

本科毕业设计说明书（论文）

**（2017届）**



论文题目 个人日常事务处理软件的设计与实现（小二，宋体，不得超过30个汉字，论文题目过长可分两行书写）

作者姓名 洪泽楠

指导教师 盛伟国

学科(专业) 软件工程1301班

所在学院 计算机科学与技术学院

提交日期 2017年6月

摘要（小二，黑体，加粗，居中，段前24，段后18,单倍行距）

随着手机在人们日常生活中占有的地位逐日增高，移动应用的开发也成了当前的热门话题，然而开发一个移动应用的成本颇高，开发一个Android端应用的同时，还需要开发一个iOS端应用。而且这两个平台的应用开发均要求开发者拥有相对复杂的开发技术。

本文提出了一种高效快速地开发跨平台移动应用的方法，即混合移动应用的开发。混合移动应用的开发允许开发者使用Web开发的技术来开发移动应用，开发出的移动应用与原生应用很像，在性能上也十分接近，而且混合移动应用开发技术正蓬勃发展，社区也在快速变得庞大，国内外已经有大量公司使用混合移动应用开发了。另外，本文还介绍了Firebase，这是一个把后端作为服务的云平台。Firebase为开发者们提供了后端的服务，将后端从我们的开发中隔离，帮我们解决开发后端过程中的框架问题，比如安全性、稳定性等，极大地降低了开发成本，缩短了产品发布时间。现在我们就能快速安全地开发移动应用了。

**关键词：混合移动应用，Ionic2，Angular2，Firebase**具体关键词，小四号宋体，段前段后0磅，1.5倍间距；关键词数量为4—6个，每一关键词之间用逗号分开，最后一个关键词不用标点符号

Abstract（小二，Times New Roman,加粗，居中，段前24，段后18,单倍行距）

Middle check system in teaching reform and bulid project is applied to realize project middle examine on-line. The use of the system will change the traditional method of project examine type, control the implementation process of project and promise the quality of project……

With the mobile phone in people’s diary life occupies a growing position, mobile application development has become a hot topic, but the development of mobile application takes a high cost. While developing an Android-side application, we need also develop an iOS application. And the development of these two platforms requires developers to have relatively complex development technology.

This paper presents an efficient and rapid development of cross-platform mobile applications, that is, the development of hybrid mobile applications. The development of hybrid mobile applications allows developers to use Web development technology to develop mobile applications, the produces developed by these technology look the same as the native applications, in terms of performance is also very close, and hybrid mobile application development technology is booming, the community is also fast become huge, there are a large number of companies at home and abroad to use a hybrid mobile application development. In addition, this article also introduced Firebase, which is a backend as a serve cloud platform. Firebase provides developers with a back-end service that isolates the backend from our development and helps us solve the development of back-end issues such as security, stability, and so on, greatly reducing development costs and reducing product release time. Now we can quickly and securely develop mobile applications.

(小四, Times New Roman, 段前段后0磅，1.5 倍行间距)

每段开头留4个字符空格，英文摘要的内容应与中文摘要基本相对应，

**Keywords：hybrid application，ionic2，angular2，firebase**全部小写，每一关键词之间逗号分开，最后一个关键词后不打标点符号；字体小四, Times New Roman, 段前段后0磅，1.5 倍行间距)

目录（以下目录内容是自动产生的，可以点击鼠标右键并选择“更新域”进行自动更新；其中关于字体大小的说明是原文带下来的，不必理会。）

摘要（小二，黑体，加粗，居中，段前24，段后18,单倍行距） I

Abstract（小二，Times New Roman,加粗，居中，段前24，段后18,单倍行距） II

第一章 绪论（１级标题，小二，黑体，加粗，段前段后18磅，1.5倍行距） 5

1.1 项目的背景与意义（２级标题，小三，宋体，加粗，段前12磅，段后6磅，1.5倍行距） 5

1.2 技术背景与发展 6

1.3 国内外技术研究现状 6

1.4 技术设计原则 7

1.4.1 实用性原则 7

1.4.2 经济性原则 7

1.4.3 技术可行性原则 7

1.5 本文的组织结构 7

第二章 相关技术介绍 8

2.1 PhoneGap and Ionic2 8

2.1.1 PhoneGap(3级标题，四号，宋体，加粗，段前６磅，1.5倍行距，建议不使用四级或更高级别目录、标题) 8

2.1.2 Ionic2(3级标题，四号，宋体，加粗，段前６磅，1.5倍行距，建议不使用四级或更高级别目录、标题) 9

2.2 Angular2 and TypeScript 10

2.2.1 Angular2 10

2.2.2 TypeScript 10

2.3 Firebase and AngularFire 11

2.3.1 Firebase 11

2.3.2 AngularFire 11

第三章 系统需求分析 12

3.1 系统需求简述 12

3.2 系统需求详细描述 12

3.2.1 日记管理系统需求目标 12

3.2.2 系统模块需求分析 13

3.2.3 系统安全和数据库需求分析 14

第四章 详细设计 14

4.1 系统开发框架结构设计 14

4.2 系统开发数据库结构设计 16

4.3 业务逻辑层详细设计 18

4.3.1登录/注册/重置密码模块详细设计 18

4.3.2增加标签/地址模块详细设计 19

4.3.3增加、修改日记模块详细设计 19

4.3.4日记显示模块的详细设计 20

4.3.5其他模块的详细设计 20

第五章 系统实现 21

5.1 系统展示和实现 21

5.1.1登录/注册/重置密码模块 21

5.1.2增加标签/地址模块 22

5.1.3增加、修改日记模块 24

5.1.4日记显示模块 25

5.1.5其他模块 28

5.2 系统测试 30

5.3 开发过程中遇到的问题 31

5.4 未实现的功能 32

第六章 总结 33

参考文献(居中，小二，黑体，加粗，段前段后18磅，1.5倍行距) 34

致谢(居中，小二，宋体，加粗，段前段后18磅，1.5倍间距) 37

附录（小二，黑体，居中，加粗，段前段后18磅，1.5倍间距） 38

附件1 毕业设计文献综述 38

附件2 毕业设计开题报告 38

附件3 毕业设计外文翻译（中文译文与外文原文） 38

图目录

图2‑1 Hibernate工作原理（图标题：五号，宋体，段前段后6磅，行间距1.5） 3

表目录

表2‑1 数据库表清单 (五号，宋体，段前段后6磅,行间距1.5) 4

# 第一章 绪论（１级标题，小二，黑体，加粗，段前段后18磅，1.5倍行距）

## 项目的背景与意义（２级标题，小三，宋体，加粗，段前12磅，段后6磅，1.5倍行距）

近年来，随着科技的迅猛发展，我们生活的节奏变得越来越快捷，时间一天一天地过去，走得太快了都没有人发觉。我们需要有一样东西来记录我们的生活，而当今手机成为了我们生活中必不可少的一部分，所以我打算实现一个个人日常事务处理软件来记录我们的生活。

个人日常事务处理软件是一款简单、高效、私密的日常记录软件。你可以用它记录生活中的某个有意义的瞬间，也可以用它记录某一刻涌现的思绪或想法。无论是短短的一两句话，还是一篇忙碌了一天的日记，你都可以记录在其中。你可以通过关键字、日期、标签甚至是记录的地点来搜索过滤出你需要的内容。它会把记录的东西保存到服务器，所以当你换一台手机登录后，你依旧可以查看到自己的日记。

