****

**WEB 高级编程**

**课程作业**

**Web高级编程大作业设计方案**

学生姓名 张宇 学号 16203132

专 业 软件工程 班级 162031班

作业项目名称 学生选课系统

指导教师及职称 张恒峰 讲师

开课学期 2018 至 2019 学年 第 2 学

[1 需求分析 2](#_Toc9369189)

[1.1 系统简介 2](#_Toc9369190)

[1.2 用例分析 2](#_Toc9369191)

[1.2.1 用例与角色 2](#_Toc9369192)

[1.4 参考文献 3](#_Toc9369193)

[2 系统设计 3](#_Toc9369194)

[2.1 实体关系建模 3](#_Toc9369195)

[2.2 数据库设计 4](#_Toc9369196)

[2.2 系统开发环境与关键技术 5](#_Toc9369197)

[2.3 界面设计 5](#_Toc9369198)

[2.4 …… 5](#_Toc9369199)

# 1 需求分析

## 1.1需求概述

学生选课系统是一个学校不可缺少的部分，它的内容对于学校的决策者和管理者来说都至关重要，所以学生选课系统应该能够为用户提供充足的信息和快捷的査询手段。

## 1.2 用例分析与描述

## 1.2.1 用例分析

本学生选课系统需要满足来自四方面的需求，这三个方面分别是学生、教师、教务和机房。

学生的需求是査询课程，查询课表，查询成绩；

教师对选课系统学生选课情况进行操作，提交开课信息，申请实验，成绩录入；

教务需要审核开课信息，审核选课信息，安排授课时间、地点。

机房需要管理机房信息、安排实验。

图片包含 文字, 地图

描述已自动生成通过上述分析，可得到系统的用例图如下：

## 1.2.1 用例描述

|  |
| --- |
| 用例名称：申请选课 |
| 描述：学生通过本系统选择课程 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:学生 |
| 前置条件：学生已登录 |
| 主事件流:   1. 学生成功登录系统 2. 选择申请课程功能 3. 选择目标课程 4. 点击选课   A1：选课人数达到上限  A2：选课时间与已有课程产生冲突   1. 选课结束 |
| 其他事件流：  A1: 选课人数达到上限  A2：选课时间与已有课程产生冲突 |
| 后置条件：将向光选课信息添加到数据库里 |
| 特殊需求：无 |

|  |
| --- |
| 用例名称：查询课表 |
| 描述：学生通过系统查询课表 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:学生 |
| 前置条件：学生已登录 |
| 主事件流:  1.学生成功登录系统  2.选择查询课表功能  A1：未选择课程，无法显示课表  5.查询结束 |
| 其他事件流：  A1：未选择课程，无法显示课表 |
| 后置条件： |
| 特殊需求： |

|  |
| --- |
| 用例名称：录入学生成绩 |
| 描述：教授通过本系统录入学生的成绩 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:教授 |
| 前置条件：教授已登录 |
| 主事件流:   1. 教授成功登录本系统 2. 选择录入学生成绩功能   A1：未开设课程   1. 录入成绩 2. 结束 |
| 其他事件流：  A1：未开设课程 |
| 后置条件： |
| 特殊需求： |

|  |
| --- |
| 用例名称：安排授课时间和地点 |
| 描述：教务处通过本系统安排课程的授课时间和地点 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:教务 |
| 前置条件：成功登录 |
| 主事件流:   1. 成功登录系统 2. 选择安排授课时间和地点按钮 3. 选择需要安排的课程 4. 安排时间和地点   A1:时间冲突  A2：地点冲突  5.安排结束 |
| 其他事件流：  A1:时间冲突  A2：地点冲突 |
| 后置条件：将安排的信息储存到数据库中 |
| 特殊需求： |

|  |
| --- |
| 用例名称：审核选课信息 |
| 描述：教务处通过本系统审核学生的选课信息 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:教务处 |
| 前置条件：成功登录 |
| 主事件流:   1. 成功登录系统后 2. 选择审核学生选课信息按钮 3. 进行审核 4. 结束审核 |
| 其他事件流： 无 |
| 后置条件：无 |
| 特殊需求：无 |

|  |
| --- |
| 用例名称：安排实验 |
| 描述：机房通过本系统接受到安排试演的需求 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:机房 |
| 前置条件：已登录 |
| 主事件流:   1. 机房成功登录系统 2. 选择安排实验按钮 3. 根据信息安排实验   A1:实验时间或地点冲突   1. 结束 |
| 其他事件流：  A1:实验时间或地点冲突 |
| 后置条件： |
| 特殊需求： |

