSql 数据库，相比 Memcahe 拥有更多种结构。并且 Redis 的性能十分不错，对关系型数据库又一个很好的补充。Redis 还支持主从同步，使得多结构的数据库群成为了可能。

### 3.2.4 Redis 的数据结构

用于防止恶意攻击的键，设置有效期为 15 分钟，验证次数最大为 10

key: CheckEvil:+ip 地址或手机号 value: 验证次数

该键为有序列表，用于统计用户的兴趣排名，可支持百万以上用户实时排行

key: userStrong:+年月日 value: 兴趣名+该兴趣名拥有的得分

该键为集合，集合的特性是只能存一个相同的数据，可快速计算大小，很适合投票和点赞系统

key: circleLike:+附近动态帖子 ID value: 用户 ID+: +点赞的用户名

使用Redis的目的是为了弥补普通关系型数据库的不足,就比如有序列表和集合,如果用关系型数据来做,不仅想对麻烦而且速度也是远远比 Redis 的慢。

就算mysql加B-tree索引,但是Redis是直接使用的hash进行的计算和排序(hash的打表排序就是在插入时,已经完成排序了),速度差距,不言而喻。

4. 系统的实现

4.1 前端界面设计原理

4.1.1 界面设计结构

前端界面的设计是使用 AngularJS，使前端符合 MVVM 设计思想，所以代码结构会有主页层，应用层，控制器层，服务层。不同的层次也有着不同的功能，而主页层主要是负责应用 js 文件和定义应用控制范围。应用层主要是分发模块和路由定义绑定功能，而路由功能则是绑定控制器和模板，使界面根据链接来显示。控制器层则是调用对应的服务和实现界面功能。服务层主要是负责数据请求和数据处理的功能，流程图可参考图 4-1。

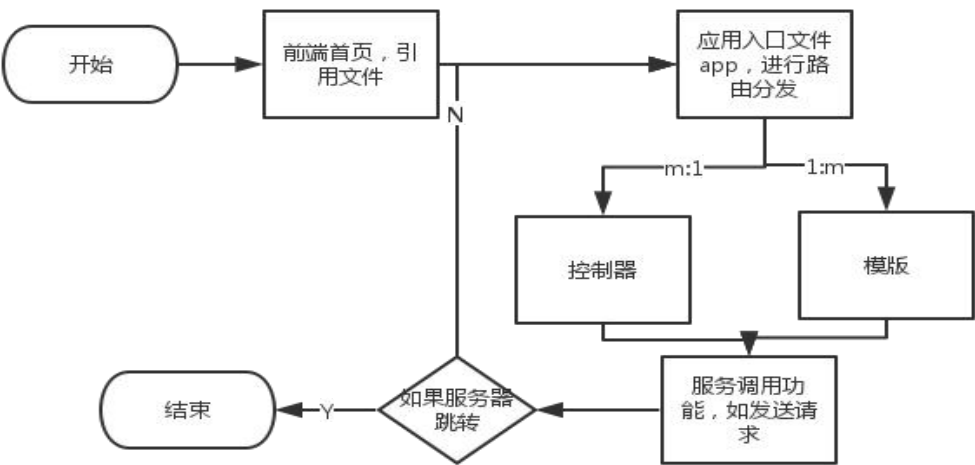


图 4-1 前端运行流程

4.1.2 欢迎界面

可滑动的欢迎界面，左边是 iphone6s 的截图，右边是 Google Nexus 4 的截图。这两张图应该算是目前苹果和安卓的代表作品。保证的跨平台性。通过 Web 技术的响应式布局使得界面可以在不同的界面上实现技术。同时如果是图片则放入不同分辨率的图片通过判断界面分辨率来使用分辨率最接近的图片，使界面不会发生变形。



图 4-2 不同手机运行

通过监听页面滑动界面来实现滑动，使用 HTML5 的过渡特效实现页面滑动，同时如果已经登陆会通过检测是否登陆来跳转进入聊天界面，如果未登录则判断是否打开过欢迎界面，如果打开过则，直接进入登陆界面。