另一方面，我打算设计实现个人日常事务处理软件还有另一个原因。随着生活节奏的不断加快，手机移动应用的需求不断上升，开发人员成为了稀缺的资源，开发应用的人跟不上应用的需求，或者说开发一块移动应用的速度太慢了，而且还要开发Android端应用和iOS端应用。正好巧合之间听说了混合移动应用的开发，可以用Web开发的技术来开发移动端应用，而且基本上只需要编写一份代码，就能打包出在Android平台和iOS平台分别使用的安装包。这真的是太棒了。可是我对这个技术了解得并不多，而且也听说了，用这种技术开发出来的应用比原生应用的性能要差，有些交互性高的应用会有明显卡顿现象。所以我打算尝试一下，用这种技术来开发应用，如果开发速度上比原生要快，而且性能上也还可以的话，或许我们真的可以用这种技术来开发移动应用了，那样的话真是太棒了。

## 技术背景与发展

随着移动互联网的热潮刮起，众多公司纷纷进入移动开发领域。然而，市场上移动开发人员太少了，这就导致了人才的疯狂争抢。市场机制下移动开发人员的待遇扶摇直上，最后很多企业变得负担不起一个好的跨平台移动开发的专业团队。而正好Html5的出现让网页应用有了新的曙光，Html5开发的网页应用让很多公司都怦然心动，但是网页应用相比原生应用来说，差太多了，一方面流畅性方面远远不足，而且网页应用对手机底层设备的调用也是一个问题；另一方面，用户的入口习惯也使得网页应用跟原生应用无法相比。正是这样焦灼之时，兼备了网页应用低成本快速开发，友兼备了原生应用各种特性的混合应用杀出战场，并且吸引了不少人的注意。大幅度降低开发成本和缩短发布时间，又继承了原生应用大部分特性，让混合应用在移动开发领域占得了一席之地。

从网页应用到混合应用的转换，需要有一个工具，这个工具可以把网页应用打包成安装包，最重要的是，这个工具可以允许网页应用去访问手机底层的设备接口，从而来使用这些设备，比如说拍照、定位、重力感应等等。自从混合应用开发的兴起以来，现在市场上已经有多家此类的打包工具了。最著名的有AppCan，Appcelerator，Kerkee，WeX5，APICloud，PhoneGap，NativeScript，Kinvey，ExMobi。而我在该项目中要使用的Ionic框架，就是基于PhoneGap打包转换工具的。如此之多的工具拔地而起，正说明混合应用的开发浪潮正越卷越高。

## 国内外技术研究现状

上一节提到了用来打包转换的工具有如此之多，其中WeX5就是一款由中国社区创造的开源框架。而且在如此之多的打包工具之上，各个工具百花齐放，在不同方面做出了独特的改进，比如AppCan，它就是着重解决了混合应用“不流畅”，”体验差 “的问题，做出来一个独特的应用引擎，使得开发出的应用交互性能和原生应用媲美；APICloud从”云“和”端“两个方向为开发者提供了API，简化了移动应用开发技术，让一个移动应用开发周期从一个月减为7天。混合移动应用刚出来的时候，的确技术比较疏浅，开发出来的应用在许多方面跟原生应用无法比，但是随着这方面技术的发展，在世界各地广大程序员的努力之下，混合应用正茁壮成长，变得越来越优秀。

## 技术设计原则

### 1.4.1 实用性原则

在移动开发人员稀缺的情况下，出现一种能快速开发移动应用的技术，又能降低开发成本，缩短产品发布时间，又能满足大部分原生应用功能，值得尝试一下。

### 1.4.2 经济性原则

使用混合移动开发技术来开发应用，既能降低开发成本，又能缩短产品发布时间，可以说是非常经济的行为。

### 1.4.3 技术可行性原则

从技术上来说，目前国内外已经有大量移动应用采用混合开发的模式了，而且在国内外，这方面的社区活跃度比较高，技术正以蓬勃之势发展，所以用混合应用开发日常事务处理软件在技术上是可行的。

## 1.5 本文的组织结构

本文共分为六章，以项目设计开发为主线，来讨论系统开发过程的流程，各章内容如下：

第一章，介绍了开发该项目的背景和意义，以及开发该项目所使用的技术的背景和发展，国内外的研究现状。

第二章，详细介绍了项目中所使用的技术，以及以自己的理解分析了这些技术的优劣。

第三章，介绍了该项目的需求分析，项目需要完成的功能，项目的基本情况。

第四章，详细介绍了项目开发过程中的框架设计，数据库设计，以及具体功能模块的设计。

第五章，展示了系统实现的截图以及实现的部分代码，系统测试，系统开发过程中遇到的问题和系统未完成的部分功能。

第六章，对本文、本项目所做的工作进行总结展望，对项目使用的混合移动应用开发技术的讨论，以及对它前景的看好。最后对自己在开发项目过程中所得所知进行客观的阐述。

# 第二章 相关技术介绍

## PhoneGap and Ionic2

### 2.1.1 PhoneGap(3级标题，四号，宋体，加粗，段前６磅，1.5倍行距，建议不使用四级或更高级别目录、标题)

PhoneGap是一个开源的移动设备开发框架，它的目的是让开发者能够使用Html、CSS、JavaScript等Web技术开发跨平台的移动设备应用技术。由Nitobi公司开发，在Adobe收购Nitobi公司后，将其捐献给Apache基金委员会，保留其商标并改名为Cordova。PhoneGap，现在一般我们叫它Cordova就是一个将网页应用打包成混合应用的工具。

### 2.1.2 Ionic2(3级标题，四号，宋体，加粗，段前６磅，1.5倍行距，建议不使用四级或更高级别目录、标题)

Ionic是一款基于Angular和Apache Cordova的开源混合移动应用开发框架。简单来说就是，Apache Cordova能把网页应用打包成混合应用，Ionic集成了Apache Cordova，Ionic也能将网页应用打包成混合应用。另外呢，这个网页应用不是用纯Javascript，而是用近年来比较火热的前端MV\*框架—Angular来写的，你可以享受Angular框架带来的好处。Ionic2是Ionic的第二个版本，随着Angular从1到2进阶过程中质的跨越，Ionic2也享受到了其带来的非常非常多的便利，这点我能深深体会到。在实习期间，我在公司做过Angular1.x版本的项目，用Angular1.x写的代码结构非常混乱，很难管理，而且语法很不清晰。这次毕业设计期间用Angular2写代码真的是各种清晰，结构又明朗，写大一点的项目也能够很好地管理。当然Ionic2自己从1到2也有很大的进步。首先，Ionic2自定义了丰富的组件，这些组件涉及了我们在开发过程中的基本需求。另外，Ionic2有自己的UI库，而且还提供了一套自己的ionicons，这些ionicons模仿了原生应用中常用的图标。不过有一点我要吐槽一下，这些ionicons的确给我们带来了极大的便利，在iOS端倒还好，在Android端，这些ionicons不好看，没有原生图标来得清晰。但是这不是什么大问题，因为有Html5，CSS3和JavaScript，我们能自己写一个。我没有写，因为比较懒，而且能力有限。Ionic2也给我们提供了一系列方便的指令 ，只要敲几下指令，比如ionic serve，我们就能在网页上查看编写的代码的效果了;ionic run android我们就能在Android上运行应用程序了。

