Android iOS App Update:

The android NTL Lake Condition has updated to V2.0, where a new UI has been introduced, and user using scheme also gets altered a lot. The new user interface looks like this.

“I need to add the ”

From the developer side, this is something that you definitely need to know:

**Newly added class:**

LakeInfoAdapter: This adapter is for managing the fragment, specifically LakeConditions instance to show up correctly for main ViewPager, pageRootCcontroller.

UserSettingKey: This class is for storing the key for user’s several preference. By using those keys the developer can access to the user preference in SharePreference, such as which lake they want to be set as Homepage, or which Unit they want to see the lake information in.

MetricUnitConverter: This interface defines some methods that will translate the data from metric to other units, especially British Unit. The new class “MetricToBritishConverter” and “MetricToMetricConverter” are the implementation for this interface.

WeatherDataFilter: This class, based on its name, is used to clean the WeatherData instance. It will check whether the information in WeatherData is out of the reasonable bounds. If yes, the filter will set the data into a missing value, which will be recognized as missing when the data get displayed.

LakeTermWiki: This LakeTermWiki stores the explanation for each term displayed. You can link a view with one of the explain. Then as long as you click on that view, a SnackBar or Toast will show up to tell you the explanation.

LakeMaps: This LakeMaps works like an ordered dictionary, so that the first inserted item will stay in the first place. You can also get the key and value based on the position. For this program, I also write a function that can move one entry into the first position so that I can put the home page the first.

**Heavily changed class:**

Main Activity: it now acts as a main controller that holds the different components such as “PageViewer”, “TabLayout”, “FragmentAdapter”. It also works with some application status such as UnitType, network status. It also interacts with a current displayed fragment to keep it updated.

Lake Conditions: This class now serves the purpose of Fragment. It now controls a heavy information page, which relies on LakeData instance to get populated. Each instance now has its own WeatherOps instance, which will update the data through the network or from the local database for this particular lake

**Newly added layout:**

Fragment\_lakeinfo.xml: the view file for Fragment, lake Conditions

Content\_main\_tablayout.xml: the view file that holds the “tablayout”, “PageViewr” and “HomeButton”

**How did I create ViewPager?**

ViewPager requires Fragment to work, which is essentially a template view. The FragmentPagerAdapter will assign the data to the Fragment and then the Fragment will use the data to generate the view. In my implementation, LakeConditions is the Fragment, and LakeInfoAdapter is the FragmentPagerAdapter. The LakeInfoAdapter will use the lakeID to initialize the corresponding Fragments based on the user’s choice of the Homepage. After that, the LakeConditions itself will do an operation to update the data. After cleaning the data and converting the units, the information on the Fragment will get populated to show before the user. It’s not that complicated.

The ViewPager can work seamlessly with Tablayout, so that a tab with the page title will show up and be used to navigate.

**Dependency:**

Since I use some Google library for a more fancy user interface and user interaction, I include some extra dependencies for the managing the code. It can be a good start to understand what I have done. Cautious: all of the library has version number 22, I tried 23, but it seems not working:

Android.support:appcompat-v7: AppCompatActivity and ActionBar for the material design, the UI implementation idea I adopted from Google .

android.support:design: I used some widgets from this package, for example floating action buttons and snackbars for my user interface

android.support:support-v4: Fragment, Fragment Manager

You can learn more about those extra libraries from this website:

<https://developer.android.com/topic/libraries/support-library/features.html>

You can also learn more about material design from this web:

<https://material.google.com/>