CA2 Report: Aaron Power

Database:

To utilize the database we create a “Gateway” which provides methods for doing CRUD(Create Read Update Delete) operations. We then create a subclass of a ContentProvider which maps it to a URI(Uniform Resource Identifier) and parses request to the URI and calls the appropriate methods from the Gateway. This would normally be fine except if when we wanted the resources the database it would be called on the main UI Thread. This would make the app freeze while it is waiting for the database to respond. This is solved by creating a subclass of AsyncQueryHandler.

Interfaces:

An interface is set of method signatures defining the method's name, parameters, and return type. The only interface used within the application is the “AsyncQueryListener” interface. This interface is used so when AsyncUserContentProvider query has finished instead of using it's super classes methods it loops through and arraylist of AsyncQueryListeners and calls each one's individual query complete method. This allows for different activities to behave differently once a query has been completed.

CRUD:

While no single AsyncQueryListener fully utilizes every single type of query every single one is used in certain areas. When the user views the leaderboard, the application gets every users and displays them sorted by score in a descending order. When a user signs up they are added the database. If the user is already signed in they can update their score, or delete their account.

Abstract Classes:

If the application had two activities that behaved the same when a delete query was called, we would create an abstract class that would define the onDeleteComplete method behavior, but leave the other methods undefined. This means that while that class would not be instantiated, other classes could inherit the the abstract class and define the remaining three methods. This reduces code re use.

Nested Classes:

A nested class is a class contained within another class. The nested class can access and call any of the outer classes variables or methods including private members. However the outer class cannot access the inner classes private members. In this application most fragments are nested classes within their activity. For example having the Dialog Fragments as nested classes of the MainActivity allows the Fragment to to change the users score without having to start an activity of their own.