**《家教通》架构设计说明书1.0**

架构设计是需求分析到软件实现的桥梁，也是决定软件质量的关键。

架构的本质是呈现三大能力：即系统如何面向最终用户提供支撑能力、如何面向外部系统提供交互能力、如何面向企业数据提供处理能力。

1. **文档概述**

**1.1背景与目标**

家教信息管理平台目前处于形式多样化、经营尚未规范化的两难境地。各地家教网星罗棋布，几乎遍布国内每个地级市，其中每个城市都有不同的各类中小型家教网站，主要以家庭教育网站为主，且每个家庭网站注册时需要收取一定的费用。

本家教管理系统是建立在互联网基础上的一种信息交流平台，面向有需要的学生与家长。无论是找家教的学生还是做家教的老师，都可以通过该平台进行信息方面的交流。任何学生、老师都可以自由地查看、搜索、发布该家教管理系统的信息。该平台是区别于传统的家教中介的服务平台。

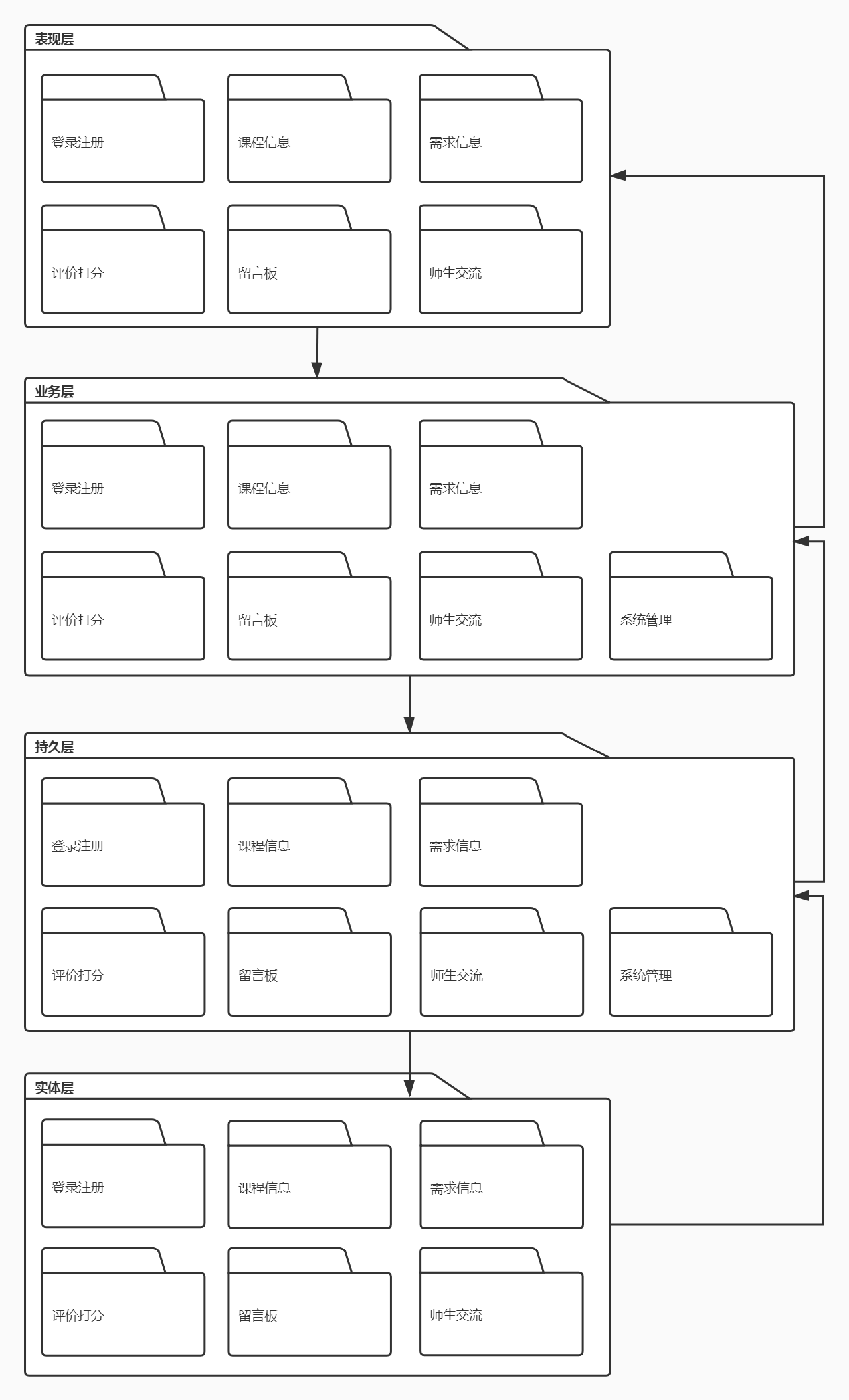
**1.2参考文档**

《家教通》--软件开发计划书1.0 《家教通》系统需求规格说明书2.0 《家教通》进度计划1.0 《家教通》需求分析模型--E-R图1.0

1. **逻辑架构**

系统如何面向最终用户提供支撑能力：这一点是要从系统自身的能力来看，即本系统到底应该具备哪些功能，各功能间如何协作以满足支撑最终用户的使用，其实就是系统的功能架构或逻辑架构，和系统从功能粒度上划分了几个功能模块或子系统，各模块或子系统之间的内部接口关系如何等问题。在纵向维度上，随着架构设计理念的不断发展， 逻辑架构模型从最初的展示-数据两层模型，到展示-逻辑-数据（所谓的MVC）三层模型,甚至到展示-调用接口-逻辑-数据接口-数据五层模型，不同层次表明系统内部设计的精细程度，因此在逻辑架构设计中也需要针对实际情况加上这种分层设计的内容。

