02/02/2024

* Talked to Lisa about entering code talk
* Struggled to think about ideas

03/02/2024

* Came up with idea around getting notified for trains

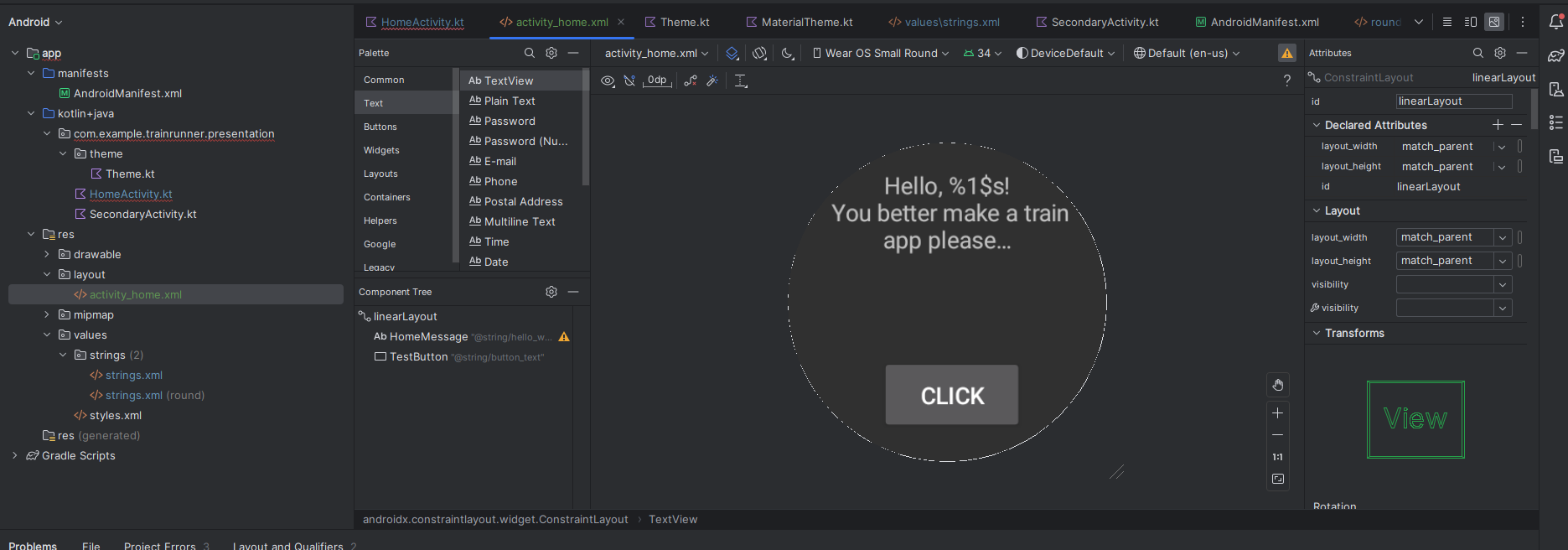
04/02/2024

* Installed latest Android Studio
* Found page on getting Wear OS app going in emulator <https://developer.android.com/training/wearables/get-started/creating>
* Tried setting up emulator with Tiramisu, but laptop too poos to install (didn’t meet system requirements)
* Set up physical watch for debugging
  + adb pair *IP\_ADDRESS*:*PORT*
  + adb connect 192.168.1.100:5555
* Used Bing image generator to make an icon
* Added icon into app
* Set up Metlink API
* Hit endpoints to get initial information

22/02/2024

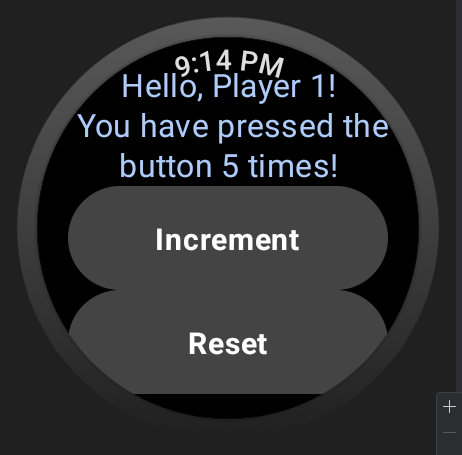
* IRL testing with Herm

24/02/2022

* Began trying to understand how the base app works in more detail.
* Activity
  + What is an activity, seem to be quite important? Is it a class? Generally, one activity implements one screen in an app. <https://developer.android.com/guide/components/activities/intro-activities>
* Composable
  + A function within activities which can take a parameter and be responsible for rendering components. When a composable is executed, it is immutable and needs to be recalled with different parameter values to be updated.
* Frontend
  + Tried to make a button appear on screen. App was initially made with Compose, so buttons and stuff are defined in the activity. I then looked at tutorial (see linked) to add a button and that used a XML approach. **Began getting confused as I was struggling to see how this compose stuff feed into the XML part! This will be a good talking point in presentation**
    - Tutorial: <https://www.youtube.com/watch?v=r1P7slDbtd4>
    - Compose vs XML: <https://www.reddit.com/r/Kotlin/comments/11vm9o8/compose_ui_or_layouts/>
  + Google seems to recommend compose going forward, although I got button working with XML :’(
  + 
* Got counter button done which updates message with count. Also added reset button {add photo}

