

CHENGDU COLLEGE OF
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

学 号	1640610315
作 者 姓 名	李小平
指 导 教 师	何凯霖
教 师 单 位	电子科技大学成都学院

摘要

传统视频生产与传播成本较高，不利于信息的传播。短视频则大大降低了生产传播门槛，即拍即传，随时分享。短视频实现了制作方式上最简单化，一部手机就可以完成拍摄、制作、上传分享。短视频内容简单明了。随着 5G 时代的来临，网络速度不断提升，在快节奏的生活使得用户在单个娱乐内容所占时间越来越短，短视频则更符合碎片化的浏览趋势，充分利用用户的零碎时间，让用户更直观便捷的获取信息，主动抓取更有吸引力、有创意的视频，加快信息的传播速度。在此过程中，iOS 作为移动设备 iPhone 和 iPad 的操作系统，由于具有优越的性能和独特的内存管理机制，对移动互联网的普及也起到了很大的推动作用。

在此毕业设计中，我主要开发一款基于 iOS 与短视频相关的 APP：OTunes 短视频社交 APP。本 App 后台使用 redis + ffmpeg 开发，数据库使用一个关系数据库 MySQL。其中 APP 基本功能：用户登录、注册、修改个人信息、播放视频、发布视频、拍摄视频、点赞、评论、关注、分享到第三方平台等功能。主要通过调用网络接口与服务器进行数据交互。

此项目的源代码在 ARC 环境下编译运行，利用系统新特性自动管理内存，提高开发效率；整个项目用 MVC 框架对程序进行分离解耦，对视图控制器进行优化；嵌入了友盟 SDK，实现第三方平台分享功能；登录、注册界面使用 Storyboard 进行可视化开发；

关键词： Objective-C，短视频，MVC、可视化编程

ABSTRACT

The cost of traditional video production and dissemination is relatively high, which is not conducive to the dissemination of information. Short video greatly reduces the threshold of production and communication, instant shooting, sharing at any time. Short video achieves the simplest production method, one mobile phone can complete shooting, production, upload and share. The short video content is simple and clear. With the advent of the 5G era, the speed of the network continues to increase. In a fast-paced life, users occupy less and less time in a single entertainment content, and short videos are more in line with the fragmented browsing trend, making full use of users' fragmented time to let Users can obtain information more intuitively and conveniently, actively capture more attractive and creative videos, and speed up the spread of information. In this process, iOS as the operating system of mobile devices iPhone and iPad, due to its superior performance and unique memory management mechanism, has also played a big role in promoting the popularity of mobile Internet.

In this graduation project, I mainly develop an iOS-based short video-related app: OTunes short video social APP. The App background is developed using redis + ffmpeg, and the database uses a relational database MySQL. Among them, the basic functions of APP: user login, registration, modification of personal information, playing video, posting video, shooting video, like, comment, follow, share to third-party platforms and other functions. Mainly exchange data with the server by calling the network interface.

The source code of this project is compiled and run in the ARC environment, and the new features of the system are used to automatically manage memory to improve development efficiency. The entire project uses the MVC framework to separate and decouple the program and optimize the view controller; the Youmen SDK is embedded to achieve Third-party platform sharing function;

Keywords: Short video, social, MVC

目 录

摘要.....	I
ABSTRACT.....	II
第一章 绪论.....	1
1.1 选题的目的与意义.....	1
1.2 国内外研究现状.....	1
1.2.1 国内研究现状.....	1
1.2.2 国外研究现状.....	2
1.3 本报告主要工作.....	2
1.4 复杂工程问题归纳.....	2
1.5 报告章节安排.....	3
第二章 相关技术研究.....	4
2.1 相关技术介绍.....	4
2.2 知识技能学习情况.....	4
2.3 本章小结.....	4
第三章 系统需求分析.....	5
3.1 系统特点.....	5
3.2 功能需求.....	5
3.3 非功能需求.....	8
3.3.1 性能需求.....	8
3.3.2 易用性需求.....	8
3.3.3 安全性需求.....	8
3.4 可行性研究.....	8
3.5 本章小结.....	9
第四章 系统概要设计.....	10
4.1 系统总体架构设计.....	10
第五章 系统详细设计.....	12
5.1 数据库设计.....	12
5.2 模块设计.....	14
5.2.1 系统登录模块设计.....	14
5.3.2 用户个人中心模块设计.....	15

5.3 本章小结.....	17
第六章 系统实现.....	18
6.1 登录模块实现.....	18
6.2 个人中心模块实现.....	18
6.3 本章小结.....	24
第七章 系统测试.....	25
7.1 测试环境.....	25
7.2 功能测试.....	25
7.2.1 登录模块功能测试.....	25
7.2.2 用户中心模块测试.....	28
7.3 本章小结.....	35
第八章 工程能力培养.....	36
8.1 工程计划管控和执行情况.....	36
8.2 对本工程领域发展的认识.....	37
8.3 工程协作与交流.....	38
8.4 本章小结.....	39
第九章 结束语.....	40
9.1 全文总结.....	40
9.2 不足与下一步工作.....	40
参考文献.....	41
致谢.....	42

第一章 绪论

1.1 选题的目的与意义

传统视频生产与传播成本较高,不利于信息的传播。短视频则大大降低了生产传播门槛,即拍即传,随时分享。短视频实现了制作方式上最简单化,一部手机就可以完成拍摄、制作、上传分享。短视频内容简单明了。现在快节奏的生活使得用户在单个娱乐内容所占时间越来越短,短视频则更符合碎片化的浏览趋势,充分利用用户的零碎时间,让用户更直观便捷的获取信息,主动抓取更有吸引力、有创意的视频,加快信息的传播速度。结合以上趋势,本设计主要开发一款移动端短视频 APP,实现在移动端完成视频的拍摄与发布,方便用户记录生活小细节。满足用户在闲暇的时间里打发无聊时间。将会作多种屏幕尺寸的适配,能够支持主流屏幕尺寸的 iPhone 机型,包括 4.7 英寸, 5.8 英寸, 6.1 英寸和 6.5 英寸的屏幕。

1.2 国内外研究现状

1.2.1 国内研究现状

由于短视频进入国内的时间并不长,以“短视频”为关键词在读秀上检索时发现,国内目前并没有短视频相关的学术论著,有关短视频的书籍多是跟网红、制作、流量变现、自媒体等有关,缺乏学术价值。学术论文方面,在中国知网 CSSCI (中国社会科学引文索引) 数据库以“短视频”进行主题检索,截至 2020 年 3 月 30 日,共检索到 1090 条结果,从数据结果来看,从 2014 年开始,相关文献越来越多,且呈逐年递增的趋势,到了 2019 年达到最高,这与近两年短视频应用的快速发展有着直接关系。研究领域方面:传播学领域,如赵昱、王勇泽的《短视频的传播现状分析》(2015 年),作者通过对秒拍、微视、美拍三款流行短视频应用平台进行调查,总结分析出短视频的传播特点与传播现状;短视频应用平台、短视频的发展现状及监管问题、受众分析、内容生产、媒介融合、版权问题等方面。旦大学新闻与传播学院高存玲的《移动端短视频 APP “使用与满足”研究——以快手 APP 为例》,作者运用“使用与满足”理论从受众角度解释分析短视频流行的原因,她认为,“移动端短视频应用能够激发用户的‘需求链’,并不断满足这一需求链上各层次的需求。这是其流行的根本原因。”;王晓红、包圆圆、吕强在《移动短视频的发展现状及趋势观察》(2015 年)从发展现状、问题和趋势等四个方面探讨了移动短视频应用在我国未来发展的路径。他们认为短视频发展主要存在以下问题:1.尚未建立合理的内容生产系统;2.定位不明确,内容同质化现象严重;3.盈利模式左右摇摆,缺乏成功目标,并对未来移动短视频的发展提出了两点思考:1.工具性视频功能的开发是否会成为新竞争点?2.微信加入“小视频”功能是否会改变市场格局?。

1.2.2 国外研究现状

国外最早对短视频进行研究的是 Gibbs 和 Colin,他们在文章《Short-form may be long-tail for mobile video》对短视频应用进行了介绍,并预言移动短视频应用势必成为媒体市场必争之地。(1)此外,在百度学术以“short Video”进行外文主题搜索时,一共搜到 181000 条信息,其中多集中于光学工程、信息与通信工程、临床医学、教育学方面,有关新闻与传播方面的资料非常少。仅有的一些文章也多从短视频应用出发,进行个案研究,如 Luck Chitwood 的《Social video success for brands on Vine and Instagram: your 6-to-15 seconds of fame》,作者通过对 Vine and Instagram 这两大短视频应用进行研究从而探讨短视频应用的价值。(2)研究领域方面:营销学领域,如美国学者 Humayyun Khan 的《7 creative ways brands are using Vine》,作者对移动短视频应用 Vine 的商业价值进行深入分析,认为短视频在营销方面发挥着重要作用。

1.3 本报告主要工作

本文主要介绍了 OTunes 短视频程序的设计与实现过程,包括:

1. OTunes 短视频程序的需求分析

OTunes 短视频需求分析包括功能性需求和非功能性需求两个方面,功能性需求主要包括:用户注册账号、登录、推荐视频、附近视频、视频拍摄上传、修改个人信息、消息接受处理等。非功能性需求包括性能需求

