

Design Rationale

Group Name: AndrewAndBassel

In our design, we tried to follow as many design principles as possible. Our focus was on extensibility and robustness. First principle taken into consideration was the DRY principle (do not repeat yourself) in a sense that we tried to use abstraction as much as possible to avoid repeating even a single line of code. This is demonstrated in the abstraction of entities like user, contract, bid, subject and message. Such classes were created to avoid repeating code every time a user for example was mentioned in a certain part of the system. Another design taken into consideration was the SRP principle (single responsibility principle) in a sense that every class one and only one reason to change. In the NewRequestWindow class, for example, the sole purpose of the class is to initialize a new request layout. Another principle taken into consideration was the encapsulation principle which implies that since any piece of software is expected to change at some point, code expected to change should be encapsulated. Following this principle, we decided to make variables and methods private by default and then we started upgrading access rights step by step from private to protected and not to public straight away to maintain proper encapsulated code.