**Challenges Faced and How We Overcame Them**

* **Conflict Between Bottom Navigation and Drawer Navigation**:
  + **Challenge**: The NavController was shared between the bottom navigation and drawer navigation, causing overlaps and unexpected behavior, such as the Settings page interfering with bottom navigation tabs.
  + **Solution**: Consolidated all destinations into a single navigation graph and used dynamic logic to manage navigation flow effectively.
* **Maintaining Bottom Navigation State**:
  + **Challenge**: Clicking on a drawer item replaced a bottom navigation tab, and switching back to a tab led to the wrong page.
  + **Solution**: Implemented logic to treat drawer navigation destinations independently while keeping the bottom navigation state intact.
* **Visibility of Bottom Navigation on Drawer Pages**:
  + **Challenge**: The bottom navigation bar was displayed on pages accessed via the drawer, leading to UI inconsistency.
  + **Solution**: Added a listener (OnDestinationChangedListener) to hide the bottom navigation bar when navigating to drawer-specific destinations.
* **Managing Drawer Behavior**:
  + **Challenge**: The drawer remained open after a menu item was selected, leading to a poor user experience.
  + **Solution**: Used DrawerLayout.closeDrawer() to automatically close the drawer after a selection.
* **Fragment Navigation and Crashes**:
  + **Challenge**: Incorrect navigation logic caused app crashes or failed transitions to specific fragments.
  + **Solution**: Carefully reviewed navigation IDs, ensured proper graph setup, and re-tested transitions for each scenario.

 **Button Alignment Issues:**

* **Challenge:** Buttons in the toolbar were misaligned, with the first button sticking to the left edge and inconsistent spacing between buttons.
* **Solution:** Used proper ConstraintLayout constraints to define relative positioning between the buttons and the parent layout. Added margins and balanced constraints to ensure even spacing.

 **Duplicate Menu Button:**

* **Challenge:** A default menu button appeared alongside our custom buttons, disrupting the toolbar layout.
* **Solution:** Disabled the default setDisplayHomeAsUpEnabled behavior in NavController and explicitly managed the visibility of custom buttons using onDestinationChangedListener.

 **Bottom Navigation Bar Not Responding:**

* **Challenge:** The bottom navigation bar stopped functioning correctly and did not switch fragments.
* **Solution:** Ensured the BottomNavigationView was properly linked to the NavController using setupWithNavController and verified the IDs in the mobile\_navigation.xml file matched the menu items in bottom\_nav\_menu.xml.

 **Button Spacing Collapsing on Visibility Changes:**

* **Challenge:** When hiding toolbar buttons, the remaining buttons shifted unexpectedly due to layout recalculations.
* **Solution:** Replaced View.GONE with View.INVISIBLE to maintain button positions, and used guidelines in the ConstraintLayout to ensure consistent spacing.

 **Fragment Overlap in Toolbar:**

* **Challenge:** Toolbar elements like buttons and titles overlapped when navigating to certain fragments (e.g., settings).
* **Solution:** Managed button and title visibility dynamically based on the destination using onDestinationChangedListener.