1.Contents of the electronic submission:

- The entire android project is attached as a zip file. It includes the following packages:
    - (new files/directories are bolded)
    - businessLogic:
        - **OpenWeatherMapProvider.java**
        - **PreferencesService.java**
        - **ReminderTimeProcessing.java**
        - TransitListGenerator.java
        - TransitListPopulator.java
        - **UserPreference.java**
        - **WeatherProvider.java**
    - objects:
        - BusRoute.java
        - BusRouteSchedule.java
        - BusStop.java
        - BusStopSchedule.java
        - BusVariant.java
        - Location.java
        - **ReminderParams.java**
        - ScheduledStop.java
        - Street.java
        - **Temperature.java**
        - Time.java
        - TransitListItem.java
        - **Weather.java**
        - **WeatherCondition.java**
    - persistence:
        - **preferences:**
            - **DataAccessObject.java**
            - **DataAccessStub.java**
            - **Preferences.java**
        - transitAPI:
            - ApiListenerCallback.java
            - TransitAPI.java
            - TransitAPIClient.java
            - TransitAPIProvider.java
            - TransitAPIResponse.java
        - **weatherAPI:**
            - **WeatherAPI.java**
            - **WeatherAPICallback.java**
            - **WeatherAPIClient.java**
            - **WeatherAPIProvider.java**
            - **WeatherAPIResponse.java**

  - presentation:
    - **AlarmReceiver.java**
    - **BusDetailedViewFragment.java**
    - **BusInfoDisplayAdapter.java**
    - **BusListViewHolder.java**
    - **BusStatus.java**
    - **BusTimesDisplayAdapter.java**
    - **LocationSettings.java**
    - MainActivity.java
    - **MainListViewFragment.java**
    - **MapManager.java**
    - **Messages.java**
    - **OnBusStopClickListener.java**
    - **OnReminderButtonClick.java**
    - **OptionsMenu.java**
    - **WeatherPresenter.java**
- test/java:
  - comp3350/WinnipegTransitGo:
    - tests:
      - businessLogic:
        - **ReminderTimeProcessingTest.java**
        - TransitListGeneratorTest.java
        - **UserPreferenceTest.java**
      - objects :
        - TransitListItemTest.java
      - **persistence/preferences:**
        - **PreferencesTest.java**
    - AllTests.java
    - ExampleUnitTest.java

2.Log file:

- The individual logs and the team meeting logs are located in the log.pdf file

3.Overview of new features for this release:

- List of nearby bus schedules will now automatically refresh itself at regular intervals without any user input required (this information is at the bottom half of the screen)
- Weather status (ie: sunny, cloudy, rainy, etc) and temperature for current location are now displayed at the top of the map and continually updated
- User can now modify how close a bus stop needs to be to their current location before appearing on the map. They do this by changing the length of the radius that defines a circle which surrounds the user's location on the map. Any bus stop that falls within this circle will appear.
    - This information is saved in the persistence layer
- Users can now select individual busses and request to be reminded about them when they will arrive soon
    - The user will be reminded even when the app is not running.

