**四川师范大学本科毕业设计**

|  |
| --- |
| 基于iOS的美妆电商类APP实现 |

|  |  |
| --- | --- |
| **学生姓名** | **＊＊＊** |
| **院系名称** | **计算机科学学院** |
| **专业名称** | **软件工程** |
| **班 级** | **2＊＊＊级＊班** |
| **学 号** | **＊＊＊** |
| **指导教师** | **李贵洋** |
| **完成时间** | **2016年5月6日** |

**基于iOS的美妆电商类APP实现**

学生：＊＊＊ 指导教师：李贵洋

内容摘要：此为内容摘要，大概300字

关键词：iOS开发 移动智能客户端 电子商务

**THE IMPLEMENTATION OF ELECTRICITY BUSINESS APPLICATION FOR IOS**

**Abstract:** say something lalalalala

**Key Words: iOS development Mobile intelligent terminal Electricity business**

# 目录

1. 概述 ………………………………………………………………………………
   1. 研究的目的和意义 ………………………………………………………
   2. 研究的背景 ………………………………………………………………
      1. 国内现状 …………………………………………………………
      2. 国外现状 …………………………………………………………
   3. 主要贡献 …………………………………………………………………
   4. 文章的结构 ………………………………………………………………

2. 预备知识及原理说明 ……………………………………………………………

2.1 Cocoa Touch …………………………………………………………………

2.2开发语言及开发工具 ………………………………………………

2.3设计开发模式MVC …………………………………………

2.4 依赖管理工具CocoaPods

2.5 代码管理工具 Git

2.4 iOS 第三方库 ………………………………………………………………

2.5 版本控制软件git …………………………

3. 软件系统分析 ………………………………………….

4．软件设计和实现 ………………………………………

5. 基于iOS的电商美妆app系统测试和运行 ……………………………………

6. 成果与展望 ………………………………………………………………………

7. 致谢 ………………………………………………………………………………

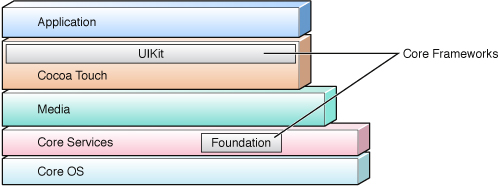
参考文献 ………………………………………………………………………………

**基于iOS的美妆电商类APP实现**

# 1 概述

# 2预备知识及原理说明

# 2.1 Cocoa Touch

****

**iOS 系统架构**

Cocoa不是一种编程语言（它可以运行多种编程语言），它也不是一个开发工具（通过命令行我们仍然可以开发Cocoa程序），它是创建Mac OS X和IOS程序的原生面向对象API，为这两者应用提供了编程环境。

Cocoa Touch 是针对于iOS的应用开发环境，它包括Objective-C 运行环境和两个核心框架：foundation框架和ApplicationKit（UIKit）框架。框架的功能类似于动态库，既可以在运行时动态的载入应用程序的地址空间，但框架作为一个对计算机的捆绑，而非独立文件，除了可执行代码外，也包含了资源，头文件和文档。

foundation框架作为通用的面向对象函数库，主要定义了一些基础类，提供了字符串，数值的管理，容器及其枚举，以及一些其他的与图形用户界面 没有直接关系的功能；UIKit框架提供了在屏幕上绘制的机制，捕获事件，和创建通用用户界面元素。UIKit也通过管理显示在屏幕上的组件来组织复杂的项目。使用UIKit可以：构建和管理你的用户界面，捕获触摸和基于移动的事件，呈现文字和web内容，优化你的多任务程序，创建定制的用户界面元素。

