Architectural patterns and styles

Object-Oriented

**Definition**: A design paradigm based on division of responsibilities for an application or system into individual reusable and self-sufficient objects, each containing the data and the behaviour relevant to the object.

***Reason for inclusion:***

By implementing the very widely used Object-Oriented design strategy, the program will gain a variety of benefits in terms of the **reduced complexity** and **code volume** provided by the **abstraction**, **polymorphism**, **decoupling** and **inheritance** attributes of the architectural pattern.

A system of this complexity would be incredibly hard to coordinate in any system that does not permit object-oriented design. Object-Orientation is a de facto requirement in all but the most high-level systems wherein memory availability is not extremely limited..

***Main Benefits***

* Understandable
* Reusable
* Extensible
* Testable
* Highly Cohesive

Client/Server

**Definition**: Segregates the system into two applications, where the client makes requests to the server. In many cases, the server is a database with application logic represented as stored procedures.

***Reason for inclusion:***

Sensitive and personal data pertaining to the name, emails and other client information must be kept secure. Client/Server architecture provides a far higher degree of **security** and **data integrity**.

The nature of the Buzz forum system necessitates a means of **centralised data access** and a system featuring **high maintainability**. The Client/Server architecture ensures that changes made to the system are immediately visible to all clients.

Component-Based Architecture

**Definition**: Decomposes application design into reusable functional or logical components that expose well-defined communication interfaces.

***Reason for inclusion:***

The possible future needs of the system necessitate that individual functionalities be modified and/or extended to meet changing requirements. Moreover, should specific functionality be required by secondary systems or other, independent, projects; the ability to copy some part of the functionality of this system may well save significant costs and time. To achieve these goals of **reusability**, **replaceability**, **extensibility** and **encapsulation** the Component-Based Architecture style must be appropriately used.

***Main Benefits***

* Ease of deployment
* Reduced future costs
* Ease of development
* Reusability
* Mitigation of complex concerns

Layered Architecture

**Definition**: Partitions the concerns of the application into stacked groups (layers).

***Reason for inclusion:***

Provides **abstraction**, allowing individual client objects to be varied, interchanged and **reused** as necessary, and **loose coupling** the extent that, if necessary, the implementation of the forum could be completely modified without having to manipulate the Student object hierarchy at all (**interchangeability**).

Stratifying the object hierarchy of the Buzz forum system will also allow for a significantly improved development environment, wherein a developer need only worry about the integration of his section with the work of others concerned with the same layer of complexity.

***Main Benefits***

* Abstraction
* Isolation
* Manageability
* Performance
* Reusability
* Testability