## 1.4 参考文献

[1] 齐志昌，谭庆平，宁洪.软件工程(第二版)[M].北京.高等教育出版社.2004

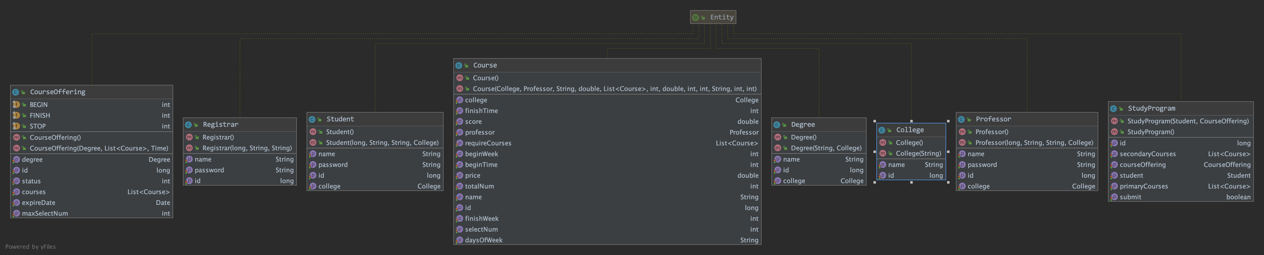
[2] 王珊，萨师煊.数据库系统概论（第四版）[M].北京.高等教育出版社.2006

# 2 系统设计

* 1. 实体关系建模
  2. 数据库设计
  3. 界面设计
     1. 页面跳转关系
     2. 页面设计
  4. ……

## 2.1 实体关系建模

实体关系建模是系统设计的第一步，以宠物诊所来说，「宠物」（Pet）和「兽医」（Vet）都是经常出现的名词，它们也都是从需求描述中识别出来的「实体」。

s

## 2.2 数据库设计

根据以上给出的实体关系模型可以得到以下数据库物理物理模型。

图片包含 黑色, 屏幕截图, 墙壁

描述已自动生成

表格详细情况如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| college | id | bigint(20) | auto\_increment | NULL | 否 |  |
| college | name | varchar(255) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| course | id | bigint(20) | auto\_increment | NULL | 否 |  |
| course | begin\_time | int(11) |  | NULL | 否 |  |
| course | begin\_week | int(11) |  | NULL | 否 |  |
| course | days\_of\_week | varchar(255) |  | NULL | 否 |  |
| course | finish\_time | int(11) |  | NULL | 否 |  |
| course | finish\_week | int(11) |  | NULL | 否 |  |
| course | name | varchar(255) |  | NULL | 否 |  |
| course | price | double |  | NULL | 否 |  |
| course | score | double |  | NULL | 否 |  |
| course | select\_num | int(11) |  | NULL | 是 |  |
| course | total\_num | int(11) |  | NULL | 否 |  |
| course | college\_id | bigint(20) |  | NULL | 否 |  |
| course | professor\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| course\_offering | id | bigint(20) | auto\_increment | NULL | 否 |  |
| course\_offering | expire\_date | datetime(6) |  | NULL | 否 |  |
| course\_offering | status | int(11) |  | NULL | 否 |  |
| course\_offering | degree\_id | bigint(20) |  | NULL | 否 |  |
| course\_offering | max\_select\_num | int(11) |  | NULL | 是 |  |
| course\_offering\_courses | course\_offering\_id | bigint(20) |  | NULL | 否 |  |
| course\_offering\_courses | courses\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| course\_require\_courses | course\_id | bigint(20) |  | NULL | 否 |  |
| course\_require\_courses | require\_courses\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| degree | id | bigint(20) | auto\_increment | NULL | 否 |  |
| degree | name | varchar(255) |  | NULL | 否 |  |
| degree | college\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| professor | id | bigint(20) |  | NULL | 否 |  |
| professor | name | varchar(255) |  | NULL | 否 |  |
| professor | password | varchar(255) |  | NULL | 否 |  |
| professor | college\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| registrar | id | bigint(20) |  | NULL | 否 |  |
| registrar | name | varchar(255) |  | NULL | 否 |  |
| registrar | password | varchar(255) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| student | id | bigint(20) |  | NULL | 否 |  |
| student | name | varchar(255) |  | NULL | 否 |  |
| student | password | varchar(255) |  | NULL | 否 |  |
| student | college\_id | bigint(20) |  | NULL | 是 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| study\_program | id | bigint(20) | auto\_increment | NULL | 否 |  |
| study\_program | submit | bit(1) |  | NULL | 否 |  |
| study\_program | course\_offering\_id | bigint(20) |  | NULL | 否 |  |
| study\_program | student\_id | bigint(20) |  | NULL | 是 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| study\_program\_primary\_courses | study\_program\_id | bigint(20) |  | NULL | 否 |  |
| study\_program\_primary\_courses | primary\_courses\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| study\_program\_secondary\_courses | study\_program\_id | bigint(20) |  | NULL | 否 |  |
| study\_program\_secondary\_courses | secondary\_courses\_id | bigint(20) |  | NULL | 否 |  |