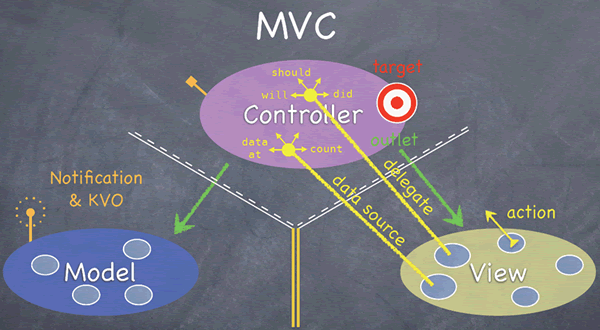
**2.2 开发语言及开发工具**

20世纪八十年代初，Brad Cox在其公司Stepstone发明Objective-C，它以一种叫做SmallTalk-80的语言为基础。Objective-C建立在C语言之上，意味着它是在C语言基础上添加了扩展而创造出来的能够创建和操作对象的一门新的程序设计语言。对Objective-C最主要的描述是他1986年出版的《 Object-oriented Programming, An Evolutionary Approach》。1988年，NeXT Software公司获得了Objective-C语言的授权，并开发出了Objective-C的语言库和一个名为NEXTSTEP的开发环境。1992年，自由软件基金会的GNU开发环境增加了对Objective-C的支持。1994年，NeXT Computer公司和升阳微系统（Sun Microsystem）联合发布了一个针对 NEXTSTEP系统的标准典范，名为OPENSTEP。OPENSTEP在自由软件基金会的实现名称为GNUstep。1996年12月20日，苹果公司宣布收购NeXT Software公司，NEXTSTEP/OPENSTEP环境成为苹果操作系统下一个主要发行版本OS X的基础。这个开发环境的该版本被苹果公司称为Cocoa。（互动百科）

苹果公司于2008年3月6日发布了iPhone和iPod touch的应用程序开发包,其中包括Xcode开发工具、iPhone SDK 和iPhone手机模拟器。第一个Beta版本是iPhone SDK 1.2b1(build 5A147p),发布后立 能使用, 是同时 出 的App Store所需要的 件更新 到2008年7月11日 发布。编写本书时,iOS SDK 7.0.4版本 经发布。  iOS开发工具主要是Xcode。自从Xcode 3.1发布以后,Xcode 成为iPhone 件开发工具包的开发 。Xcode 可以开发Mac OS X和iOS应用程序,其版本是与SDK相 对应的。例如,Xcode 3.2.5与iOS SDK 4.2对应,Xcode 4.1 与iOS SDK 4.3对应,Xcode 4.2与iOS SDK 5对应,Xcode 4.5和Xcode 4.6与iOS SDK 6对应,Xcode 5与iOS SDK 7 对应。  在Xcode 4.1之前,还有一个配 使用的工具Interface Builder, 是Xcode 件的一部分,用来设计 体和视 图,通过 可以 所见 所得 地 控件并定 事件等,其数据以XML的形式 存储在xib文件中。在Xcode 4.1 之后, Interface Builder成为了Xcode的一部分,与Xcode集成在一 。 （iOS开发指南：从零基础到App Store上架（第2版 ）

**2.3设计开发模式MVC**

MVC，全称是 Model View Controller，是模型 (model)－视图 (view)－控制器 (controller) 的缩写。它表示的是一种常见的客户端软件开发框架。MVC 的概念最早出现在二十世纪八十年代的施乐帕克实验室中，当时施乐帕克为 Smalltalk 发明了这种软件设计模式。现在，MVC 已经成为主流的客户端编程框架，在 iOS 开发中，系统为我们实现好了公共的视图类：UIView，和控制器类：UIViewController。大多数时候，我们都需要继承这些类来实现我们的程序逻辑，因此，我们几乎逃避不开 MVC 这种设计模式。



iOS系统下的MVC关系图

Paul Hegarty，斯坦福大学公开课：iOS 7应用开发[EB/OL]. 网易公开课.2014.

Controller能够访问Model和View，Model和View不能互相访问；当View与用户交互产生事件时，使用target-action方式来处理；当View需要处理一些特殊UI逻辑或获取数据源时，通过delegate或data source方式交给Controller来处理；Model不能直接与Controller通信，当Model有数据更新时，可以通过Notification或KVO (Key Value Observing)来通知Controller更新View。

另外由MVC还衍生出了一些其他的架构：MVVM，MVCS，VIPER等。

MVVM是基于胖model的架构思路建立，在胖model中拆出两部分：model和viewModel，viewModel让Controller只需关注于数据调配，它去负责数据加工并通过通知机制让view响应viewModel的改变，总之不管是胖model还是viewModel都是给controller减负。

（PS：胖model：即包含了部分业务逻辑，controller从其拿到数据之后不用做额外操作或很少操作就能使用，瘦model：只负责数据的表达，然后配套helper类或方法来对弱业务做抽象，其他一律扔给controller。独立性强。但一定程度上违背了DRY，即Don’t repeat yourself）