2. OTunes 短视频程序的设计

针对之前的需求分析,对本程序进行全局上的设计,描述整个程序的架构和各个模块之间的关系,之后对各个功能模块进行详细设计与实现。

1.4 复杂工程问题归纳

第一个重要问题是如何解决用户与视频发布者的关系,以及如何给用户推荐不重复的视频,用户下次还想继续观看发布者的其他视频或者该视频。如何让用户每次打开程序都能随意操作界面,如果使用非强制型登录,则需要用户需要在需要登录的操作时,弹出登录界面,这样虽然能解决问题,但在操作流畅性方面体验感是否让用户反感,这是一个亟待解决的一个问题。

第二个重要问题是如何拍摄与发布视频,很多时候用户并不一定打开我们的应用来拍摄视频,也可能采取其他设备来拍摄视频,我们应该如何导入用户的视频,视频内容

过大导致不能上传，用户的视频内容也有可能包含非法信息，如何保证视频的合法性，我们需要获取用户相册的权限来获取本地的视频。视频发布后需要后台先查看其内容，即先审核视频内容，不满足条件就删除该视频。这样前端就获取不到相关的视频信息。

1.5 报告章节安排

本文第二章主要介绍了实现课题的关键技术和方法。

第三章主要描述程序的需求分析包括系统特点、功能性需求和性能需求。

第四章主要论述如何通过需求分析对该程序进行设计，包括程序架构设计和功能模块设计。

第五章主要介绍程序的实现过程以及各个功能模块使用到的方法定义、程序流程图和核心处理环节的关键代码。

第六章主要介绍系统的测试过程，包括测试环境、功能测试和性能测试。

第七章陈述了本人在毕业设计实现过程中的工程能力与职业素养培养。

第八章为结束语，总结以上的工作，指出不足之处和对下一步工作。

第二章 相关技术研究

2.1 相关技术介绍

针对 iOS 系统, 选用了 MVC 的架构模式: 模型 model (负责封装数据、存储和处理数据运算等工作)、视图 View (负责数据展示、监听用户触摸等工作) 以及控制器 Controller (负责业务逻辑、事件响应、数据加工等工作)。在 iOS 中, m 和 v 之间禁止通信, 必须由 c 控制器层来协调 m 和 v 之间的变化。c 对 m 和 v 的访问是不受限的, 但 m 和 v 不允许直接接触控制器层。而是由多种 Callbacks 方式来通知控制器。

网络请求框架 AFNetworking 是一款开源的网络请求框架, 是对 iOS 原生 NSURLConnection 和 NSURLSession 做了封装, 使网络请求更加便捷高效, 支持返回数据序列化、文件上传、断点下载、多线程、网络状态判断等诸多优点。

布局框架: Masonry 是一个对 iOS 系统 NSLayoutConstraint 进行封装的第三方自动布局框架。采用链式编程的方式提供给开发者 API。Masonry 采取了链式编程的方式, 代码理解起来非常清晰易懂, 代码量少。

iOS 可视化编程技术: Xib 和 Storyboard, Xib 侧重于单文件 (单独的控制器或者视图) 编辑, StoryBoard 侧重于多页面关联。Story 可以直观的梳理出页面间的逻辑, 并且所有页面跳转逻辑均可在 `-(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender` 方法完成, 方便界面间数据统一管理。

2.2 知识技能学习情况

iOS 前端开发。

2.3 本章小结

本章介绍了毕业设计项目开始之前的知识学习情况, 包括网络请求框架的选用调研, QMUIKit 框架的调研与使用, 并对后台框架和数据库进行了一定的了解和学习。

第三章 系统需求分析

3.1 系统特点

OTunes 应用首先具有和一般视频应用播放视频的功能，此外它还拥有普通视频应用程序不具备的社交功能，拍摄上传作品的功能，此外，它可以对用户根据喜好进行推荐视频的功能。

3.2 功能需求

根据应用本身的系统结构，从功能的角度对系统的功能进行了初步的模块划分，主要将系统分为

1. 登录及相关模块，其包含注册、登录、退出登录三个功能点。
2. 用户中心模块，其包含查看个人信息、添加修改个人基础信息、查看自己发布的作品、查看其他用户信息、查看其他用户发布的作品功能五个功能点。

用例图如图 3-1。

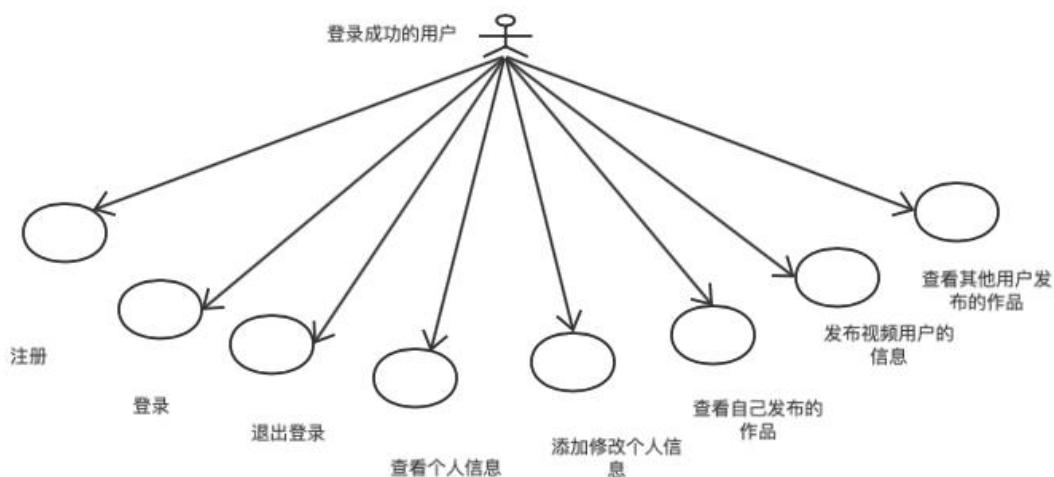


图 3-1 OTunes 个人权限用例图

根据用例图，可以进一步细化每个用例之间的业务关系，从业务层面针对本系统进行需求分析，目的是理清系统的操作逻辑和各个功能之间的耦合关系。系统业务流程图

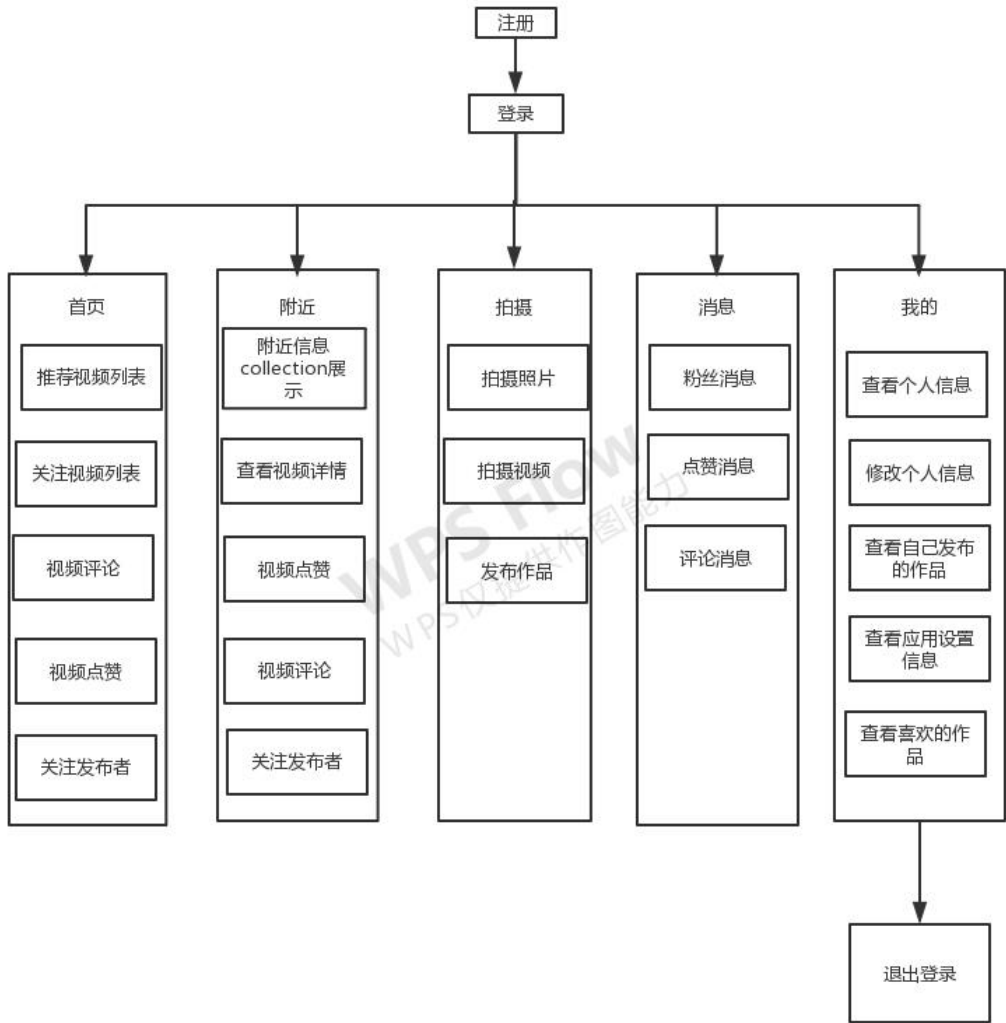


图 3-2 系统业务流程图

用户需要登录才能使用本应用，在登录界面用户可以选择注册完成操作后跳转到登录界面。用户登录后，可以查看和修改个人信息。此外，所有用户都可以拥有对视频的点赞、评论、分享的权限。拍摄、发布视频。下面将各个功能的基本业务流程作简要说明：

