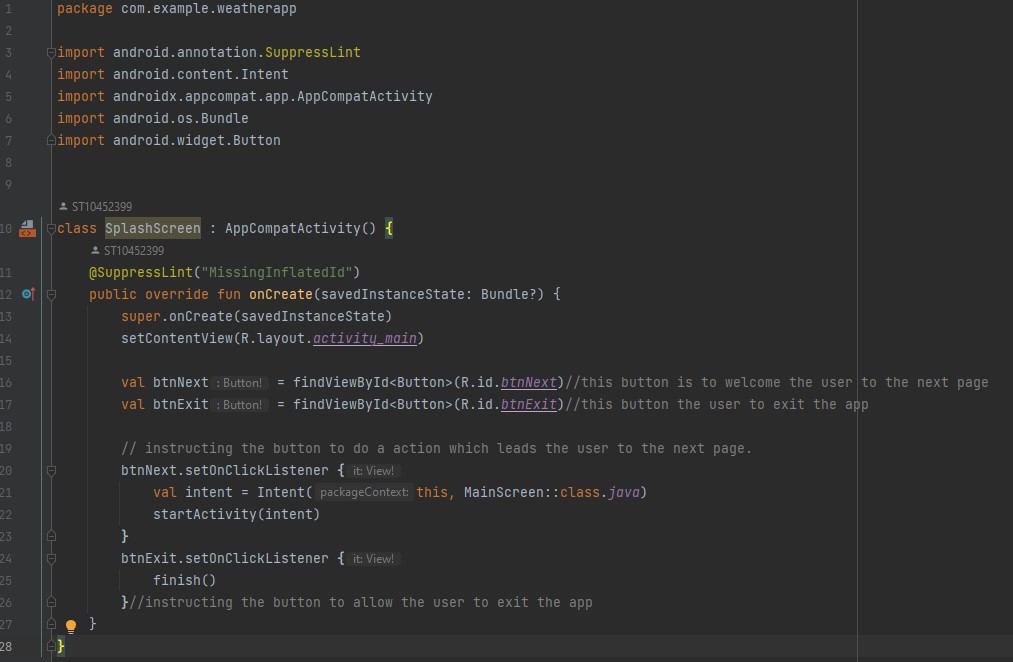
Anthony Malunga

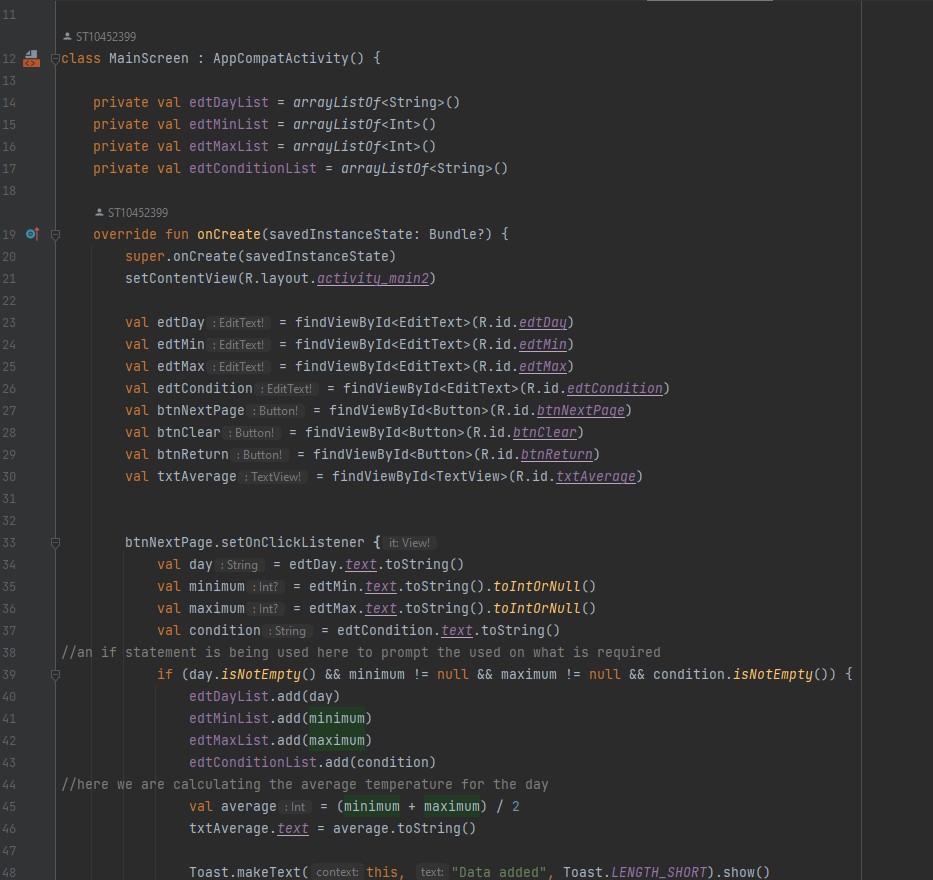
St10452399

**GitHub Link:**

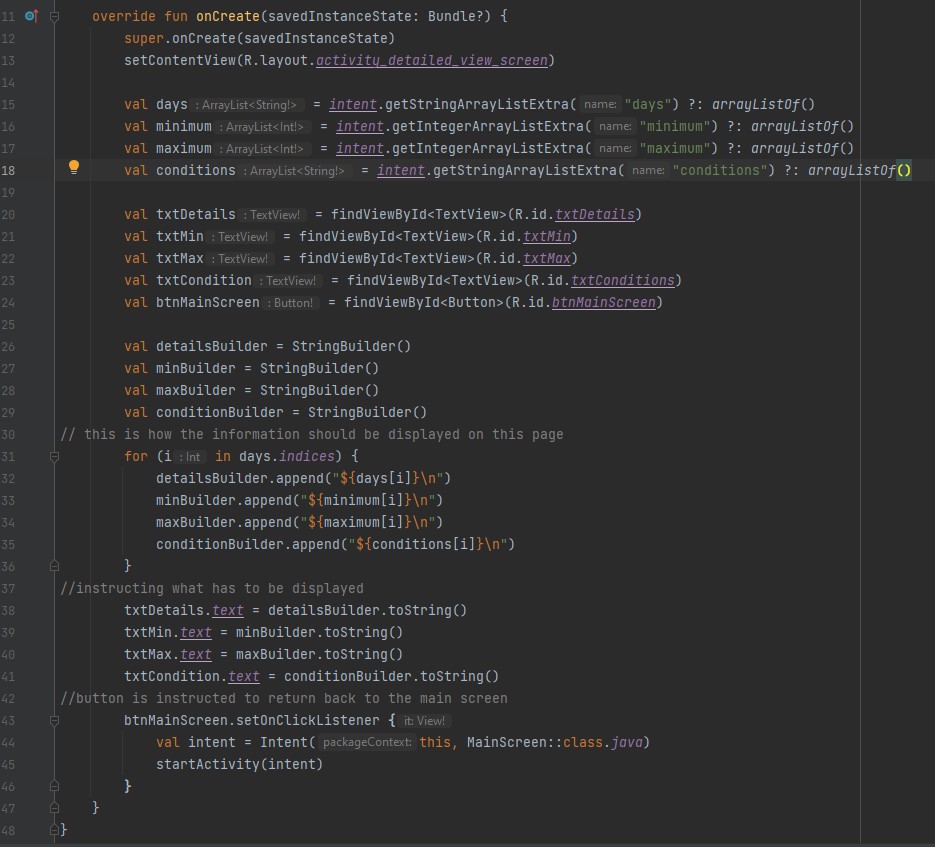
**https://github.com/ST10452399/haw1-imad5112-practicum-submission-ST10452399.git**



The `SplashScreen` class on this app acts as an entry point(welcome page), displaying an initial screen with two buttons: "Next" and "Exit." In the `onCreate` method, it sets the layout and references these buttons. When the "Next" button is clicked, the button is instructed to navigate the user to the `MainScreen` activity and starts this activity. The "Exit" button ends the activity, effectively closing the app. This screen provides a simple way to either proceed to the main functionality of the app or exit it entirely, offering a straightforward user experience for initial navigation.



The `MainScreen` class on this app collects weather data, displays the average temperature, and navigates to a screen that will display more information. It initializes four lists to store days, minimum and maximum temperatures, and conditions. On creation, it sets the layout and references UI elements. When the "Next Page" button is clicked, it validates and adds user input to the lists, calculates and displays the average temperature, and clears the input fields. If any field is empty, it shows a warning. The "Clear" button clears the input fields and average temperature display. The "Next Page" button sends the data to `DetailedViewScreen`, while the "Return" button navigates to `SplashScreen`.



