一：概述

该项目是用于解决学生快速查找网课地址，通过搜索网课名称和网课内容关键字进行网课查询，给出的查询信息包括课程所在的所有网课网站地址以及教学老师 学校、课时、网课结束时间。也可以在登录页面后，点击想学的内容，系统会自动给出用户想学习的内容有关的网课地址信息

二：目的

本系统设计说明书是关于校园网课查询系统的设计，主要描述了系统运行环境设计，包括系统运行环境结构设计、软件平台配置清单；描述了系统结构设计，包括软件总体结构设计、软件技术架构设计、软件功能结构设计、软件功能模块清单；描述了功能模块设计。 本系统设计说明书依托于“网课查询系统”项目前期的选题报告、原型设计、以及需求设计说明书作为编写的依据，按照未来实际的开发环境和生产环境进行设计。 本系统设计说明书可以为之后的开发和部署提供参考

读者对象：

（1）产品经理：产品经理维护并在之后的开发中保持对文档的完善，并依据此文档进行项目管理、任务分配、产品验收。

（2）设计师：设计师可根据本文档中相应的功能模块进行界面 UI 设计和图片美化。

（3）开发人员：开发员通过该文档了解到系统设计，以此来进行编码。

（4）测试人员：根据本文档编写测试用例，并对产品进行功能性测试和非功能性测试。

（5）推广人员：通过本文档了解预期产品的功能和使用场景及方式。

（6）用户：了解预期产品的使用环境和功能模块，并与需求分析人员一起对需求进行协商。

（7）其他人员：可以据此文档了解此产品的系统设计和功能。

三：开发环境

开发软件:

apache-tomcat-7.0.104-windows-x64

eclipse-jee-luna-SR2-win32-x86_64

mysql6.7

AxureRP-Pro8.0-Chinese

jdk-8-64 位

四. 架构设计

本系统基于 ssh 架构完成

SSH 为 struts+spring+hibernate 的一个集成框架，是目前较流行的一种 JAVA Web 应用程序开源框架。

Struts

Struts 对 Model，View 和 Controller 都提供了对应的组件。

ActionServlet，这个类是 Struts 的核心控制器，负责拦截来自用户的请求。

Action，这个类通常由用户提供，该控制器负责接收来自 ActionServlet 的请求，并根据该请求调用模型的业务逻辑方法处理请求，并将处理结果返回给 JSP 页面显示。

Model 部分：

由 ActionForm 和 JavaBean 组成，其中 ActionForm 用于封装用户的请求参数，封装成 ActionForm 对象，该对象被 ActionServlet 转发给 Action，Action 根据 ActionForm 里面的请求参数处理用户的请求。

JavaBean 则封装了底层的业务逻辑，包括数据库访问等。

View 部分：

该部分采用 JSP（或 HTML、PHP……）实现。

Struts 提供了丰富的标签库，通过标签库可以减少脚本的使用，自定义的标签库可以实现与 Model 的有效交互，并增加了现实功能。对应上图的 JSP 部分。

Controller 组件：

Controller 组件有两个部分组成——系统核心控制器，业务逻辑控制器。

系统核心控制器，对应上图的 ActionServlet。该控制器由 Struts 框架提供，继承 HttpServlet 类，因此可以配置成标注的 Servlet。该控制器负责拦截所有的 HTTP 请求，然后根据用户请求决定是否要转给业务逻辑控制器。

业务逻辑控制器，负责处理用户请求，本身不具备处理能力，而是调用 Model 来完成处理。对应 Action 部分。

Spring

Spring 是一个开源框架，它由 Rod Johnson 创建。它是为了解决企业应用开发的复杂性而创建的。Spring 使用基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而，Spring 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何 Java 应用都可以从 Spring 中受益。

目的：解决企业应用开发的复杂性

功能：使用基本的 JavaBean 代替 EJB，并提供了更多的企业应用功能

范围：任何 Java 应用

简单来说，Spring 是一个轻量级的控制反转(IoC)和面向切面(AOP)的容器框架。

轻量——从大小与开销两方面而言 Spring 都是轻量的。完整的 Spring 框架可以在一个大小只有 1MB 多的 JAR 文件里发布。并且 Spring 所需的处理开销也是微不足道的。此外，Spring 是非侵入式的：典型地，Spring 应用中的对象不依赖于 Spring 的特定类。

控制反转——Spring 通过一种称作控制反转（IoC）的技术促进了松耦合。当应用了 IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。你可以认为 IoC 与 JNDI 相反——不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。

面向切面——Spring 提供了面向切面编程的丰富支持，允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和事务（transaction）管理）进行内聚性的开发。应用对象只实现它们应该做的——完成业务逻辑——仅此而已。它们并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。