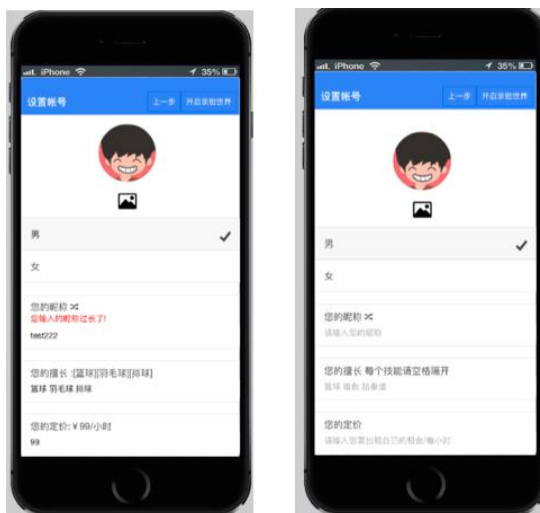
4.1.3 注册界面

注册界面主要可以手机验证界面和注册信息填写界面和更换手机号界面，着三个界面分别右如下功能，验证界面如图 4-3 的功能则为验证手机是否为该用户所持有。填写用户信息界面则为图 4-4。



图 4-3 手机验证界面

通过与后端 Json 交互，告诉后端发送短信，后端使用阿里大鱼接口发送短信。然后通过使用的短信模板发送给用户验证信息。如果是该用户所持有则进入下一步，而如果用户之前注册过则直接进入聊天界面。



(a) 填写用户信息的提示 (b) 点击性别后出现的界面

图 4-4 信息填写界面

如果用户验证手机成功则可以填写用户信息,需要填写的用户信息有性别,昵称,擅长,定价,填写以上信息后则可以直接进入聊天界面。更换手机界面则就是简单的验证自己之前的手机号后,再验证新手机号,如果之前的手机号注册过,则用新帐号替换旧手机帐号,无论是否新手机帐号有过注册。

### 获取短信代码

```
public JsonBack getRegSms(HttpServletRequest request, HttpSession
session)
{
    String phone=(String)request.getParameter("phone");
    if(this.checkPhone(phone)){
        jsonBack.setCode("-1");
        jsonBack.setMsg("指定参数没有传入");
        jsonBack.setRes("");
        return jsonBack;
    }
    // 获取随机的短信密码
    String password=this.getPassword();
    session.setAttribute("phone", phone);
    session.setAttribute("password", password);
    if (this.checkEvil(request,phone)) {
        jsonBack.setCode("-3");
        jsonBack.setMsg("你操作过于频繁, 请 1 小时后再试!");
        jsonBack.setRes("");
        return jsonBack;
    }
    //发送短信代码
```

```

SendSms sendSms=new SendSms();
// 验证用户发送次数及是否发送。及发送成功返回 True 结果
if ( sendSms.authUser(phone, password) ) {
    jsonBack.setCode("0");
    jsonBack.setMsg("发送短信成功!");
    jsonBack.setRes("");
} else {
    jsonBack.setCode("-2");
    jsonBack.setMsg("发送短信失败!");
    jsonBack.setRes(null);
}

return jsonBack;
}

```

### 注册代码

```

/ **
    将 POST 数据映射到 userData 变量上
* /
@RequestMapping(value="/regin", method = RequestMethod.POST)
@ResponseBody
public JsonBack regin(HttpServletResponse response, HttpServletRequest
request, HttpSession session, @RequestBody UserMongo userData)
{
    //,@RequestBody List<String> strong, @RequestBody Localtion
    localtion
    String phone=(String)session.getAttribute("phone");
    String password=(String)session.getAttribute("password");
    String pwd=userData.getPassword();
    // 判断密码是否正确
    if(password==null){
        jsonBack.setCode("-4");
        jsonBack.setMsg("注册超时! 请重新获取短信!");
        jsonBack.setRes(null);
        return jsonBack;
    }
    if(!password.equals(pwd)){
        jsonBack.setCode("-1");
        jsonBack.setMsg("手机验证码错误!");
        jsonBack.setRes(null);
        return jsonBack;
    }
    // 设置用户数据
    userData.setPhone(phone);
    userData.setModifyTime(System.currentTimeMillis()/1000);
}