This is a detailed view screen for the app, which displays weather information for multiple days and includes a button to return to the main screen. In the `onCreate` method, `setContentView(R.layout.activity\_detailed\_view\_screen)` sets the layout for this screen using the `activity\_detailed\_view\_screen.xml` file. The activity receives data from the previous screen (likely the main screen) through an `Intent`. `intent.getStringArrayListExtra("days")` retrieves a list of days, `intent.getIntegerArrayListExtra("minimum")` retrieves a list of minimum temperatures, `intent.getIntegerArrayListExtra("maximum")` retrieves a list of maximum temperatures, and `intent.getStringArrayListExtra("conditions")` retrieves a list of weather conditions. If any data is missing, empty lists are used instead. The method `findViewById<TextView>(R.id.txtDetails)`, `findViewById<TextView>(R.id.txtMin)`, `findViewById<TextView>(R.id.txtMax)`, and `findViewById<TextView>(R.id.txtConditions)` finds and references the corresponding `TextView` elements in the layout, while `findViewById<Button>(R.id.btnMainScreen)` finds the button that will be used to return to the main screen. `StringBuilder` objects are used to build the strings that will be displayed in the text views. A loop runs through the lists of days, minimum temperatures, maximum temperatures, and conditions, and for each day, it appends the relevant information to the `StringBuilder` objects. The constructed strings are set as the text for the corresponding `TextView` elements (`txtDetails`, `txtMin`, `txtMax`, and `txtCondition`). An `OnClickListener` is set for the `btnMainScreen` button. When the button is clicked, an `Intent` is created to start the `MainScreen` activity, and the app navigates back to the main screen. This activity displays detailed weather information (days, minimum and maximum temperatures, and conditions) in a formatted manner and includes a button to return to the main screen.