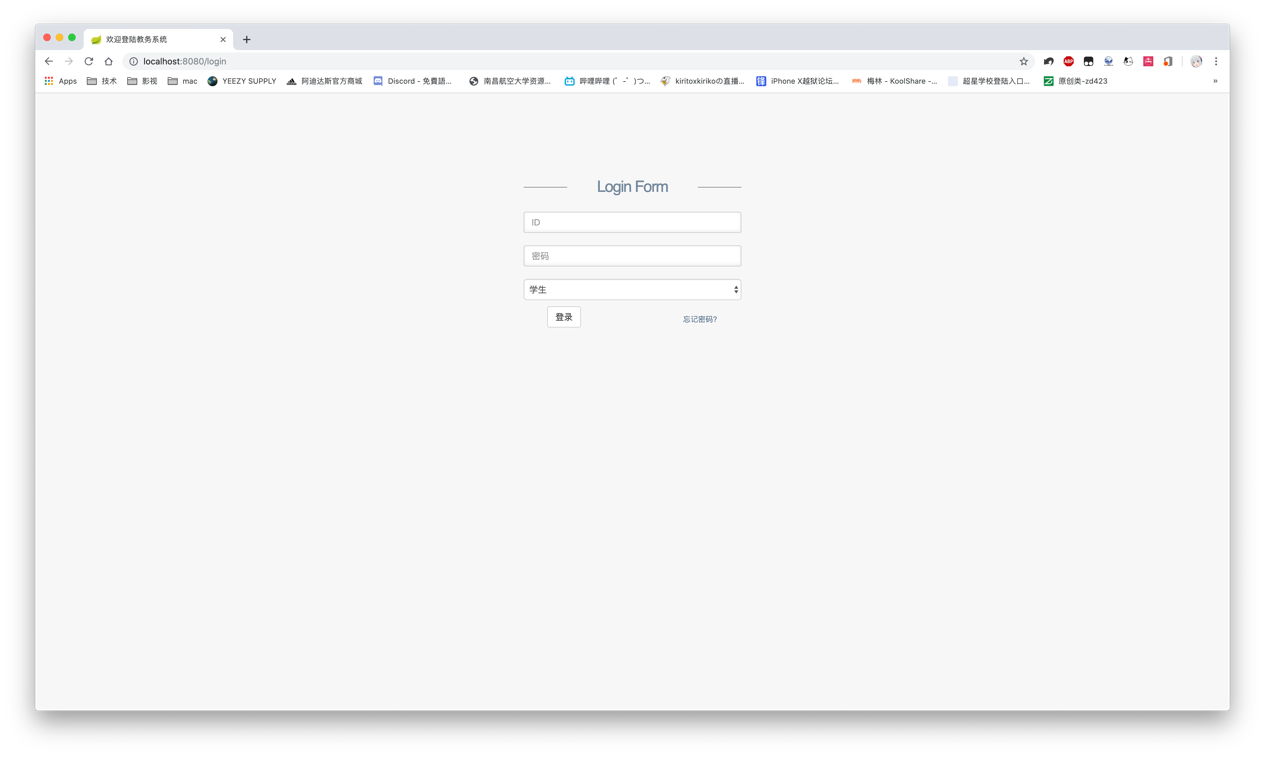
## 2.3 系统开发环境与关键技术

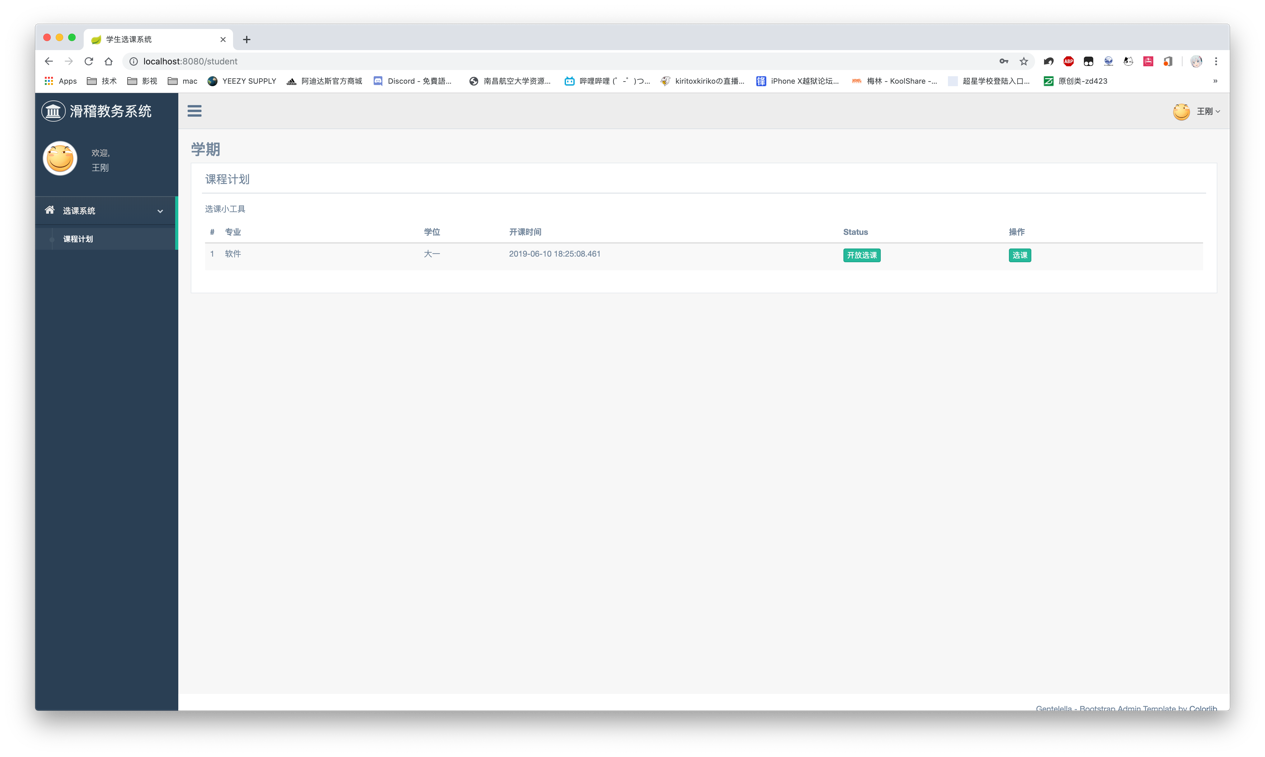
….. The developer will need to obtain the following tools externally, all of which are freely available:

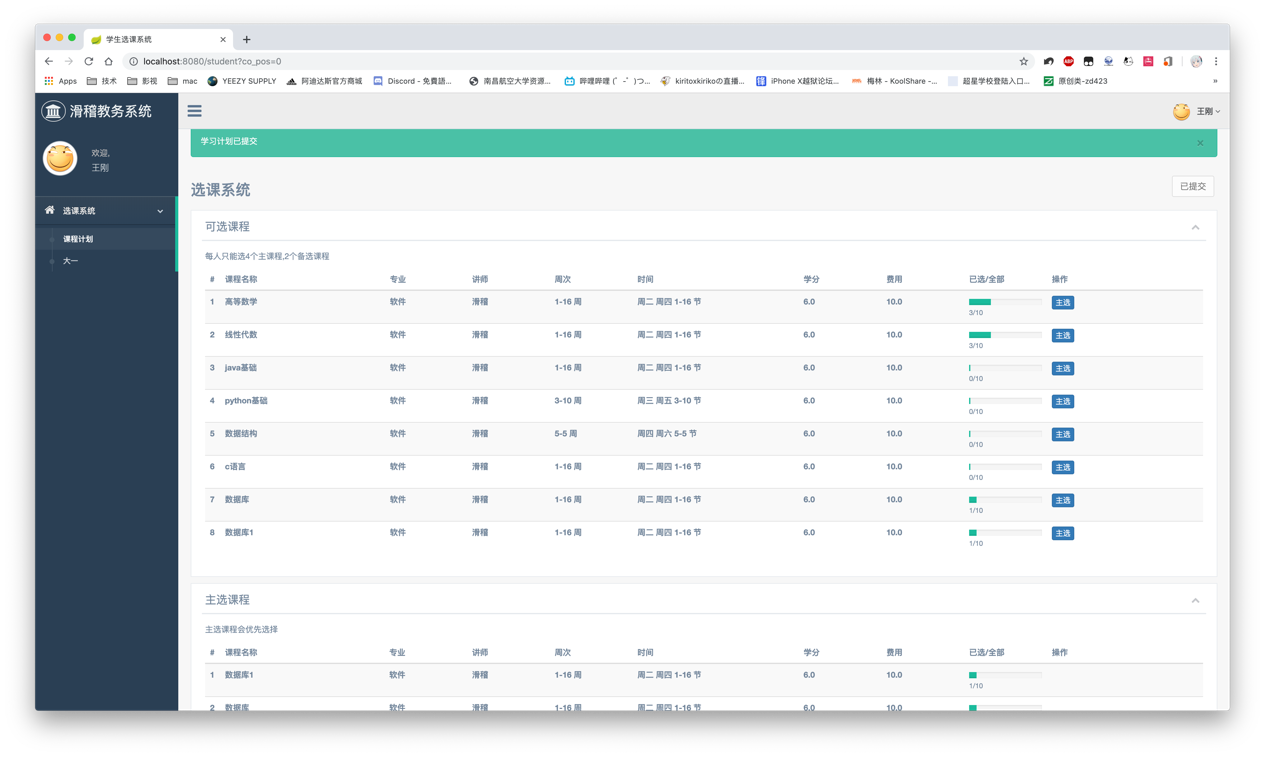
* Java SDK 1.5.x
* Ant 1.7.x
* Tomcat 5.x.x, or some other web application container
* JUnit 4.4 - needed to run the tests
* …

(Optional) MySQL 5.x with MySQL Connector/J 5.x

## 2.4 界面设计







## 2.5 ……

对系统的系统的可扩展性、可靠性设计等展开讨论

# 1 系统分析

## 1.1 需求概述

项目背景

The Spring Framework is a collection of small, well-focused, loosely coupled Java frameworks that can be used independently or collectively to build industrial strength applications of many different types. The PetClinic sample application is designed to show how the Spring application frameworks can be used to build simple, but powerful database-oriented applications. It will demonstrate the use of Spring's core functionality. …

Since the purpose of the sample application is tutorial in nature, the implementation presented here will of course provide only a small subset of the functionality that would be needed by a real world version of a PetClinic application. …