```

```

        userData.setVipRefreshTime(System.currentTimeMillis()/1000);
        userData.setVipDeadLine(System.currentTimeMillis()/1000);
        if (
            userData.isEmptyPhone() || userData.isEmptyPassword() ||
            userData.isEmptyAvatar() || userData.isEmptySex() ||
            userData.isEmptyName() || userData.isEmptyStrong() ||
            userData.isEmptyRent()
        )
        {
            jsonBack.setCode("-2");
            jsonBack.setMsg("请正确填写完全部信息!");
            jsonBack.setRes(null);
            return jsonBack;
        }
        // 判断用户是否存在
        userMongo=userServiceMongo.findOneByPhone(phone);
        //System.out.println(userMongo);
        if (userMongo==null) {
            //userMongo.setLocaltion(localtion);
            userServiceRedis.setStrongCount("userStrong",
            userData.getStrong());
            userData.converAvatar(request, FileType.AVATAR);
            userServiceMongo.insert(userData);
            MyUser.makeLoginInfo(response, session, userData);
            jsonBack.setCode("0");
            jsonBack.setMsg("注册成功!");
            jsonBack.setRes(userData);
        } else {
            userServiceRedis.setStrongCount("userStrong",
            userMongo.getStrong());
            / * *
            将登陆状态写入服务器中
            * /
            MyUser.makeLoginInfo(response, session, userMongo);
            jsonBack.setCode("1");
            jsonBack.setMsg("账号已存在, 即将取出账号!");
            jsonBack.setRes(userMongo);
        }
        return jsonBack;
    }
}

```

#### 4.1.4 实时聊天技术的实现

主要是用于显示好友聊天列表和显示消息数量, 点击好友头像可进入聊天, 实现

好友信息实时发送和实时接受功能，次要功能有查看好友资料，删除当前聊天记录。界面如图 3-5。



(a) 聊天显示列表 (b) 滑动用户头像可以出现的功能

图 4-5 聊天界面

实时聊天的实现是通过 HTML5 的新特性，WebSocket 来实现的。而 webSocket 和普通的轮询方法优势在于不需要定时发送是否有新消息的请求，节约了发送 web 头文件的流量而且还不需要轮询时间的等待，只要发出信息就可以实现接收。原理图如 4-6