## Angular2 and TypeScript

### 2.2.1 Angular2

AngularJS是一款开源JavaScript库，由Google维护，用来协作单一页面应用程序运行的。它通过MV\*模式功能来增强基于浏览器的应用，最主要的特点就是数据的双向绑定。以我自己的理解来说，一方面，从效果上来说，它通过数据的双向绑定，以及MVC的模式，使得应用程序的数据变化能够很快地在页面上渲染出来，从而带来很好的用户体验；另一方面，从开发层面来讲，AngularJS可以通过HTML层来修改DOM元素，即你可以通过代码通过HTML层面来增加自定义的DOM节点，然后在HTML页面中使用。这边这么讲是因为这几年同样火热的另一个由Facebook维护的React框架，它是通过JavaScript层面来修改DOM元素的，即自定义DOM元素，然后渲染在HTML页面中。回到Angular，比如说像我在项目中要用到一个日历，我就是用JavaScript，Html和CSS三个文件，封装出一个日历组件，然后向外提供一个接口叫ion-calendar，之后我需要用到这个日历时，只需要在对应的HTML页面相应的位置处使用<ion-calendar></ion-calendar>就行了。Angular2相对于Angular1.x，是经历了脱胎换骨的转变的，由于变化太大，也不一一详细介绍，就说说我觉得平时最常接触的几点：Angular2抛开了Angular1.x中复杂混乱的$scope作用域的概念，而使用zone.js来记录检测变化；以组件化的形式去组件应用程序，将应用程序的各个部分都写成”零件”的形式，并且有相应的接口，在使用时将” 零件”组装起来就行了；Angular2是设计包含移动端的，所以对我这次的项目来说帮助非常大；Angular2默认基于TypeScript来编写，这个好处我在下一节将提到。

### 2.2.2 TypeScript

TypeScript是一中由微软开发的自由和开源的编程语言，是JavaScript一个严格超集，并添加了可选的静态类型和基于类的面向对象编程。不知道对其他人来说怎么样，对于向我这样接触JavaScript时间不是太久的人来说，TypeScript简直是一个救星。TypeScript的语法和Java或者C#非常类似，向我这种写惯了Java的人来说就是非常方便，非常贴心。静态类型可能对我影响不大，就是在写的时候，如果你确定它是那个类型了，给它定义了该类型，然后在使用的时候编辑器会提示你它的一些API，方便使用。基于类的面向对象编程就太有用了。原来JavaScript不适合大型应用的开发，为了帮助大型应用开发时的管理，还特地加入了模块化。但是像我这种不太搞得懂JavaScript模块的人就挺麻烦的。基于类的面向对象编程就可以让我像Java那样写代码，并且对类进行引用和管理。

## Firebase and AngularFire

### 2.3.1 Firebase

Firebase是一个将后端作为服务的云平台，是一个移动端和网页应用的开发平台。被Google收购后，在2016年Google I/O大会上以全新的样貌出现，吸引了一大批企业的注意。在我的理解中，对于像我这个项目这样小的应用来说，后端的服务就是那么几点：数据安全方面的处理，提供服务性能的改善，最普遍的就是数据在数据库的存储以及增删改查了。这些的话Firebase全部帮我们做好了，你只需要调用几个API就行进行相应的操作了。而且Firebase主打的是实时数据库，对于构建实时性应用非常友好。

### 2.3.2 AngularFire

AngularFire是为了方便Angular框架的使用者，对Firebase一些部分API的封装，可以用Angular的风格来操作Firebase。这个的确很有用，可以省去你去学习Firebase接口方法的时间，而且AngularFire对TypeScript也十分友好。但是AngularFire在有些方面还存在不足，在我开发我的项目的时候AngularFire还未提供在Firebase存储、访问图片、视频等的方式，所以最后我还是使用了Firebase原生的API。

# 第三章 系统需求分析

## 3.1 系统需求简述

个人日常事务处理软件可以方便地记录生活中的点滴。软件需要有登陆、注册、忘记密码等功能。用户登录软件后，希望能够上传个人头像，能够增加日记的标签，比如业务、家庭等，用来标记不同的日记以便筛选是查看，同样的还有增加地址的功能。最重要的就是增加日记的功能，在主界面可以快速的增加一篇日记，日记中可以编辑标题、正文，还能修改此条日记的日期时间，最下面还能够选择对应的标签，应用能通过GPS自动生成一个地址，如果你不想使用自动生成的这个地址，也可以手动修改，最后还能够拍照，上传日记的照片。对日记还能修改、删除、查看等。你还可以通过标签、地址、日期来筛选出你所需要的日记。另外，还可以通过直接搜索日记的内容来筛选日记。该项目功能比较简单，但是覆盖到一款个人日常事务记录软件的基本需求。

## 3.2 系统需求详细描述

### 3.2.1 日记管理系统需求目标

日记管理软件的目标是能够增加和管理日记，软件要有基本的用户登录注册功能。为了能方便地管理日记，我们是根据日记的日期并按照月份来显示日记的。另外，我们需要增加标签和地址，当用户的日记数量达到一定数目时， 我们通过下滑主页来查找日记并不方便，所以我们在这时候就需要通过日记中的标签、地址和具体日期来筛选出日记了。当然，要是记得某一篇日记部分内容，还可以通过搜索关键字来找到对应的日记。这也是该软件比较方便的地方，另外，我们需要给用户一个良好的界面效果，让用户在使用的过程中能有一个好心情。

### 3.2.2 系统模块需求分析

登录注册模块

作为一个给用户用来记录日常事务的软件，我们需要给用户提供登录注册 功能，因为我们会把数据记录在服务器上而不是 本地，所以当用户换了手机，用户还是能通过邮箱密码来登录软件，并查看以前的日记的。我们为粗心的用户提供了重置密码的功能，所以用户在注册时最好以真实的邮箱来注册，到时候会发邮件到该邮箱，然后指导用户重置密码。

增加、修改日记模块

增加和修改日记是软件最常用的功能，为了给用户带来方便的体验，软件给用户提供了返回时自动保存日记的功能，当增加了日记，或者修改了日记之后，只要按返回键或者手机实体返回键，日记就会自动保存到服务器上。如果不想保存新日记，可以在事后删除日记；如果是不想保存修改，千万不要按返回键，只要在后台关闭该应用就好了。另外，新增日记后，不需要刷新页面，就能看到新增加的那条日记了，这也是托Firebase实时性和Angular2双向数据绑定的福。

筛选日记模块

通过关键字、日期、标签、地址筛选日记，这部分是软件比较关键的功能，直接在搜索框内输入关键字，不需要点击任何按钮，主页面上的日记就被成功筛选了。下拉日期、标签、地址栏，点击其中某一条，比如说标签中的业务，主页面上的日记就被业务这一标签筛选了，想要取消筛选，也只需要在该标签上再点一下就行。另外，日期和其他几个不一样，关闭日期栏，通过日期的筛选功能就关闭了。这是为了图开发过程中的一点便利，但应该不会影响功能。

### 3.2.3 系统安全和数据库需求分析

Firebase身份验证集成实时数据库，在开发过程中，我们控制了用户只能对自己的日记进行访问。在用户成功登录后，会返回一个auth的变量，这个变量存储了该用户的唯一标识符以及相关信息。而且数据库存储的时候，也设置了一道防线，只有用户唯一标识符和数据存储时的标识符对应时才有权限访问。关于这个在开发时还碰到过好几个问题，不过都解决了。

# 第四章 详细设计

## 4.1 系统开发框架结构设计

为了开发这个项目，我们先要确定用什么框架，然后如何将这些框架串起来，使项目能走通。在这个项目中，我选择了前后端分离，然后通过REST协议进行前端与后端之间的通信。前端我使用了Ionic2框架，后端我选择了Firebase服务，使用遵循了REST协议的Firebase API进行前后端的数据交互工作。