本系统的逻辑视图如下所示：



其中各功能模块详细说明见文档《家教通》系统需求规格说明书2.0

\*

|  |  |  |
| --- | --- | --- |
| 包 | 用途 | 实现 |
| 表现层 | 展示系统主要信息，获取来自用户的数据输入提供给后端业务处理，显示后端系统处理返回的数据。 | bootstrap前端框架，主要使用HTML5和CSS技术，和实现前端验证等网页行为的JavaScript脚本。 |
| 业务层 | 实现系统中各个模块具体的业务逻辑，在系统各个模块之间传递数据。 | Java面向对象编程设计，包含控制层注解、接口编程和相关XML文件配置等。 |
| 持久层 | 处理与后端数据存储区的交互，包括数据库、消息处理系统等。 | 使用mybatis定制化 SQL查询语句，配置简单的 XML，将接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java对象)映射成数据库中的记录。 |
| 实体层 | 创建数据表中各个实体的对象供业务逻辑实现相关功能。 | 创建数据表中的实体Java对象，供业务层逻辑实现和持久层数据库读取。 |

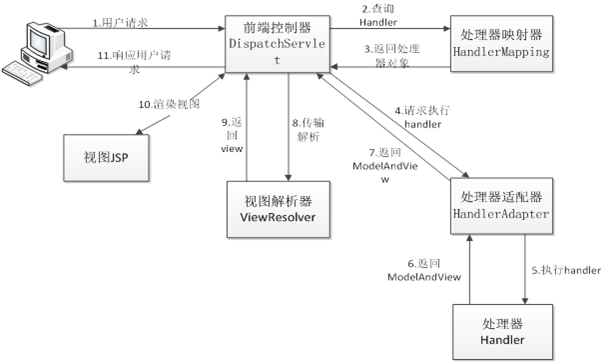
包内关系待更新。

1. **技术架构**

本系统使用spring+springMVC+mybatis（SSM）后端框架开发，框架主要分为控制层（controller），业务层（service），数据访问层（DAO，持久层），实体层（pojo），主要运行流程为：

