**Design One T**

**he benefits for design one lies in its simplicity. It incorporates everything necessary, like: different account permission levels; the necessary menus that a user has to interact without exposing too much of the implementation side of things. This means that it becomes a lot easier to prototype the design. This is because it is concise and only describes the necessary processes without worrying about implementations, for example it doesn’t describe how any of the encryption should work. This means that when we come to prototyping this design it becomes a lot easier to fully represent the design in the prototype. However, because the design doesn’t go into the minute details, multiple implementations can have substantial semantic differences. This means that depending on the implementor, the whole application could act differently, due to different interpretations of what to do that isn’t described. For example some implementers could encrypt user data and some couldn’t, meaning that one would be susceptible to attacks and the other wouldn’t. However it does provide a great high level view of all the processes, relations and transmissions that need to take place throughout the application. Design Two This design is great, because it provides extreme detail into how every single process should be implemented. It includes everything from design one, as well as encryption, classroom persistence and security infrastructures. This means that it accounts for much more and is much more secure in general. This is great because it doesn’t leave many design decisions to the implementers and all they have to do is follow the design. If they do this successfully the whole application will be secure and there is less room for error. However, it also means that it is harder to prototype, because there is so much detail needed for a full prototype. However, you can implement a subset of the design in the prototype to enable a quick and brief prototype to demonstrate certain features, although this means that sometimes the prototype won’t accurately represent the design. Conclusion In conclusion, we have decided to go with design two. This is because of all the extra detail the design provides, meaning it accounts for more situations (such as its encryption accounting for man in the middle attacks). This means that we can trust the design more and multiple implementations of the same design should have very little semantic difference, due to the large detail in the design.**

**Tradeoff Analysis**

**权衡分析**

**此设计包含软件里所有必要的组件，此系统依赖于与其他几个子系统的交互并考虑了用户之间的权限区别。首先Classroom Operation system 组件中包含了系统的基本功能，用户可以根据自己的需要来选择课程，教室，虚拟环境，以及语言偏好。其次在File Management system中包含了学生和老师的上传/下载文件的功能，老师有更高的权限能选择访问每个学生的上传文档。其次在Assistant System中包含了Teaching Assistant和AI Assistant来帮助学生解决问题。其次在Management system中包括了整个系统的信息数据管理，管理员可以访问，修改和改变系统数据。最后Extra Package是这个系统的扩展包，可以获得其他组件的数据内存大小扩充。其中整个系统的用户数据也被单独区分储存和分配。这个组件图描述了整个软件的实现的必要过程，在原型中更容易实现。**

**此部署图是基本根据组件图1里的功能来设计的。因为在这个系统中会有很多用户共同使用服务器，所以设计了负载均衡器来防止同时过载并解决了在使用软件时出现**[**高并发**](https://zh.m.wikipedia.org/wiki/%E5%B9%B6%E5%8F%91%E6%80%A7)**和**[**高可用**](https://zh.m.wikipedia.org/wiki/%E9%AB%98%E5%8F%AF%E7%94%A8%E6%80%A7)**从而导致服务器崩溃的问题。在软件中用户信息数据很重要，此设计将数据存储在几个区域服务器群集，每个服务器数据库都会进行定期备份并且定期与每个服务器主数据库同步，提高了系统和服务器的可维护性和稳定性。当数据库出现错误或故障时，储存的备份数据可以用于恢复主数据库。**

**设计2**

**此设计和第一个设计是相似的，因为在一个很多用户一起使用的软件中安全性是很重要的，所以此设计在设计1的基础上还加入了安全性和维护性。在用户访问参加课程或上传/下载文档时可以进行加密和解密来防止数据泄露，从而保护数据。因为加密和解密是比较集中的，所以会增加对系统的维护难度和系统的复杂性从而降低系统的效率。但它确保了信息数据在子系统之间传递时的安全性，因为安全性在教学系统和对用户隐私保护中是非常重要的。**

设计2

**此设计是根据第二个组件图来设计的，为了以确保高度安全的传输数据，服务器在设计1的基础上增加了额外的安全性，维护性的模块。因为数据加密会给网络造成一定的影响，并影响服务器的运行速度。所以将安全性和维护性分配给每个服务器群集从而增加效率。此设计拥有负载均衡器和备份数据库，所以可以既高效又安全的在每个子服务器和与主服务器之间传递数据信息。**

**Tradeoff Analysis：**

**Component Diagram Design 1：**

**This design includes all the necessary components of the software. The system relies on interactions with several other subsystems and takes into account differences in permissions between users. First, the Classroom Operations System component contains the basic functionality of the system. Users can choose courses, classrooms, virtual environments, and language preferences according to their needs. Secondly, the File Management system includes the ability to Upload/Download files for students and teachers. Teachers have a higher degree of access to the files uploaded by each student, with the option to choose. Thirdly, Teaching Assistants and AI Assistants are included in the assistant system to help students solve problems. Second, the management system includes information data management for the entire system, enabling administrators to access, modify and change system data. The last Extra Package is an expansion pack for the system that can obtain data memory size expansions for other components. The user data of the whole system is also stored and distributed separately. This component diagram describes the process necessary to implement the entire software, which is easier to implement in a prototype.**

**Deployment Diagram Design 1:**

**This deployment diagram is designed based on the functionality of the Component in Design 1. Since many users share servers in this system, load-balancers are designed to prevent simultaneous overload and address the high concurrency and high availability issues when using the software, which can cause servers to crash. User information data is very important in software. This design will store the data in several regional server clusters. Each server database will have regular backup and regular synchronization with the main database of each server, improving the maintainability and stability of the system and the servers. In the event of a database error or failure, the stored backup data can be used to restore the primary database.**

**Tradeoff Analysis：**

**Component Diagram Design 2：**

**​ This design is similar to Design 1 in that security is important in software that is used by many users together, so it adds security and maintainability to Design 1. Data can be protected by encryption and decryption to prevent data leakage when users access courses or upload/download documents. Because encryption and decryption are centralized, it increases the difficulty of system maintenance and the complexity of the system, thus reducing its efficiency. However, it ensures the security of information data as it is passed between subsystems, as security is very important in pedagogical systems as well as the protection of user privacy.**

**Deployment Diagram Design 2:**

**This Deployment Diagram design is based on the functionality of the component diagram in Design 2. To ensure highly secure data transmission, the server has added additional security and maintainability modules based on Design 1. Because data encryption has a certain impact on the network and affects the speed of the server. Therefore, security and maintenance are distributed to each server cluster to increase efficiency. The design has a load balancer and a backup database so that data can be efficiently and securely transferred between each child server and the primary server.**