**五、软件架构设计**

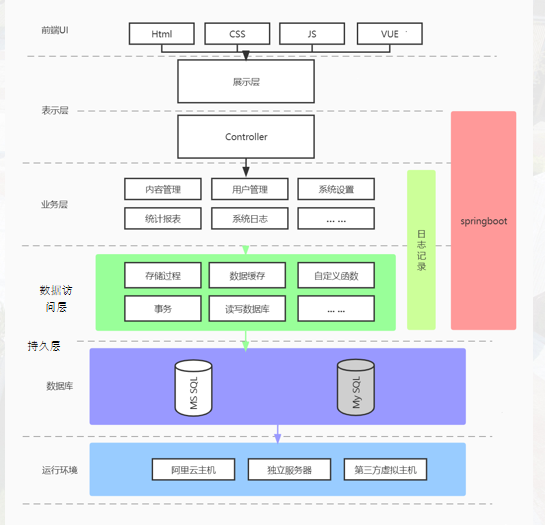
**1、整体架构**

整体采用3层B/S架构设计，包括表示层（可细分为展示层和控制层），业务（逻辑）层，持久层（可细分为数据访问层和数据库层）；后端基于Springboot的三层框架（Controller/Service/Dao） + Pojo层实现整合，涵盖了控制层、业务逻辑层、数据访问层。

（1）表示层：用于接收客户端请求，向客户端响应结果。其中包括了展示层和控制层，展示层用于结果的呈现，控制层Controller负责接收请求。我们以MVC为设计模式，Model用于数据的封装，View用于数据展示，Controller用于处理程序逻辑，并基于VUE框架，使用CSS、HTML、JS实现对应的页面设计。

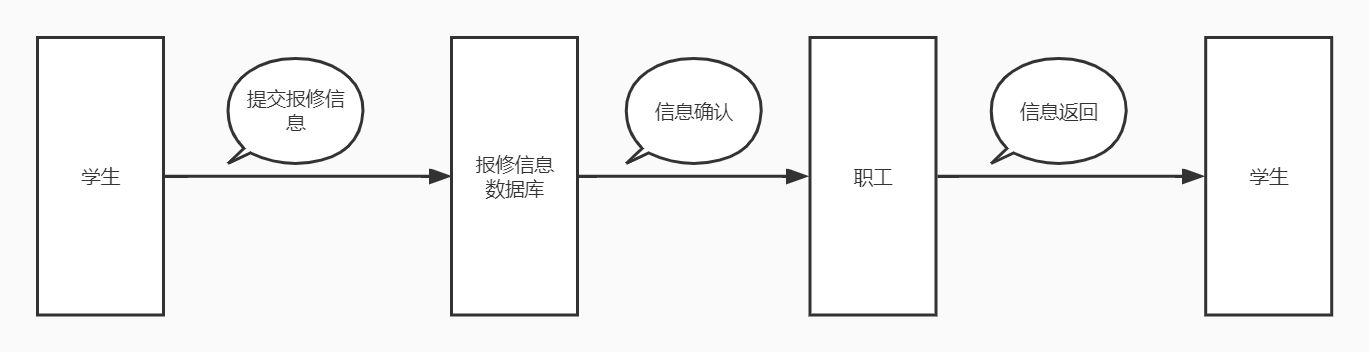
（2）业务层：Service 层。它负责业务逻辑处理，与软件需求相关。

（3）持久层：负责数据持久化，包括数据库层和数据访问层（Dao）数据库是对数据进行持久化的载体，数据访问层是业务层和持久层交互的接口，业务层通过数据访问层将数据持久化到数据库中。



**2、事件控制**

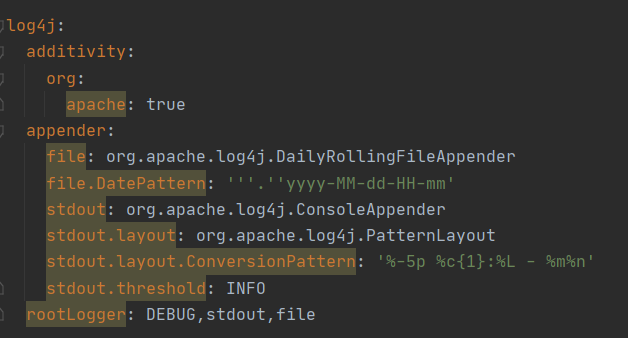
报修检修遵循事件体系结构风格，学生作为事件源，发布报修信息产生事件，在宿管访问的时候触发了访问、互动等响应。采用有独立事件派遣模块的派遣机制，学生发布报修信息后存入数据库，宿舍报修信息数据库通过全广播模式触发访客的响应。



**3、质量属性提升策略**

**（1）可用性：**设置异常捕获和抛出机制，在用户异常操作时保证系统正常运行。

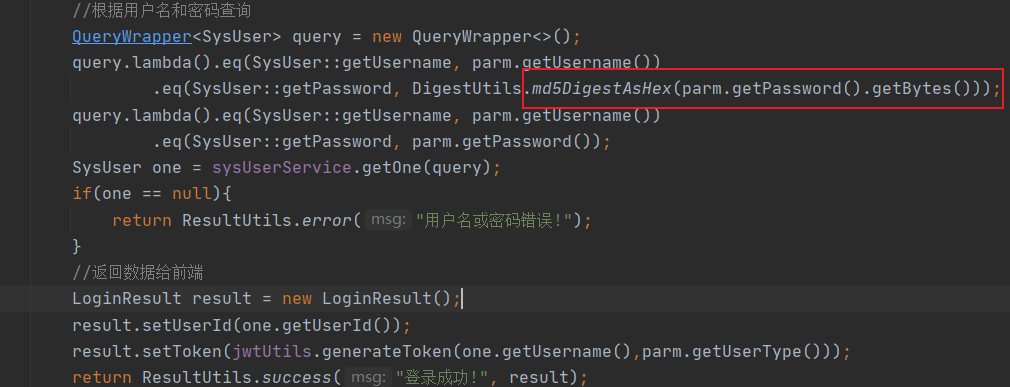
**（2）可修改性：**程序模块化，不同的模块之间透明，便于快速定位和修改，并减少对其他模块的影响；设置统一的配置文件，减少需要修改的代码量。



**（3）性能：**优化算法和程序代码，减少计算复杂程度和资源占用量；增加可用资源。

**（4）可测试性：**黑盒测试，快速检验功能完整性；白盒测试，利用JUnit、Postman、LoadRunner、JMeter等技术或软件对代码、接口、性能进行测试。

**（5）安全性：**身份授权和认证，密码加密，数据备份。



**（6）易用性：**简化用户界面，简化操作流程，增加系统提示和引导。