

# 《毕业论文》检测报告

PaperOK

总相似率：27.15%

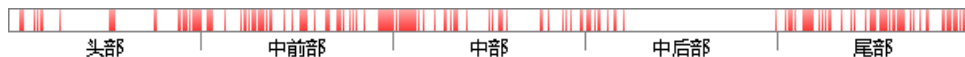
## 基本信息

文档名称	毕业设计论文
报告编号	698fa726-20df-4018-b53d-24358058428e
文档字数	23221
提交人姓名	佚名
提交方式	粘贴文本检测
检测范围	学位论文库（含硕博）、学术期刊库、会议论文库、法律法规库、互联网资源库、自建比对库
提交时间	2020-05-11 00:32:18

## 检测报告指标详情

原创率	抄袭率	引用率	字数统计	参考文献字数
72.85%	27.15%	0%	23221	-

## 相似片段位置图



注：红色部分为重度相似，橙色部分为中度相似，蓝色部分为引用部分

## 相似片段详情（仅显示前10条）

序号	篇名	来源	命中率
1	浅谈软件工程技术现状和发展趋势	学术期刊库	3.2%
2	移动短视频对新闻生产与传播机制的变革	学位论文库	1.23%
3	IOS平台上基于SIP协议的即时通讯软件设计与实现	学位论文库	1.16%
4	...the Cheng Cycle? offers higher part-load efficiency...	互联网资源库	0.8%
5	浅析软件工程的发展趋势	学术期刊库	0.72%
6	SSDLC实践:安全设计评审 - FreeBuf互联网安全新媒体平台	互联网资源库	0.63%
7	软件开发公司哪家比较好一点_百度知道	互联网资源库	0.63%
8	iOS将一个viewController的view加到self.view上_冰凉的..._CSDN博客	互联网资源库	0.62%
9	使用与满足理论视角下移动短视频的发展策略研究——以秒拍APP为例	学位论文库	0.53%
10	移动短视频的传播及运营方式研究——以快手app为例	学位论文库	0.52%

## 文档原文标注

电子科技大学成都学院

CHENGDU COLLEGE OF

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

毕业设计论文-技术版

论文题目 OTunes短视频App

程序设计与实现

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 + fgmpeg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 + fgmpeg, 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 工程能力的培养	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，方便用户记录生活中有趣或者珍贵的事并发布分享平台，供其他人一起分享。满足用户在闲暇的时间里打发无聊时间，也在信息传播上更加快捷、方便。针对iOS系统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:

#### 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:接口请求时间,播放的流畅性等。

#### 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非法视频信息。

### 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AFNetworking网络请求框架，Masonry自动布局框架、QMUIKit框架的调研与使用，Xib、Storyboard可视化编程说我区与各自的优缺点。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. 用户输入账号与密码
2. 前端先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 后台返回登录结果

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：填写账号、密码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（账号、密码）
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OTunes短视频的各个方面的内容，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：view视图层，用于显示和渲染数据。以一定的形式向用户呈现数据并允许用户对数据进行操作；controller控制器层：业务模块流程的控制。在Model和View之间加入一个Controller的主要目的就是“去耦”，即消除Model和View之间的耦合性；view视图层：业务模块的逻辑应用设计；model数据模型层：与网络接口数据进行交互，为应用程序提供独立的，不依赖于view和Controller子系统的数据模型。

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，然后通过KVO（属性观察）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，然后加入到了到一个可变数组中。Controller创建时就会请求该数据，通过属性传递给子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：

图5-1网络视频数据model

图5-2模型赋值

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tokrn，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

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.videoPath]];

? ? originVideoURL = [NSURL URLWithString:[NSString
stringWithFormat:@"%s%@",kAddressUrl,currentPlaySmallVideoModel.videoPath]];

? ? 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 播放下一个

发布视频流程：

- 1.用户可以选择从本地相册获取视频、图片或者拍摄视频、图片素材，
- 2.前端将用户名、视频或图片数据数据、标题信息后一并传给后端，
- 3.将视频的所有信息放入数据库保存，后台审核后视频信息后，
- 4.通过接口返回给前端，
- 5.。

