Department of
Computer Science and Engineering

Title: XML Creation using Linear Layout

Mobile Application Development
CSE 402

Green University of Bangladesh

1 Objective(s)

- To develop the android mobile application which consist of GUI components for Login Page creation.
- To implement different button operations.

2 Problem analysis

In Web Programming we use HTML to design the front-end of a web page. The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. XML stands for extensible markup language. XML is case-sensitive, requires each tag be closed, and preserves whitespace. Android applications use XML to create layout files. Here, we will use a similar markup language like HTML named XML to design the User Interface of a simple mobile application. Layout xml files are used to define the actual UI(User interface) of our application. It holds all the elements(views) or the tools that we want to use in our application. Like the TextView's, Button's and other UI elements. To design login page coding in xml needs to be done.

3 Login Page Design

```
<!-- Login page design using XML -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorPrimary"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Log-in Page"
        android:textAlignment="center"
        android:textSize="40dp"
        android:textColor="@color/colorPrimaryDark"
        android:paddingBottom="10dp"
        android:paddingTop="20dp"/>

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter Name"
        android:paddingLeft="10dp"
        android:paddingStart="20dp"/>

    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Name" />

    <TextView
```

```
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter Password"
        android:paddingStart="10dp"
        android:paddingLeft="20dp"
        android:paddingBottom="20dp"/>

    <EditText
        android:id="@+id/etPass"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/btnOk"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ok" />

    <Button
        android:id="@+id/btnReset"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Reset" />
</LinearLayout>
```

4 Input/Output

Run the code and observe the output in the virtual device.

5 Nested Layout Code Implementation

```
<!-- Nested Layout Code Implementation using XML -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/system_notification_accent_color"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/btnOk"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
            android:textAlignment="gravity"
            android:text="Ok" />

        <Button
            android:id="@+id/btnReset"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Reset"
            android:textAlignment="center"
            android:background="@android:color/holo_green_dark"
            android:textColor="@android:color/holo_red_dark"/>
    </LinearLayout>
</LinearLayout>
```

6 Input/Output

Run the code and observe the output in the virtual device.

7 Discussion & Conclusion

From this experiment we learnt about the different components of Android XML like the important attributes available for those components, nested layout etc.

8 Lab Task (Please implement yourself and show the output to the instructor)

1. Implement nested layout structure for Name and Password field.

8.1 Problem analysis

Create EditText fields for name and its input in a single row. Do the same for password field.

9 Lab Exercise (Submit as a report)

- Implement the Reset button having both an image and a text.
- Create field for taking input of email address in a single row.
- Make the overall login page with 3 EditText fields and two buttons.

10 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.