MVCS是从controller中把负责数据存储部分提取出来，交给另外一个对象去做，这个对象就是store。

VIPER，即视图 (View)，交互器 (Interactor)，展示器 (Presenter)，实体 (Entity) 以及路由 (Routing)。将一个应用程序的逻辑结构划分为不同的责任层。这使得它更容易隔离依赖项 (如数据库)，也更容易测试各层间的边界处的交互。

**2.4 依赖管理工具CocoaPods**

随着iOS开发者的增多，业界也出现了为iOS程序提供依赖管理的工具，经过多年发展，现在已经成为iOS开发事实上的依赖管理标准工具。开发iOS项目不可避免地要使用第三方开源库，CocoaPods使得我们可以节省设置和更新第三方开源库的时间。

引用第三方库，一般需要：把这些第三方开源库的源代码复制到项目中，或设置成git的submodule；需要手工地将这些开源库所依赖的系统框架一一添加到项目依赖中，比如一个网络库需要添加的框架有CFNetwork, SystemConfiguration, MobileCoreServices, CoreGraphic, zlib; 对于某些开源库需要设置-licucore或者-fno-objc-arc等编译参数；最后还得管理这些依赖包的更新。

使用CocoaPods之后，我们只需要将用到的第三方开源库的源码放到一个名为Podfile的文件中，然后执行Pod install。CocoaPods就会自动将这些第三方开源库下载下来，并且为我们的工程设置好相应的系统依赖和编译参数，执行pod update便可更新指定的第三方库。《iOS开发进阶》唐巧著. 北京：电子工业出版社，2015.1 ISBN 978-7-121-24745-3

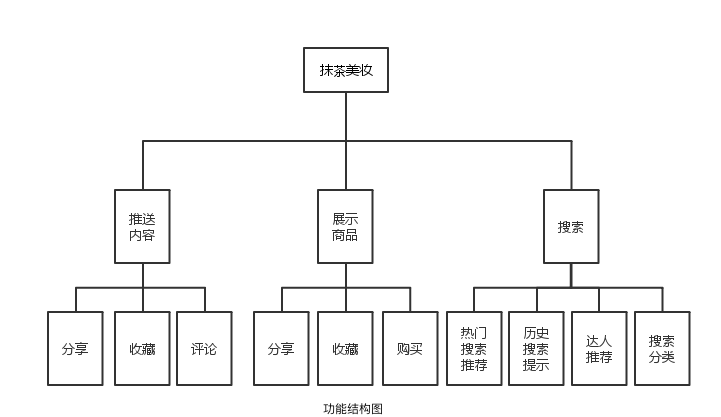
**2.5版本控制软件git**

git是一个免费的、分布式的版本控制工具，或是一个强调了速度快的源代码管理工具。Git最初被Linus Torvalds开发出来用于管理Linux内核的开发。每一个Git的工作目录都是一个完全独立的代码库，并拥有完整的历史记录和版本追踪能力，不依赖于网络和中心服务器。git和其他版本控制系统（如CVS）有不少的差别，git本身关心文件的整体性是否有改变，但多数的CVS或Subversion系统则在乎文件内容的差异。因此git更像一个文件系统，直接在本机上获取数据，不必连接到主机端获取数据。

# 3 软件分析

**3.1 需求分析**

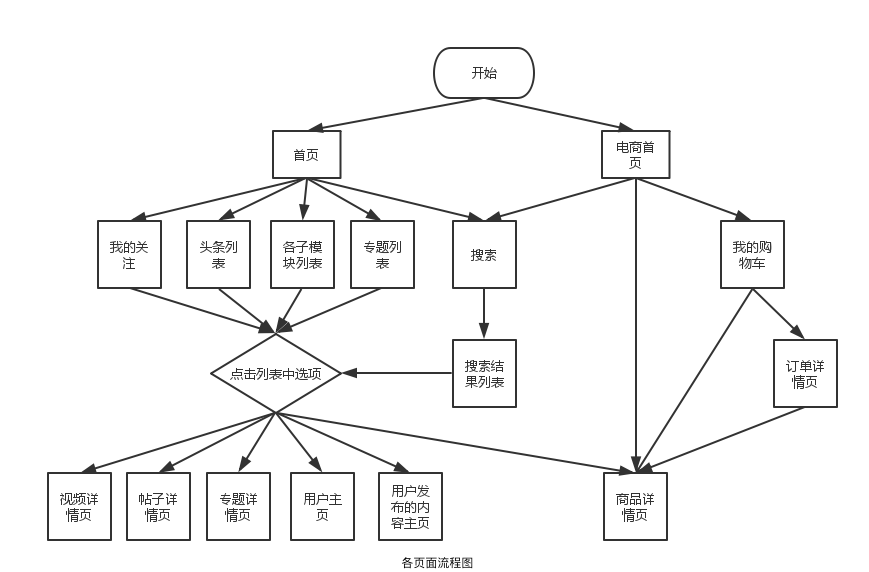
为了实现两大需求，第一是实现动态显示从服务端获取的指定信息，如视频，帖子，专题等，并支持和用户间的互动，比如收藏内容，关注发布者，评论信息等；第二是线上商品销售，主要针对美妆类商品，支持动态展示商品，加入购物车，下单结算等电商模式；编写一个基于iOS的客户端。