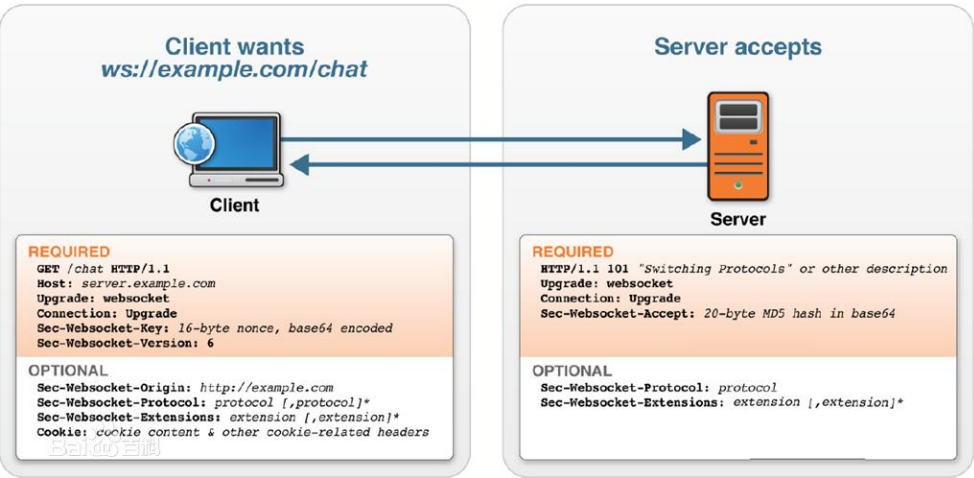


图 4-6 WebSocket 原理图

在 java web 和前端的交互中，有多种类型，所以必须要用 Json 将类转换为字符串来实现数据交互，所以我定义了如下了 POJO 类，来惊醒接受和发送 WebSocket 的信息。这个类也通过了类型来分别当前 socket 数据的作用，代码如下。

```
/**
 * WebSocket 的接受类
 */

public class MyWebSocketMsg {
    private String sockType;
```



```

private String jsonData;

public String getSockType() {
    return sockType;
}

public void setSockType(String sockType) {
    this.sockType = sockType;
}

public String getJsonData() {
    return jsonData;
}

public void setJsonData(String jsonData) {
    this.jsonData = jsonData;
}
}

```

而要真正使用这个类，则是通过如下方法进行调用，通过循环来查找要发送的目标，并且在 POJO 中，也定义了对应的用户发送的唯一标示 ID。数据交互代码如下。

```

/ **
    通过实现 WebSocket 接口实现对应的 WebSocket 功能
* /
public class MyWebSocketHandler implements WebSocketHandler {
    // json 转换变量
    private Gson gson=new Gson();
    // websocket 储存的 Session 数据
    private static final ArrayList<WebSocketSession> users=new
ArrayList<WebSocketSession>();
    // 地址的服务类，用于进行数据操作
    @Autowired
    private AddressServiceMongo addressServiceMongo;
    / **
    实现 WebSocket 的连接关闭后的操作
    * /
    @Override
    public void afterConnectionClosed(WebSocketSession session, CloseStatus
arg1) throws Exception {
        users.remove(session);
    }
    / **
    接受到信息后的操作
    * /
    public void handleMessage(WebSocketSession session, WebSocketMessage<?>
message) throws Exception {

```

```

        MyWebSocketMsg sockMsg=gson.fromJson((String)message.getPayload(),
MyWebSocketMsg.class);
        if (sockMsg.getSockType()==null || sockMsg.getSockType().equals(""))
        {
            return;
        }
        / **
        获取 Socket 的类型，是否是添加好友类型
        * /

        if (sockMsg.getSockType().equals("AddFriend")) {
            AddressMongo
addressMongo=gson.fromJson(sockMsg.getJsonData(), AddressMongo.class);
            AddressMongo
toAddressMongo=addressServiceMongo.findOneByMyIdAndFriendId(addressMongo.g
etFriendId(), addressMongo.getMyId());
            sockMsg.setJsonData(gson.toJson(toAddressMongo));
            TextMessage returnMessage = new
TextMessage(gson.toJson(sockMsg));
            sendMessageToUser(toAddressMongo.getMyId(), returnMessage);
        } else if (sockMsg.getSockType().equals("SendMsg")) {
            ChatMongo chatMongo=gson.fromJson(sockMsg.getJsonData(),
ChatMongo.class);
            ChatMongo toChatMongo=new ChatMongo();
            String myId=(String)session.getAttributes().get("id");
            toChatMongo.setType(chatMongo.getType());
            toChatMongo.setContent(chatMongo.getContent());
            toChatMongo.setMyId(chatMongo.getFriendId());
            toChatMongo.setFriendId(myId);
            toChatMongo.setSendTime(System.currentTimeMillis()/1000);
            sockMsg.setJsonData(gson.toJson(toChatMongo));
        }
        / **
        用于 WebSocket 的文本封装
        * /
        TextMessage returnMessage = new TextMessage(gson.toJson(sockMsg));
        sendMessageToUser(toChatMongo.getMyId(), returnMessage);
    } else {
        return;
    }
}

```

## 4.2 用户相关操作的实现

4. 2. 1 好友管理界面



(a) 好友管理界面      (b) 滑动头像界面      (c) 快速定位联系人界面

图 4-7 联系人界面

联系人在打开软件时会有一个判断是否最后添加联系人是否有变化，如果最后添加联系人的时间在更新时间后，则会更新联系人数据。主要是用于好友的增删，完善好友资料，对好友有加入黑名单的操作，通过添加好友，还可以实现通过好友昵称和好友手机号码来查询对方，从而进行好友添加。点击好友的首字母可以实现好友首字母查询，可以直接查询该首字母的好友。如图 4-7 所示。