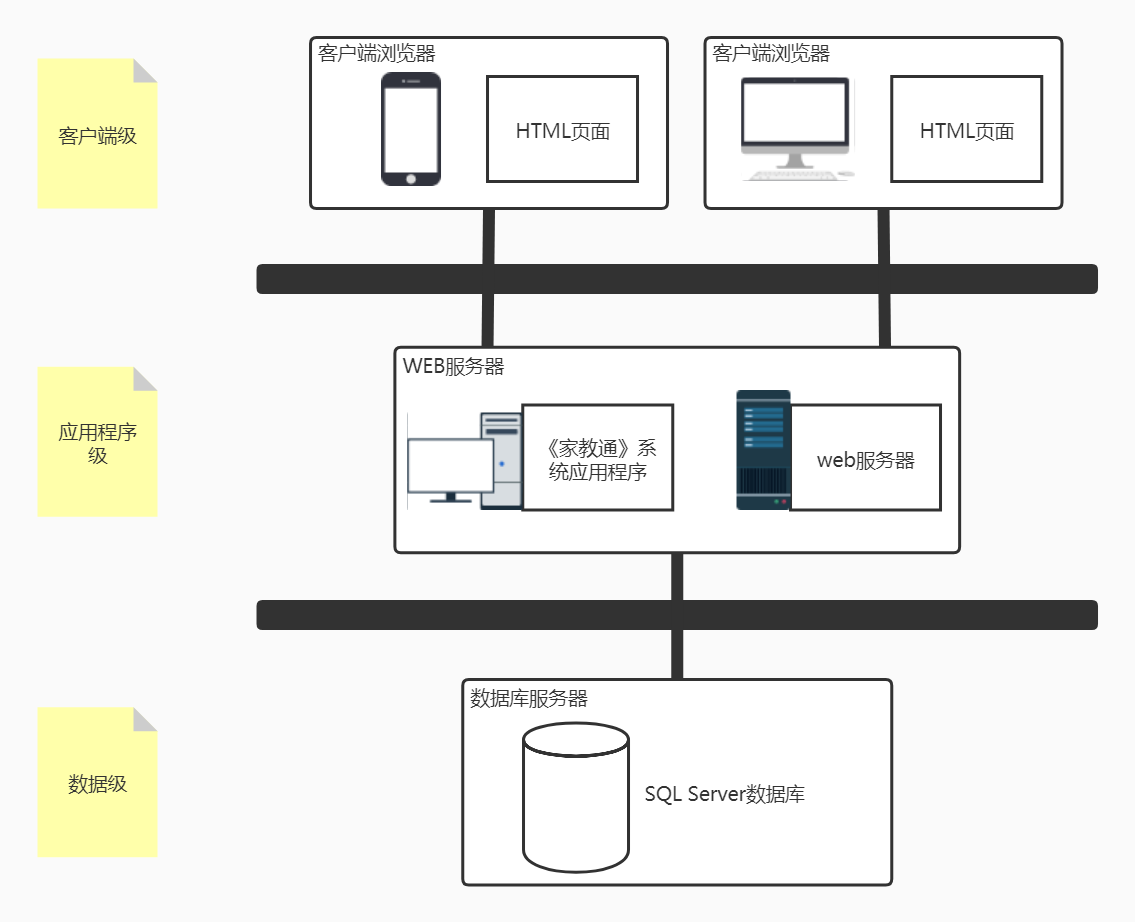
1. 1.前端页面jsp（view）发送请求
2. 2.通过核心控制器DispatcherServlet调用请求解析器HandlendMapping对请求进行解析，通过映射关系匹配到Controller层
3. 3.在控制层调用业务逻辑层（service）进行具体业务操作，业务逻辑层调用数据持久层（DAO）获取相关数据，然后返回控制层，请求完成获取一个结果，在控制层设置一个要跳转的视图
4. 4.核心控制器调用视图解析器ViewResolver解析视图，匹配相应的页面实现页面跳转

SSM框架主要处理流程如下图所示：



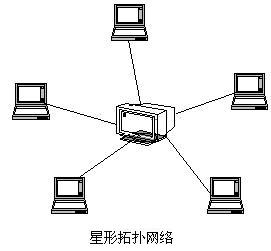
1. **部署架构**
2. 系统如何部署，网络拓扑上有何要求，对硬件服务器有何要求，需要几台，是否需要优化服务器参数；

系统主要围绕三个物理级构造应用程序：客户端、应用程序和数据库。如下图所示：



其中客户端级主要是用户的移动端，如手机、平板等，和电脑端网页，应用程序级为服务器的搭建和系统部署，数据级为数据库服务器的搭建。

在项目上线运行的初级阶段，网络结构主要使用星形拓扑结构。星型结构表现为一个中心，多个分节点。它结构简单，连接方便，管理和维护都相对容易，而且扩展性强，网络延迟时间较小，传输误差低，中心无故障，一般网络没问题。但是中心故障，网络就出问题，同时共享能力差，通信线路利用率不高。此时只需要使用一台服务器设备，对该服务器设备性能要求比较高。星形拓扑结构如下图所示：



在物理部署上，这三层结构可以部署在同一个物理机器上，但是随着系统业务的发展，必然要对已经分层的模块分离部署，将三层结构分别部署在不同的服务器上，使得网站可以承受更大的用户访问量。本系统目前将三层结构部署再同一台物理设备上。

1. **非功能性设计**

**5.1性能**

一般来说，服务器性能主要由三部分决定：服务器的硬件配置；服务器操作系统；应用程序。

服务器硬件配置主要有：CPU;RAM;网络带宽；硬盘。一般来说CPU的数量越多，主频越高，那服务器的性能也就会越好。内存的大小也是影响服务器性能的一个重要因素。内存太小，系统进程要被阻塞，应用程序会变得缓慢，甚至是失去响应；如果太大的话，也是造成一种浪费。相对其他硬件资源，网络带宽在价格上更贵，这需要我们合理预估服务器的可服务器能力，需要占用的带宽资源。硬盘自带的读写缓存大小，对于硬盘读写速度至关重要。读写速度快的硬盘，通常都带有较大的读写缓存。