具体产品功能需求如下：

1. 对于新用户，有个用法引导，对于老用户升级到新版本后，针对新功能也有个引导提示，所有引导只显示一次，除非卸载后重新安装。
2. 首页模块中，顶部导航栏有支持横滑的选项卡，选项卡元素包括关注、头条、护肤、彩妆、时尚、专题，除了关注，头条，专题这三块是固定显示之外，其他选项内容是动态从服务端获取。每个选项卡对应不同页面，每个页面显示样式不同。打开软件进入首页后默认主页面，即跳转到哪个选项由服务端确定。
3. 首页所有页面支持下拉刷新以及上拉加载更多，进入主页面后，第一次以及超出指定时间间隔后滑动至或点击选项卡跳转至其它页面后支持自动刷新。
4. 首页中的关注页面，主要显示用户所关注的用户发布的内容，以及显示推荐用户，便于用户关注。如果进入该页面后，用户没登陆，得提醒用户登陆，并提供入口。
5. 首页中的头条页面，也是核心页面，它是具体所发布的信息的入口，它主要由两大显示块组成，一是顶部的banner,即支持横滑，也支持自动轮播的推荐内容入口，点击不同内容跳到不同的详情页；其他的就是以表视图形式展示信息，并支持不同类型的cell显示，其中视频类的cell点击播放按钮要支持在当前界面播放，当视频滑出屏幕后要关闭视频，针对视频和帖子类型的cell会有发布者的信息，如果用户没关注该发布者，则提供关注按钮，用户点击“关注”即可关注该用户，点击发布者头像会跳到发布者主页。
6. 首页其他各选项即是所有信息的分类显示，其中专题分类以feed流表视图的形式展示，其他的分类以瀑布流的形式展示。点击后都会跳转至对应详情页。
7. 首页每个cell里可能会包含推荐的商品，如果有则有相应按钮，点击弹出弹层，展示商品列表，如果该商品有库存则点击商品跳转至商品详情页，如果无货，则提供“种草”，即可加入心愿单。
8. 专题详情页所包含的内容有封面图，标题，分享入口，发布者信息，具体内容，各评论信息。其中服务端返回的内容部分包括：图片文字，推荐的商品信息（可加入心愿单或加入购物车或跳至详情页购买），视频（支持当前播放），其他相关专题入口。
9. 电商模块中主页模式与首页有点相似，只是是各商品的展现，点击具体商品跳至相应商品详情页。
10. 商品详情页从上到下包括的内容为，支持横滑的商品图片及视频，商品名，商品售价，简介，部分评论，详细信息，其他专题及商品推荐。以及各入口，包括分享，客服入口，查看购物车，加入购物车。点击加入购物车要判断该商品是否有多种规格，若无则直接加入购物车，若有，则弹出弹层，供用户选择，其中不能点击没库存的规格，选择完后再加入购物车。
11. 搜索模块，搜索的入口在首页导航栏上，即选项卡的右边。搜索页面中：导航栏中有搜索输入框，可搜索出发布的各种信息，下边界面上则有四个版块，热门搜索：提供热门的搜索标签，可直接点击进行搜索该标签内容，历史搜索：所有的历史搜索词都以标签形式展示，用户可直接点击搜索，另外有删除历史搜索纪录按钮，若没有历史搜索则无该版块，推荐用户：点击可进入相应用户所有发布内容的界面，其他推荐：该版块下把所有搜索内容分成有五类：妆品分类，妆品功效，基础彩妆，基础护肤，化妆风格，点击跳转的页面把这五类又细分了很多块，并且点击不同类，跳转的页面对应相应的分类。