4. 2. 2 附近动态界面

可以在发送附近的动态，可以查看附近的人发送的动态，还有下拉刷新的功能，想看最新的动态只要下拉就可看到最新的信息。功能界面如 4-8 所示。



(a) 附近动态界面      (b) 下拉滑动刷新

图 4-8 附近动态界面

```
修改获取动态代码
@RequestMapping("/getCircle")
```

```

@ResponseBody
public JsonBack getCircle(HttpServletRequest request, HttpSession
session){
// 获取发送获取动态时间
String sendTimeStr=(String)request.getParameter("sendTime");
long sendTime = Long.parseLong(sendTimeStr);
if ((System.currentTimeMillis()/1000)-sendTime>3600*24*30*3) {
    sendTime=(System.currentTimeMillis()/1000)-3600*24*30*3;
}
// 将附近动态打成列表进行返回
List<DynamicCircleMongo> dCircleList=
dynamicCircleServiceMongo.findListBySendTime(sendTime, 0, 50);
if (dCircleList==null) {
    jsonBack.setCode("-1");
    jsonBack.setMsg("没有任何信息更新");
    jsonBack.setRes(null);
} else {
    jsonBack.setCode("0");
    jsonBack.setMsg("获取成功");
    jsonBack.setRes(dCircleList);
}
return jsonBack;
}

```

#### 4.2.3 修改好友资料界面

修改好友资料可以通过各个方法来进入，可在联系人处和在聊天界面滑动头像，点击查看 / 修改好友资料来进入好友的修改和查看资料，完善好友资料后，如果好友在两周内上线，则不可以修改好友资料。界面如图 3-8 所示。



图 4-9 修改好友资料界面

修改用户信息代码

```

    @RequestMapping(value="/upUserInfo", method = RequestMethod.POST)
    @ResponseBody
    public JsonBack upUserInfo(HttpServletResponse response,
    HttpServletRequest request, HttpSession session, @RequestBody UserMongo
    userData)
    {
        // 判断用户是否是本人
        userMongo=userServiceMongo.findOneById(userData.getId());
        / **
        如果是本人则可以修改, 如果是其它人, 则需要判断资料最近是否被修改和本人
        用户是否最近使用过这个软件
        * /
        if (userMongo!=null) {
            String id=(String)session.getAttribute("id");
            //System.out.println(userData.getName());
            if (userData.getId().equals(id)) {

                // 用户数据修改和更新
                userMongo.setAvatar(userData.getAvatar());
                userMongo.setSign(userData.getSign());
                userMongo.setSex(userData.getSex());
                userMongo.setName(userData.getName());
                userMongo.setStrong(userData.getStrong());
                userMongo.setRent(userData.getRent());
                userMongo.converAvatar(request, FileType.AVATAR);
                userServiceMongo.update(userMongo);
            } else {
                if
                (userMongo.getModifyTime()+14*24*3600-(System.currentTimeMillis()/1000)<0)
                {

                    // 用户数据修改和更新
                    userMongo.setAvatar(userData.getAvatar());
                    userMongo.setSign(userData.getSign());
                    userMongo.setSex(userData.getSex());
                    userMongo.setName(userData.getName());
                    userMongo.setStrong(userData.getStrong());

                    userMongo.setModifyTime(System.currentTimeMillis()/1000);

                    userMongo.setVipRefreshTime(System.currentTimeMillis()/1000);

                    userMongo.setVipDeadLine(System.currentTimeMillis()/1000);
                    userMongo.converAvatar(request, FileType.AVATAR);
                    userServiceMongo.update(userMongo);
                }
            }
        }
    }