需求说明

The application requirement is for an information system that is accessible through a web browser. The users of the application are employees of the clinic who in the course of their work need to view and manage information regarding the veterinarians, the clients, and their pets. …

Here is a list of requirements:

* Support only one type of users – clinic employees
* Clinic employees can manage vets and their specialties
* Clinic employees can manage owners, pets, and visits
* Clinic employees can view all user activities
* …

业务规则

Here is a list of business rules:

* An owner may not have multiple pets with the same case-insensitive name.
* …

## 1.2 用例分析与描述

### 1.2.1 用例分析

先分析系统有哪些角色，然后说明这些角色能够完成哪些操作，最后形成用例图

Clinic Employee:

* View a list of veterinarians and their specialties
* View information pertaining to a pet owner
* Update the information pertaining to a pet owner
* Add a new pet owner to the system
* View information pertaining to a pet
* Update the information pertaining to a pet
* Add a new pet to the system
* View information pertaining to a pet's visitation history
* Add information pertaining to a visit to the pet's visitation history
* …

Vet: …

Owner: …

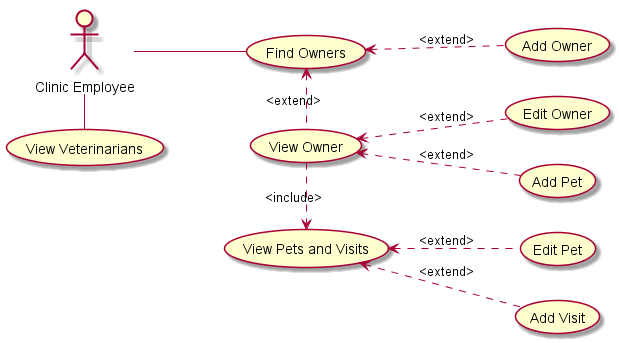


图1.1 宠物医院系统用例图

### 1.3.2 用例描述

给出每个用例的说明与详细流程

|  |  |
| --- | --- |
| 用例编号 | UC01 |
| 用例名称 |  |
| 简要说明 |  |
| 执行者 |  |
| 前置条件 |  |
| 后置条件 |  |
| 基本事件流 |  |
| 扩展事件流 |  |

（每一章都必须另起一页）

# 2 系统设计

## 2.1 多层结构设计

说明如何进行分层，各层运用哪些技术，各层分别由哪些package、页面以及其他资源构成，画出系统的组件图

The Presentation Layer is implemented as a Java EE Web Application and provides a very thin and concise Model-View-Controller type user interface to the Business and Persistence Layers. …

Since the PetClinic application is all about database access and there is very little business logic in the application outside of that, there is no separation of the primary Business and Persistence Layer API's. While this design technique should not be used for an application with more complex business logic, it is acceptable here because all of the non-persistence related business rules have been implemented in business objects and have not leaked into the Persistence Layer. The most important facet of the design is that the Business and Persistence Layers are COMPLETELY independent of the Presentation Layer. …

The Persistence Layer can be configured to use either HSQL or MySQL with any one of the following data access technologies aided by infrastructure provided by Spring:

