The MVC is kind of like our project but not exactly the same. For our project, the model can be the Business Plan, and the controller can be the client-server. And when we start to create the actual view using scene builder, that will be the view. However, it is not exactly the MVC in the book. The model in the book is only one object, but if the Business Plan is the model, then we will have many models that will be selected by the users.

The model contains all the logic and stores all the data and states, which is exactly what our Business Plan does. The view is just used to display and call the controller to make changes to the model, which is what I thought a JavaFX view should do. Since we use RMI in order to get a server-client relationship, our controller will be both of them, although I think the server will be more like a controller since it has an attribute that is used to reference all of the Business Plan. However, if we create a view, the client will have an attribute to reference the view. From this point of view, the client is more like a controller, so I just treat them as an entity. From the book, the view will have a reference to the model. And the server and view will be registered as observers, since both of them have direct contact with the model. Besides the methods, we have for now, we also need to add methods to register and remove observers in Business Plan in order to be a complete MVC.

A screenshot of a cell phone

Description automatically generated