```

```
        } else {  
            jsonBack.setCode("-2");  
            jsonBack.setMsg("修改时间未到，不能修改");  
            jsonBack.setRes(null);  
            return jsonBack;  
        }  
    }  
    // 更新成功，返回数据  
    jsonBack.setCode("0");  
    jsonBack.setMsg("修改用户成功");  
    jsonBack.setRes(null);  
} else {  
    jsonBack.setCode("-1");  
    jsonBack.setMsg("用户不存在");  
    jsonBack.setRes(null);  
}  
return jsonBack;  
}
```

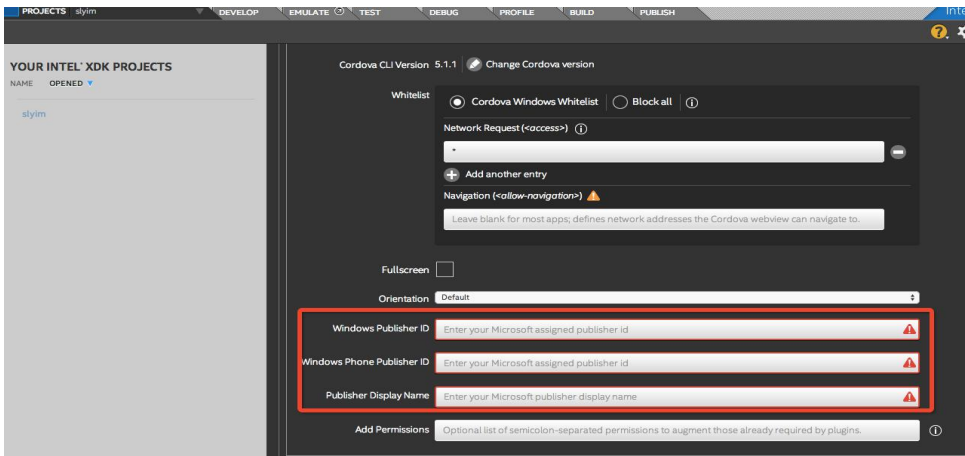
为什么允许他人修改资料，主要是为了节约新用户的注册时间，如果别人帮助你完善了资料就只要验证用户手机号码，不需要填写用户信息。其次增加用户的互动和上线频率，如果你长时间不上线，那就可以让其它用户帮你修改资料，达到自身的资料获得最新的更新。

5. 软件测试和编译

5.1 编译客户端

5.1.1 编译准备

在 XDK 中安卓不需要设置,即可实现在 Build 选项卡中实现编译,而 IOS 和 Windows 则需要通过获取生成开发者证书来实现编译。由于 XDK 是不需要本地环境就可以实现编译,所以编译过程需要全程联网,让 Intel 提供的服务器来帮助自己编译各个平台的程序。开发者证书的填写位置如图 5-1 和图 5-2。5-1 为 IOS 证书填写位置,5-2 为 windows 推送 ID 填写位置,图片如下。



