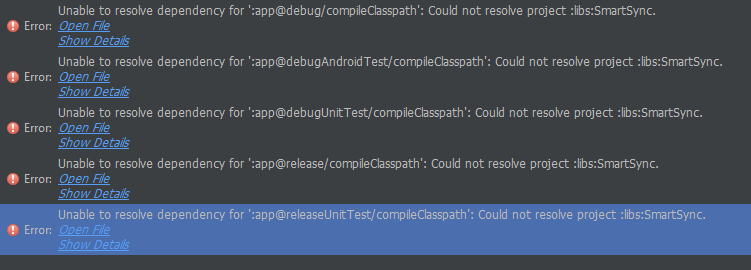
**Obstacles:**

1. SOQL has problems with nested queries (won’t do more than 1 level)
2. Dependency and path issues created bugs (Salesforce application that made queries used their own newer dependencies that clashed with ours)
3. Occasional build errors that aren’t resolved with cleaning and rebuilding
4. Null pointer errors (rendering home fragment when app starts, rendering fragments with queries without the need for a button)
   1. Doing fragment transactions in MainActivity (network calls in soql to get records) causes loads of problems due to onSaveInstanceState() placed before onStop()
5. Application with Salesforce implementation won’t work on most people’s laptops, even with proper installations for the environment. Gets the following errors:



1. Callbackhandler error in ConnectivityManager and incessant need to verify Salesforce account with pin sent to phone number of system admin
2. Error posting in Salesforce using SalesforceSDK’s methods even though rest request was successful

**Workarounds:**

1. Bunch several requests to imitate nested queries (e.g. sendBedSoqlRequest1, sendBedSoqlRequest2, etc.)
2. Fix all packages to v 27.1.0 and in the future resolve any mixing version errors by consolidating versions
3. Deleting build folders and have android studio rebuild from scratch
4. Null pointer fixes:
   1. Starting up app bug: moved fragment rendering and instantiation to onPostResume in MainActivity since we cannot be sure that an activity's existing fragments have resumed in the activity's onResume and should avoid fragment transactions until onResumeFragments & onPostResume, when their state has restored and they have actually resumed
   2. Removing button to get data: moved query to onViewCreated to make sure view is rendered before editing
5. Doing query based work solely on Monica’s laptop due to error with unable to find matching configuration with SmartSync [Temp Workaround]
   1. Created a bash script that will setup the environment needed. Proceed to follow instructions on downloading template app using forcedroid, overwriting template app folders with our project folders (edit AndroidManifest.xml and build.gradle in app level with dependencies, and under app folder in module, replace assets folder, src folder, and build folder with our modified ones.
6. Add IP address to trusted ip range in Salesforce for temporary workaround or get the connected app working to fix issue with logging in Salesforce
7. Using Postman API to figure out the kinks needed to post (certain fields are required, the format, etc.)

Possible future improvements:

* Have home and lottery preserve states when switching screens