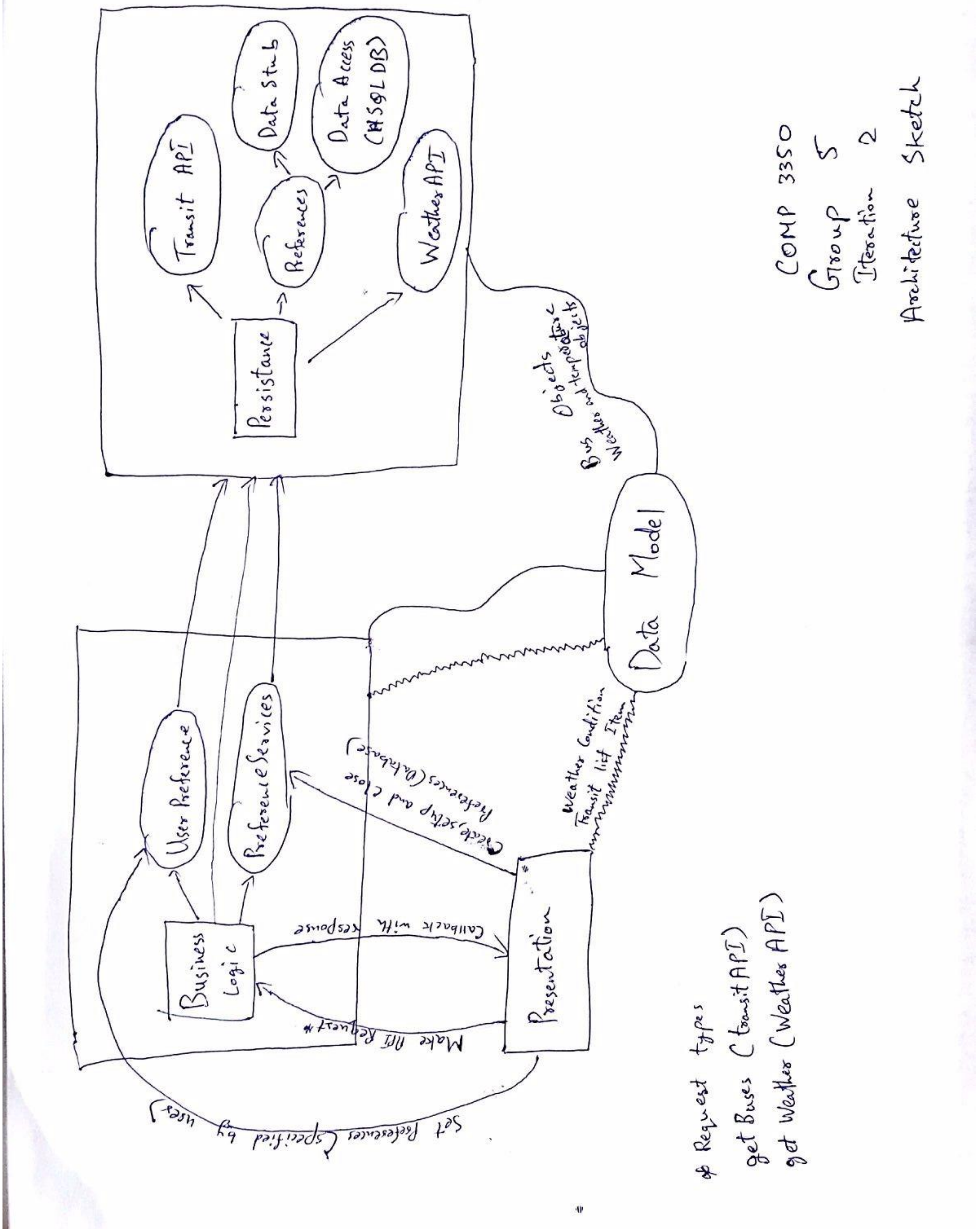
4.Bugs and Known issues:

- No known issues or bugs for this iteration

5.Tutorial on using the new features:

- Weather:
    - No user action is required. Weather information is overlaid on the map at the top-left.
- Setting Radius:
    - Tap on the upper-right of the screen (the three dots) and select “Set Radius”
- Bus Arrival Reminders
    - Tap on the row for the upcoming bus you want to be reminded about
    - Select a time
        - The time is how many minutes prior to the bus’s arrival you want to be notified

6.Updated architecture sketch:



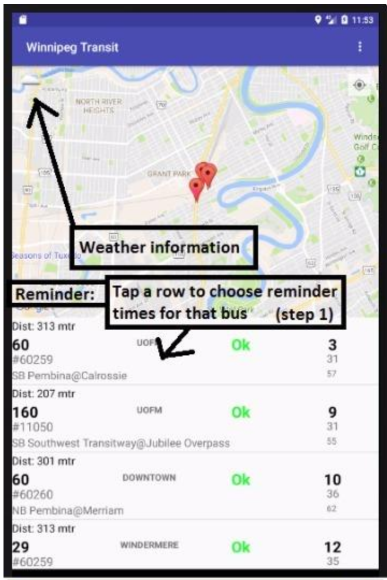
7.Unit tests:

- We cannot call the public methods to test in the business logic because they are creating new threads, this is why we are using reflection to test the private methods.
- We don't have tests for "OpenWeatherMapProvider" since it is only used for passthrough of API data to presentation.
- We do not have high test coverage for the following reasons:
    - Many of our methods make API calls that require running on separate threads which are simple pass throughs of data from the API.
    - Most of our data model classes do not require tests since they only provide getters and setters. In addition, we never create any instances of these objects since they are all created automatically for us by the API.

