# 架构的分析设计

## 系统特点：

1、各个功能点之间联系不紧密（除了一个审批单据十分特殊）；各功能点之间的联系主要就是共享数据库中的数据。

2、系统分为两大部分：客户端（client）与服务器端（server），两者有明显分层关系

Client：与系统用户交互，接受用户的各种查询、输入，将其进行基本处理（包括格式检查、权限检查）后向服务器发起数据库操作请求

Server：可以同时与多个Client交互，可以直接接触数据库，处理其操作请求（之前应该再次权限检查），并给出反馈

## 分层方式：

Client自底向上：GUI呈示层，输入处理层，功能内核层

Server自底向上：功能内核层，数据层（可能使用专业数据库）

Client与Server之间存在连接件RMI

各层详细功能：

GUI呈示层：接受系统用户的查询、输入；向系统用户在GUI上显示系统反馈

输入处理层：监听GUI呈示层的状态，在用户进行操作时传递查询请求到功能内核，用户进行输入时进行格式检查、正确性检查（期间可能向上请求查询数据），直接给出驳回反馈或将输入向上传递；向GUI呈示层传递来自功能内核层的高级反馈

功能内核层：接受输入处理层的数据请求并反馈，接受输入处理层的操作请求检查后（主要为权限检查）包装并向Server传递，向输入处理层传递数据、反馈；向Server请求数据，调用Server的功能，接受Server的数据与反馈

功能内核层：拥有与数据库交互的许多功能，呈示为实际功能需求的形式供下层调用；处理来自数据库的数据，解释包装后传递给下层

数据层：希望由专业数据库管理，使用专业数据库的接口进行信息的增删改查

## 选择分层的理由：

1、系统两部分之间具有明显分层特征，Client与Server各自也较容易划分层级

2、逻辑清晰，易于分析。

3、由于分层方式将软件分块较多，团队分工容易，而且每个人还可以选择自己比较拿手的部分写

4、项目前期需求分析较为详细，所以层次之间接口定义可能较为稳定，将来修改的可能性低

5、之前没有尝试过使用分层设计开发项目，本着学习课程的目的，希望能够尝试分层设计