(1)登录

角色：用户

用例描述：输入账号密码进入应用

前置条件：用户进入登录界面

后置条件：用户登录验证通过则跳转页面，否则继续停留在登录界面用户进入登录界面。

基本事件流:

1. 用户输入账号密码
2. 前端对输入信息进行格式校验
3. 后台判断用户名和密码是否匹配
4. 返回登录结果

登录泳道图如图 3-3。

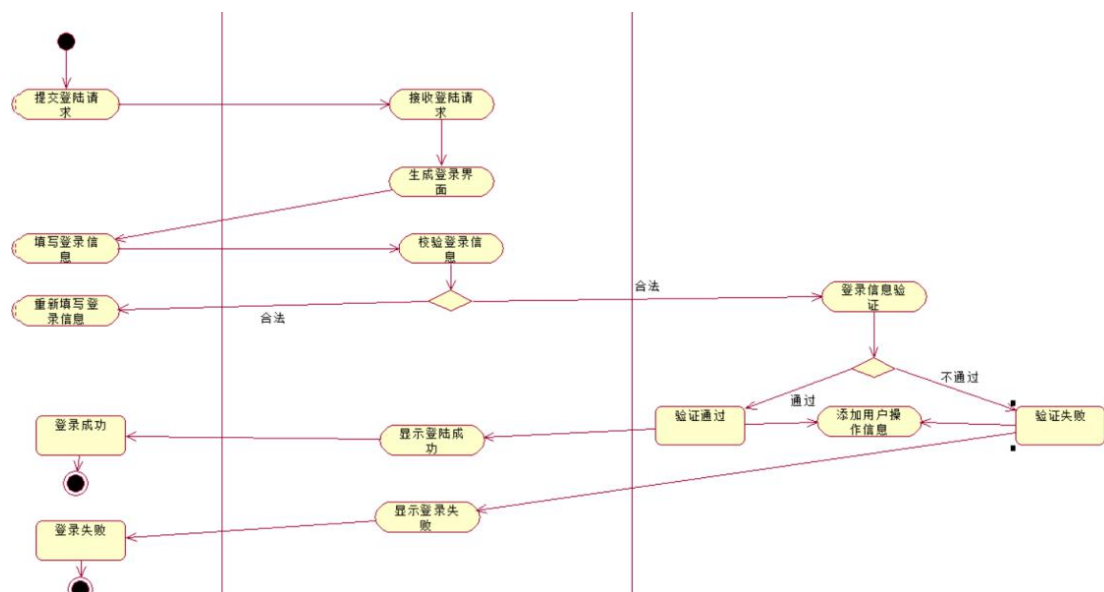


图 3-3 登录流程图

(2)注册

角色：用户。

用例描述：填写基本信息注册账号以使用本系统。

前置条件：用户进入注册界面。

后置条件：用户注册成功则生成新的账号，否则提示用户注册失败继续注册。

1. 用户输入注册所必需的信息
2. 前端对输入信息进行格式校验
3. 后台对用户数据进行存储，生成新账号
4. 返回注册结果

注册泳道图如图 3-4。

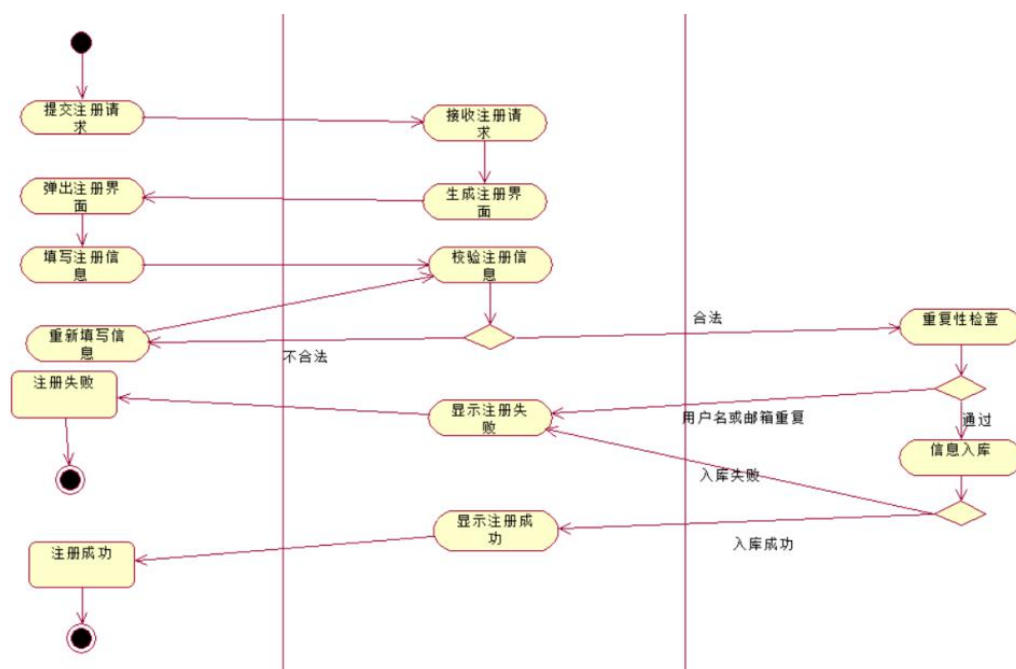


图 3-4 注册流程图

3.3 非功能需求

3.3.1 性能需求

用户对注册、登录、退出登录、查看个人信息、修改个人信息、点赞、评论、关注这几个接口的访问相应时间不超过 1 秒，发布视频、获取推荐视频、查看个人作品、喜欢作品由于包含了较多的内容查询的部分，故响应时间可以略微延长，但应不超过 1.5 秒。

3.3.2 易用性需求

用户使用应用基本无障碍，完成各项功能的流程简单并符合用户的操作逻辑。

3.3.3 安全性需求

只有在本应用注册过的用户才能访问数据、只有获得授权的用户才能对某些敏感信息进行修改。

3.4 可行性研究

本节描述该系统在社会、经济、环境、法律和技术等方面的可行性论证。

3.5 本章小结

本章主要介绍了 OTunes 短视频应用的需求分析部分，包括系统特点、功能需求、性能需求三个方面，比较详细地分析了高校专利情报系统的各个方面的内容，接下来就是根据需求分析进行系统总体设计和详细设计，这些内容将在第四章进行展开详细说明。

第四章 系统概要设计

4.1 系统总体架构设计



图 4-1 iOS 系统架构图

应用基于 MVC 设计模式，整个系统可以分为：视图层，用于显示和渲染数据；controller 层：业务模块流程的控制；view 层：业务模块的逻辑应用设计；model 层：与网络接口数据进行交互。

