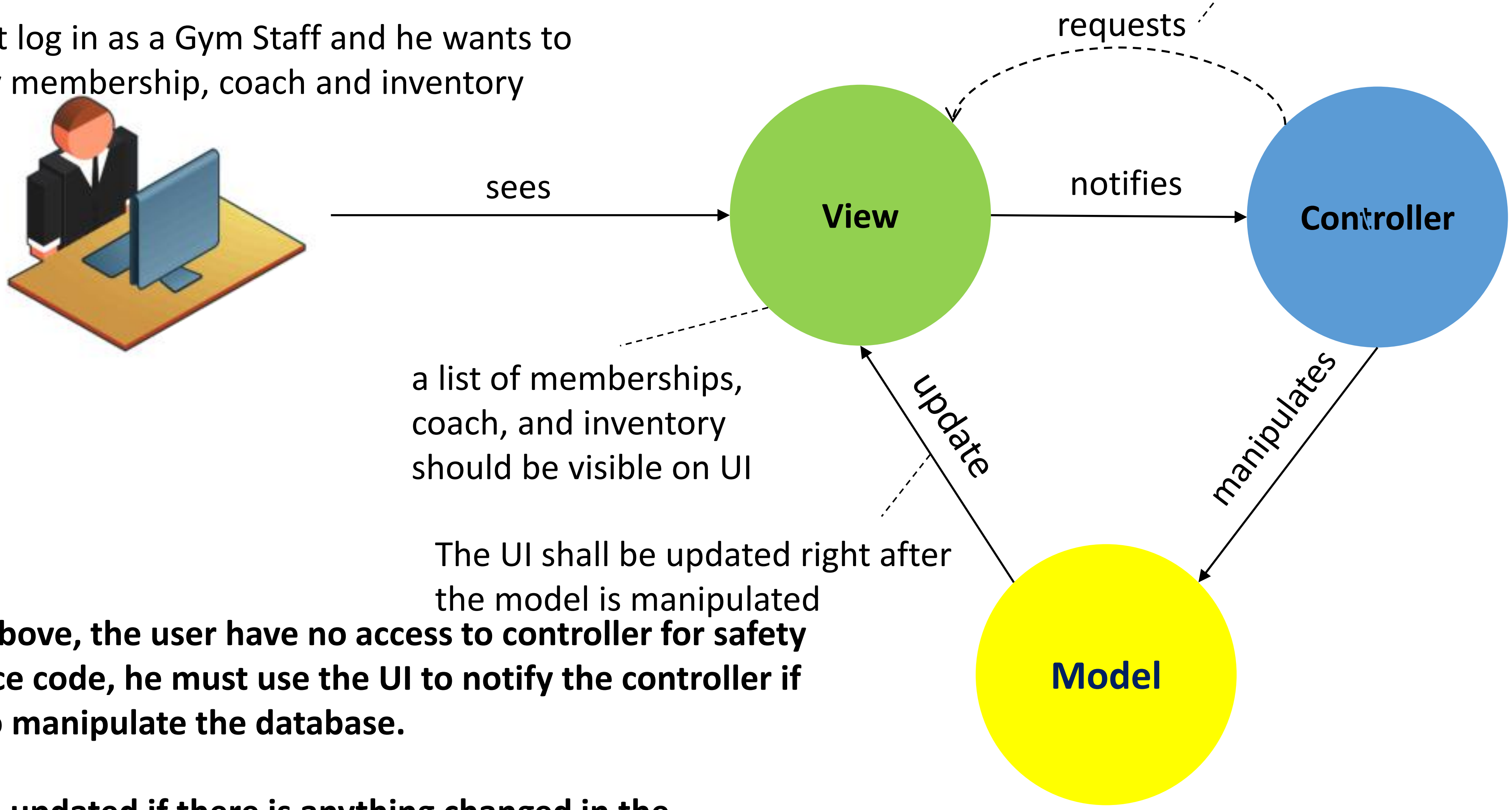
NOTE: we construct architecture sketch by using MVC
view represents the UI that client sees
Controller represents the internal component of the software
Model represents the database

To be specific, controller will request the UI for further info to better locate the item for example, which one should be removed

If client log in as a Gym Staff and he wants to modify membership, coach and inventory