1. 环境的配置安装

开始Ionic2项目之前，首先我们需要下载安装Node.js，在这过程中默认会安装NPM，但是如果没有，则需要单独安装NPM包管理工具。因为在开发Ionic2项目的过程中需要安装依赖，而且会用到Ionic2的一些命令行工具，这就需要node.js和NPM的帮助了。在安装完最新的Node.js之后，我们需要安装Cordova和Ionic，因为安装了Node.js和NPM，所以接下来我们只要在终端输入以下命令就可以了：

npm install –g cordova ionic

这样，cordova和ionic就安装完成了。为了编译到Android平台，我们还需要安装JDK。最后，我们需要根据不同平台安装相应的sdk，如Android端的Android SDK和iOS端的iOS SDK。终于，我们的环境已经安装布局完成了，接下去就可以搭建架构了。

2. 创建项目

因为我们装了Ionic，我们可以直接使用Ionic的命令行来开始项目，在终端输入以下命令:

ionic start myApp tutorial --v2

这个指令会创建一个叫myApp的Ionic项目，tutorial是可选参数，表示根据tutorial这个模板去生成项目，如果不指定，默认根据tab模板生成。--v2是指生成Ionic2项目。这样我们的Ionic2项目就创建好了，这是一个基本为空的项目，内容都是模板自带的。

3. 打包成安装包

为了打包成不同平台的安装包，我们需要将平台相关的SDK信息添加到项目中。由于我们已经安装了Android和iOS相应的SDK，我们分别需要运行以下指令来添加平台信息：

ionic platform add android

ionic platform add ios

这样就添加了平台到项目了，也能够打包成相应的安装包了。运行：

ionic build android

这样就能生成Android安装包了。因为我使用的是Windows电脑，不能安装iOS的 SDK，所以我在虚拟机里面下载了一个mac操作系统。但同时遇到一个问题，使用xcode打包生成iOS安装包时，需要有iOS开发者账号才行，简单起见，我选择了在iOS模拟器上来运行软件。

4. 与Firebase的交互

第一步，我们需要去Firebase官网注册开发账号 ，通过这个账号可以进入Firebase控制台配置项目信息。第二步，在控制台中点击添加项目，输入项目的名字，点击创建就行了。第三步，在Overview一栏，选择“将Firebase添加到您的网页 应用”，会出现一段配置代码。第四步，将这段配置代码加到您的项目中就行了。我这里用的是Ionic2开发，所以将这段配置代码加在app.module.ts中，并且加上AngularFireModule.initializeApp(config);这样，前后端的架构就搭建好了，配置也做好了，接下去只要调用Firebase API就能进行前后端数据交互了。

## 4.2 系统开发数据库结构设计

用户的注册 和登录部分托管于Firebase，所以不需要设计专门的数据库结构，甚至，你还可以快速地增加用Facebook登录或者Twitter登录，我这里没有那么做，用户可以通过邮箱和密码来注册登录。Firebase采用的是NoSQL，数据存储不需要固定的表格模式。比如说在项目中需要用到的标签，地址，头像，日记等，可以通过下面的结构来进行存储：

{

“picture”: “mmexport1494508746834”,

“tags”: [

{

“0”: “enterment”

},

{

“1”: “business”

}

],

“locates”: [

{

“0”: “浙江工业大学”

},

{

“1”: “留下”

}

],

“diarys”: [

{

“0”: {

“tag”: “business”,

“location”: “浙江工业大学”,

“text\_title”: “title”,

“text\_content”: “content”,

“date”: {

“date”: “19”,

“day”: “星期五”,

“hours”: “22”,

“minutes”: “42”,

“month”: “五月”,

“year”: “2017”

}

},

“1”: {

“tag”: “enterment”,

“location”: “留下”,

“text\_title”: “second title”,

“text\_content”: “second content”,

“date”: {

“date”: “23”,

“day”: “星期二”,

“hours”: “15”,

“minutes”: “08”,

“month”: “五月”,

“year”: “2017”

}

}

}

]

}

NoSQL的数据库结构设计去掉了关系型数据库的关系型特征，数据之间没有关系，这样有利于数据的拓展；NoSQL具有灵活的数据模型，无需事先为要存储的数据建立字段，随时都可以存储自定义的数据格式；NoSQL数据库有非常高的读写性能，尤其是在大量数据的情况下，表现得非常优秀。

抛开数据库结构设计，我在存储用户数据是，会在存储路径中加上用户的uid，即在Firebase数据库中存储的形式为:

users: {

uid: {

//这里的内容是上面的数据库结构

}

}

所以，只有用户本身才能够读取到uid下面的内容，这样就确保了数据的安全性与隐私性。

## 4.3 业务逻辑层详细设计

### 4.3.1登录/注册/重置密码模块详细设计

当用户进入软件时，应该判断用户是否登录，如果登录了，则进入主界面；如果用户没有登录，则进入登录/注册界面。在该页面中，可以通过HTML5 radio的组件来替代类似a标签在当前页面中选中某块片段的功能。点击sign in，则出现登录页面；若点击sign up，则出现注册页面。这样就可以在一个页面内完成登录注册功能了。另外，在登录那块片段下面还应该有一个重置密码的按钮，点击后进入一个重置密码的页面，输入用户的邮箱，然后Firebase服务就会发送一封请求重置密码的邮件给该用户的邮箱，用户只需要根据邮件的指示操作就能重新设置密码了。

### 4.3.2增加标签/地址模块详细设计

增加标签和增加地址模块是同一个类型的部分，应该一起考虑设计。增加标签和地址，然后供用户在增加或修改日记时选用。这样的话设计就相对简单了，因为增加标签和地址是单独的两个东西，在数据库中存储时，只需要存在该用户的uid下就行了 ，然后在增加或修改日记时，读取用户路径下的标签和地址字段，就能得到所有该用户的标签和地址以供用户选择了。但是地址相对与标签，有另外一个情况，当用户增加新的日记时，软件会通过手机GPS来自动生成一个当前地址，然后作为该日记的地址。而且，这时候会进行判断，如果用户地址集合中没有改地址，则会将改地址增加到地址集合中；如果已经有了，则不做处理。这样的话也比较好设计，只要在新增的日记页面获取当前地址时，遍历地址集合，判断是否存在，若不存在则将它加进去，若存在则不处理。

### 4.3.3增加、修改日记模块详细设计

这个部分在设计时，我觉得有一点想法很好，不管时增加日记还是修改日记，它们都可以使用同一个页面模板。即增加日记时，这个模板的内容是空的，而在修改日记时，可以把选中的那条日记作为参数传到该模板页面，然后根据参数的内容，也就是被修改日记的内容，将空模板对应部分填充好就行了。这样设计可以将增加页面和修改页面做统一管理，可以很大减少工作量。确定这一点后，就可以设计该空模板的格局了。首先会有一个增加图片的功能，同拍照功能，然后将图片上传保存到服务器。这个要借助到Ionic2照相机和文件管理的插件，来调用手机设备进行拍照，或者通过文件管理设备进行照片的选择。然后是选择时间的功能，这个Ionic2提供了自己的组件，我可以使用这个组件来完成选择时间的功能，然后显示在模板页中。日记标题和日记内容可以借助Angular2的双向数据绑定和JavaScript获取DOM节点内容的方法来获取标题和内容。最后需要有一个选择标签和地址的功能，就像前面所说的，可以将该用户的所有标签和地址获取到，然后以弹窗选择的形式提供用户选择标签和地址。地址还要有一个通过手机GPS自动获取当前地址的功能。这个也得借助Ionic2的插件，来调用手机的GPS功能，然后获取当前的地址。

