**1.3设计策略**

（1）关键需求决定架构。关键需求决定架构有两个方面的涵义：一方面，功能需求与非功能需求数量众多，应该控制架构设计时需要详细分析的用例个数；另一方面，不同非功能需求之间往往具有相互制约性，应该权衡非功能需求之间的关系，找到影响架构的重点非功能需求。关键需求决定架构的策略有利于集中精力深入分析最为重要的需求。当架构工程师把全部精力扑在相对较少的关键需求上时，可以更为深入的分析这些需求，有利于得到透彻的认识，从而设计出合理的架构。

（2）全面认识需求。从不同级别、不同类别梳理列表归纳总结，建立跟踪矩阵与影响分析表。一方面，将所有需求从不同的级别（组织级、用户级、开发级）分层梳理列表归纳总结，建立跟踪矩阵，既避免因遗漏需求而造成软件系统达不到要求，也避免开发人员一厢情愿也为用户制造没有实际意义的无用功能，同时，有助于软件架构设计人员对软件系统要求有全面的认识，清晰理清需求之间、需求与软件系统目的目标与核心业务、商业理由间的因果关系。另一方面，将所有需求划分为不同的类型（功能需求或用例、质量属性、约束与限制）进行梳理列表归纳总结，建立影响分析表，找出不同需求类型之间的相互支持、相互制约关系的影响，并在深思熟虑之后作出合适的需求权衡和取舍。

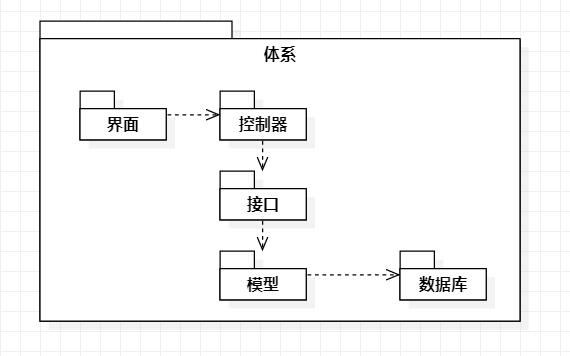
（3）多视图探寻架构。一次只从某一立场、某一视角出发围绕少数概念和技术展开，并分析对其它部分、其它立场视角分析结果的关系与影响。

（4）尽早验证架构。架构设计是现代软件开发中最为关键的一环，架构设计是否合理将直接影响到软件系统最终是否成功。架构验证必须精挑细选能够触发主要设计决策参与执行的、或有较高技术风险的、或最影响用户满意度的一些功能，来尽早验证架构方案并作出调整，避免后期出现大规模返工。

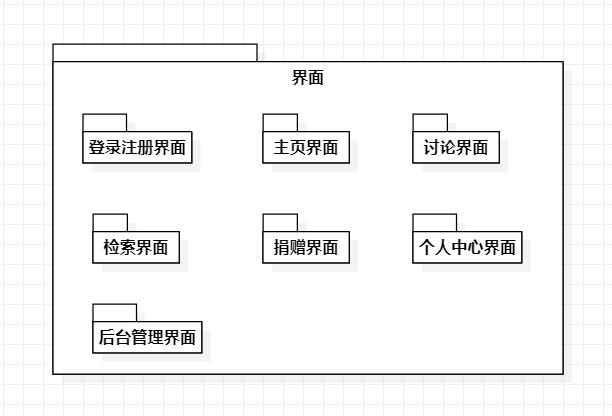
**1.4系统架构**

（2）包图

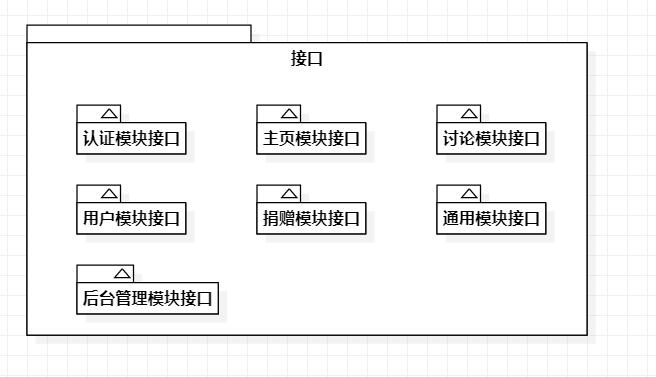
系统包图



界面包图

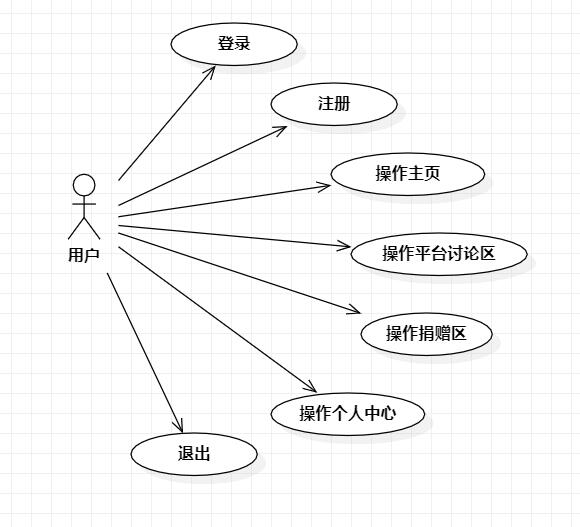


接口包图



（3）模型视图

用户用例图



管理员用例图