当前主流的服务器操作系统则主要分为:Windows server、UNIX、Linux、NetWare这四大阵容。不同的系统有不同的特点，要根据情况来判断。

对于应用程序本身的设计来说，只要选用合适的数据结构和内存操作方式，那么至少在性能上差距不是很大，避免对常用内存数据的多次创建和释放。也可增加服务器的性能。

**5.2高可用性**

高可用性是指通过设计减少系统不能有效运行的时间。在业务系统的非功能设计中中高可用性是一个重要的组成部分。

系统不能有效运行主要指业务系统停机，业务系统的停机包括计划内停机和计划外停机两种。计划内停机是指管理员有意识安排的停机，比如在对硬件进行升级、对软件进行升级、更换损坏的硬件、对系统进行备份、系统的新功能测试时，可能需要停止业务系统的运行。计划外停机是指非人为的、因外界环境变化而引起的停机，比如当硬件出现重大故障、应用程序停止运行、计算机所运行的机房环境遭到灾难性的破坏时所引起的业务系统停止运行。而大部分情况下，业务系统的停机都是计划内停机，少有因为故障、外界因素等的计划外停机。

为使得系统具备高可用性，可以从硬件和软件两个方面来考虑：

硬件方面：多CPU，在计算机中至少有两颗CPU。通过多CPU，不仅可以保证多个进程能够真正实现并发执行，而且可以保证当一颗CPU出现故障时，整个计算机仍然能够运行；冗余电源，在计算机中至少有两个电源，当其中一个电源出现故障时，其他电源仍然能够向计算机持续供电；冗余网卡，在计算机中一般有两个以上网卡，用户可以通过任何一个网卡访问计算机中的服务。在网卡上可以指定可“漂移”的IP地址，当一个网卡出现故障时，这个网卡上的IP地址就漂移到另外一个网卡上。通过多个网卡还可以实现数据流量的均衡。

软件方面：稳定的文件系统，文件系统的稳定，对整个计算机系统的稳定起着至关重要的作用。目前各种操作系统一般都通过日志机制来保证文件系统的稳定性，如AIX的JFS/JFS2文件系统、HP-UX的VxFS文件系统、Solaris的ZFS文件系统、Linux的ext3文件系统等；数据的备份，备份无疑是保证数据安全的一种重要措施，当因为系统出现故障而导致数据丢失时，利用备份可以对数据进行恢复。

**5.3可扩展性**

可扩展性是指应用程序应对需求变化的扩展能力，少量修改就可以，无须重构或者重建。程序中设计业务层接口和接口实现类，当有需求变化时，只需修改相应的接口及接口实现方法即可。

**5.4可维护性**

可维护性是衡量一个系统的可修复(恢复)性和可改进性的难易程度。所谓可修复性是指在系统发生故障后能够排除(或抑制)故障予以修复，并返回到原来正常运行状态的可能性。而可改进性则是系统具有接受对现有功能的改进，增加新功能的可能性。

应用程序使用严格的编程规范和代码静态检查工具，在保证程序能够稳定运行的情况下提供故障快速检查并排除能力。

**5.5安全性**

为防止他人恶意的使用，或偷窥系统内部实现，保证数据安全和代码可控的运行，需要考虑系统的安全性。

应用程序中的具体实现使用final、static、private定义各对象，数据库中加密用户关键信息如登录密码等。

**5.6可移植性**

可移植性并不是指所写的程序不作修改就可以在任何计算机上运行，而是指当条件有变化时，程序无需作很多修改就可运行。

使程序可移植的第一步就是使用标准库函数；第二步是尽可能使所写的程序适用于所有的编译程序，而不是仅仅适用于你所使用的编译程序；第三步是把不可移植的代码分离出来。

**5.7UI设计**

UI设计是指对软件的人机交互、操作逻辑、界面美观的整体设计,在保证系统功能完整运行的情况下，还应做到用户界面的友好和美观。

用户界面设计原则：界面直观，对用户透明，用户在使用软件后对界面上的功能按钮一目了然，对大部分功能无需查看用户手册就可以方便的使用本系统；始终以用户为功能实现的中心，用户界面应当体现由用户来控制应用如何工作、如何响应，而不是把开发人员的意愿作为界面的运行流程。

1. **其他说明**
2. 本应用程序符合党中央的各项规定，始终以新时代特色社会主义思想为引导。