网络数据获取与解析主要是利用 AFNetworking 三方库，通过序列化解析得到 json 类型数据，然后将值赋给 model 的属性。

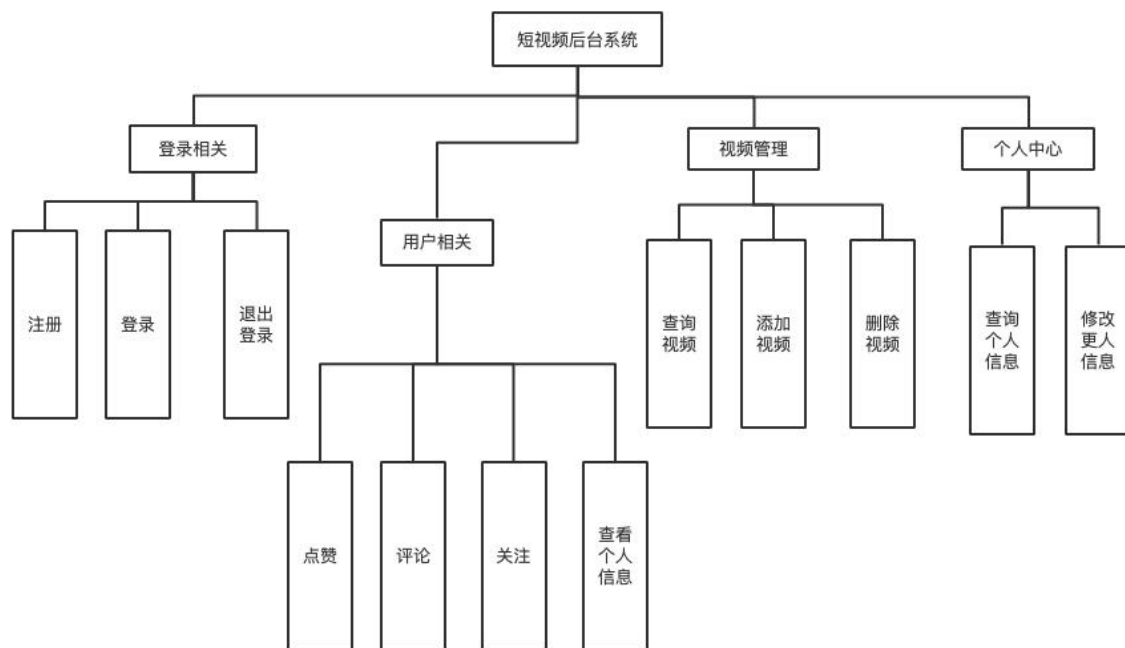


图 4-2 系统结构图

按照之前的需求分析，整个系统分为四大模块，这四大模块将整个系统进行了功能上的划分，也明确了每一个模块中的功能点分布情况。

第五章 系统详细设计

5.1 数据库设计

对于视频数据来源一块，我的设计是先使用 Charles 抓取了其他应用的部分接口，并保存到了本地，然后利用 JSON 格式化工具处理了数据。然后创建了一个 model。由于抓取的数据是数组，我对每一组数据初始化了一个模型，然后加入到了一个可变数组中。Controller 创建时就会请求该数据，通过属性传递给 Controller，利用 tableViewcell 的复用到达预加载的效果以及 tableView 的 pagingEnabled=YES 的方式达到翻页的效果。通过监听 scrollView 的 contentOffset.y / cell 的高度判断是否应该播放下一个视频。网络视频数据 model 如图 5-1 所示，模型赋值 如图 5-2，cell 赋值 如图 5-3 网络视频数据 model 如表 5-1:

```

1  //
2  // KB_HomeVideoDetailModel.h
3  // Project
4  //
5  // Created by hi kobe on 2020/4/15.
6  // Copyright © 2020 hiKobe@lsirCode. All rights reserved.
7  //
8
9  #import <Foundation/Foundation.h>
10
11  NS_ASSUME_NONNULL_BEGIN
12
13  @interface KB_HomeVideoDetailModel : NSObject
14  @property (nonatomic, strong) NSString *id;
15  @property (nonatomic, strong) NSString *userId;
16  @property (nonatomic, strong) NSString *audioId;
17  @property (nonatomic, strong) NSString *videoDesc;
18  @property (nonatomic, strong) NSString *videoPath;
19  @property (nonatomic, strong) NSString *videoCategory;
20  @property (nonatomic, assign) float videoSeconds;
21  @property (nonatomic, assign) CGFloat videoWidth;
22  @property (nonatomic, assign) CGFloat videoHeight;
23  @property (nonatomic, strong) NSString *coverPath;
24  @property (nonatomic, assign) NSInteger likeCounts;
25  @property (nonatomic, assign) NSInteger status;
26  @property (nonatomic, assign) long long createTime;
27  @property (nonatomic, strong) NSString *nickName;
28  @property (nonatomic, strong) NSString *face_image;
29  @property (nonatomic, strong) NSString *videoFilter;
30  @property (nonatomic, assign) BOOL isFocus;
31  @end
32  |
33  NS_ASSUME_NONNULL_END
34

```

图 5-1 网络视频数据 model

```

188 - (void)getDataList{
189     if (self.page == 1) {
190         [self showEmptyViewWithLoading];
191     }
192     NSString *url = [NSString
193         stringWithFormat:@"video/showAll?page=%d&isSaveRecord=0&category=food",@(self.page)];
194     [RequestApi requestAPIWithParams:nil andRequestUrl:url completedBlock:^(ApiResponseModel
195         *apiResponseModel, BOOL isSuccess) {
196         if (isSuccess) {
197             [self hideEmptyView];
198             [self.collectionView.mj_header endRefreshing];
199             NSMutableArray *datas = [NSArray arrayWithClass:[KB_HomeVideoDetailModel class]
200                 json:apiResponseModel.data[@"rows"]].mutableCopy;
201             if (self.page == 1) {
202                 [self.dataArray removeAllObjects];
203                 self.dataArray = datas;
204             }else{
205                 [self.dataArray addObjectsFromArray:datas];
206             }
207             if (datas.count == 5) {
208                 //有下一页
209                 self.collectionView.mj_footer.hidden = NO;
210                 [self.collectionView.mj_footer endRefreshing];
211                 self.page++;
212             }else{
213                 [self endRefreshWithFooterHidden];
214             }
215             if (self.dataArray.count == 0) {
216                 [self showNoDataEmptyViewWithText:@"暂无附近数据" detailText:@"请前往首页观看更多视频"];
217             }else{
218                 [self.collectionView reloadData];
219             }
220         } else {
221             [self hideEmptyView];
222             self.collectionView.mj_footer.hidden = YES;
223             [self.collectionView.mj_header endRefreshing];
224             [self showEmptyViewWithImage:UIImageMake(@"404") text:@"加载失败"
225                 buttonTitle:@"点击重试" buttonAction:@selector(getDataList)];
226         }
227     }
228 }
229 }
230 }
231 }
232 }
233 }
234 }

```

图 5-2 模型赋值

```

#pragma mark - UITableViewDataSource & UITableViewDelegate
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return 1;
}
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:(NSInteger)section {
    return self.modelArray.count;
}
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    KB_NearVideoPlayCell *cell = [tableView dequeueReusableCellWithIdentifier:NearVideoCellIdentifier
        forIndexPath:indexPath];
    cell.delegate = self;
    cell.videoModel = self.modelArray[indexPath.row];
    return cell;
}
- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexPath {
    return SCREEN_HEIGHT - TabBarHeight;
}

```

图 5-3 cell 赋值

表 5-1 video Model 设计

名	类型
rid	UInt64
name	NSString
Comment_num	NSInteger
score	NSInteger
uid	NSString
artist	NSString
Header_url	NSString
Video_url	NSString
Cover_url	NSString
aspect	CGFloat

。

5.2 模块设计

5.2.1 应用登录模块设计

此模块功能覆盖用户登录本系统及其相关内容，主要包括：用户注册、登录、退出登录三个功能点。系统登录子模块组成图如图 5-4:



图 5-4 登录相关子模块组成图

注册：注册需要用户输入用户名、密码两条信息，其中用户名是不可重复的。用户完成注册后可能返回提示四种提示信息：注册成功、注册失败（网络错误）、用户名重复，注册失败。

登录：用户输入用户名和密码进入应用，可能返回的字段有：登录成功、密码错误、用户不存在。登录成功后台返回 token,并保存在本地缓存中。用于请求接口的 header 头。

退出登录：用户点击退出登录按钮，先调用网络接口：可能返回的字段有：成功、失败。成功后则清除本地缓存的 token，弹出登录即界面；失败则不做处理。登录相关网络操作如图 5-5

```
#pragma mark - 登录按钮消息通道
- (RACSignal *)signInSignal
{
    @weakify(self)
    return [RACSignal createSignal:^(RACDisposable *(id<RACSubscriber> subscriber) {
        @strongify(self)
        NSMutableDictionary *params = [NSMutableDictionary dictionary];
        params[@"username"] = self.userNameTextField.text;
        params[@"password"] = self.passwordTextField.text;
        [SVProgressHUD show];
        [UserCenter clearUserCenter];
        [RequestApi requestAPIWithParams:params andRequestUrl:@"login"
         completedBlock:^(ApiResponseModel *apiResponseModel, BOOL isSuccess) {
            if (isSuccess) {
                [SVProgressHUD dismiss];
                // 保存用户名密码
                User_Center.username = params[@"username"];
                if (self.rememberPassword_btn.isSelected) {
                    User_Center.pass = params[@"password"];
                }

                // 用户中心
                [UserCenter resetUserCenterWithDictionary:apiResponseModel.data];
                User_Center.userToken = apiResponseModel.data[@"userToken"];
                [UserCenter save];
                [subscriber sendNext:@(YES)];
                [subscriber sendCompleted];
            } else {
                [SVProgressHUD showErrorWithStatus:apiResponseModel.msg];
                [subscriber sendNext:@(NO)];
                [subscriber sendCompleted];
            }
        }];
        return nil;
    }]);
}
```

图 5-5 登录网络请求

5.3.2 用户个人中心模块设计

用户在登录本应用后，可以进入个人中心进行与个人相关信息查看、修改。所有用户均为普通用户，只能修改自己的个人信息和查看其他用户相关信息。此模块主要包含：查看个人信息，，增、改、查个人信息。模块功能如图 5-6 所示。

