**In MainActivity.java**

When the user clicks the login button it calls the loginMethod() which starts a new LoginTask(). The method: LogIn(boolean) is called when the LoginTask finishes execution and needs to update the main thread.

**LoginTask extends AsyncTask<…>**

This is a private class in MainActivity.java with a preExecute, DoInBackground, and postExecute method. PreExecute is called when this class is created, DoInBackground handles the operations the task was made to do, and postExecute is responsible for updating the main thread after DoInBackground finishes execution.

This is called when the loginMethod() executes. LoginTask is a background task that handles the database connection and operation, afterwards when it finishes its background task it calls the LogIn(Boolean) function which updates the main thread, if the user was found in the database it switches views to the dashboard, otherwise displays an error message. If the user was found in the database it saves the MRN, Login\_ID, and Login\_PW into a User object.

In the DoInBackground(String… params) method in the LoginTask class, this is where all of the database operations are handled. It uses a URL to establish a URL connection to a SQL database, sends the HTTP request to the desired URL and receives a JSON response. The JSON response is subsequently parsed for the desired information such as checking the SUCCESS field to see if a user was found in the databse and the PATIENT field to retrieve the MRN which is needed for subsequent database accesses in other aspects of the app.

**In DataModels/User**

This class represents the user of this application and follows the singleton design pattern. When the user logs in it saves their MRN, Login\_ID, and Login\_PW, and is used when other parts of the app need to query the database. The queries use the MRN provided by this singleton which is used to read from or insert into the database.

**In DataModels/BloodPressureDataItem**

This was meant to be the true data model for data read from the weight sensor to then be saved in the database. This was not used, but we used the dummy data models instead.

**In DataModels/PatientWeightScaleDataItem**

This was meant to be the true data model for data read from the weight sensor to then be saved in the database. This was not used, but we used the dummy data models instead.

**In DataModels/SensorDataItem**

This was meant to be the interface for the BloodPressureDataItem and PatientWeightScaleDataItem to define common data fields and functionality.

**In DataModels/Patient**  
 This was meant to model a patient in the userbase based on the data model design that was provided by the stakeholder. This was for testing purposes only and can be deleted.

**In DataModels/PatientWeightComputeItem**

This was meant to model a patient weight compute entry into the database and was based on the design provided by the stakeholder. This was for testing purposes only and can be deleted.

**In DataModels/PatientBPComputeItem**

This was meant to model an patient blood pressure compute entry into the database and was based on the design provided by the stakeholder. This was for testing purposes only and can be deleted.