WORKSHEET 1

Application "PetKitty"

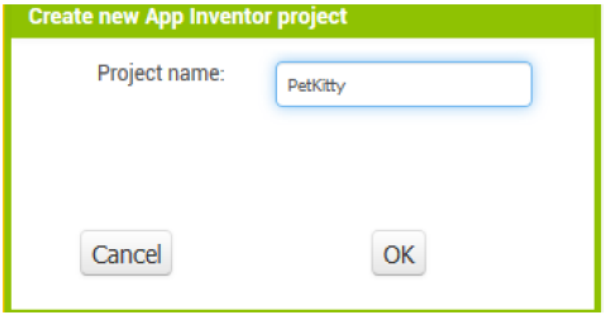
In order to familiarize yourself with the App Inventor environment, you will initially create a very simple application. In this application, an image of a cat will appear as a button. When the button is pressed, the sound of the cat meowing will be played, while at the same time the device will vibrate for 1 second.

Step 1 (Accessing the App Inventor environment)

Visit the App Inventor website (http://ai2.appinventor.mit.edu) and sign in using the Google account that has been provided to you.

Step 2 (Creating and naming a new project)

From the main menu of the application, select **Start New Project** and name the project **PetKitty**.



Step 3 (Configuring the initial application screen)

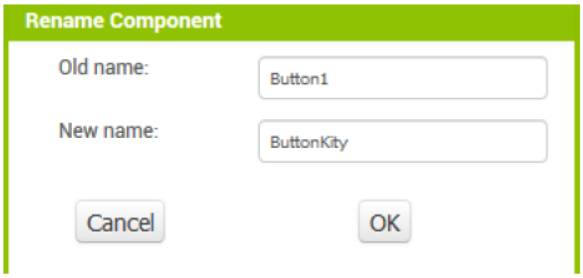
In the project you have created, the first screen of the application already exists as a basic component, named **Screen1**. Modify the **Properties** of this screen according to the following table:

Component	**Properties**
Screen1	BackgroundColor: Orange
	AlignHorizontal: Center (3)
	Title: Kitty

Step 4 (Creating and configuring a button)

Place a **Button** component on **Screen1** from the **User Interface** group of the **Palette**, using the "drag and drop" method.

Then, from the **Components** panel, select **Button1**, click **Rename**, and rename it from **Button1** to **ButtonKitty**.



Next, modify the **Properties** of the **ButtonKitty** component according to the following table:

Component	Properties
ButtonKitty	Text: (empty) Image: Upload File → Browse → kitty.png (The file *kitty.png* is located in your class folder)

Note: You can first upload multimedia files through the **Media** panel and then select them from the corresponding **Image** property of the component.

Step 5 (Creating and configuring a label)

Place a **Label** component on **Screen1**, below the cat button, from the **User Interface** group of the **Palette**, using the "drag and drop" method.
Rename the label from **Label1** to **LabelKitty**, and then modify its **Properties** according to the following table:

Component	Properties
LabelKitty	BackgroundColor: Green FontSize: 28 Text: Pet the kitty TextColor: Blue

---

Step 6 (Adding and configuring sound)

From the **Media** group of the **Palette**, add a **Sound** component to **Screen1** using the "drag and drop" method.
Note that this component belongs to the **Non-visible components** of the application and appears below the screen.

Rename the sound component from **Sound1** to **SoundKitty**, and modify its **Properties** according to the following table:

Component	**Properties**
SoundKitty	MinimumInterval: 300
	Source:
	Upload File → Browse → meow.mp3
	(The file *meow.mp3* is located in your class folder)

The final design of your application screen should look as follows:



Step 7 (Programming behaviors)

Switch to the **Blocks Editor** to program the behaviors of your application components. Create the following code:



The use of the above code is interpreted as follows:

When the **ButtonKitty** button is clicked:

1. The **SoundKitty** sound is played, and
2. The device vibrates for **1000 milliseconds** (i.e., for 1 second).

Step 8 (Saving the project)

From the **Projects** menu, select **Save project** to save all changes online.
Then, from the same menu, export your project to your computer folder by selecting **Export selected project (.aia) to my computer**.

Step 9 (Packaging the application)

In this step, you will save/package your project in **.apk** format so that it is ready to be used on an Android device.
To do this, select from the **Build** menu the option
App (save .apk to my computer).

Step 10 (Live testing the application on an Android device via Wi-Fi)

From the **Connect** menu, select **AI Companion** in order to generate a QR code to be scanned by the Android device.

Note that this option requires:

- The computer used for programming and the Android device to be connected to the **same network**
- The Android device to be connected **wirelessly via Wi-Fi**
- The **MIT AI2 Companion** application to be installed on the Android device (required only the first time)

After successfully scanning the QR code, you can run the application directly on your device and verify that it works correctly.