Design Patterns Used

One design pattern our group used is the Model–View–Controller pattern. Model–View–Controller is an important design pattern for our framework. The Android Application Framework allows us to make the distinction between model, view, and controller by using XML files to design the layouts of the activities and using Java files for event handlers and data management.

The View is implemented in the xml layout files. It is the physical appearance of the our app. The View is also implemented in Java files that contain the Activities and Fragments, it allows the user interact with many features of the app, notify the controller user’s request so controller can execute and pass along to model, at the end the View generates the output representation back to the user. The Controller is implemented in the AccountManager and MealPlanManager, which pass information between the Activities/Fragments and the DBManager when an event occurs. For instance, when a user updates their account information, the ManageAccountFragment passes the new information to a method in AccountManager, which formats the information and passes it along to the DBManager. The DBManager is the 2nd level of Controller that talks to the other Managers and the database/PHP server. When information is passed from the AccountManager to the DBManager, the DBManager saves the information into the database through a web server. The Model contains the database, PHP server, and Account and MealPlan objects. The database stores all the information about Accounts and MealPlans. The database notify its associate view to display an update state to the user. One example could be user hit refresh button that display the most updated state of all the meal plans.

Another design pattern our group made use of is the adapter pattern. We made an ExpandListGroup class. This class holds information about a meal plan to be displayed graphically. The ExpandableListView extends the BaseExpandableListAdapter used to provide data and Views from some data to an expandable list view. This adapter takes a list of ExpandListGroups, which is essentially just data, and an activity, and displays the ExpandListGroups to the activity. There is an inner class in our app’s home fragment which uses the adapter to display the Expandable list to the activity. The interface of the existing class that can be used with our ExpandListAdapter is the ExpandableListView.