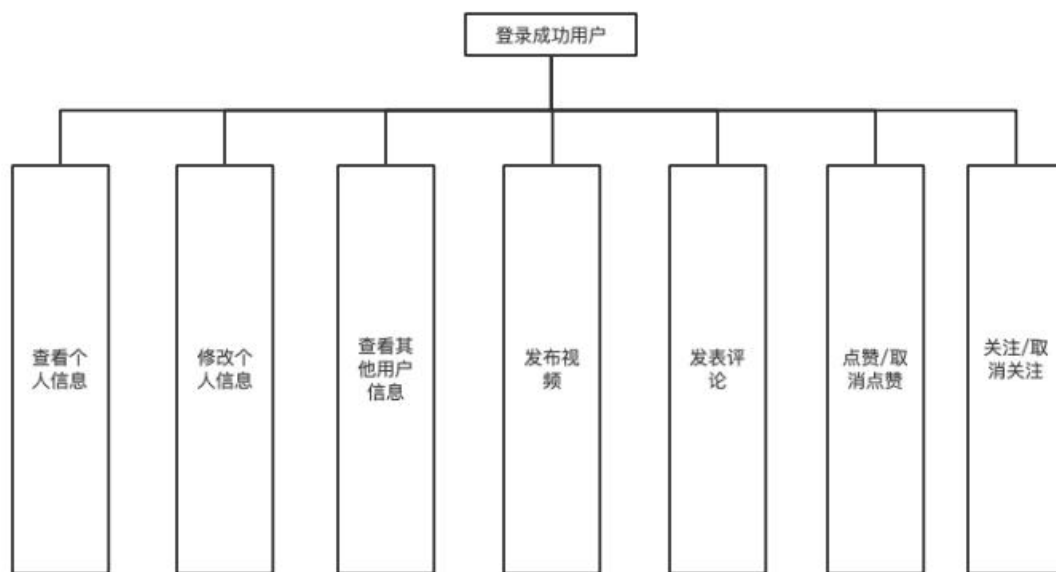


图 5-6 用户个人中心子模块组成图

查看个人信息：用户点击我的查看个人信息。返回结果：用户个人基本信息，信息包括用户名、性别、出生日期、昵称、简介、头像、已发布作品、地区。通过调用的接口获取相关网络数据。根据返回数据展示。

修改个人信息：包括更换头像、更换昵称、修改简介、性别、生日、地区等信息。每一条信息的修改都会调用接口。并保存至服务器。

查看其他用户信息：包括头像、昵称、简介、性别，年龄、地区、作品等信息。主要通过该视频用户的 Id 获取网络数据。

发布视频：用户可以发布自己拍摄的视频，或者本地相册视频，也可以添加视频简介。发布视频会上传至服务器。

其他操作：对视频发表自己的评论，点赞/取消点赞，关注用户/取消关注。主要与服务器交互，每次操作都会调用接口。

5.3 本章小结

本章根据之前的需求分析，先确定了应用的整体架构和系统结构，之后划分了系统模块，并对每一个模块进行了简要说明，并针对每个模块中涵盖的功能点进行了详细设计，并对每个功能点都给出了相应的泳道图，明确了每个功能点的实现流程，接下来的第五章，我将基于其设计对每个功能点、每个模块的实现过程进行解释说明。

第六章 系统实现

6.1 登录模块实现

此模块含有注册、登录、退出登录三个功能点，用户必须先注册才能正常进入应用。由于这些功能是绝大多数应用都拥有的，在此省略实现过程。

6.2 个人中心模块实现

用户个人中心模块主要包括观看推荐、关注、附近视频，对喜欢的视频点赞，对想要长期关注视频发布者进行关注，视频分享功能。本模块主要讲述是视频播放相关的实现。

关注用户主要通过用户登录时 token 作为网络请求 head 视频用户 userid 作为参数，请求后台接口，点赞则是请求参数为视频的 id。

视频播放流程：

1. 视频数据都是通过接口获取，然后通过属性传递给 cell。
2. cell 拿到后台返回的视频网址 URL，将视频展现在 imageView 上，由于视频是使用了缓存机制，所以当视频播放完后，会循环播放，但不会请求网络数据。
3. 弱网环境下，则会有等待的效果，也可以点击屏幕来暂停视频播放。
4. 播放下一首则是通过监听 scrollView 滚动的 contentOffset.y 值，获取将要播放的信息。

播放视频方法名：playIndex:

播放下一个方法名：preLoadIndex:

输入参数：currentIndex+1(下一个)、currentIndex - 1 (上一个)

程序流程图如图 6-1。

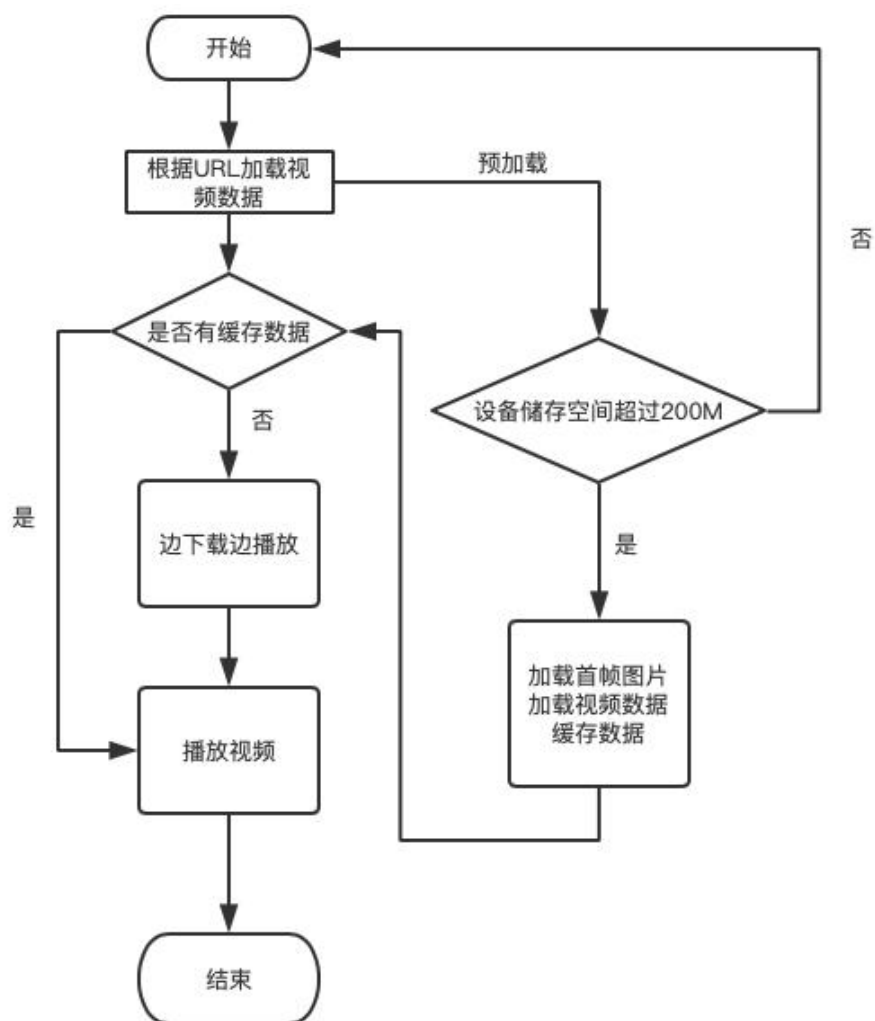


图 6-1 播放视频流程图

核心代码如代码 6-1

```
- (void)playIndex:(NSInteger)currentIndex {  
    DLog(@"播放下一个");  
    KB_NearVideoPlayCell *currentCell = [self.tableView  
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:currentIndex  
inSection:0]];
```



```
NSString *artist = nil;

NSString *title = nil;

NSString *cover_url = nil;

NSURL *videoURL = nil;

NSURL *originVideoURL = nil;

BOOL useDownAndPlay = NO;

AVLayerVideoGravity videoGravity = AVLayerVideoGravityResizeAspect;


KB_HomeVideoDetailModel *currentPlaySmallVideoModel =

self.modelArray[currentIndex];


artist = currentPlaySmallVideoModel.nickName;

title = currentPlaySmallVideoModel.videoDesc;

// 首帧图

cover_url = [NSString

stringWithFormat:@"https://www.lotcloudy.com/scetc-show-videos-mini-api

-0.0.1-SNAPSHOT%@",currentPlaySmallVideoModel.coverPath];

// 视频地址

videoURL = [NSURL URLWithString:[NSString

stringWithFormat:@"%s%@",kAddressUrl,currentPlaySmallVideoModel.videoP
```

```
ath]];

    originVideoURL = [NSURL URLWithString:[NSString
stringWithFormat:@"%@@%@",kAddressUrl,currentPlaySmallVideoModel.videoP
ath]];

    useDownAndPlay = YES;

    if((currentPlaySmallVideoModel.videoHeight /
currentPlaySmallVideoModel.videoWidth) >= 1.4) {

        videoGravity = AVLayerVideoGravityResizeAspectFill;

    } else {

        videoGravity = AVLayerVideoGravityResizeAspect;

    }

    self.fatherView = currentCell.playerFatherView;

    self.videoPlayerManager.playerModel.videoGravity = videoGravity;

    self.videoPlayerManager.playerModel.fatherView      = self.fatherView;

    self.videoPlayerManager.playerModel.title           = title;

    self.videoPlayerManager.playerModel.artist = artist;

    self.videoPlayerManager.playerModel.placeholderImageURLString =
cover_url;

    self.videoPlayerManager.playerModel.videoURL        = videoURL;

    self.videoPlayerManager.originVideoURL = originVideoURL;
```

```
self.videoPlayerManager.playerModel.useDownAndPlay = YES;

//如果设备存储空间不足 200M,那么不要边下边播

if([self deviceFreeMemorySize] < 200) {

    self.videoPlayerManager.playerModel.useDownAndPlay = NO;

}

[self.videoPlayerManager resetToPlayNewVideo];

}
```

