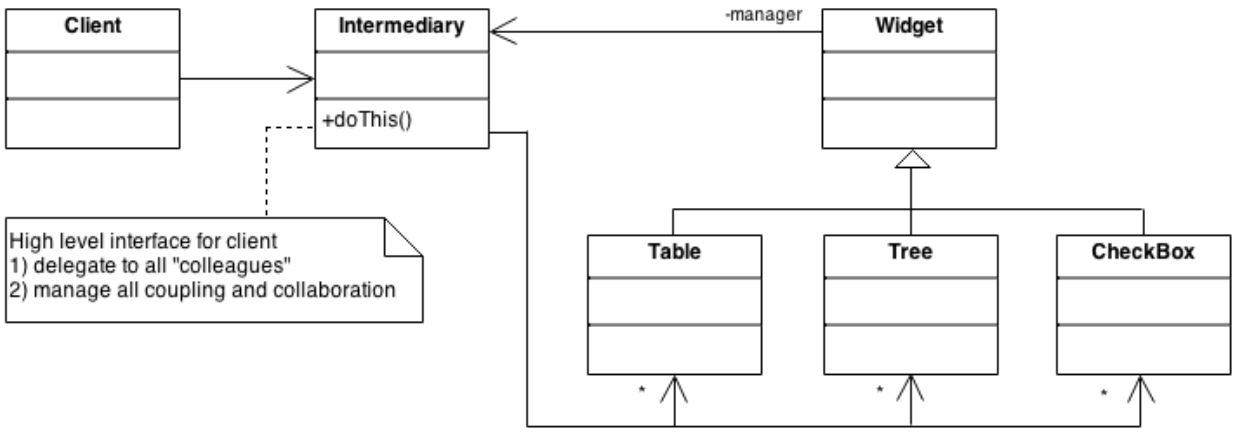
Design Pattern Description

For this deliverable, the “mediator” design pattern was chosen. Such a system considers at least four objects/subobjects and their interactions with one another. Here, we see a client object communicating to a given widget through an intermediary. That widget then communicates to various subwidgets. Needless to say, this fits completely with our application.

The structure of our system revolves around the primary interaction between the user and a database controller. Unlike the “adapter” pattern, which connects the user to a database controller through an intermediary object, our system establishes a connection directly from the user to the database controller. That database controller then delivers content functionality back to the user. Please consider the diagram below, outlining a default “mediator” design pattern:



https://sourcemaking.com/design_patterns/mediator

Our *client* exists as the UserAccount object, which is extended by the Student, Teacher, and Physician account objects. This then connects to an *intermediary* separating the client from the content, the database controller itself - such is labeled as the FirebaseController. This Firebase

controller connects to the *widget* example LoginOrAccountCreate, which is constructed from two *subwidgets*, Login and CreateAccount.

This was the only design pattern we found that allowed for the database controller itself acting as an intermediary between a given client object and the content it was to access. It was to our fortune, that this design pattern implements subwidgets, which work perfectly with our login scheme, as well as other features, like the shop and the leaderboard which have yet to be developed.