基于JavaEE和Android的大学生成长档案的设计

摘要

第一章 绪论

1.1引言

第二章 系统总体设计

第三章 Android客户端设计

第四章 JAVA EE服务端设计

本文所述基于Java EE和Android的大学生成长档案的设计，服务端主要负责了全部的业务逻辑。服务器端设计符合MVC设计思想。系统架构为当下流行的SSH 的轻量级JAVA EE 架构，即为Spring ,Spring MVC , Hibernate，所使用的的数据库为Mysql，web容器为Tomact。

4.1：相关技术介绍

本平台开发采用的是SSH整合技术，SSH指Spring，Spring MVC ,Hibernate三大框架的整合。

4.1.1 Spring

Spring的基于控制反转（IOC）的核心机制，基于面向切面编程（AOP）实务管理，整合多种持久层技术（Hibernate），整合MVC框架（Spring MVC）等。Spring贯穿于项目中的持久层，业务层和表现层等。是java应用的整体解决方案。有以下优点：

1、用Spring的IOC容器，将对象之间的依赖关系交给Spring，降低组件之间的耦合性，让我们更专注于应用逻辑

2、对第三方主流持久层框架很好的集成支持，简化底层数据库的访问流程。如Hibernate ,JPA, Mybatis等

3、.Spring的高度可开放性，并不强制依赖于Spring，开发者可以自由选择Spring部分或全部

4.1.2 Spring MVC

Spring MVC 是当今最主流的web mvc 框架，提供了MVC(模型 - 视图 - 控制器)架构和用于开发灵活和松散耦合的Web应用程序的组件。 MVC模式导致应用程序的不同方面(输入逻辑，业务逻辑和UI逻辑)分离，同时提供这些元素之间的松散耦合。

模型（Model）：封装了应用程序数据，由POJO类组成。

视图(View)：负责渲染模型数据，由浏览器，手机端等展示。

控制器(**Controller**)：负责处理用户请求并构建适当的模型，并将其传递给视图进行渲染。

4.1.3 Hibernate

目前主流的数据库是关系型的数据库，比如：Mysql,Oracle。二维表结构易于理解；使用sql语言操作方便；丰富的完整性减小数据冗余，易于维护；支持表关联，实现复杂的查询。目前主流的编程语言是面向对象型的编程语言，比如：JAVA。面向对象建模和操作；继承和多态；相对面向过程容易理解，简单易用等。所以当我们使用面向对象的编程语言进行开发时，也是基于对象的分析，设计和开发，但是当实现数据持久化，进行数据库访问时，又不得不采用关系型数据库的访问方式，因此，编程语言和数据库之间具有设计模式上的区别和技术上的差异，发展不协调。ORM（**Object Relational Mapping**）框架应运而生，可以将关系型数据库转为面向对象模型，方便开发和维护。

Hibernate是典型的java EE持久层的解决方案，也是流行的ORM框架之一。Hibernate是对JDBC的进一步封装，从对象（Object）映射到关系(Relation)，再从关系映射到对象。这样在操作数据库的时候，不需要和复杂的SQL打交道，只需要操作po（persistant object），实现对数据表中的数据进行增，删，改，查等操作。