代码 6-1 下一个

收藏夹移除专利流程:

- 1.用户在某个收藏夹下选定专利点击确定移除
- 2.前端获取用户名、收藏夹名、要移除专利的专利号数据并传给后端
- 3.后台通过用户名和收藏夹名找到该收藏夹
- 4.从该收藏夹下移除专利,
- 5.添加操作日志, 返回操作结果。

从收藏夹中移除专利方法名: `deleteFromFavorite`

输入参数: `userName`(用户名), `favoName`(收藏夹名称), `patentNumber`(专利号)

输出参数: “delete successfully”(成功删除)、“error”(删除错误)

程序流程图如图 6-2。

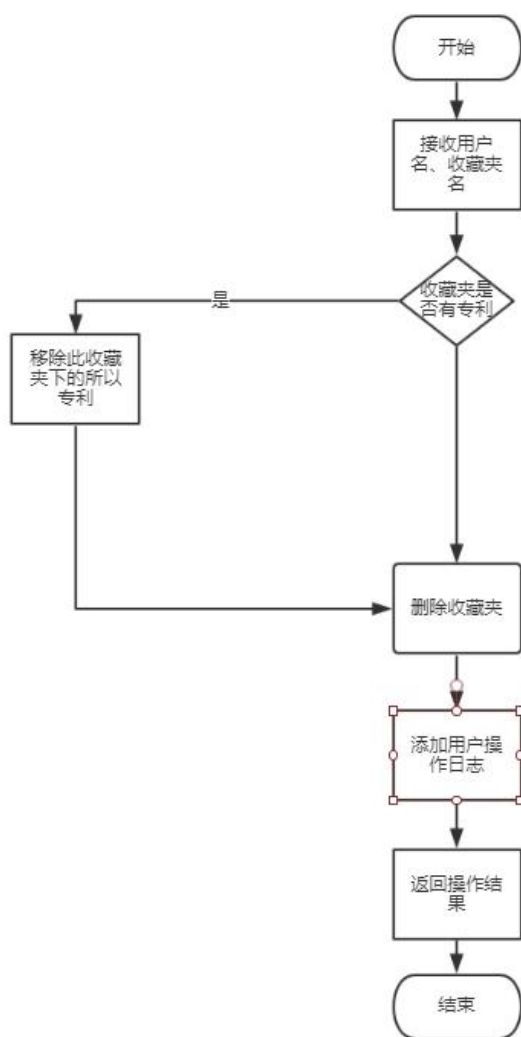


图 6-2 从收藏夹中移除专利流程图

核心代码如代码 6-2

代码 6-2 从收藏夹中移除专利

```
Date time = new Date();

int i = favoriteDao.deleteFromFavorite(favoName, userName, patentNumber);

if (i != 0) {

    userDao.insertUserAction(userName, "deleteFromFavoriteFolder", "success", time);

    return "delete successfully";

}

else {

    userDao.insertUserAction(userName, "deleteFromFavoriteFolder", "fail", time);
```

```
return "error";
```

6.3 本章小结

本章讲述 OTunes 短视频的实现过程，分别从登录、首页、附近、拍摄、消息、我的五大模块并参照第四章的设计部分，对每个功能点的实现过程进行了比较详细的说明。

第七章 系统测试

7.1 测试环境

硬件: iPhone 手机

操作系统: windows、MacOS

系统部署情况: 后台管理器部署在 Tomcat 远程服务器上, 前段部署在 Xcode

调试环境: Xcode

测试平台: 前端在 iPhone 手机

7.2 功能测试

7.2.1 登录模块功能测试

测试短视频播放器登录模块中注册账号、登录密码、退出登录的流程是否存在问题, 同时设计了一些不属于基本操作流程的测试用例, 测试系统是否会反馈错误信息。

表 7-1 登录模块功能测试

测试目的	测试用户从注册、登录、退出登录整个流程是否可以通过				
编写人	李小平	时间	2020-4-15	状态	检测结果为合格
前置条件	用户注册、登录界面				
序号	测试步骤			期待结果	是否通过
1	用户点击注册, 输入账号、密码进行注册			操作成功, 用户注册成功	通过, 和预期结果一致
2	用户第二次输入相同账号密码进行注册			操作失败, 用户已经存在	通过, 和预期结果一致
3	用户再次输入相同账号不同密码进行注册			操作失败, 用户已经存在	通过, 和预期结果一致
4	用户输入不同账号相同密码进行注册			操作成功, 用户注册成功	通过, 和预期结果一致
5	用户输入不同账号不同密码进行注册			操作成功, 用户注册成功	通过, 和预期结果一致
6	用户输入注册成功的账号密码进行登录			操作成功, 用户成功登录	通过, 和预期结果一致

7	用户输入未注册的账号密码进行登录	操作失败，用户未注册	通过，和预期结果一致
8	用户输入注册成功的账号，错误的密码进行登录	操作失败，密码错误	通过，和预期结果一致
9	用户输入注册成功的账号密码进行登录，但未区分大小写	操作失败，未成功登录	通过，和预期结果一致
10	用户输入密码后勾选“记住密码”，退出登录界面后，再进入登录界面查看密码是否存在	密码依然存在	通过，和预期结果一致
11	用户在勾选“记住密码”，退出登录界面后，再次进入登录界面，点击登录，是否能成功登录	操作成功，能成功登录	通过，和预期结果一致
12	用户取消勾选“记住密码”，退出登录界面后，再进入登录界面查看密码是否存在	密码不存在	通过，和预期结果一致
13	用户成功登录后退出当前登录	操作成功，退出登录	通过，和预期结果一致
14	用户退出登录后，立即使用该账号再次进行登录	操作成功，用户成功登录	通过，和预期结果一致
15	用户退出登录后，立即更换账号进行登录	操作成功，用户成功登录	通过，和预期结果一致



图 7-2 引导页界面



图 7-3 登录界面



图 7-4 注册界面

7.2.2 用户中心模块测试

测试用户使用所有功能是否正常，系统能否根据用户操作正常进行

表 7-2 用户操作中心模块功能测试

测试目的	测试用户从登录后能否正常使用用户中心的各项功能				
编写人	李小平	时间	2020-4-16	状态	检测结果为合格
前置条件	用户已成功登录				
序号	测试步骤			预期结果	实际结果
1	用户首页短视频进行播放			操作成功，短视频成功播放	通过，和预期结果一致
2	用户点击暂停短视频播放			操作成功，短视频暂停播放	通过，和预期结果一致
3	用户滑动进度条快进短视频			操作成功，短视频快进成功	通过，和预期结果一致
4	用户滑动进度条后退短视频			操作成功，短视频后退成功	通过，和预期结果一致
5	用户滑动进度条快进/后退短视频，是否会出现卡顿现象（排除网络因素）			未出现卡顿现象	通过，和预期结果一致
6	用户增大短视频播放音量			操作成功，短视频音量增大	通过，和预期结果一致
7	用户减小短视频播放音量			操作成功，短视频音量减小	通过，和预期结果一致
8	用户关闭音量，短视频是否还有声音			短视频没有声音	通过，和预期结果一致
9	用户将音量调至最大，是否出现播放声音不清晰、声音卡顿			未出现声音不清晰、声音卡顿现象	通过，和预期结果一致
10	用户切换到下一个短视频			操作成功，成功切换	通过，和预期结果一致
11	用户对首页短视频进行点赞			操作成功，能够成功点赞	通过，和预期结果一致
12	用户对首页已点赞的短视频取消点赞			操作成功，能够成功取消点赞	通过，和预期结果一致
13	用户对首页短视频进行关注			操作成功，关	通过，和预

		注成功	期结果一致
14	用户对首页短视频取消关注	操作成功，取消关注成功	通过，和预期结果一致
15	用户对首页短视频进行评论	操作成功，能够成功评论	通过，和预期结果一致
16	用户删除自己在短视频的评论	操作成功，能够成功删除评论	失败，不能删除评论，和预期结果不符
17	用户查看他人的评论	操作成功，能够查看他人的评论	通过，和预期结果一致
18	用户删除他人的评论	操作失败，不能删除他人的评论	通过，和预期结果一致
19	用户分享首页短视频到微信	操作成功，成功分享到微信	通过，和预期结果一致
20	用户在“附近”界面查看附近的短视频列表	能成功查看到附近的短视频	通过，和预期结果一致
21	用户在“附近”界面点开短视频进行播放/暂停播放	操作成功，短视频成功播放/暂停播放	通过，和预期结果一致
22	用户对“附近”界面的短视频进行点赞/取消点赞	操作成功，能够成功点赞/取消点赞	通过，和预期结果一致
23	用户对“附近”界面的短视频进行关注/取消关注	操作成功，能够成功关注/取消关注	通过，和预期结果一致
22	用户对“附近”界面的短视频进行评论	操作成功，能够成功评论	通过，和预期结果一致
23	用户对“附近”界面的短视频删除自己的评论	操作成功，能够成功删除评论	失败，不能删除评论，和预期结果不符
24	用户对“附近”界面的短视频进行分享到微信	操作成功，成功分享到微信	通过，和预期结果一致
25	用户上传本地保存的短视频	操作成功，短视频上传成功	通过，和预期结果一致
26	用户使用前置摄像头拍摄短视频后进行上传	操作成功，短视频上传成功	通过，和预期结果一致

