**\*shared preferences:** require set/get methods that allow for accessing and modifying data even when app is destroyed (stores it in preferences)

**Firebase:**

* Firebase URL specified in google-services.json
  + File path: capstone\_sam\_alex/app/google-services.json
  + To change firebase project, configure project in firebase, get the new URL and replace in the json file

**LoginActivity:**

* Initiates Firebase Authentication
* Stores UserId as a shared preferences variable
* onStart() method:
  + if user is already signed in from previous login, then automatically switch to StartActivity
* activity also handles “sign out” information it receives from the StartActivity

**StartActivity:**

* gets the UserId from the LoginActivity
* Contains sign out button
  + The only place in the app that the user can sign out and return to the LoginActivity
* Contains contact button that displays a Dialog Message box
* Stores night counter as a shared preferences variable
* Start drinking button switches to the MainActivity
  + Passes Integer id and String startActivity
    - Used in MainActivity as a flag that tells the app that it is coming from the StartActivity
* Stores current screen as a shared preferences variable
  + Every time the app is launched, the start screen checks to see what the currentScreen variable is and switches to that particular activity

**MainActivity (Timer Activity):**

* If location is not turned on, upon entering MainActivity, app reminds user to turn on location
* Gets flags from StartActivity that tell the app that it is coming from the StartActivity
* Shared preferences variable stored that tells the app that the current screen is “main”
* If the app is coming from the Start activity:
  + Create the Morning Questionnaire alarm (that goes off the next morning)
  + Create alarms that will go off 30, 60, 90, 120 mins from the current time (initialTime)
  + Start the UI service using initialTime
  + Start the Location service
* If the app is coming from the question screen (Main2Activity):
  + Cancel all current alarms
  + Stop current UI service
  + Get new current time (notificationTime)
  + Create new alarms that will go off 30, 60, 90, 120 mins from the notificationTime
  + Start UI service using notificationTime
* Button to go back to Start:
  + Stop UI service
  + Cancel all alarms
  + Change currentScreen shared preferences variable to “start”
  + Unregister receivers to avoid memory leak
  + Stop Location Service
* Alarm Manager methods
  + Once alarm is created, it goes to TimerReceiver class and does not trigger until the appropriate time
  + Controlled by the broadcast ID associated to each alarm
    - Allows us to know what alarm is what
    - Gives finer control over specific alarms in TimerReceiver
* onResume()
  + after app is closed or destroyed, this method is called when app is reopened

**Main2Activity (Questions Screen):**

* Creates all the buttons for the questions
  + Writes all the answers to the questions to the database
* Gets the shared preferences variable for the UserID and night counter
* Increments numberDrinks shared preferences variable each time activity is launched

**MorningQS:**

* Creates all the buttons for the questions and writes to the database
* Certain buttons are made visible only if answer to previous questions is “yes”
* If entered the activity through the notification, the activity cancels the notification once onCreate() is called
* Gets the shared preferences variable for the UserID and night counter

**MorningReport:**

* Reads the type and size of drink from the database for ONLY the previous night
* Types of analysis performed:
  + Number of Drinks consumed (shared preferences variable)
  + Number of Calories consumed
  + Percent of type of drinks consumed (Pie Chart)
  + Number of different locations visited
  + Amount of alcohol consumed (in Liters)

**BackgroundLocationService:**

* Uses Google API for location
* Gets UserID from shared preferences variable and sends it to LocationUpdates

**LocationUpdates (service):**

* Gets the userID from BackgroundLocationService
* Assigns latitude and longitude to variables and writes to the database
* Checks to see if the currentScreen shared preferences variable is “main”
  + Only if it is main, then write to database
  + Otherwise don’t write

**BroadcastService:**

* UI service
* Service is registered in the MainActivity:
  + startedTimeInMillis is a shared preferences variable (set in MainActivity) that is equal to time the timer started

**ButtonReceiver:**

* receiver for if the user presses “No” in the notification
* if this receiver is triggered, the notification is cancelled

**TimerReceiver:**

* receives broadcastInt from MainActivity
* Controls when the notification pops up, when it stops popping up, and what happens when the morning questionnaire alarm goes off

**NotificationService:**

* Receives instructions from the TimerReceiver class
* Creates a notification when triggered by the TimerReceiver

**Couple points to note:**

1. The shared preferences variable for the currentScreen is very important. To limit user interaction and access, we automatically switch screens for the user when the app is reopened. To clarify in more detail, once the user clicks “Start Drinking” (in the StartActivity), the switching of the screens is driven by the alarms. When all four alarms go off (indicating user inactivity for 2 hours), the app automatically switches from the Timer screen (MainActivity) to the Start screen (StartActivity).

Since the alarms are running in a service, the morning questionnaire alarm, which is set to go off the following morning, is still running. When the morning questionnaire alarm goes off, a notification pops up and the app automatically switches to the Morning questionnaire after which the user is able to answer all the questions, view their morning report, and return to the start screen.

All the code that handles the screen switching is in the StartActivity and the shared preferences variable is reset in each activity to indicate what the last launched activity is.

1. There is a Presentation branch in the Git repository in which the time between notifications is set to 1 minute and location updates to every 10 seconds. This is for testing and demo purposes. The master contains the code in which the times are set to 30 minutes and location updates to every 10 minutes.

Instead of hardcoding the times in every class/method, we created public variables at the top of the BackgroundLocationService, BroadcastService, and MainActivity. If times ever need to be changed, just change the values (in milliseconds).