5-1 IOS 编译需求

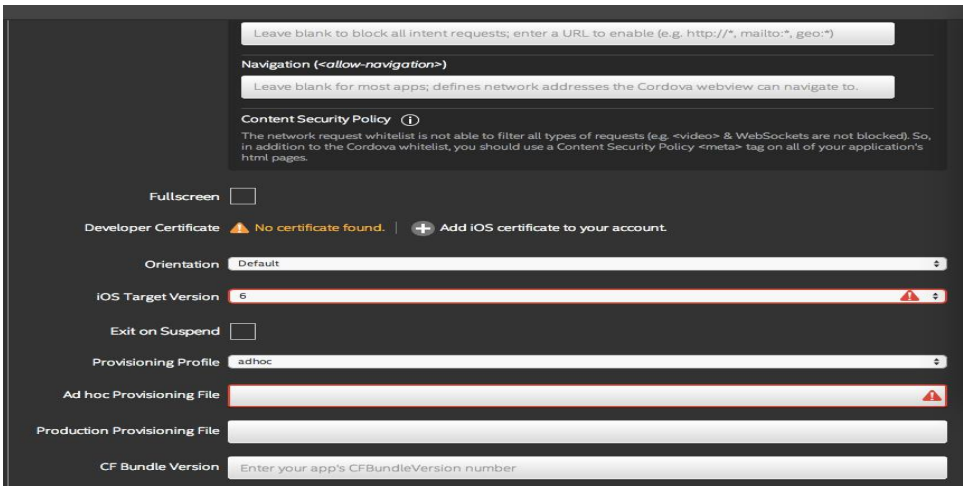


图 5-2 WindowsPhone 编译需求

### 5.1.2 应用编译

编译则极其简单，只需要在 Build 的选项卡，勾选对应你可以编译的任务，如安卓平台的话，只要勾上安卓的卡片后，再点击 Start Builds 后就可以编译了，编译后可以直接点击卡片的白色箭头下载和安装。如图 5-3，图为安卓编译完成。

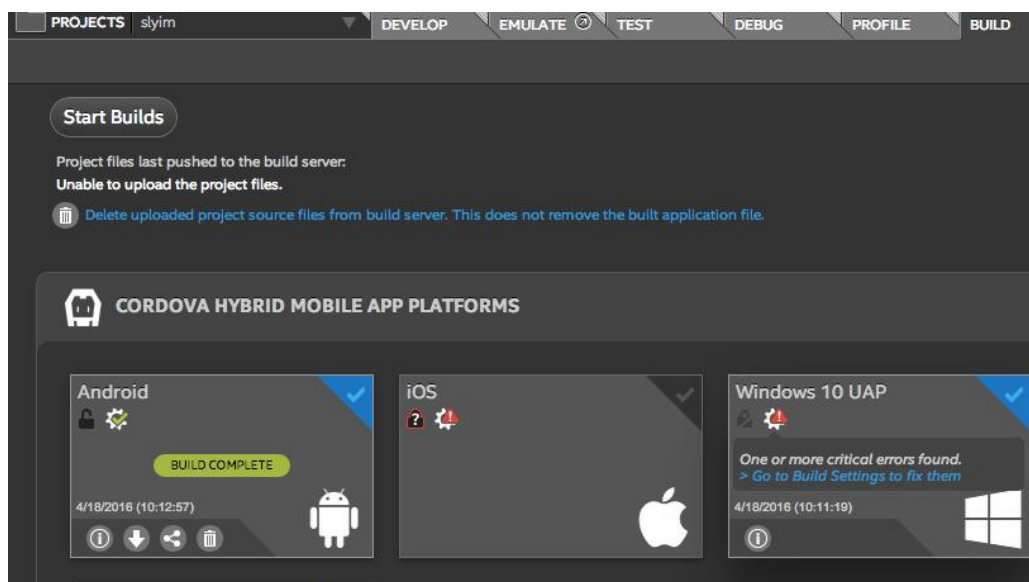


图 5-3 编译界面

## 5.2 编译服务器

### 5.2.1 编译准备

首先准备号服务器环境，基本来说要先安装 jdk 后，再进行 jetty 的源码编译后，再将 MongoDB 和 Redis 数据库安装完成后，即可进行下一步编译。

### 5.2.2 编译

打开 STS，右击项目，选择 Run As 再选择 Maven Install 进行将服务器文件编译，编译后文件会出现在项目的 target 目录中，将 war 文件改名为 root.war 后放入之前编译好的 jetty 的 webapp 目录中即可，输入对应服务器的域名+8080 即可运行。



## 参考文献

- [1] (美)Jeremy Wilken . Ionic in Action: Hybrid Mobile Apps with Ionic and Angularjs . Manning Publications . 2015-10-01
- [2] 许令波. 深入分析 JavaWeb 技术内幕 (修订版). 电子工业出版社. 2014-08-01
- [3] 陈亚辉, 李新锋. JSP 网络开发逐步深入. 清华大学出版社. 2010-07-01
- [4] (美)维斯 著, 冯舜玺 译. 数据结构与算法分析:C 语言描述 (原书第 2 版). 2004-1-1
- [5] (美)Mark Pollack. Spring Data: Modern Data Access for Enterprise Java. 东南大学出版社. 2013-6-20
- [6] (美)Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, William Markito. The Java EE 7 Tutorial. 机械工业出版社. 2015-05-01
- [7] (美)埃克尔. Java 编程思想. 机械工业出版社. 2007-06-01
- [8] (美)巴克 著, 万波 译. BeginningJavaObjects 中文版从概念到代码. 人民邮电出版社. 2007-1-1
- [9] (英)Jason Lengstorf 著, Phil Leggetter 著, 肖智清 译. 构建实时 Web 应用: 基于 HTML5 WebSocket、PHP 和 jQuery. 机械工业出版社. 2013-09-01
- [10] (美)Nicholas S. Williams 著; 王肖峰 译. Java Web 高级编程: 涵盖 WebSockets、Spring Framework、JPA. 2015-06-01