27	用户使用后置摄像头拍摄短视频后进行上传	操作成功，短视频上传成功	通过，和预期结果一致
28	用户使用后置摄像头并开启闪光灯拍摄短视频后进行上传	操作成功，短视频上传成功	通过，和预期结果一致
29	用户拍摄短视频，查看计时对应时间是否正确	计时对应时间正确	通过，和预期结果一致
30	用户在“消息”界面点击粉丝查看他人给自己点的关注	能够查看他人给自己点的关注	通过，和预期结果一致
31	用户在“消息”界面点击喜欢查看他人给自己的点赞	能够查看他人给自己的点赞	通过，和预期结果一致
32	用户在“我的”界面能否查看自己上传的短视频	能够查看自己的上传的短视频	通过，和预期结果一致
33	用户在“我的”界面能否查看自己视频的获赞数、关注数、粉丝数	能够查看获赞数、关注数、粉丝数	通过，和预期结果一致
34	用户在“我的”界面点击“编辑资料”查看个人资料	能够查看个人资料	通过，和预期结果一致
35	用户在“编辑资料”界面，点击头像，选择拍照进行更换	操作成功，头像更换成功	通过，和预期结果一致
36	用户在“编辑资料”界面，点击头像，选择图片进行更换	操作成功，头像更换成功	通过，和预期结果一致
37	用户在“编辑资料”界面，点击账号对昵称进行修改	操作成功，昵称修改成功	通过，和预期结果一致
38	用户在“编辑资料”界面，点击简介进行编辑	操作成功，成功编辑简介	通过，和预期结果一致
39	用户在“编辑资料”界面，点击学校进行添加	操作成功，学校添加成功	通过，和预期结果一致
40	用户在“编辑资料”界面，点击性别进行修改	操作成功，性别修改成功	通过，和预期结果一致
41	用户在“编辑资料”界面，点击生日进行选择	只能选择以前日期，不能选择未来日期	失败，能选择未来日期，和预期结果不符
42	用户在“编辑资料”界面，点击地区进行选择	操作成功，地区选择成功	通过，和预期结果一致

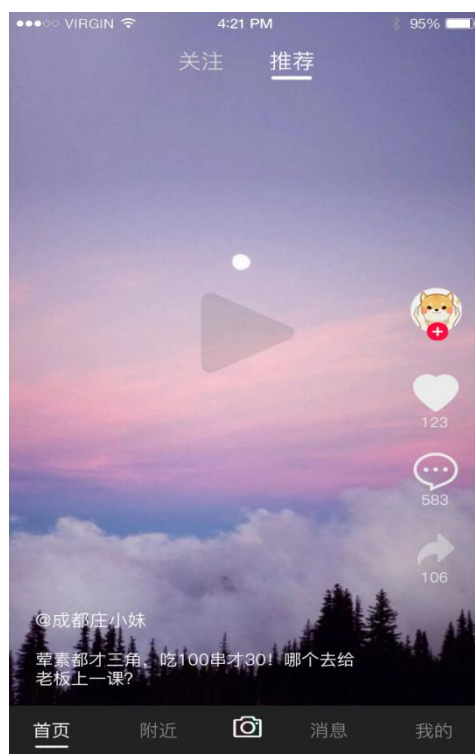


图 7-5 首页界面

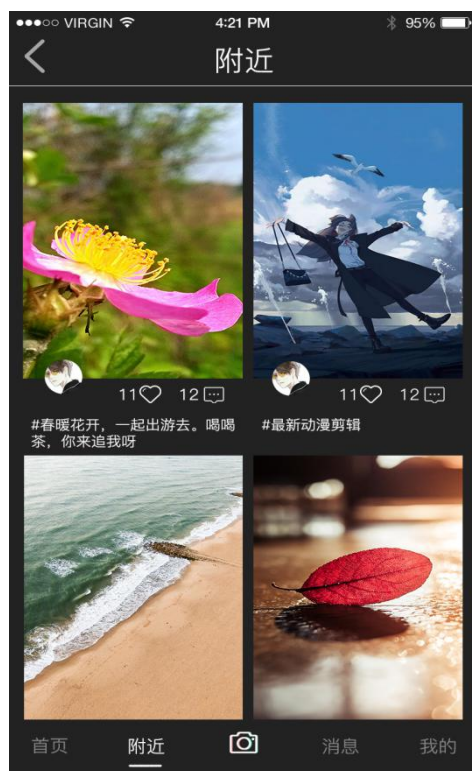


图 7-6 附近短视频界面

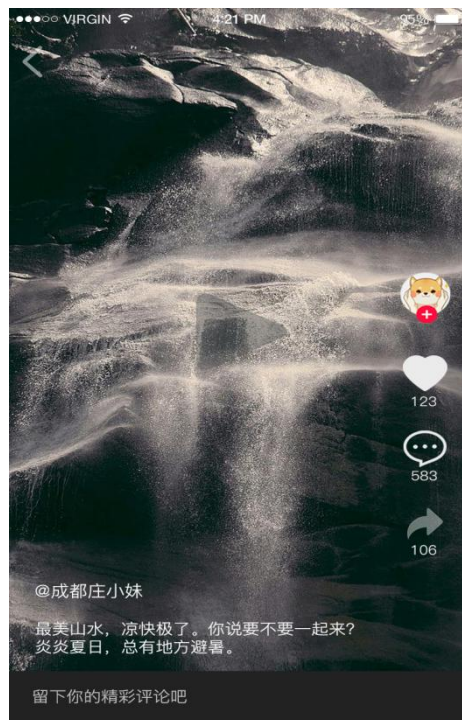


图 7-7 附近短视频播放界面

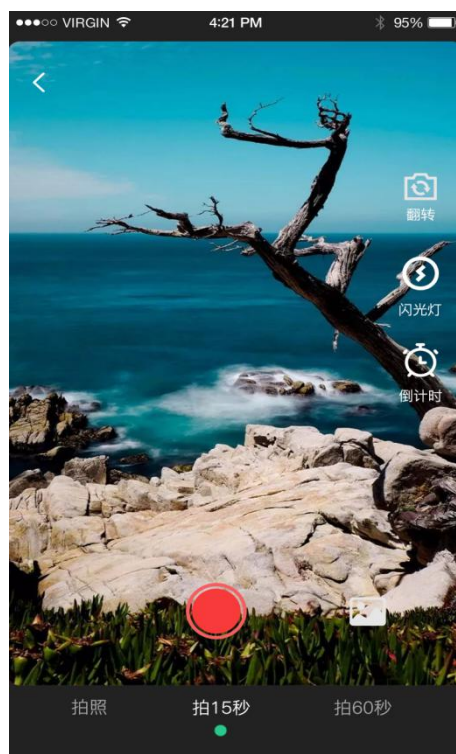


图 7-8 拍摄视频界面



图 7-9 消息界面



图 5-10 我的界面

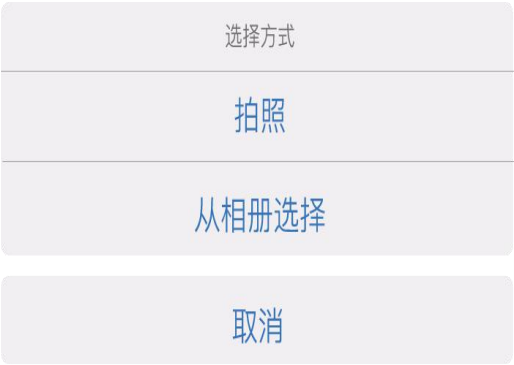


图 7-1 更换头像界面



图 7-12 编辑资料界面

7.3 本章小结

本章介绍了根据第三章的需求分析对各个模块功能进行测试的过程，其中测试包括功能性测试，用于测试每项功能点是否满足之前的需求；性能测试用于测试用户在使用每个功能时请求页面的相应时间是否达到之前的性能需求，经过测试，本系统在功能和性能上已经达到了之前的需求。

第八章 工程能力培养

8.1 工程计划管控和执行情况

基本管理原则：每位成员既是积极的建言者，又是负责的合作者，同时也是决策的制定者。决策应在充分的讨论基础上由大家共同做出，一旦决策做出就必须被及时有效的执行。禁止再有异议。

目标 1：按时按量完成项目的基本功能，按时发布产品及文档，这是本团队的最高目标。

目标 2：遵循规范化的项目运作标准，文档严谨完整，代码注释充分，便于后续维护，这是第二目标。

目标 3：产品运行稳定，界面友好，用户易操作，尽量从用户的角度去看问题，并提出解决问题的方案。

目标 4：注重团队建设，成员分工合理，团队成员合作默契，气氛融洽。每周的讨论会积极建言。在开发过程中积极协作。

目标 5：项目设计和开发上尽量有创新，有亮点。

实现目标的方案：

1. 每周例会上各位组员积极对当前的开发工作进行积极的评审和建言，由组长做最后的作口头总结。

2. 小组成员都要密切监控风险状态，发现风险后提交风险报告。并由组长做出临时处理决定。然后在该周的例会上由小组成员共同讨论对风险的处理意见。并形成风险处理的日志做为以后的经验。