8.Code debt and clarification:

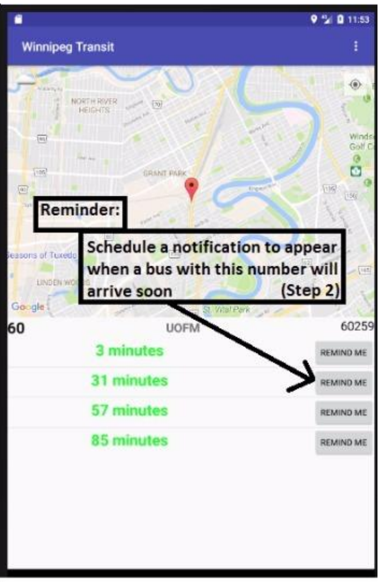
- We realize that "OpenWeatherMapProvider" should be part of persistence since it only provides pass through of API data to the presentation. We intend to fix this for the next iteration and have a factory method for the API in the business logic instead.
- SimpleDateFormat belongs in the business logic as it is used to calculate fields (not for presentation)
    - We include this because this was mentioned in the feedback for iteration 1.

9. Illustration of the new features:

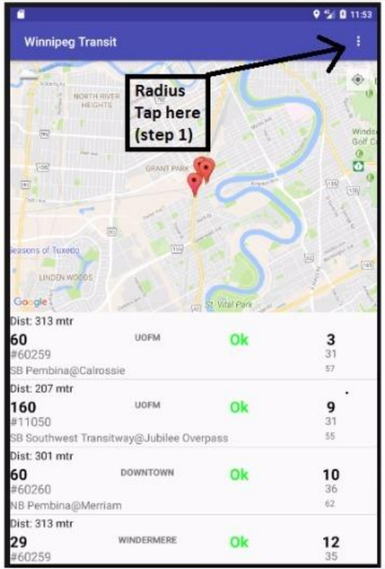

Weather information

Reminder: Tap a row to choose reminder times for that bus (step 1)

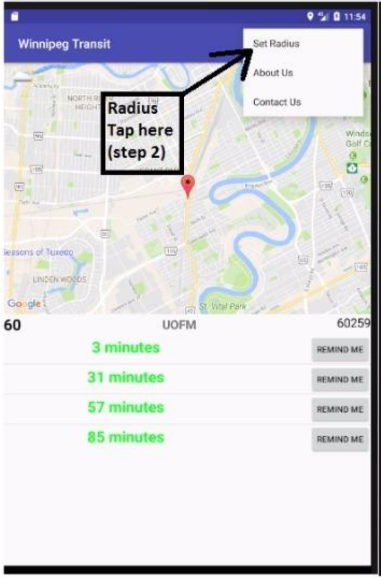
Winnipeg Transit
Dist: 313 mtr
60 #60259 SB Pembina@Calrossie UOFM Ok 3 31 57
Dist: 207 mtr
160 #11050 SB Southwest Transitway@Jubilee Overpass UOFM Ok 9 31 55
Dist: 301 mtr
60 #60260 NB Pembina@Merriam DOWNTOWN Ok 10 36 62
Dist: 313 mtr
29 #60259 WINDERMERE Ok 12 35

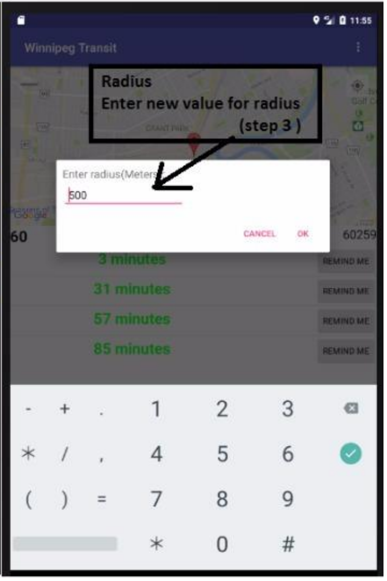

Reminder:

Schedule a notification to appear when a bus with this number will arrive soon (Step 2)


Radius Tap here (step 1)

Winnipeg Transit
Dist: 313 mtr
60 #60259 SB Pembina@Calrossie UOFM Ok 3 31 57
Dist: 207 mtr
160 #11050 SB Southwest Transitway@Jubilee Overpass UOFM Ok 9 31 55
Dist: 301 mtr
60 #60260 NB Pembina@Merriam DOWNTOWN Ok 10 36 62
Dist: 313 mtr
29 #60259 WINDERMERE Ok 12 35


Radius Tap here (step 2)


Radius Enter new value for radius (step 3)