

## Team 29

Sudipta Halder (2021202011)

Sowmya Lahari Vajrala (2021202010)

Anjaneyulu Bairi (2021202008)

Josh Joy (2021204009)

### Design 3 (by Cooney)

**Design Pattern:** Mediator, Proxy, State, Observer, Adapter, Memento, Iterator, Singleton

#### Positive

- The network transmission protocol contains all stages (connecting, authenticating, transmitting, verifying, disconnecting)
- The STS is providing the broker with a display of his/her favorite stock. The display is provided using an Observer pattern.
- The STS in this design provides the broker with the information of pending stocks, this is done using the Observer pattern.

#### Negative

- The database should be isolated from the rest of the implementation according to the requirement. Here the database is being accessed directly which is a violation.
- Newsfeed categorization (politics, health, business) is not present.
- Daily, end-of-week, quarterly, yearly reports are missing from the UML design. Lack of separation of report generation and violation of single responsibility principle in StockTradingController class.
- Natural Language processing feature not present.
- There is no observer pattern to notify the networks change of state.

### Design 4 (by Talbot)

**Design Patterns used:** Adapter, Observer, State, Builder, Command

**Positive:**

- There is a Report builder which displays the daily, end of week, monthly and yearly reports. They have implemented it nicely using Builder design pattern.
- The network states are implemented using state patterns and the state change is notified using observer pattern in a very efficient way.

**Negative:**

- Lack of Isolated Database, database is being accessed directly.
- Lack of order database, so no chance of backup or re-execution.
- Favorite stock display is missing.
- No Pending Trades are displayed.
- News feed categorization is not present.
- Natural Language processing feature not present.

### Design 5 (by Andrew)

**Design Patterns used:** Adapter, State, Observer, Proxy, Composite, Mediator

State design pattern for changing connection state, Façade, Colleague, Observer pattern to isolate the complex database, Proxy, Adapter, Observer, Composite, Mediator design patterns are also used.

Best design, implementation of all most main features.

**Positive:**

- Database implementation is good as there is isolation present.
- Report builder for daily, weekly, yearly is implemented.
- Favorite stock display is present.
- Natural language processing is used to interpret the news feed.

**Negative:**

- News feed categorization is not present.
- Pending Trade status and trade completion alert using an observer pattern is not implemented according to the requirement.
- 

**Conclusion:** After examining the three designs presented above, we discovered that **Andrew's design 5** meets the majority of the system's requirements and hence it is the **best design** according to us among 3<sup>rd</sup>, 4<sup>th</sup> and 5<sup>th</sup> design.