### 4.3.4日记显示模块的详细设计

因为日记在数据库上的存储是杂乱无章的，到时候我们把日记从数据库上拿来之后，不能直接在主界面上显示，因为那样的话日记会很乱，不方便用户查看。先要将获取到的日记按照年月份来划分开来，然后以年为单位在主页面上显示，这样的话就能方便用户查看了。但是这不是一个很简单的工作，由于日记上存储的日期是将年、月、日、日期、时间、分钟分开存储的，要在获取数据时根据年月来获取也并不是很简单，可能得在前端获取到数据之后，将数据进行加工然后再显示了。

### 4.3.5其他模块的详细设计

其它还得有上传头像模块，这部分跟日记中的选择照片部分很类似，就是在存储头像信息的时候不一样，头像是只有一张照片的，所以在每次更换了头像之后，之前那个头像记录就会被现在的头像替换掉。还有需要退出登录的功能。这个部分虽然简单，但是有一点很明显要注意，就是退出登录之后，原来那个用户的信息要从记录中全部清除掉，所有变量也要重新清除，不能被下一个用户获取到信息，也不能影响到下一个用户的正常操作，下一个用户登录时数据也都要重新获取。Ionic2中有一个rootPage的概念，page在Ionic2是以页面栈的形式存在的，rootPage在最下面，然后push一个页面到上面，再push，需要返回时就将上面那个页面pop掉。所以我想，如果要清除所有信息，或许在退出登录的时候将登录页面设为rootPage，然后在登录成功后将主页面作为rootPage，这种结构能达到想要效果。最后还要做一个筛选模块。这个模块比较清晰，就是根据标签，地址，日期来筛选，或者通过输入关键字查询日记内容。因为标签，地址，日期和日记内容在每一条日记都有单独的存储，所以筛选比较清楚，但是有一点问题，能否同时通过标签、地址、日期和关键字筛选呢，这就是一个问题了。如果能的话是在前面那个筛选结果的基础上再筛选还是重新根据两个条件来筛选最初始的那个数据。我可能会选择后者，记录上一次筛选的那个条件的值，然后再下一次的时候通过两个条件同时筛选，这样写可能比较简单吧。

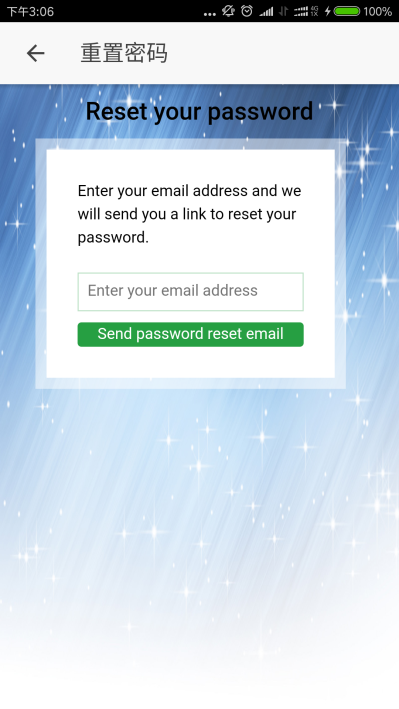
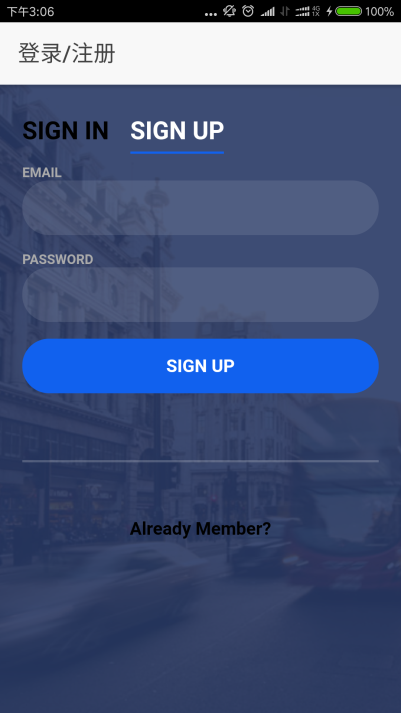
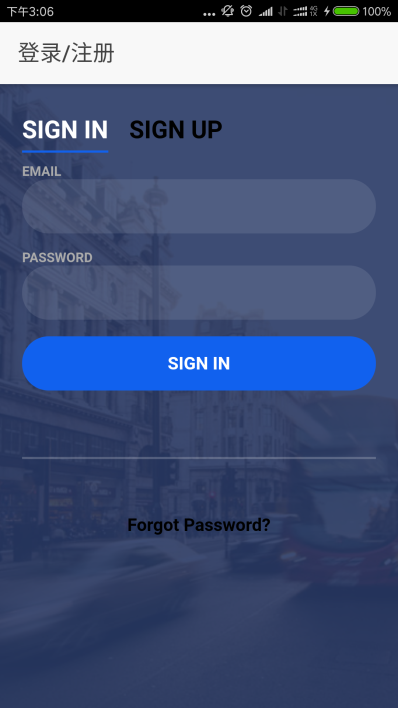
# 第五章 系统实现

## 5.1 系统展示和实现

### 5.1.1登录/注册/重置密码模块

如图为登录/注册/重置密码模块，在该模块实现中，主要利用HTML5中radio来作为控制选择，label标签来显示对应的登录或者注册界面。在该页面中，在点击登录或注册按钮时，读取邮箱和密码的值，然后调用Firebase中对应的API就可以了，我把登录注册对Firebase API的调用部分写成了一个service，调用时只要将邮箱密码作为参数传过去就行了，而且Firebase对重置密码也有相应的API。部分主要代码如下：

|  |
| --- |
| // 登录  this.angfire.auth.login({  email: email,  password: password  }, {  provider: AuthProviders.Password,  method: AuthMethods.Password  });  // 注册  this.angfire.auth.createUser({  email, password  });  // 重置密码  this.authservice.resetPassword(this.email); |



### 5.1.2增加标签/地址模块

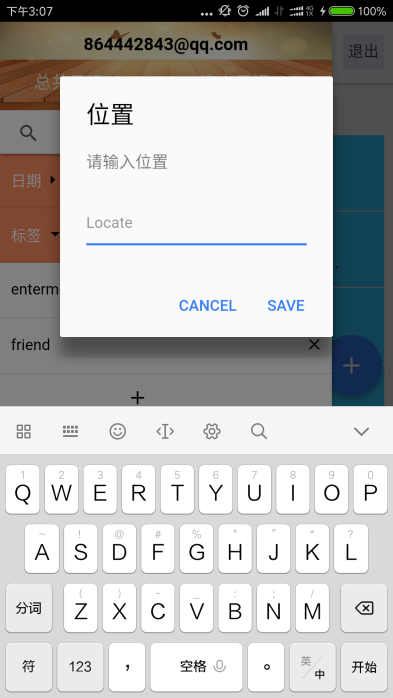
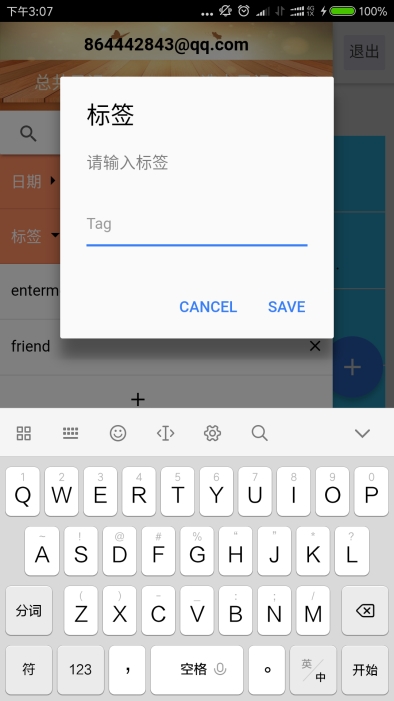
在Firebase中有一个特别厉害的东西，代码如下：

|  |
| --- |
| this.database = this.angfire.database.object('users/' + this.user\_uid);  this.database.subscribe(response => {  // 省略处理数据的逻辑  }); |

