|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **备选方案分析表一**  备选方案标识号： A001 制作日期：2014-4-8 制作人：张舒贤、张天、薛笑雨  备选方案基本描述：富客户端的桌面应用+有服务器的C/S+MVC风格+推送消息+实名制注册   1. 备选方案软件类型   富客户端的桌面应用   1. 部署约束   有服务器的C/S架构  主要特征：C/S架构即客户端和服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到Client端和Server端来实现，降低了系统的通讯开销。目前大多数应用软件系统都是Client/Server形式的两层结构，内部的和外部的用户都可以访问新的和现有的应用系统，通过现有应用系统中的逻辑可以扩展出新的应用系统。  服务器端的特征：  1.被动的角色（从）。  2.等待来自用户端的要求。  3.处理要求并传回结果。  客户端的特征：  1.主动的角色（主）。  2.发送要求。  3.等待直到收到回应。   |  |  | | --- | --- | | **主要优点** | **主要缺点** | | 1. 客户端响应速度快 2. 应用服务器运行数据负荷较轻 3. 数据的储存管理功能较为透明 | 1. 对[客户端](http://baike.baidu.com/view/930.htm)的操作系统一般也会有限制 2. 客户端需要安装专用的客户端软件 3. 需要建立两端的实时数据同步，维护成本高昂 4. 传统的C/S结构的软件需要针对不同的操作系统系统开发不同版本的软件 |  1. 架构风格   MVC风格  主要特征：  将系统功能组织为模型、视图和控制三个部件。模型封装了系统的数据和状态信息，实现业务逻辑，对外提供数据服务和执行业务逻辑。视图封装了用户交互，提供业务展现，接收用户行为。控制封装了系统的控制逻辑，根据用户行为调用需要执行的业务逻辑和数据更新，并且根据执行后的系统状态决定后续的业务展现。   |  |  | | --- | --- | | **主要优点** | **主要缺点** | | 1. 易开发性，设计机制清晰，易于团队分布式开发测试和维护。是业务程序员精于业务逻辑，而前端的程序员专注于界面开发 2. 视图和控制的可修改性，提供了高内聚、低耦合的代码功能块，利用代码的复用，并且不同模块的内部的变化不会影响其他模块   3. 易于实时性较高的系统开发的特性。 | 1. 复杂，MVC将用户任务分解成了表现、控制和模型三个部分，增加了系统的复杂性，不利于理解任务实现。 2. 模型修改困难。视图和控制都要依赖于模型，因此，模型难以修改。 |  1. 和架构相关的技术细节 2. 系统登录方式：实名制注册登录   主要优点：方便好友间的识别。  缺点：实名制注册需要识别注册信息，增加了工作量。   1. 消息的发布和接收方式：推送   主要优点：可以满足消息发布和接收的要求。  缺点：需要设计用于推送的功能组件；一有新消息，处于接收状态的另一端接收消息。 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **备选方案分析表二**  备选方案标识号： A002 制作日期：2014-4-8 制作人：臧晓杰、徐敏  备选方案基本描述：富客户端的桌面应用+C/S部署+账号注册+信息推送   1. 备选方案软件类型   富客户端的桌面应用  采用富客户端的桌面应用能更充分地利用客户资源，有更好的响应能力、丰富的UI功能，改善用户体验。   1. 部署约束   C/S架构  主要特征：  可用于局域网、广域网和因特网，多用于Internet。  提供独立的客户端程序。  可以选择2层（Layer/Tier）的架构。  工作负载在服务器和客户端之间进行分配。客户端通常是一个“需要什么”的程序，一般要求服务（包括打印、信息检索和数据库访问）；服务器则是“提供什么”的程序，负责处理客户机的要求。  一个服务器可以向多个不同的客户端提供服务。   |  |  | | --- | --- | | **主要优点** | **主要缺点** | | 1. 数据具有很高的稳定性和可靠性：数据在服务器，支持更安全的控制 2. 集中数据的数据访问：数据只存储在服务器，比起其他架构风格，提供更容易管理数据的访问和更新功能 3. 系统性能较好：一般都是非开放的 4. 易于维护：系统的职责分布在已知服务器上，确保客户端不需要了解由于服务器的维修、升级或搬迁而造成的影响。 | 1. 系统扩展难度较大 2. 可移植性差 3. 客户端和服务器只能工作在特定的软、硬件平台，开发和维护成本较高 |  1. 架构风格   分层风格+O-O  主要特征：  分解软件的关注点到某个特定的层（或组）。  分解软件的责任到独立的可重复使用的对象，每个对象包含有关的数据和行为。   |  |  | | --- | --- | | **主要优点** | **主要缺点** | | 1. 易于将一个复杂系统逐步分解，简单化 2. 开发工作的重用性、继承性高，降低重复工作量 3. 结构清晰明了 4. 易于测试。具有定义明确的层接口提高了可测试性 5. 缩短了开发周期。 | 1. 效率低，上层服务依赖于最底层，相关数据可能需要若干次的传递转化 2. 很难明确精准地划分层次 3. 类和继承等特点使得程序会多很多指针操作来定位函数入口和自身要维护虚拟方法表等额外的工作 |  1. 和架构相关的技术细节 2. 系统登录方式：用学号注册   主要优点：符合学生间使用要求，区分度高，简单，易记，易于管理。  缺点：适用人群有局限性，审核难度大。   1. 信息的发布和推送方式：推送   主要优点：可以满足信息发布和接受的要求。  缺点：需要设计用于推送的功能组件；一有新消息，处于接收状态的另一端接收消息。 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **备选方案分析表三**  备选方案标识号：A003 制作日期：2014-04-08 制作人：徐仪萍  备选方案基本描述：富客户端的桌面应用+C/S部署+注册登录+信息推送   1. 备选方案软件类型   桌面应用  该游戏软件作为桌面应用能够更好配置软件运行环境。   1. 部署约束   C/S架构  主要特征：  可用于局域网、广域网和因特网，多用于Internet。  提供独立的客户端程序。  工作负载在服务器和客户端之间进行分配。客户端通常是一个“需要什么”的程序；服务器则是“提供什么”的程序，负责处理客户机的要求。   |  |  | | --- | --- | | **主要优点** | **主要缺点** | | 1. 数据具有很高的稳定性和可靠性：数据在服务器，支持更安全的控制 2. 集中数据的数据访问：数据只存储在服务器，比起其他架构风格，提供更容易管理数据的访问和更新功能 3. 系统性能较好：一般都是非开放的 4. 易于维护：系统的职责分布在已知的几台服务器上，确保客户端不需要了解由于服务的维修，升级或搬迁而造成的影响。 | 1. 系统扩展难度较大 2. 可移植性差 3. 客户端和服务器只能工作在特定的软、硬件平台，开发和维护成本较高 |  1. 架构风格   客户端/服务器   |  |  | | --- | --- | | **主要优点** | **主要缺点** | | 1. 易操作,交互性很好 2. 响应速度快 | 1） 客户端程序设计复杂，环境部署困难  2） 客户机与服务器的通信依赖于网络，可能成为整个系统运作的瓶颈。  3）服务器的负荷过重，难以管理大量的客户机，系统的性能受到很大的影响。 |  1. 和架构相关的技术细节 2. 系统登录方式：注册登录   主要优点：符合对该游戏软件的需求。  缺点：随着玩家人数的增多，对注册信息的审核和登录信息的验证需要的时间可能会加长。   1. 信息发布和接收方式：推送   主要优点：可以满足好友管理的要求。  缺点：需要设计用于推送的功能组件；一有新信息，处于接收状态的另一端接收信息。 |