INES – RUHENGERI
Faculty of Sciences and Information Technology
Department of Computer Science
SWE / YEAR 3 /DAY - 2024-2025

Assignment#1: Mobile Application Development / 100 Marks

📆 **Deadline:** Monday, 3rd February 2025, by 23:59 PM
👤 **Type:** Individual
📱 **Platform:** Kotlin (Android Studio) OR MIT App Inventor
📤 **Submission:** Upload the APK file + project source files + a short report

Scenario:

You are a junior mobile app developer at **TechStart Innovations**, a company that specializes in creating interactive mobile applications. The company is planning to launch a **Personalized Greeting App**, which will take a user's name as input and display a welcome message when a button is clicked.
Your task is to **design and implement this application** using either **Android Studio (Kotlin)** or **MIT App Inventor**. The app must have a user-friendly interface and follow good coding practices. You will **test the APK file on your phone** before submission.

Assignment Requirements:

- **App Name:** The app should be named using the **acronym of your first and last name**.
- **User Input:** A text field for the user to enter their name.
- **Button Interaction:** A button that, when clicked, displays a personalized greeting.
- **Activity Navigation:** The app must have **two screens**:

  - **Screen 1:** User enters their name and clicks a button.
  - **Screen 2:** Displays a greeting message using the entered name.
    - **Toast Message:** A Toast message should appear confirming the name was received.
    - **Intent:** Data should be passed from **Screen 1 → Screen 2**.
    - **APK Testing:** The app should be installed and tested on a personal phone.

Option 1: Implementation Using Android Studio (Kotlin)

Step 1: Set Up the Environment

1. Open **Android Studio** and select "Create a New Project".
2. Choose **Empty Activity** and click **Next**.
3. Name your project using the **acronym of your first and last name**.
4. Set the programming language to **Kotlin** and click **Finish**.

Step 2: Design the UI (activity_main.xml)

1. Open res/layout/activity_main.xml.
2. Add the following UI components inside a ConstraintLayout:
   - **EditText** (ID: etUserName) for user input.
   - **Button** (ID: btnGreet) to submit input.
3. Adjust **constraints** to position the elements properly.

Step 3: Implement Button Click and Intent (MainActivity.kt)

1. **Open MainActivity.kt and modify it as follows:**

```
package com.example.myapp // Change "myapp" to your project's package name

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val etUserName = findViewById<EditText>(R.id.etUserName)
        val btnGreet = findViewById<Button>(R.id.btnGreet)

        btnGreet.setOnClickListener {
            val name = etUserName.text.toString()

            if (name.isNotEmpty()) {
                Toast.makeText(this, "Welcome, $name!", Toast.LENGTH_SHORT).show()
                val intent = Intent(this, GreetingActivity::class.java)
                intent.putExtra("USER_NAME", name)
                startActivity(intent)
            } else {
                Toast.makeText(this, "Please enter your name", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Step 4: Create the Second Screen (GreetingActivity.kt)

1. Create a **new activity** named GreetingActivity.
2. Modify GreetingActivity.kt as follows:

```
package com.example.myapp

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class GreetingActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_greeting)

        val userName = intent.getStringExtra("USER_NAME")
        val textView = findViewById<TextView>(R.id.tvGreeting)

        textView.text = "Hello, $userName! Welcome to the app!"
    }
}
```

Step 5: Design the Second Screen (activity_greeting.xml)

1. Open activity_greeting.xml and add a TextView with ID tvGreeting.
2. Ensure proper UI alignment using **constraints**.

Step 6: Build and Test the APK

1. Click **Build → Build APK(s)** and install the .apk on your phone.
2. Open the app and **test if it works as expected**.

Option 2: Implementation Using MIT App Inventor

1. **Open MIT App Inventor**: MIT App Inventor.
2. **Create a New Project** and name it using the **acronym of your first and last name**.
3. **Screen 1 (UI Components):**
   - Add a **TextBox** for user input.
   - Add a **Button** to navigate to the next screen.
4. **Screen 2 (Greeting Screen):**
   - Add a **Label** to display the personalized greeting.
5. **Block Programming for Button Click:**
   - Use when Button.Click → open another screen with value to pass data.
   - On **Screen 2**, retrieve the name and display it in the **Label**.
6. **Test and Download APK**:
   - Use the **MIT AI Companion** app to test.

    - Export the **APK file** and install it on your phone.

Submission and Deliverables

📤 **Submit the following files via Moodle:**

- **APK file** (.apk) of your working application.
- **Project source files** (.zip of Android Studio project or .aia file from MIT App Inventor).
- **A short report (PDF)** containing:
    - **Your name and student ID**.
    - **Screenshots** of your app running on a phone.
    - **Explanation** of how your app works (1-2 paragraphs).

📅 **Deadline:** Monday, 3rd February 2025, by **23:59 PM**.
⛔ **Late submissions will not be accepted**.

Grading Criteria (Total: 100 Marks)

Component	Marks
Functional UI (proper layout and positioning)	20
Working button and Toast message	10
Data passing between activities	20
App functionality (correct greeting output)	20
APK installation and testing on phone	10
Report (screenshots, explanation)	20

Important Notes

✔ **Individual Work** – Plagiarism will lead to a **zero grade**.
✔ **APK must run on your phone** – Ensure proper functionality before submission.
✔ **Use Either Android Studio (Kotlin) OR MIT App Inventor** – No other platforms allowed.
✔ **Follow Coding Best Practices** – Use meaningful variable names and comments.