评审确认部分

1.满足质量属性

1.1可修改性：因为我们已经提出了预期的改变并对相关部分进行了单独的封装，所以未来的修改将会变得容易简单。但不排除有对其他部分有大的改动的可能性。并且我们在设计体系架构时降低了耦合度，这也为我们之后的改动提供了很大的可修改空间。同时我们也增加了单元间的接口数量来保证各单元的独立性。

1.2性能：由于我们的软件只是个新的软件，并不具有很大的用户基础，所以我们并不能实现快速反应，大的吞吐量以及大的负载。但我们采取先到先服务的方式，来保证所有请求都被处理。因为我们并没有很大的用户基础，所以采取这种方式可能带来的高优先级请求不能得到及时响应的影响会大大降低。

1.3安全性：因为我们的软件体系架构规模小，所以并不具有很高的免疫力。但我们通过了功能分段的方式来最大限度的降低外来攻击对系统的影响，并通过快速的响应来实现体系的高弹性。

1.4可靠性：我们设计了主动故障检测的机制，迅速识别出异常并处理。而如果故障发生时，我们会通过备份来实现故障的恢复，从而提高可靠性。

1.5健壮性：我们设计了一套互相怀疑的策略，但是由于我们的团队并不强大，所以我们系统架构的健壮性还是有待提高的。

1.6易使用性：我们通过设计了简洁的图形界面以及命令按键来使用户立马明白我们的功能设计，并且我们的设计满足可以检测和响应任何预期的用户输入。具有很可靠的易使用性。

1. 体系结构评估和改进

2.1测量设计质量：我们通过测量各个类之间的内聚度和耦合度来判断各个类之间的交互属性，判断各个类之间的依赖关系，得到结果之后再设计减少错误的可能性。

2.2故障树分析：我们会对出现的实效建立故障树，并通过建立割集树来判断可能引起失效的组合。从而改正故障，或增加构件和条件来预防引起故障执行的条件。

2.3安全性分析：我们严格执行了软件特征化、威胁分析、漏洞评估、风险可能性决策、风险影响决策、风险缓解计划六步来评估体系结构设计是否满足了系统的安全需要。从而提高了我们的安全性。

2.4权衡分析：我们通过比较表的方式来直观的选出最适合的设计方案。

2.5成本效益分析：我们会对改动的经济成本做出预算，并对带来的经济效益做出评估，通过我们判断的成本与效益来实现较高的投资回报。

2.6原型化：我们采取了抛弃型原型的技术，来从过往系统的设计错误中得到受益。