这段代码第一句的作用是获得Firebase数据库中’users/’ + 用户uid路径下所有数据的入口；第二局是监听并处理该入口中的数据，而且每次该入口中的数据发生变化时，第二行处理数据逻辑的代码都会再次执行。这就是Firebase体现实时性数据库的地方。在该项目中，需要实时获取数据的地方都用了这个方法。

标签模块中首先通过上面说的那种方法，获取标签集合数据并且显示，当增加标签时，首先将新的数据存储到数据库，然后subscribe方法监听到了数据库中标签集合值发生了改变，就会再次调用subscribe中的方法给标签集合赋值，这样新增加的标签不用刷新就能直接显示了。

|  |
| --- |
| // 获取标签集合的值  this.database\_tags = this.angfire.database.object('users/' + this.user\_uid + '/tags');  this.database\_tags.subscribe(response => {  this.tags = [];  for (let i of response) {  this.tags.push(i);  }  });  // 增加标签  this.tags.push(this.addTagItem); // this.addTagItem是新增加的标签值  const database\_tags = this.angfire.database.object('users/' + this.user\_uid + '/tags');  database\_tags.set(this.tags); |



### 5.1.3增加、修改日记模块

增加日记和修改日记的按钮不同，当增加日记时，直接进入DetailPage；而当修改或查看日记时，在进入DetailPage的时候，传递包含了这条日记的参数给DetailPage页面。DetailPage首先获取该参数的内容，如果该内容为空，则说明是新增日记，模板中除了日期等值外，均设为空；如果内容不为空，则通过这个传递过来的内容，在模板相应的位置显示数据。

|  |
| --- |
| // 增加日记  addTopic() {  this.navCtrl.push(DetailPage);  }  // 修改、查看日记  goToDetail(item) {  this.navCtrl.push(DetailPage, {  "item": item  });  } |

### 5.1.4日记显示模块

日记显示模块不是一个功能很大的部分，但是在这里我要拿出来说一下。当我从数据库拿回该用户全部日记时，日记是以最新修改来排序的，这并不是我们想要的效果，所以我们要先对这些日记进行处理。每一条日记的格式是这样的：

{

“tag”: “business”,

“location”: “浙江工业大学”,

“text\_title”: “title”,

“text\_content”: “content”,

“date”: {

“date”: “19”,

“day”: “星期五”,

“hours”: “22”,

“minutes”: “42”,

“month”: “五月”,

“year”: “2017”

}

}

第一步，我根据年和月来划分日记，用一个diarys\_help的数组来存储，格式将是这样的：

{

“2017五月”: [

{// 这里面是一条日记},

{// 这里面也是一条日记}

],

“2017四月”: [

{// 这里面是一条日记},

{// 这里面也是一条日记}

]

}

第二步，因为在html中遍历，要求对象是一个数组，所以我们要将这个diarys\_help对象转换成一个数组的形式来存储。先通过Object.keys(this.diarys\_help)遍历diarys\_help对象中的属性 ，然后根据每一个属性，生出一个新的对象diarys\_array\_item，它的结构是这样的：

{

“month”: “2017五月”,

“diarys”: [

{// 这里是一条日记},

{// 这里也是一条日记}

]

}

最后，只要将遍历diarys\_help属性时产生的diarys\_array\_item对象，都加到diarys\_array中去，这样就形成了一个根据年月划分日记的数组了，结构是这样的：

[

{

“month”: “2017五月”,

“diarys”: [

{// 这里是一条日记},

{// 这里也是一条日记}

]

},

{

“month”: “2017四月”,

“diarys”: [

{// 这里是一条日记},

{// 这里也是一条日记}

]

}

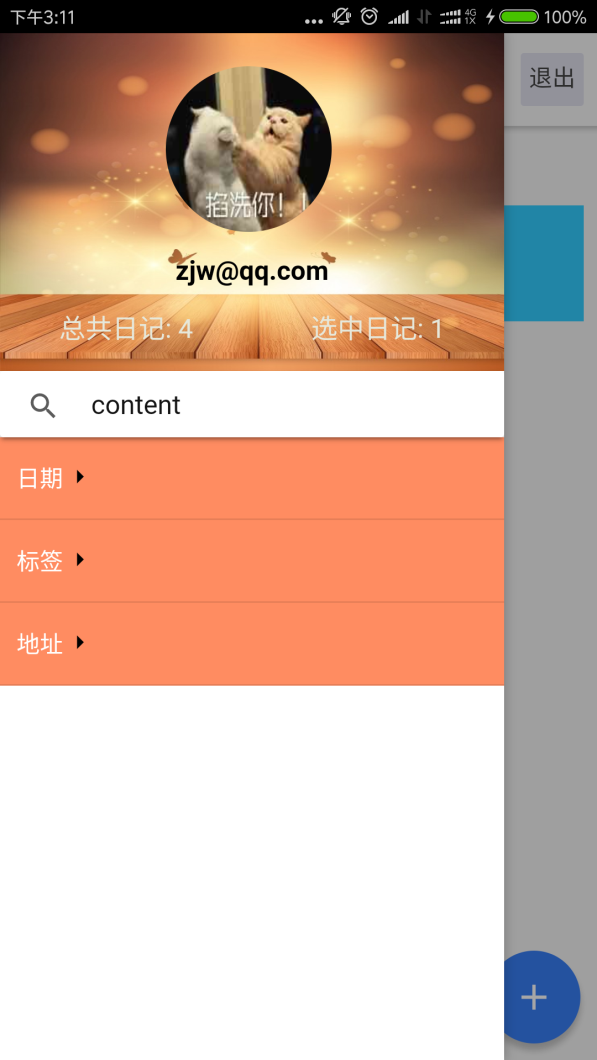
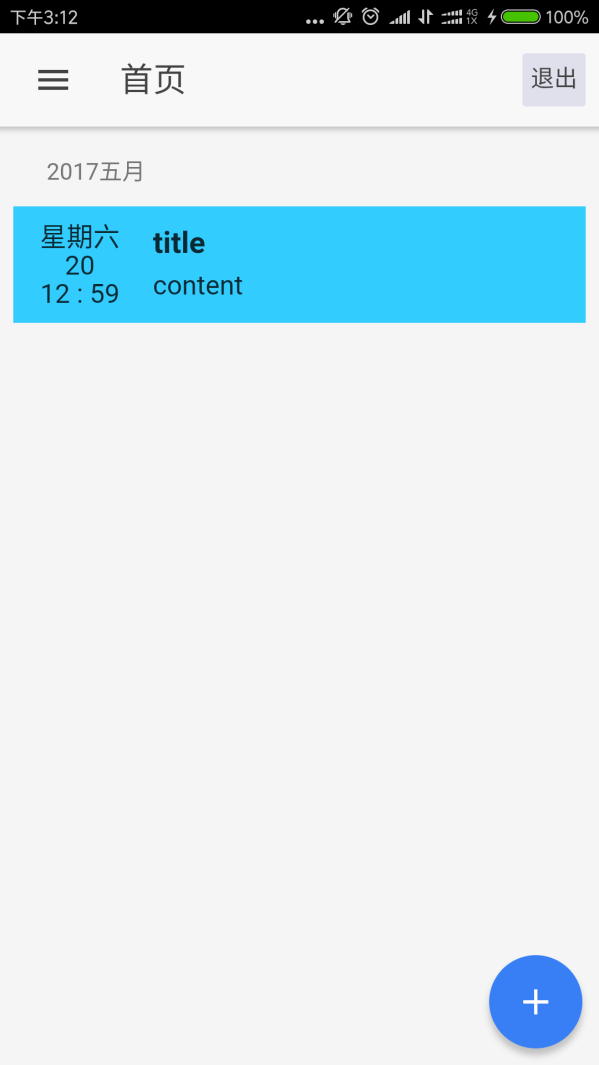
]

