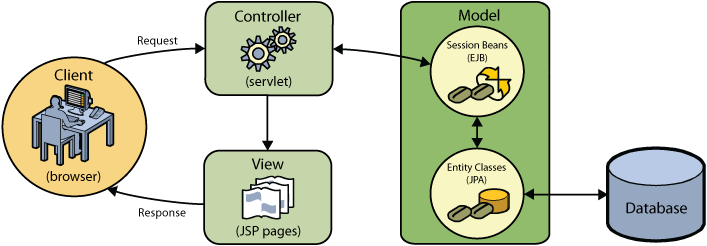
**Architectural Patterns and styles**

MVC via Java EE and layered style



View (Client tier)

…

Controller (Web tier)

…

Model (EIS tier)

…

Dependency injection

Creates new objects using dependencies, decouples code, makes it cleaner, easier to modify and easier to reuse.

Blackboard pattern

To archive scalability, blackboard multiple processes to work closer together on separate threads, introduction of this pattern will help out multiple process of the buzz system to run efficiently as the pattern emphasizes multiple process working together

Layered Architecture

System will be separated through layers; there will be User Interface layer, services layer and process layer which includes Business logic and data. User Interface layer will handle interaction like receiving input from users, the service layer will provide the human layer with services like opening a buzz space and commenting on the buzz thread and lastly process layer will process services rendered for authorization and quality check like plagiarism. Separation through layers will enhance performance, manageability and reusability.

*Client/Server*

**For communication of the server which is buzz system with users, this pattern have benefits of security as all data will be stored on the buzz system server and ease of maintenance as server is responsible of repair with client knowing of damage.**