发布视频方法名: deleteFromFavorite

输入参数:userName(用户名),favoName(视频数据),patentNumber(标题)

输出参数: “delete successfully” (发布成功)、 “error” (上传失败)

程序流程图如图6-2。

图6-2 发布视频流程图

核心代码如代码6-2

代码 6-2 发布视频网络请求

```
- (void)clickShootBtn {  
    ?? // 拍照  
  
    ?? if (self.selectedIndex != 0) {  
        ??? if (!self.camera.isRecording) {  
            ????? self.flashOnButton.hidden = YES;  
            ????? self.exchangeCamera.hidden = YES;  
            ????? self.pickerView.hidden = YES;  
            ????? self.pointLabel.hidden = YES;  
            ?????  
  
            ????? NSURL *outputURL = [[[self applicationDocumentsDirectory]  
URLByAppendingPathComponent:@"video"] URLByAppendingPathExtension:@"mp4"];  
  
            ????? [self.camera startRecordingWithOutputUrl:outputURL didRecord:^(LLSimpleCamera  
*camera, NSURL *outputFileUrl, NSError *error) {  
  
                ????? KB_PublishViewController *vc = [[KB_PublishViewController alloc] init];  
  
                ????? vc.videoUrl = outputURL;  
  
                ????? vc.image = [self getVideoFirstViewImage:outputURL];  
  
                ????? [self.navigationController pushViewController:vc animated:YES];  
  
                ????? }];  
  
            ??? } else {  
  
                ????? self.flashOnButton.hidden = NO;  
  
                ????? self.exchangeCamera.hidden = NO;  
  
                ????? self.pickerView.hidden = NO;  
  
                ????? self.pointLabel.hidden = NO;  
  
                ????? [self.camera stopRecording];  
            }  
        }  
    }  
}
```



????}

???? [SVProgressHUD showSuccessWithStatus:[NSString  
stringWithFormat:@"%@",@(self.selectedIndex)]];

?? }else {

?????

???? [self.camera capture:^(LLSimpleCamera \*camera, UIImage \*image, NSDictionary  
\*metadata, NSError \*error) {

?????? if (!error) {

???????? KB\_PublishViewController \*vc = [[KB\_PublishViewController alloc] init];

???????? vc.image = image;

???????? [self.navigationController pushViewController:vc animated:YES];

?????? }

???? } exactSeenImage:NO];

?????

?? }

}

//获取视频第一帧（用于制作封面）

-(UIImage \*)getVideoFirstViewImage:(NSURL \*)path {

???

?? AVURLAsset \*asset = [[AVURLAsset alloc] initWithURL:path options:nil];

?? AVAssetImageGenerator \*assetGen = [[AVAssetImageGenerator alloc] initWithAsset:asset];

???

?? assetGen.appliesPreferredTrackTransform = YES;

?? CMTime time = CMTimeMakeWithSeconds(0.0, 600);

?? NSError \*error = nil;

?? CMTime actualTime;

?? CGImageRef image = [assetGen copyCGImageAtTime:time actualTime:&actualTime  
error:&error];

```
?? UIImage *videoImage = [[UIImage alloc] initWithCGImage:image];  
  
?? CGImageRelease(image);  
  
?? return videoImage;  
  
}
```

## 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，有无崩溃等其他问题。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，用户收成功登录

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运行，系统能否根据用户操作正常反应

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，关注成功

通过，和预期结果一致

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，本应用程序在功能和性能上基本达到了之前的要求，基本无崩溃情况出现。 **第八章 工程能力的培养**

### 8.1 工程计划管控和执行情况

基本管理原则：由于iOS前端由我和另外一位小伙伴一起开发，为了避免代码冲突，我们原则上每天都必须在约定时间之前提交代码，由后一位提交代码的人拉取合并然后在提交，遇到冲突一起商量解决，保证提交代码可正常编译。明确分工，由于每个人的能力不一致，在遇到困难问题时先共同商讨，并制定出解决方案，每个人都必须有效执行，不再由其他异议。