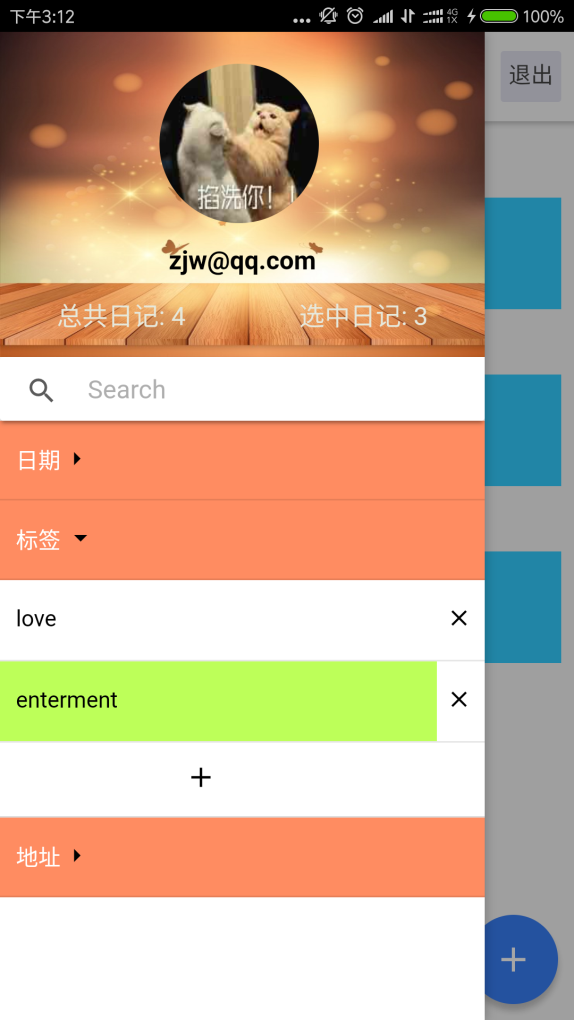
|  |
| --- |
| for (let i of this.diarys) {  if (this.diarys\_help[i.date.year + i.date.month] == undefined) {  this.diarys\_help[i.date.year + i.date.month] = [];  }  this.diarys\_help[i.date.year + i.date.month].push(i);  }  let key\_list = Object.keys(this.diarys\_help);  for (let i of key\_list) {  this.diarys\_array\_item = {  month: i,  diarys: this.diarys\_help[i]  }  this.diarys\_array.push(this.diarys\_array\_item);  } |



### 5.1.5其他模块

筛选模块只要在点击标签、地址、日期，或者搜索框内容变化时，对搜索日记进行筛选就行了。代码比较乱而杂，所以不贴出来了，主要思想就是在subscribe里面，对返回的数据库的值进行处理，如果没有筛选，则直接使用该返回值；若有筛选选项，则根据它进行筛选，留下符合要求的日记。就像前面提到过的，当多个筛选选项同时作用时怎么办，我的选择是，每次当筛选选项有变化时，也就是当你一开始选了“业务”标签，然后再选择“留下”地址时，重新对数据库返回的数据，通过这两个筛选选项进行筛选，而不是在前一个筛选结果的基础上进行“留下”地址的筛选。

## 5.2 系统测试

系统设计完成后，我根据需求分析所得到的测试用例进行测试，有以下几个case：

case 1: 登录注册测试用例

|  |
| --- |
| 准备工作: 安卓或者iOS客户端打开软件，进入登录页面。 |
| 操作: 点击sign up的tab，在邮箱密码处填上邮箱和密码，点击sign up按钮。点击sign in的tab，在邮箱密码处填上注册时的邮箱和密码，点击sign in登录按钮。 |
| 期望结果：点击sign up按钮后，提示该邮箱账号注册成功；点击sign up登录按钮后，成功进入系统主页面。 |

case 2:头像上传测试用例

|  |
| --- |
| 准备工作：用户登录成功，进入到主页面。 |
| 操作：点开toggle按钮，出现左侧边栏页面，点击头像框，选择一张照片。 |
| 期望结果：选择照片后出现头像上传成功提示，一会儿后头像框处显示刚才上传的照片。 |

case 3: 添加标签测试用例

|  |
| --- |
| 准备工作：用户成功登录，进入主页面，点开左侧边栏页面。 |
| 操作：点击标签栏，然后点击+按钮，输入一个标签名，比如“家庭”，确定。 |
| 期望结果：点击确定后，标签被成功添加，标签栏下面多了一条“家庭”标签。 |

case 4: 增加日记测试用例

|  |
| --- |
| 准备工作：用户成功登录，进入主页面。 |
| 操作：点击悬浮的+按钮，填写日记后返回。 |
| 期望结果：返回后，日记本成功添加，主页面上多出一条刚才增加的日记。 |

case 5: 日记修改测试用例

|  |
| --- |
| 准备工作：用户成功登录，进入主页面，主页面上已经有日记。 |
| 操作：点击某一篇日记，进入日记后修改标题为“title”，修改标签为“家庭”，返回。 |
| 期望结果：返回后修改的日记被保存，标题变成“title”，标签变为“家庭”。 |

以上列出了部分测试用例，根据以上以及其他一些测试用例，测试系统，系统功能正常。

## 5.3 开发过程中遇到的问题

当然，在开发的过程中也遇到了不少问题。我将这些问题记录了下来，正好可以把其中一些列出来。

问题1：在使用ionic start myApp --v2时，出现错误 Error with start undefined. Error Initializing app: There was an error with the spawned command: npminstall.