* JDBC
* ……

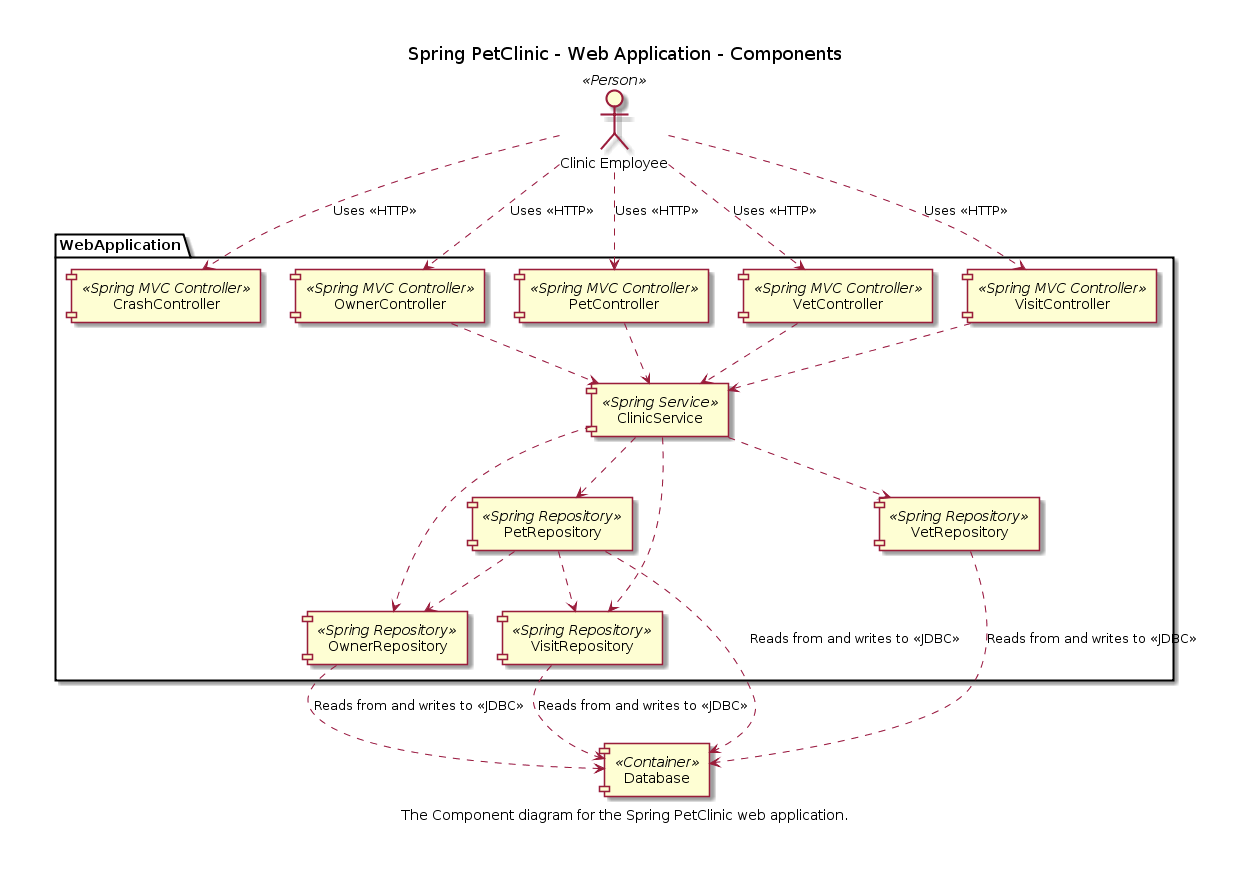


图2.1 宠物医院系统…….

## 2.2 问题域设计

### 2.2.1 域模型设计

分析系统中的实体并给出系统的领域模型

以宠物诊所来说，「宠物」（Pet）和「兽医」（Vet）都是经常出现的名词，它们也都是从需求描述中识别出来的「实体」。……

PetClinic is a fairly simple application with only seven models. The following chart shows all the models and relationships among them. …

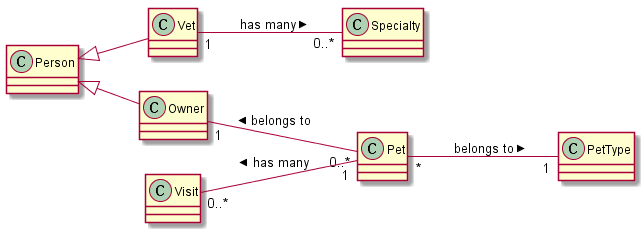


图2.2 宠物医院系统域模型

### 2.2.2 Service接口设计

通过活动图、状态图、通信图或顺序图分析得到Service接口的方法，并画出相关类图。

## 2.3 持久化设计

### 2.3.1 数据库设计

说明给出的数据库类型，并以表格形式给出数据库表结构

The sample application uses a relational database for data storage. Support has been provided for a choice of 1 of 2 database selections, MySQL or HypersonicSQL. HypersonicSQL version 1.8.0 is the default choice and a copy is provided with the application. It is possible to easily configure the application to use either database. Configuration details are provided in the Developer Instructions section.

