

Example of systems analysis and development: application architecture case study

[Business](#), [Management](#)



Payroll management systems are software's used by organizations to generate pay slips and salary records as per the employee work attendance and factoring in allowances, overtime, loans, advances and other types of deductions. The management software's consist of activities that aid in the completion of work through efficient and organized quick methods.

Generally, they improve the processing of information in a standard prescribed manner.

Payroll management systems fall in the administration section of an organization. The objectives of the administration department involve the generation of pay roll for all employees in an organized and standardized manner. In an organization with a centralized data center and branches across the United States, a pay roll management system is essential as manual management of employee records is difficult if impossible.

Pay roll management system in this scenario can best be developed using the client/server architecture. In this architecture, the system is divided into clients and servers that shares tasks. The client handles the entire system interface tasks such as input data entry, data querying, and screen presentation logic while the server stores the data and provides data access and database system management functions. Generally, the client submits a request such as employee information from the server which carries out the operation and responds to the client. The employee data file is not transferred to the client. It is only the request that is transmitted across the network. Client/server architectures are cost effective as companies have developed platforms that deliver unique combinations using internet protocols and network models.

A two tier client/server design incorporates the user interface on the server and the application logic either on the server or on the client. A three-tier design has the user interface running on the client while the data is stored on the server. In addition, there is the processing of client requests and translation into data access commands in the middle layer between the client and the server. Thus in this architecture, the middle layer is termed as the application logic. Application logic is important in three-tier designs because it provides increased performance on the data server workload. It is typically more efficient and cost effective in large scale systems spanning many organizations with branches. The communication of the data and interface layers is made possible by means of middleware.

Fig. 1 A client/server demonstration of tiers

Middleware

Data communication back and forth the n-tier system is made possible by the middleware. It is essentially defined as the software that mediates application software and network. Through the middleware, the application designer can easily link the human resource department of the pay roll system with to a Web server facilitating access through client computers via the internet or a company network. Likewise, it integrates the web-based applications with the legacy systems. A user will only have to enter the employee ID on the web-based inquiry for the pay roll system to access the pay roll records and generate the results.

An extension of the server in a fully web based application is the cloud.

Cloud computing takes the place of clients task processing. Cloud of computers acts as giant devices that perform the task queries for the users.

<https://assignbuster.com/example-of-systems-analysis-and-development-application-architecture-case-study/>

In this scenario, the cloud computing experience distributes the performance to numerous remote computers found on the cloud thereby, eliminating compatibility issues and promoting scaling.

Distributed database management system (DDBMS)

DDBMS has several advantages in the client/server architecture: First, data stored closer to the users can be easily accessed thereby reducing the network traffic. The new system should be scalable to accommodate the addition of new data without restructuring of the system. In addition, DDBMS allows the storage of data in various locations thereby distributing the risk of catastrophic failures.

Secondly, the architecture poses security concerns in terms of intrusions attacks among others. Since the architecture divides, the system into client and servers, a request by the client will be transmitted to the server through the server or wide area networks. This exposes the system to security risks. In addition, the DDBMS is more complex and difficult to manage thereby increasing the of data management. Controls and standards of data stored in remote locations is not easy to manage as the company will face centralization issues and control of effects of decentralization.

References

Shelly, G. B. (2012). Systems analysis and design (9th ed.). Boston: Thomson Course Technology.

Shivaraju. (2012). Database Systems: Design, Implementation, and Management [With Access Code]. Cengage Learning.

System. (2010). Network and System Sciences , . International Journal of Communications , 815-824.