***Design Architecture & Justification***

The chosen architectural pattern for our application is Model-View-Controller. Similar to many mobile applications, the view of our app is the device itself. The screen displays the model and interfaces with the user. When an event is registered by the device, it is sent to the controller which is our AppNavigator.js file. The controller maps the event to the proper model for processing and updating of the state.

React native allows us to develop separate components (models) for each app functionality. The controller will direct the event to the affected model such as WaterScreen.js, ProfileScreen.js, or FoodScreen.js to name a few. The component will update the view and state of itself, as well as, save the new state to the device.

Once updated, the change is re-rendered on the view to allow the user to see the response. This could include a new entry that changes graphs or tables, an update to their profile data or the removal of an erratic entry.

***Diagram***