解决：后来在网上查到，这是因为网络的问题，npm的源是国外的，国内访问有时候会出现问题，等网好的时候试试就好了。另外还有一个折中的方法，我们先使用国内的镜像源cnpm，这个可以通过 npm install –g cnpm安装，然后使用命令：cnpm start myApp --v2 --skip-npm。这个的意思是使用cnpm来下载空的项目，但是跳过项目的依赖的安装。等项目创建好了之后，进入myApp目录，运行cnpm install --save，就好了。

问题2：在使用subscribe功能的时候，有时候需要取消订阅，即unsubscribe。

this.database = this.angfire.database.object('users/' + this.user\_uid);

this.database.subscribe(response => {

// 省略处理数据的逻辑

});

取消订阅的代码写成了 this.database.unsubscribe();

解决：这是一个很容易错的问题，应该是：

let temp = this.database.subscribe(response => {});

temp.unsubscribe();

问题3：一个比较偏的问题，有一次我把电脑上的Android SDK更新了一下，然后打包项目后，安装包在手机上安装后不能运行。

解决：这个问题困扰了我很久，因为第一次接触这个框架，接触得并不多，这一类的知识也比较少。后来把项目中平台相关的信息删除了，再添加一下，就好了。因为更新了Android SDK，但是ionic项目的平台相关信息是很早之前添加的，所以两者并不统一，打包时就出现问题了。这个问题不太容易发觉，不过如果大家能在打包之前，更新一下平台相关的信息，保证项目中的信息和电脑上的SDK信息一致，就不会有这个问题了。

## 5.4 未实现的功能

项目也有几个没来得及实现的功能，比如说在日历上能显示出哪几天又日记，如果某一天有日记，那么在这一天的日历上标记一个红色的点，这样在使用日历筛选的时候也就可以快速找到那些有日记的日期了。还有，没有实现选中多条日记，一次性删除。另外，没有实现日记中一下子选中多张照片并上传的功能，因为在处理异步操作的过程中出现状况。

# 第六章 总结

本项目使用了目前比较流行的ionic2框架，通过本次系统的设计开发，我发现ionic2并不像网上所说的那样，开发出的混合移动应用项目流畅度低，交互时出现卡顿现象。相反，跟室友的Android项目比较，反而更流畅。另外，使用ionic2加Firebase开发项目的学习成本很低，作为一个接触前端不久的我，也能够快速开发出一个简易的项目。而且，产品发布时间明显缩短。虽然我的产品没有上线，但是Firebase提供的快捷服务来看，上线和产品后续的检测也非常的方便。混合移动应用和原生应用十分相似，平时我们使用的QQ、微信、微博、携程等部分页面，都大量使用了混合应用开发技术。所以，类似ionic2混合移动应用开发这种成本低，开发效率又高的技术，将会是小型企业开发应用的比较好的选择。

这次个人日常事务处理软件的设计与实现过程中，遇到了许多问题，但是也收获了很多。在遇到问题时不再像以前那样害怕逃避，而是有信心地去面对它并且解决它。另外，我了解了项目开发过程中的一个整体流程，从需求分析到项目架构设计再到编写代码，从系统实现到系统测试等等，令我逐步具备了程序员的基本意识和素质。

同时通过本次毕业设计，我发现了自己的不足，清楚还有很多知识需要我去探索，但是同样让我充满信心，相信在以后我会更加努力学习，来使自己变成厉害的人。

# 参考文献(居中，小二，黑体，加粗，段前段后18磅，1.5倍行距)

参考文献应按文中引用出现的顺序列全，附于文末。学位论文中列出的参考文献必须实事求是，论文中引用的必须列出，没引用的一律删去。

根据GB 3469 规定,以单字母方式标识一下各种参考文献类型:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 参考文献  类型 | 专著 | 论文  集 | 析出  论文 | 报纸  文章 | 期刊  文章 | 学位  论文 | 报告 | 标  准 | 专  利 | 其他  文献 |
| 文献类型  标识 | M | C | A | N | J | D | R | S | P | Z |

对于数据库,计算机程序及光盘图书等电子文献类型的参考文献,以下列字母为标识:

|  |  |  |  |
| --- | --- | --- | --- |
| 参考文献类型 | 数据库 (网上) | 计算机程序 (磁盘) | 光盘图书 |
| 文献类型标识 | DB (DB/OL) | CP (CP/DK) | M/CD |

具体格式如下：

专著格式：

[序号]著者. 书名[M]. 版次(可选项). 出版地: 出版者, 出版年份.

(英文作者采用姓在前，名在后形式)

例：

* 1. 殷剑宏, 吴开亚. 图论及其算法[M]. 合肥：中国科学技术大学出版社，2004.
  2. 党跃武，谭祥金. 信息管理导论[M]. 第二版. 北京：高等教育出版社，2006.
  3. Pedrycz W., Gomide F.. An Introduction to Fuzzy Sets [M]*.* MA: MIT Press, 1998.

期刊格式：

[序号] 作者. 题名[J]. 期刊名称, 出版年份，卷号(期号)：起止页码.

(英文作者采用姓在前，名在后形式)

例：

1. 廖建新. 移动智能网技术的研发现状及未来发展[J]. 电子学报, 2003, 31(11): 1725-1731.
2. 冯永新, 刘治国, 姜月秋. 一个应用于移动自组网络管理的簇生成算法[J]. 软件学报, 2003, 14(1): 132-138.
3. Wong V. and Leung V. Location management for next generation personal communication networks [J]. IEEE Network, 2000, 14(10): 18–24.

论文集格式：

[序号] 作者. 题名[A]. 编者(可选项). 文集名[C]. 出版者地点: 出版者, 出版年份. 起止页码(可选项)

例：

1. 赵秀珍. 关于计算机学科中几个量和单位用法的建议[A]. 中国高等学校自然科学学报研究会编. 科技编辑学论文集[C]. 北京: 北京师范大学出版社, 1997. 125-129
2. Bao F., Deng R. H., Mao W. Efficient and practical fail exchange protocols with off-line TTP [A]. Proc of the 1998 IEEE Symposium on Security and Privacy [C]. Oakland: IEEE Computer Press, 1998. 77-85

专利格式：

[序号]专利申请者. 专利名称[P]. 国别:专利号, 发布日期

例：

1. 姜锡洲. 一种热温外敷药制备方案[P]. 中国专利:？881056073, 1989-07-26.

报纸格式：

[序号] 作者. 题名[N]. 报纸名, 年-月-日( 版次 )

例：

1. 国务院新闻办公室. 中国的粮食问题[N]. 人民日报, 1996-10-25(2)

网络文献格式：

[序号] 主要责任者. 电子文献题名[电子文献及载体类型标识]（任选）. 电子文献的出处或可获得地址, 发表或更新日期/引用日期（任选）

例：

1. 王明亮. 关于中国学术期刊标准化数据库系统工程进展[EB/OL]. http://www.cajcd.edc.cn/pub/wml.txt/980810-2.html, 1998-08-16/1998-10-04

标准：[序号]起草责任者(可略). 标准代号 标准顺序号 发布年, 标准名称[S]，出版地(可略)，出版者(可略)，出版年(可略)

例：

1. 全国量和单位标准化技术委员会. GB 3100～3102-93 量和单位. 北京: 中国标准出版社, 1997
2. GB/T 16159, 汉语拼音正词法基本规则[S].

# 致谢(居中，小二，宋体，加粗，段前段后18磅，1.5倍间距)

对给予各类资助、指导、和协助完成研究工作，以及提供各种条件的单位及个人表示感谢。致谢应实事求是，切忌浮夸与庸俗之词。（小四，宋体，段前段后0磅，1.5倍行间距）

# 附录（小二，黑体，居中，加粗，段前段后18磅，1.5倍间距）

### 附件1 毕业设计文献综述

### 附件2 毕业设计开题报告

### 附件3 毕业设计外文翻译（中文译文与外文原文）

主要列入正文内过分冗长的公式推导；以备查读方便所需的辅助性数学工具或表格；重复性数据图表、程序全文及说明。

论文的附录顺次为附录1，附录2…标号。附录中的图和公式另编排序号，与正文分开。