IT2010 – Mobile Application Development BSc (Hons) in Information Technology

2nd Year

Faculty of Computing SLIIT

# – Lab Exam 02

Your Android app UI design skills will be evaluated in this lab exam. For this you need to use only **Android XML**. This project aims to assess your ability to create functional, aesthetically pleasing, and user-friendly UI designs that adhere to **Android development best practices**.

**Task Overview:**

You are tasked with designing the UI for an app idea that you are passionate about or find interesting. This can be anything from a productivity app, a social media platform, a fitness tracker, to a game. The key is to showcase your creativity, understanding of UI/UX principles, and your ability to implement these designs using XML in Android Studio.

**Requirements:**

1. Ideation (2 Marks)
   * The idea should be suitable for a mobile application.
   * Outline the target audience and the problem it solves.
   * Highlight the core features of the app.
2. Applying “60-30-10 rule” when deciding the colors. (2 Marks)
   * Primary Color(s)
   * Secondary Color(s)
   * Accent Colors(s)
   * Usage of the colors.xml file
3. Selecting suitable layouts (2 Marks)
   * Linear layouts – Horizontal or Vertical
   * Constraint layouts • Frame Layouts
   * Scroll views.
   * (you need to add adequate margin and padding)
4. Using proper views for the user interface (2 Marks)
   * TextView
   * EditText
   * Button
   * ImageView
   * Etc. (search from the internet and apply)
   * Usage of the Strings.xml file
5. Interactivity (1 Mark)
   * Use intent to navigate between the screens.
   * You don’t have to implement the actual functionalities.
6. Creativity (1 Mark)
   * Out of the box thinking
   * Overall aesthetics

**Submission Guidelines:**

1. Download the report format from Course Web, complete it with the application description, UI screenshots, and XML content (Strings and Colors), and upload the final PDF to the relavent link. The report format and upload links will be provided in due course.
2. Please note that if you do not upload the document to the correct link, you will be **penalized with 2 minus marks**.

Here’s how your wardrobe mobile app UI can be explained for each requirement:

**1. Ideation (2 Marks)**

**App Idea:**  
Your app, **"Smart Wardrobe"**, helps users organize their clothing collection, plan outfits, and get recommendations based on weather, occasion, or personal style preferences.

**Target Audience:**

* Fashion enthusiasts
* Busy professionals
* People who struggle with outfit choices
* Travelers looking for easy outfit planning

**Problem It Solves:**

* Users often struggle to find the right outfit for an occasion.
* Keeping track of clothes can be messy.
* Helps plan outfits in advance and avoid last-minute decisions.

**Core Features:**

* Add and categorize clothing items (e.g., shirts, pants, dresses, accessories).
* Outfit planner for special occasions.
* AI-based outfit suggestions based on weather and personal preferences.
* Search and filter wardrobe by color, type, or occasion.
* Wishlist & favorite outfits section.

**2. Applying “60-30-10 rule” for Colors (2 Marks)**

This rule ensures a balanced and visually appealing UI by using:

* **60% Primary Color (dominant)** → Light Grey / White (#F5F5F5) for the background.
* **30% Secondary Color (supporting elements)** → Blue (#3A86FF) for buttons, tabs, and navigation.
* **10% Accent Color (highlight important actions)** → Coral (#FF6B6B) for action buttons (like "Add Outfit", "Save").

**Usage of colors.xml:**

<color name="primaryColor">#F5F5F5</color>

<color name="secondaryColor">#3A86FF</color>

<color name="accentColor">#FF6B6B</color>

<color name="textColor">#333333</color>

**3. Selecting Suitable Layouts (2 Marks)**

You will use a mix of layouts for better UI structuring:

* **ConstraintLayout** → Main activity (better for flexible positioning).
* **LinearLayout (Vertical)** → List of clothing categories.
* **FrameLayout** → Used for a collapsible sidebar menu or modal dialogs.
* **ScrollView** → In outfit planner to allow scrolling when there are many outfits.

**Example Usage:**

<LinearLayout

android:orientation="vertical"

android:padding="16dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:text="My Wardrobe"

android:textSize="24sp"

android:textColor="@color/textColor"/>

<ScrollView>

<LinearLayout android:orientation="vertical">

<!-- List of outfits here -->

</LinearLayout>

</ScrollView>

</LinearLayout>

**4. Using Proper Views (2 Marks)**

You will use various UI elements:

* **TextView** → Display outfit names and categories.
* **EditText** → Allow users to enter clothing details.
* **Button** → Add, Edit, Delete outfits.
* **ImageView** → Display clothing images.
* **RecyclerView** → Display a grid/list of wardrobe items dynamically.

**Usage of strings.xml:**

<string name="app\_name">Smart Wardrobe</string>

<string name="add\_outfit">Add Outfit</string>

<string name="edit\_outfit">Edit Outfit</string>

<string name="delete\_outfit">Delete Outfit</string>

**5. Interactivity (1 Mark)**

* **Intents** will be used to navigate between screens, such as:
  + Main screen → Wardrobe screen
  + Wardrobe screen → Outfit planner
  + Outfit planner → Recommendations

**Example of an Intent in Java:**

Intent intent = new Intent(MainActivity.this, WardrobeActivity.class);

startActivity(intent);

**6. Creativity (1 Mark)**

To enhance creativity, you can:

* Use **rounded buttons and cards** for a modern UI.
* Add **animations** for smooth transitions between screens.
* Use **custom icons** for clothing categories.
* Implement **dark mode support**.

This explanation covers all aspects of your lab exam requirements. Let me know if you need further refinements! 🚀