目标 1：每个组员必须按时完成分配的任务，再由组长检查质量，不合格则需要重新编写。 **以达到按时按量完成任务的目标为宗旨。**

目标 2：命名、注释必须规范化，便于后续更新维护，重要方法必须加以注释。 **这是本工程的基本目标。**

**目标3：程序能稳定运行，崩溃率需低于0.10%（崩溃率=崩溃次数/每次启动次数），界面整体配色一致，无显示bug.良好的交互，尽量从用户的角度去分析处理问题，并积极提出比较好的解决方案。**

**目标4：根据个人能力分工，保证每个小伙伴的工作量。**注重团队成员的学习氛围，遇到问题，积极的与相关人员之间的沟通，每周的讨论会积极参与并发表良好的见解，以达到增强表达能力的目的。 **开发过程中必须积极配合，遇到问题及时处理。**

目标5：程序在设计样式上要有一致性，开发上尽量展现出一些有创新或亮点的功能。

实现目标的方案：

1. 每周讨论会上每个成员积极对自己的工作作出总结，包括自己完成了什么功能，质量上是否满足需求。并上报当前工作的进度，以达到监控程序进度的目的，最后组长再发表总结，成员记录会议重要的事情。

2. 每一位成员都有责任和义务监控风险状态，一但发现风险积极上报。再由组长做出临时的处理意见。然后在该周的讨论会上，每个小组成员发表对风险的处理意见。并记录，以避免在以后再次出现。

3. 在程序开发的过程中，小组成员应该积极沟通。多交流，以避免出现重大问题。保证程序按时完成。

**评审机制：**评审的目的是为了提高软件质量。每周讨论会上小组成员讨论，达成一致意见后也可邀请其他组的组长参加评议。每个成员严格执行所制定的决策。组长负责监督每个成员是否对决策严格执行。在以后的会议上持续监控，并观测其实施效果。如果不满足，则需要重新评估并制定应对措施。每一项目阶段结束之前，组长组织一次阶段评审会，评估整个阶段的工作效率和质量。尽量与每周讨论会意见一致。

## 8.2 对本工程领域发展的认识

软件工程技术以其惊人的速度不断壮大，从上世纪六十年代末，随着计算机应用领域的不断壮大，人们对软件的需求量一几何倍速增长，对软件的正确性也提出了更高的要求，并且在开发周期上也不断缩短。但是，当时的软件编写还只是一种手工活动，过多的依赖程序员的个人能力和技巧，导致了软件成产周期长，可靠性以及可维护性也极差。软件开发远远不能满足社会的需求，从而爆发了一场“软件危机”。从而，计算机软件工程应运而生。软件工程的发展为各行各业创造了良好的条件和基础，不仅提高了工作效率、生活效率，也促进了经济和社会进度的发展。在企业中，正常工作生产的运行、人员管理很大程度需要一款合适的软件来满足需求，软件的使用在一定程度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简易性、模块性、共享性、全球化发展。

### 软件的发展和本质：

从软硬件的技术体系结构来看，软件和硬件并不分家，软件本身就是通过硬件的设计来体现的，比如早期进行程序设计都需要操作各种硬件设备来完成。后来操作系统的出现促使软件和硬件实现了两条发展轨迹，硬件提供计算能力，而软件提升计算方法。从目前大数据、云计算、物联网和人工智能的发展趋势来看，未来软件的定义将有三方面的变化，其一是软件即硬件概念，对于用户来说，硬件将逐渐透明化，用户掌握了软件也就掌握了硬件资源的管理能力；其二是软件即服务的概念，实际上这也是云计算的主要服务方式之一；其三是软件即智能的概念，在人工智能技术的推动下，未来软件将是人工智能技术的重要体现方式。

### 2. 软件工程技术发展现状：

随着信息技术的不断发展，计算机软件工程现代化技术被划分为应用软件、系统软件与支持软件3种形式。在实际应用中，计算机软件工程具备运行效率高、且技术较为成熟等优势。这些优势使得计算机软件工程在应用过程中可以更好的姿态保护计算机系统，并支持整个系统获得高效、安全运行。随着时间地推移，计算机软件工程现代化技术运行状态逐步趋于稳定。然而，由于我国此项技术起步较晚，与发达国家相比，仍有着较大的发展空间。比如，从生产水平来看，我国的计算机工程软件技术生产水平仍有待进一步提升。过低的生产水平在一定程度上制约了计算机工程现代化技术的发展速度。为此，加快计算机软件工程现代化技术发展速度与发展水平，就需要给予此项工作足够的重视，并制定出适宜的发展对策。