3. 在项目进行的过程当中，组员之间应该多进行各种形式的非正式沟通，以使沟通更加的方便、快捷。

评审机制：每周例会上小组讨论形成一致意见后并，并邀请团长和其他组长参加评议。对于重大的风险处即为通过，相关负责人针对改进意见开展下一周工作，严格执行例会上所制定的决策。小组会议持续评估其成效。每一项目阶段结束之前（里程碑前后），组织一次阶段评审会，评估整个阶段的工作效率和成果质量。尽量与项目例会合理意见，应该由团长及其他组长组成评审团对处理意见进行审议和评估。并以评审团的决议作为重要参考来制定决策。

8.2 对本工程领域发展的认识

软件工程的发展为各行各业创造了良好的条件和基础，提高了工作效率。在企业中，正常工作的运行、管理很大程度依赖于合适的软件，软件的使用在一定层面上帮助企业提升了核心竞争力。软件工程作为计算机领域中重要内容，开发了计算机使用功能，主要研究软件构件技术、网络计算机智能化、标准化等内容。目前，社会的发展已离不开软件工程的帮助，软件工程的发展趋势将朝着模块性、开放性、全球化发展。

1.软件的发展和本质:

软件工程技术发展迅速，从 20 世纪 60 年代开始，经历 10 年左右时间，计算机结构化程序设计技术已实现，在这之后，计算机软件工程环境得到广泛的发展。到 20 世纪 90 年代，我国计算机软件工程技术发展到优化性能，软件使用的对象和方式开始发生变化。软件的发展伴随着事物的发展变化，软件消除硬件和编程语言的异构性作用，开发了不同操作系统，更轻松解决异构问题。软件的本质是演化性和构造性，软件的演化是软件的开发将伴随计算机科技的发展而进步，软件的性能设计是服务于人们的需求，同时依赖于信息科学技术的快速发展，在短短数十年，软件工程的发展已达到一定规模。

2.软件工程技术发展现状:

软件的构件是软件工程的基础，也是核心内容，重用则是软件的方法。通过构件的重用这种方式，软件开发人员可减少写代码工作，将精力集中于问题定义。构件的开放需要以开发工具为支撑，包括集成开发、应用管理、构件库管理等，在不同行业领域中，构件库有不同的业务构件。构件模型和软件体系结构是构件设计中重要问题，构件模型确定了软件系统构件的基本逻辑，将构件和构件之间的相互作用分开，进一步提高构件的独立性。

3.中间件技术:

中间件技术作为支撑存在于软件重用和构件化开发运行中，对软件重用和软件构件技术的发展有很大影响。中间件是软件系统集成的重要技术，提供给构件可运行空间。构件只有基于中间平台才会有作用，中间件技术包括数据库访问、通信、分布对象中间件、远程过程调用和事务处理，具有分布交互模式的关键特征，在异构环境中间起到协同作用。设计者在构件软件开发中主要关注于业务逻辑本身，对分布式应用中与业务无直接关系的问题无需投入大量精力。中间件可以屏蔽多种系统资源，在不同的操作系统和硬件平台中，实现接口和协议规范，开发并实现分布式系统与环境隔离的构件开发模式。

标准化技术:

标准化是软件重用和软件构件技术的发展需要，支撑着软件产业发展。软件工程标准分为软件标准和软件过程标准。当前，国际流行两大软件工程标准框架，其一为美国国防部提出的软件工程标准指南，涉及软件工程环境、软件工程的安全性、程序设计语言等，并针对性提出强制要求和推荐采用的支持软件系统工程化标准。其二为国际标准化组织的软件工程标准框架，提出了软件工程对标准的要求。软件工程标准化将朝着软件生存周期中整体软件开发流程标准化和具体系统的标准化方向发展，对多功能软件开发流程和开发环境和专用工具系统制定标准，促进软件标准化和软件过程标准化结合协调发展软件工程发展趋势：

1.全球化：

随着全球化发展，软件开发也经历着不同的开发环境，传统的开发理念对目前软件开发以有些不适应，软件开发未来发展将朝着 全球性发展。信息化的发展，使得软件开发不仅局限于国内发展，而且在国外开展也应用颇多。通过全球化发展，促使软件外包公司发展迅速，软件外包公司利用全球各国技术资源，将软件开发涉及的各部分内容分别完成。

2.开放计算：

开放性计算包含开放标准、 开源软件、开放架构三个内容，将开放标准进行统一，可以实现不同企业开发的不同软件可相互使用，软件开发工具也可集成应用。软件工程具有一定的开放性，开放性计算可帮助不同区域的软件开发员更便利地开发软件，加强相互之间的沟通，同时对后期的维护工作更加方便且专业。通过开放性构架的使用对 IT 环境中复杂的矛盾得到有效解决，提高软件工程的灵活性

3.模块化：

软件工程的全球化发展为软件开发带来发展机遇，同时也带给了软件工程发展的各种挑战。软件公司相互之间竞争激烈，模块化的发展提升了软件开发的效率，对提高企业竞争力有着重要的作用，这帮助解决了传统问题缺失灵活性的弊端，应用科学的方法对软件构件和与之相关的软件构件进行分类，将之模块化。

8.3 工程协作与交流

协作主要是利用 github、sourcetree 代码对代码进行管理，一定程度上避免了代码冲突问题。在于后台小伙伴讨论接口问题时，发表了自己的见解，也了解了一些后台相关的知识。代码的规范性、可读性上主要体现在文档的规范与注释的填写。优化了一些组件的代码。降低了耦合度。在解决难题时，我们组的小伙伴一起讨论如何实现，需要注意什么。如何降低崩溃方面做了很多的交流。再通过不断的交流后，终于完成了 OTunes 应用的开发

8.4 本章小结

本章介绍了本人在毕业设计过程中的工程能力与职业素养培养的情况。从工程计划管控、职业素养与工程伦理学习、对软件工程实践及软件工程领域发展等三大方面进行了比较全面的学习和提升，经过本次毕业设计，我的工程素养和道德素养也得到了进一步的提升。代码质量，规范性上也得到了一定的提升

第九章 结束语

9.1 全文总结

本文从选题理由、课题背景、前期知识技术调研、需求分析、设计再到实现、测试介绍了此课题以及其完整开发流程:

1. 首先我对整个项目进行了需求分析和设计,为后续的开发提供了明确的方向和蓝图。
2. 完成需求分析和设计后,我针对设计中的每一个模块进行开发,每一个模块中的功能点逐个实现。
3. 登录模块实现了注册、登录、退出登录功能,且测试通过。
4. 视频播放上利用 UITableView 的 cell 复用性不仅优化了播放性能,避免内存消耗过大的问题,也在一定程度上优化了与用户的操作交互。
5. 在发布视频方面,用户不仅可以利用本应用程序选择拍照或者拍摄视频,也能选取相册视频进行上传,一定程度上优化了视频的获取方式。
7. 本应用程序视频接口主要是调用推荐、关注、附近等网络接口,满足了一定范围内用户的观看需要。

9.2 不足与下一步工作

本应用程序不足之处就是,拍摄视频效果简单,后期不能编辑图片,上传文件较大不利于信息交互。由于开发时间限制,编辑技术具有一定的复杂性,同时性能要求也是一个难点,背景音乐选择上也比较单调。增加拍摄的趣味性,这将是下一步要做的工作。

参考文献

- [1] 胡辉. IOS 环境下使用 MVC 模式进行 APP 开发的设计思路探索[J].数字技术与应用,2015, 06:204-206.
- [2] 崔维, 李晖, 刘佳佳, 王艳娟. 基于 IOS 的企业 APP 客户端的开发与实现[J].科技创新与应用,2014, 24:30-31.
- [3] 张超. 基于 iOS 平台播放器的设计与实现[D].河北师范大学,2015.
- [4] 牛丽, 殷凡, 熊志勇. iOS 平台移动学习资源构建及终端实现研究[J].中国教育信息化,2015, 12:18-21.
- [5] 李爱军. 基于 iOS 平台的录音软件系统的设计与实现[J].办公自动化,2014,13:57-59.
- [6] 王志军. iOS 设备无缝调用媒体资源[J]. 电脑爱好者, 2014, 02:75.
- [7] 钱露. 基于 iOS 平台的小型社交网络的关键技术的研究与系统实现[D].北京邮电大学, 2015.
- [8] 田妞. WWDC2013 亮点汇[J].中国质量万里行, 2013(07).
- [9] 关东升. iOS 开发指南 (第五版) , 2017:283-301.
- [10] 黄海. iOS 开发基础教程, 2018:191-208.

致谢

论文是在何凯霖老师的精心指导下完成的。他帮我完善了报告的内容，帮助我修正了许多错误，给予了我很大帮助。在此，我向何凯霖老师表示衷心的感谢。

企业方导师谢曦高级工程师也给了我许多建议，项目的需求分析和设计中何凯霖老师也给了我很多的帮助，在此表示感谢。同时，我还要感谢我家人的支持，衷心感谢所有关心和帮助过我的人们。