容器——Spring 包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器，你可以配置你的每个 bean 如何被创建——基于一个可配置原型（prototype），你的 bean 可以创建一个单独的实例或者每次需要时都生成一个新的实例——以及它们是如何相互关联的。然而，Spring 不应该被混同于传统的重量级的 EJB 容器，它们经常是庞大与笨重的，难以使用。

框架——Spring 可以将简单的组件配置、组合成为复杂的应用。在 Spring 中，应用对象被声明式地组合，典型地是在一个 XML 文件里。Spring 也提供了很多基础功能（事务管理、持久化框架集成等等），将应用逻辑的开发留给了你。

所有 Spring 的这些特征使你能够编写更干净、更可管理、并且更易于测试的代码。它们也为 Spring 中的各种模块提供了基础支持。

Hibernate

Hibernate 是一个开放源代码的对象关系映射框架，它对 JDBC 进行了非常轻量级的对象封装，使得 Java 程序员可以随心所欲的使用对象编程思维来操纵数据库。 Hibernate 可以应用在任何使用 JDBC 的场合，既可以在 Java 的客户端程序使用，也可以在 Servlet/JSP 的 Web 应用中使用，最具革命意义的是，Hibernate 可以在应用 EJB 的 J2EE 架构中取代 CMP，完成数据持久化的重任。

Hibernate 的核心接口一共有 5 个，分别为:Session、SessionFactory、Transaction、Query 和 Configuration。这 5 个核心接口在任何开发中都会用到。通过这些接口，不仅可以对持久化对象进行存取，还能够进行事务控制。下面对这五个核心接口分别加以介绍。

·Session 接口:Session 接口负责执行被持久化对象的 CRUD 操作(CRUD 的任务是完成与数据库的交流，包含了很多常见的 SQL 语句。)。但需要注意的是 Session 对象是非线程安全的。同时，Hibernate 的 session 不同于 JSP 应用中的 HttpSession。这里当使用 session 这个术语时，其实指的是 Hibernate 中的 session，而以后会将 HttpSession 对象称为用户 session。

·SessionFactory 接口:SessionFactory 接口负责初始化 Hibernate。它充当数据存储源的代理，并负责创建 Session 对象。这里用到了工厂模式。需要注意的是 SessionFactory 并不是轻量级的，因为一般情况下，一个项目通常只需要一个 SessionFactory 就够，当需要操作多个数据库时，可以为每个数据库指定一个 SessionFactory。

·Configuration 接口:Configuration 接口负责配置并启动 Hibernate，创建 SessionFactory 对象。在 Hibernate 的启动的过程中，Configuration 类的实例首先定位映射文档位置、读取配置，然后创建 SessionFactory 对象。

·Transaction 接口:Transaction 接口负责事务相关的操作。它是可选的，开发人员也可以设计编写自己的底层事务处理代码。

·Query 和 Criteria 接口:Query 和 Criteria 接口负责执行各种数据库查询。它可以使用 HQL 语言或 SQL 语句两种表达方式。

架构体系

1、在表示层中，首先通过 JSP 页面实现交互界面，负责传送请求(Request)和接收响应(Response)，然后 Struts 根据配置文件(struts-config.xml)将 ActionServlet 接收到的 Request 委派给相应的 Action 处理。

2、在业务层中，管理服务组件的 Spring IoC 容器负责向 Action 提供业务模型(Model)组件和该组件的协作对象数据处理(DAO)组件完成业务逻辑，并提供事务处理、缓冲池等容器组件以提升系统性能和保证数据的完整性。

3、在持久层中，则依赖于 Hibernate 的对象化映射和数据库交互，处理 DAO 组件请求的数据，并返回处理结果。

五：功能模块设计

非功能性需求

*性能：在非高峰时间屏幕响应时间在 3 秒内，高峰时间屏幕响应时间在 5 秒内，系统可以满足 500 人同时在线使用

*系统可靠性：系统具有较高的可靠性 7*24 小时可以使用，在系统出错时会报告详细错误信息给后台管理员

*可扩展性：可实现负载均衡，日后若信息量较大，则系统可相应增加服务器实现扩展

*易用性：系统界面美观简洁，功能操作一目了然，用户很容易找到他们期望进行的各种操作

*安全性：未经授权的用户禁止登陆，系统管理员可以对用户进行特定的操作

程序模块及功能描述

* 用户在反馈模块中输入反馈内容，可提交至后台，后台人员进行处理。

* 用户在详情模块中可以查看开发团队、版本号、软件信息等信息。

* 用户在设置中可以查看用户隐私说明。

* 在功能模块中，用户可以设置一些例如开启状态栏通知等权限，程序对设备上的相关功能进行设置

性能

*数据精度：数据不出错，保留较 高精度

*响应时间：具有一定的实时性， 1S 以内能接受到数据

*适应性：能灵活适应不同系统 和浏览器