**3.2 软件流程**



**软件各界面关系，以及跳转关系图（部分）**

**4 软件设计**

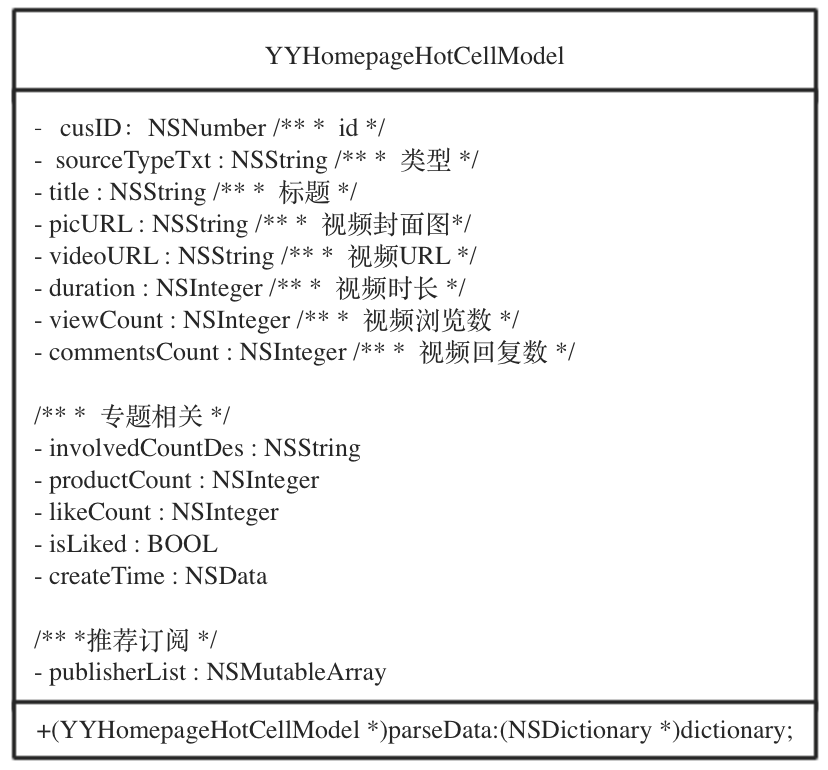
**4.1 基于MVC的架构设计**

**app设计中，除了第三库，工具类的引用，主要把程序代码分为了三个部分，模型（model），视图控制器（controller），视图（view），部分把controller里的内容分了个VM出来，以减轻视图控制器的负担。**

****

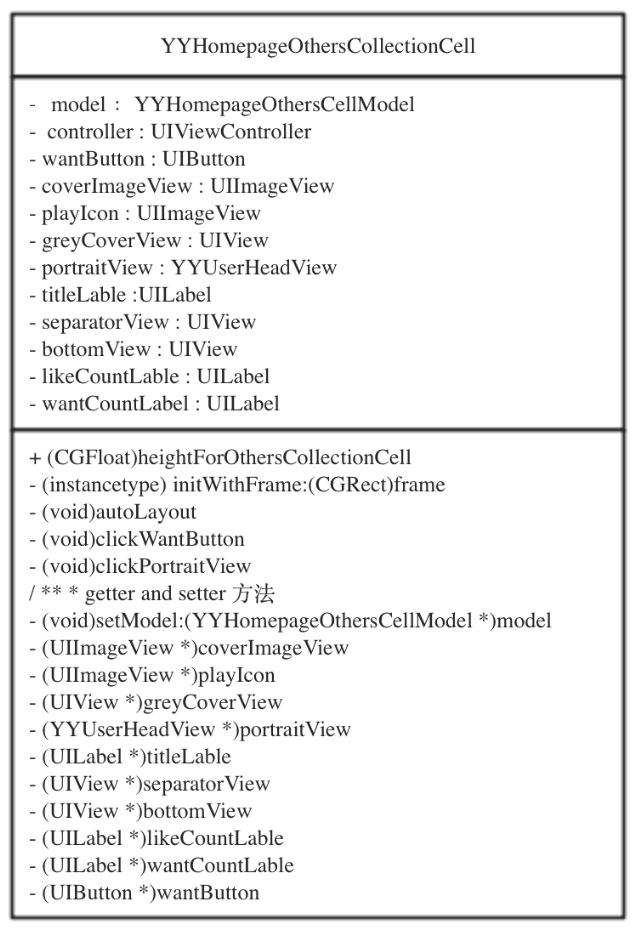
