The Project 2 Facebook uses MVC as the base of its design. The models hold the state data and methods for their modification, the view present GUIs to the user, and the controllers control the transmission of data and instructions between the two.

The first MVC set which governs the login page is mostly independent from the others. The LoginController does not contain a single LoginModel, but rather a Map of LoginModel-AccountModel pairs. The user can enter a username and password, which the controller turns into a LoginModel to use as a key for the map. If a match is found, an OwnProfileView and OwnProfileController are created for the corresponding AccountModel. If not, an error message is displayed.

The single AccountModel class is the model basis for multiple controller/view pairs. The BasicProfile set displays only a profile picture, name, search bar, setting button, and logout button. The Own, Friend, and Stranger sets extend the Basic versions to each add further functionality. Inheritance is favored over composition in this instance because the child classes need to use all of the parent's functionality, just with more on top.

The AccountModel class is composed of the NewsFeedModel class, which was created to keep track of not only the posts of others but also the identity of the poster. Doing so allows the program to initialize every other AccountModel's wall properly from the news feed file. Interestingly, due to the implementation of each AccountModel's post() method also posting to the news feed of its friends, NewsFeedModel.initialize() does not directly initialize its values but rather tells various AccountModels to post and in doing so has its values filled.

The second version of MVC is used, in which the controllers act as intermediaries to completely decouple the models and views. As such, the controllers have a large amount of the program's logic. Their constructors ensure that they are Observers to the AccountModel's various fields and also that the View's data and Listeners are properly initialized. The controllers all take in the entire FacebookDatabase as a constructor argument because, since this project is not server-based, they must be able to access all the AccountModels to access from the search bar and also tell the database to save.

A Controller can open another FriendProfile set or StrangerProfile set in a new window by using the search bar (or the friends list for the OwnProfile set). The decision to allow all views to have a search bar, settings button, and logout button was based on actual Facebook functionality. Therefore, one's profile picture can be changed even from a StrangerProfileView. This necessitates that the controllers keep track of both the model to display (for the picture and name) and the model of the user, but doing so was already necessary for the follow/unfollow functions.

The controllers read and save their data in JSON format to specified files. JSON was chosen because it is simple and lightweight. Although a JSON parser is used to read the information back out of the files, saving the data is done with a unique implementation because the output of the built-in JSON writer was not well-formatted for humans.