## 总 结

经过几个月的编写，发现 Hybrid App 的强大之处，发现 Hybrid App 的难点很多是在于数据同步和数据处理，但得益于 AngularJS 强大的数据处理能力，我不得不对创作者表达由衷的倾佩，加之 Nodejs 及其社区地极度快速地崛起，或许下一高级语言中的 C 即将到了。早在大一就接触了服务器的开发，但一直都是在使用 PHP，但刚准备做毕业设计的时候，我觉得要选一个稍微有挑战性的语言，据说 Java Web 是学习难度最大的一门编程语言，于是我决定从 java Web 入手，但看到 java 的注解的优势和 java 的安全优势，发现 java Web 在服务器安全，包括目前的大数据分析（目前主流的大数据分析都是 java 写的）面前确实是无可替代。但与目前互联网开发的速度也确实格格不入，如果用 PHP 或许后台可以用两周内完成，但 Java 却用了我两个月，层次是在是太多了。但就性能来说 java 的架构就算是如此庞大，但执行速度发现并没有比 PHP 要慢。随着移动平台的兴起，移动开发也随之大量发展，但很多之前没有暴露的问题也随之而来。如手机操作系统很多，光目前市面上多见的便有 IOS, Android, Window Phone, BlackBerry, Symbian, FireFox。想要克服个平台的差异，一次性开发多个平台使用，是目前各个手机生产商和开发者要重要考虑的问题，如黑莓兼容安卓，苹果准备将 swift 引入安卓开发，微软在后面的 visual studio 大力推广的跨平台工具 xamarin 也将免费。可是为什么这些巨头想要多个平台兼容呢？就是因为差异性这么简单，差异性暴露了开发人员稀缺，多平台导致开发进度缓慢，费用极高，每次开发不同的平台就需要一个不同的团队。因为聊天软件在跨平台的需求很高，多一个平台意味着整个平台的用户，就比如 Window Phone 平台，虽然只有百分之一的用户，但在全球就有两千万人在使用。所以本文围绕着使用跨平台技术设计和实现即时通讯聊天工具。而且我相信跨平台开发就是移动开发的未来！

## 致 谢

经过了几个月的努力奋斗，论文最终在老师，同学，及所有关心过和鼓励过的朋友的支持下完成了，衷心感谢。如果没有老师和同学和朋友们的支持下，我不可能那么顺利地学校的完成最后一次任务，衷心再一次感谢。

从最初的初稿到最后的定稿，都是由老师孜孜不倦的教导下才完成的，如果没有老师那么不厌其烦的指导，我也不可能如此顺利的完成。高小辉老师是我在大学中遇到的最认真勤奋和最负责的老师之一。也是我最敬佩的老师之一，老师用实际行动言传身教，告诉我什么是真正的为人师表，严谨，向上，负责的精神，让我一生受用。

同时在写论文的时候，也有身边的很多同学和朋友的帮助，如果没有他们，我也将完成地更加艰难，从他们身上学习到的不仅仅是知识，更是思想，热血和梦想，没有他们的帮助，我也可能不能如此坚持完成论文。在今后的日子里，我也将保持这一份热血的心，进行工作，学习。

而完成这次论文，也让我学会了坚持。告诉我失败并不可怕，可怕的是不敢再一次尝试。以后遇到困难，我也将坚持坚持再坚持，直到成功不放弃。

最后再一次衷心感谢高小辉老师，感谢大家的鼓励和帮助！