26/02/2024

* Android Emulator:
  + Was getting annoying having to use my watch when debugging for smaller things. Bit of a delay when waiting to debug over wifi, etc.
  + Got emulator working by enabling the Virtualization Technology setting in my laptop’s BIOS and then installing a random deprecated hypervisor thing called HAXM, which made the emulators start to work!
* At this point – I have gotten a page/activity with two buttons and a feedback which tells the player how many times they have pressed the button!



28/02/2024

* Trying to get a new activity going and navigation working
* Cloned the the wear os sample repo to try and spin up one of the sample projects, get a feel for it https://github.com/android/wear-os-samples/tree/main/ComposeAdvanced

03/03/2024

* Got navigation working for 3 pages (home, settings and edit) thanks to: <https://www.youtube.com/watch?v=jt5sJEnDsSQ>
* See 01-NavigationFirstCut video

05/03/2024

* Drew in notebook of watch face states

09/03/2024

* Get navigation working over two different activities/files
  + Sort of just need to just reference lol, goes hand in hand with next item
* Split/abstract out navigation from main activity because the main activity was getting too large have functionality + navigation e.g.
  + Inspired by ComposeAdvnaced sample project
  + Also one activity per page is not true
* To get swipe gesture back – changed **rememberNavController** to instead use **rememberSwipeDismissableNavController (hopefully this works)**
* For scrollable, came across the com.google.android.horologist library, which has features not yet released but often required: <https://github.com/google/horologist>
  + Used for rememberColumnState

10/03/2024

* Found SVG image site, easy to import
* Made own Chip component (will I keep it? Navigation doesn’t seem to work)
* Made settings page, route page, edit route page, etc.

11/03/2024

* Added + button to routes page

16/03/2024

* Got to a point where navigation and screens done with hardcoded data
* Started trying to load data into database and load station list when adding a route
* Copied an example from a sample project to try and load a static list of station names to populate a dropdown list. App kept crashing on startup
* A screen shot of a computer

  Description automatically generated
* Fixed with adding to manifest file: android:name="com.example.trainrunner.presentation.BaseApplication"
  + <https://stackoverflow.com/questions/62496492/android-app-application-cannot-be-cast-to-my-application>

17/03/2024

* Learnt about room, doa and model https://developer.android.com/training/data-storage/room
* Setting up legit SQL lite database: <https://www.youtube.com/watch?v=D7PW4P3FmnU>
* At this point had a “database” file full of array information, but this isn’t very practical or scalable
  + A screenshot of a computer screen

    Description automatically generated
* Instead, started migration to SQL lite, e,g, using room, doa, model, repository, etc.
* Similar sort of set up with side loading in BaseApplication

18/03/2024

* Planned on how better table structure in Database room. Add routeNotification table to be used as way for app to know when to send notifications.
* Found out about database inspector in Android studio!
  + https://developer.android.com/studio/inspect/database#view-data
* Potential way to preload stations
  + <https://stackoverflow.com/questions/68380598/how-to-add-1300-rows-to-sqlite-table>

19/03/2024

* ViewModel does not need to map to a database table 1:1
* Split out Route notification data into RouteNotification table

22/03/2024

* Tidied the add route screen, added icons
* Got the add route screen to populate from database, e.g. using state

NOTE for presso:

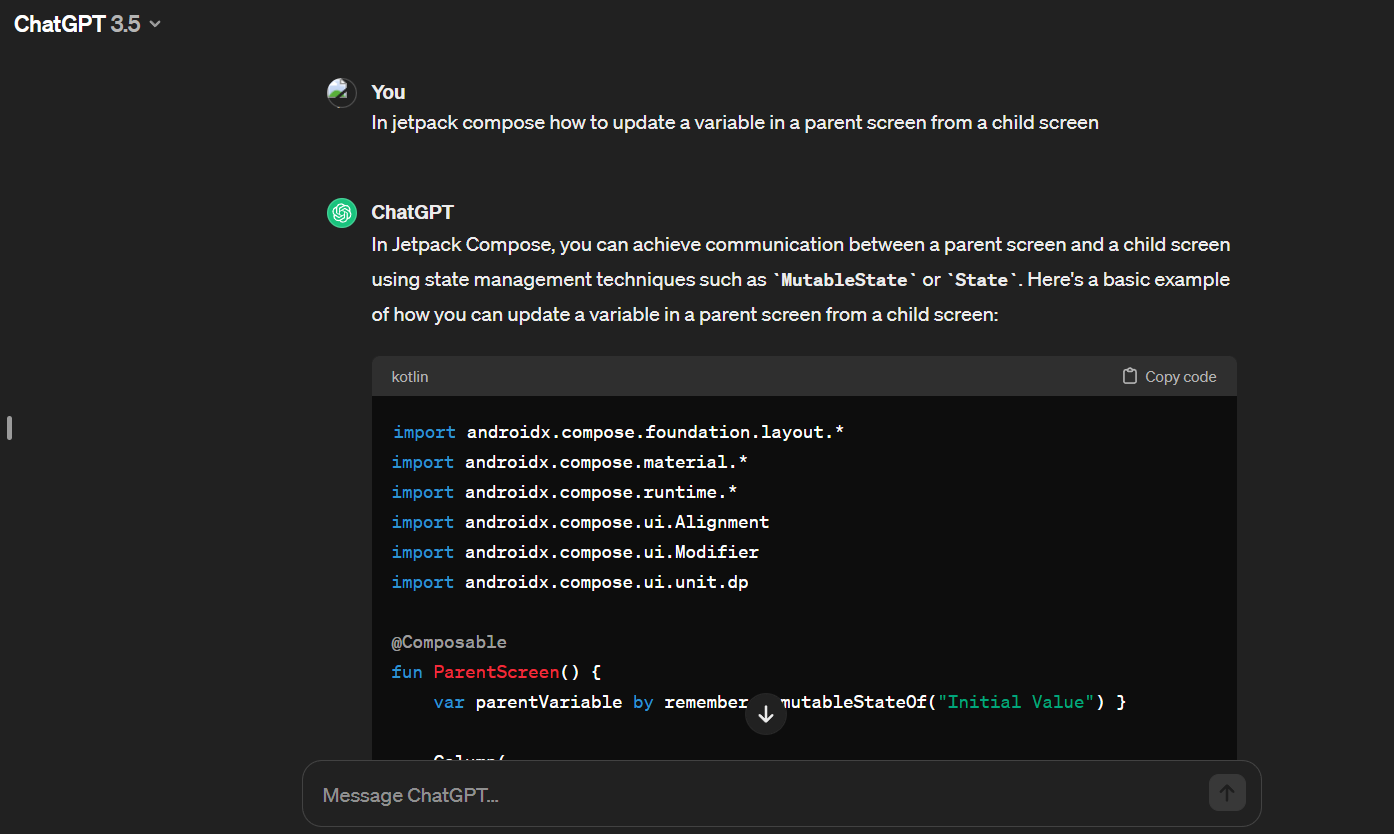
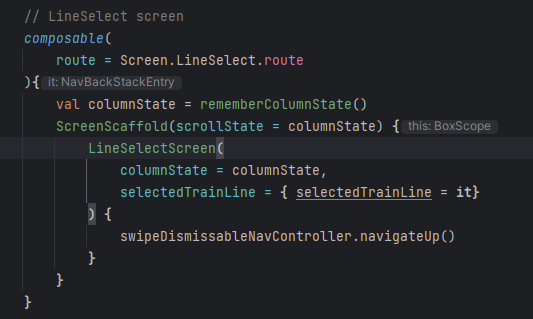
Process I used for getting going for database stuff was one step at a time. First was to set up database, models, daos, etc. Step 2 was creating a ViewModel for AddRoute screen. At this point I realized that ViewModels were how you should be passing data into a Composable function that renders. Step 3 was leveraging the ViewModel factory so that a ViewModel was created and loaded data on it’s creation. Step 4 was then using ViewModel state property (state contains the defined object for that page) to pull out what I wanted, e.g. station one code, station two code, if route was enabled, etc.

* App started crashing when adding delete (I had hardcoded routeId being passed in to AddRoute page). This was because .collectLatest was returning null and my RouteState object had non-nullable properties. Found by looking at debug logs and then putting in a breakpoint and debugging
* Created powershell script to translate station information from Metlink into code so I could paste in and have static data
* Pasted in static station code and made it load in from TrainRunnerApp, so it is only invoked when the user opens the app and it truncates the Station table and readds everything in. Probably not most efficient, I wonder if there is a way to be able to leave data in tables?

24/03/2024

* Added Metlink routes/lines as static data on startup to database
* Hooked up train line list to add route screen button so that the user goes to the available train lines

25/03/2024

* Tried to use MutableStateOf to hold train line values so it can be passed back to AddRoute page
* ChatGPT Saves the day!
  + 
  + Had a problem where I had the selectedTrainLine variable in TrainRunnerApp and wanted to populate the value from the LineSelectScreen and then pass the value back into the AddRouteScreen. Ended up passing in a lambda function to the LineSelectScreen and then in the LineSelectScreen the Lambda is invoked and the new value from the database is passed back through
  + 
  + A computer screen with text

    Description automatically generated
* Applied the same to Station selection screens
* Also made it so the route for station selection takes a parameter
* Got it so the selected station codes get saved to the route table!!!

Next up:

* Networking
  + Seeing if you can hit metlink api directly from app
  + <https://developer.android.com/training/wearables/data/network-access>
  + Also can traffic be proxied through phone viua bluetooth as my watch does not have 5G/data?