**4.2 关键类**

模型代表类



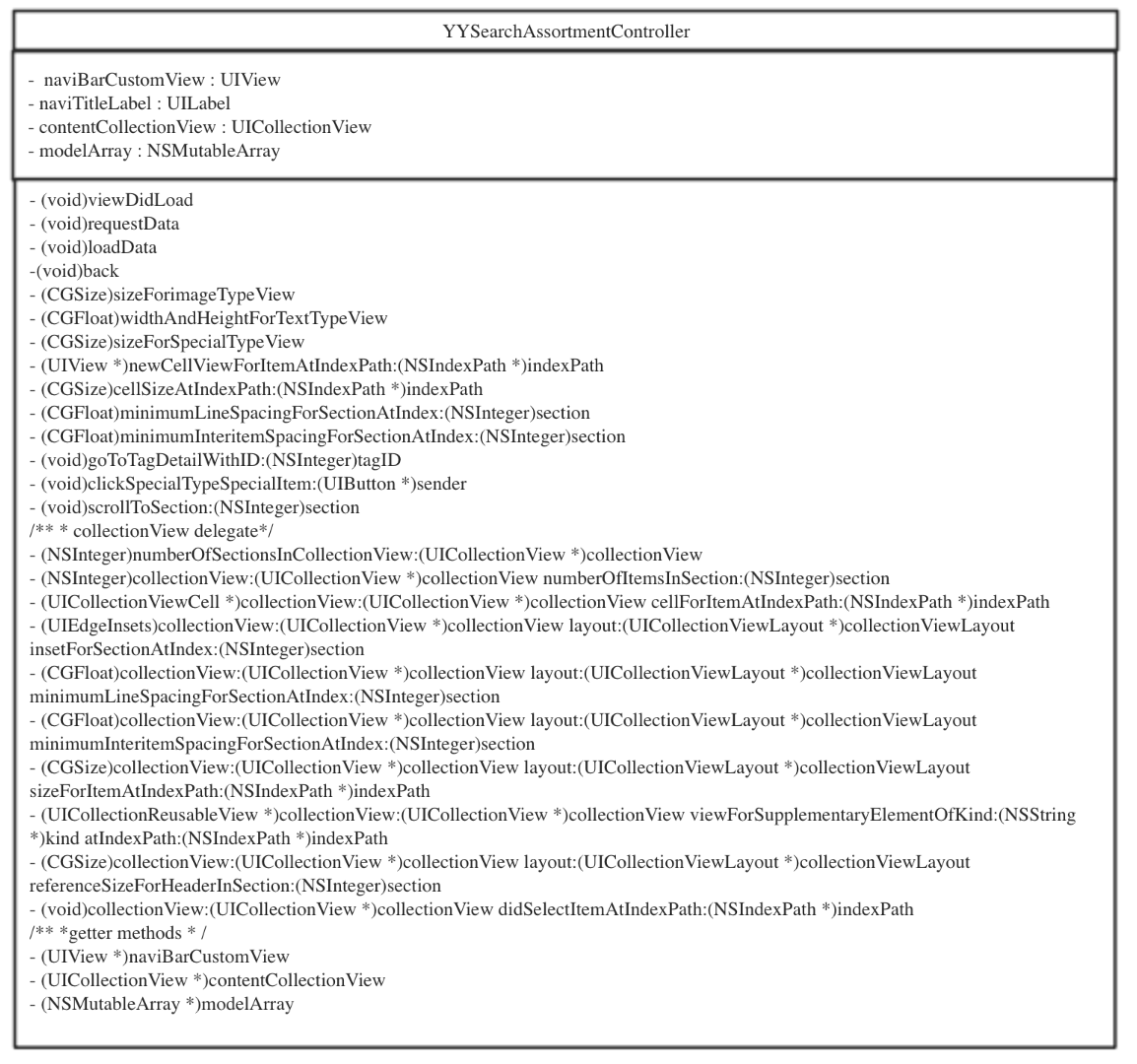
图x-x 为首页头条中视频帖子类cell的模型类图，类方法作用是转化字典数据，因为从服务器中返回的数据默认为字典类型。

view代表类图



图x-x 为首页子分类模块中的cell视图类图

视图控制器 代表类图：



**4.3 重要模块详细设计**

4.3.1 首页流程图