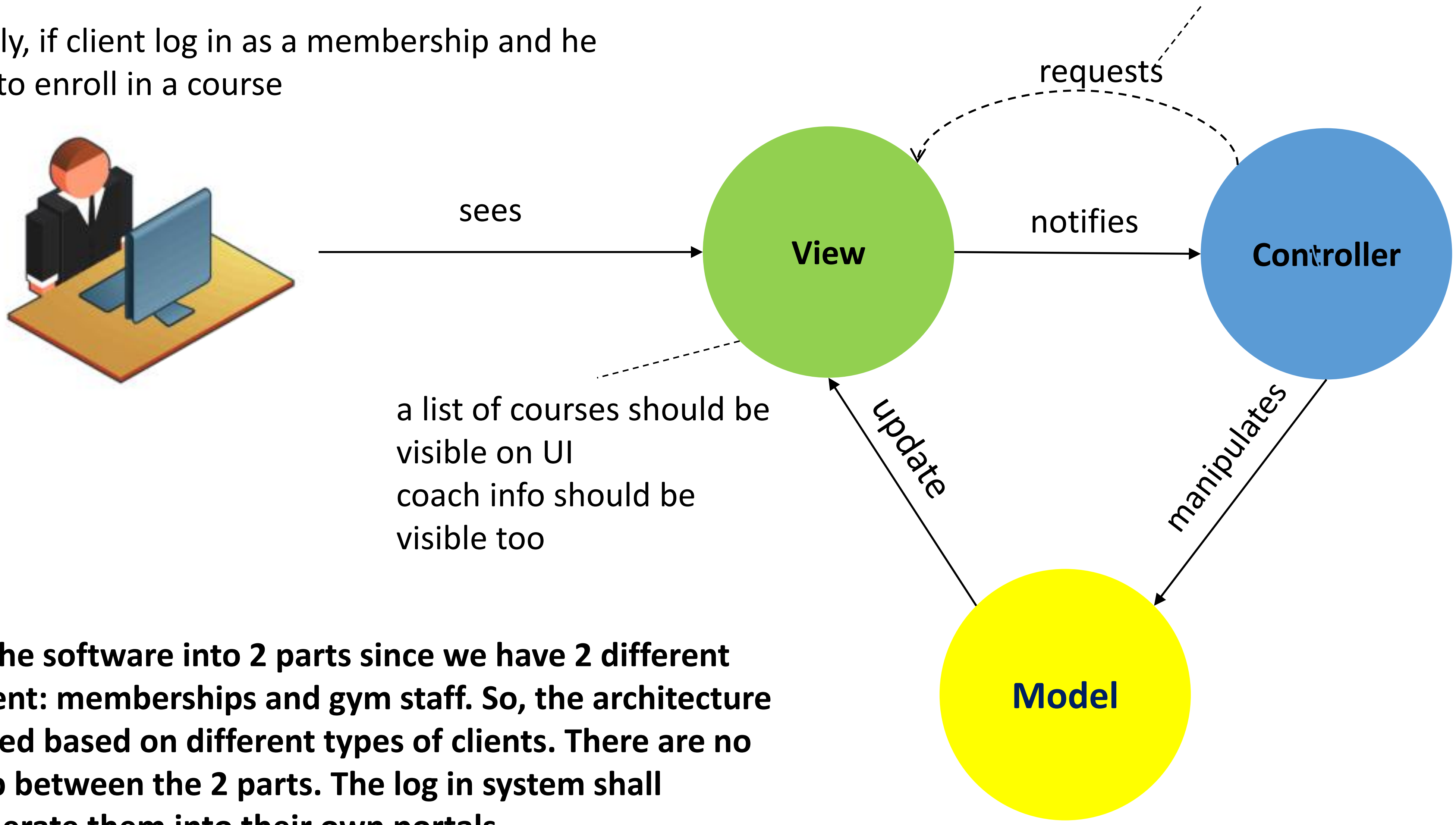
a list of memberships, coach, and inventory should be visible on UI

The UI shall be updated right after the model is manipulated

As shown above, the user have no access to controller for safety of the source code, he must use the UI to notify the controller if he wants to manipulate the database.

UI is always updated if there is anything changed in the database. We will design an update button on the main page so that the user can update the page manfully as well.

Similarly, if client log in as a membership and he wants to enroll in a course



pop a reminder to the client if fund is not enough for the course

a list of courses should be visible on UI
coach info should be visible too

We divide the software into 2 parts since we have 2 different types of client: memberships and gym staff. So, the architecture is constructed based on different types of clients. There are no relationship between the 2 parts. The log in system shall perfectly sperate them into their own portals.