### 标准化技术：



在计算机科学领域，技术标准最成熟，也是和学术界最接近的是密码学。在软件行业中，最有趣的是很多人对造轮子非常感兴趣（包括我），不过有一些轮子我不建议自己造。这就是加密算法，不止一个人在聊天中谈起对信息安全的看法时说，要是我开发一个自己的加密算法只有我自己的知道（甚至很多真的这么干了），肯定是天底下最安全的。这是一种非常朴素的信息安全认识，自己的创造的“加密算法”也只是根据特定规则对信息的混淆和变换，这甚至落后于凯撒密码。

现代密码学已经有大量的对称加密、非对称加密、HASH公开算法，甚至建立了一套完整的通信信息安全基础设施（PKI），保证信息安全的是密钥而不是加密算法。与其闭门造车，不如选择RSA 等大师们的成果。

软件工程发展趋势：

### 1.全球化：

21世纪，互联网的诞生加快了全球化的进程。世界各国的经济发展逐渐朝着全球化的方向发展，文化也逐渐朝着多元化的方向迈进。在多种力量推动下，未来软件工程技术也将朝着全球化的方向发展。在高信息化、高科技化的时代，软件工程已经不再是依靠某一个国家能够完成的任务，而是需要多个国家共同协作，共同完成某一项软件技术的开发。我国每年在此领域外包产生的收益是非常可观的，在此前提下，我们必须加快软件工程全球化的速度，推动软件工程技术更多的国家合作，并实现共赢的目的。软件工程全球化也是经济全球化的一部分。

### 2.共享性：

朝着开放性的趋势发展开放性就是软件工程技术的共享性，这与软件工程技术的全球化具有明显区别。软件工程技术的全球化是指多个国家共同合作打造成一款新型的软件工程，但是各自双方有权保留技术经验。软件工程技术的开放性指的是彼此双方或者多方可以互相交流软件工程技术的经验，学习他人的长处来弥补自身的短处。开放性软件工程技术具有全面统一的技术开发标准，在一定程度上减少了各国在IT环境下的多种矛盾，有利于多方合作交流。

### 3.模块化：

模块化发展是软件工程技术未来发展的另一趋势。在未来，企业的发展与竞争要比如今更加残酷与激烈，而软件工程技术作为一门专业的学科，拥有自己的逻辑体系与理论框架。企业在软件开发的过程中可以直接理论已有的理论框架进行技术开发，既能提高了软件开发的效率，也提高了企业的实际竞争力。比如：企业想要开发一款便于用户售后服务的软件，企业的软件工程技术开发人员只需要利用已有的理论框架，对其中售后服务的内容进行有效填充就可以满足企业对软件的需求。

简化化:

软件工程技术另一发展趋势就是简化化。这里所说的简化化是指将软件工程技术开发过程中的代码简化化。将软件工程技术过程中的代码转化成能够容易识别并记忆的代码是软件工程技术发展的一大趋势。在这个趋势下，软件工程人员不断提升自身的专业素质，加强自身专业知识的储备。简化化的软件工程技术产品能够使用户在使用的过程中更加便捷，除此之外，还可以帮助用户在最短的时间内掌握软件的特征以及软件的操作方式。？

## 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

本章介绍了个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。**对UIKit框架更加熟悉了解，也能更加灵活的运用，对组件封装更加得心应手。

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，且测试通过并且无崩溃现象出现。**
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，充分利用tableviewcell复用来实现视频数据预加载的效果。
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，提供给用户更多选择。
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

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.

**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.

张超. **基于iOS平台播放器的设计与实现[D]**.河北师范大学,2015.

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.

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.

王志军. **iOS设备无缝调用媒体资源[J]**. 电脑爱好者 , 2014 , 02:75.

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.

田妞. WWDC2013亮点汇[J].中国质量万里行, 2013(07).

关东升. iOS开发指南 (第五版), 2017:283-301.

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。