The following is an overview of the database schema used in PetClinic. Detailed field descriptions can be found in the "initDB.txt" SQL script files in the database-specific "db" sub-directories. All "id" key fields are of Java type int. …

### 2.3.2 数据访问设计

分析并给出数据库配置以及repository类的类图。

## 2.4 界面设计

### 2.4.1 页面链接关系

（分析页面之间的连接关系以及url所对应的Controller方法。）

The following table shows routes created for resource pets:

表2.1 URL对应功能说明（pets）

|  |  |  |  |
| --- | --- | --- | --- |
| HTTP verb | URL | action（Controller类的方法） | used for |
| GET | /pets/new | add | return an HTML form for creating a new pet |
| POST | /pets/new | create | create a new pet |
| GET | /pets/1/edit | edit | return an HTML form for editing a pet |
| POST | /pets/1/edit | update | update a specific pet |

The models will look like this:

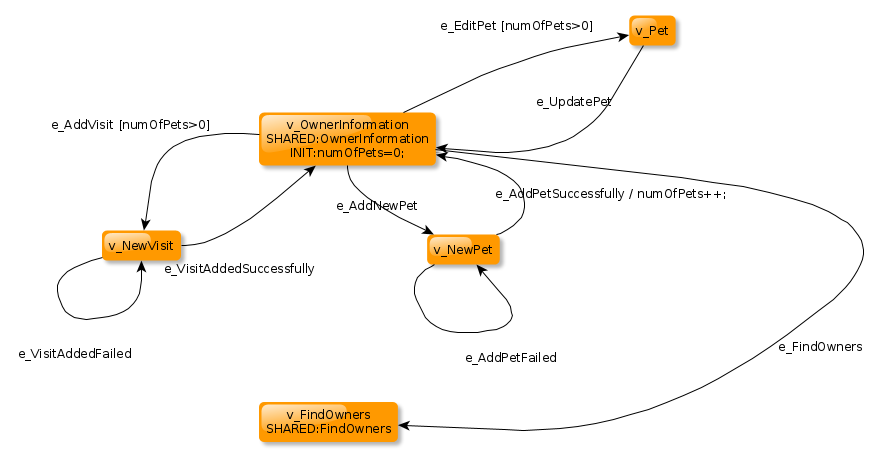


图2.2 宠物医院系统…….

### 2.4.2 页面设计

（给出页面设计及其操作流程。）

You can then access petclinic here: http://localhost:8080/

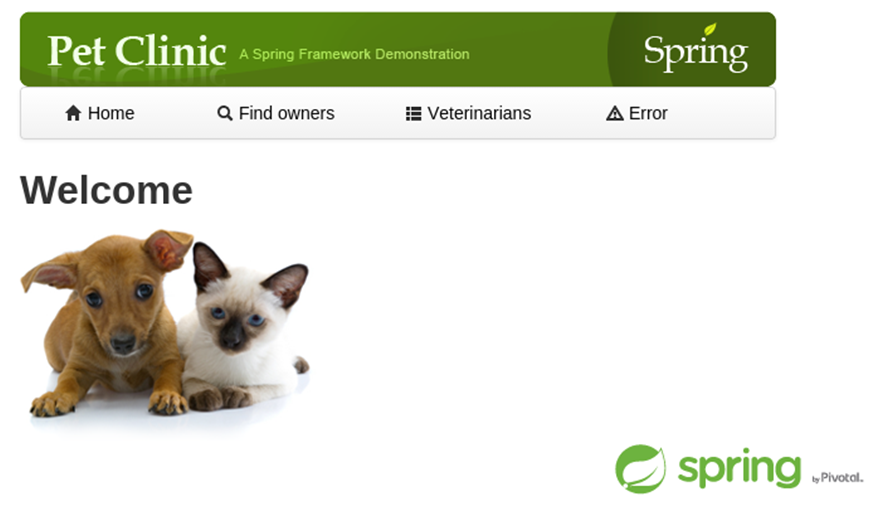


图2.3 宠物医院系统…….