MOBILE APPLICATION DEVELOPMENT

LABORATORY MANUAL AY2018_19_ E2CSE_SEM2

Created by

Mr. Satyanandaram N

MSIT,IIIT-Hyd

IIIT-RK Valley, RGUKT-AP

INDEX

References:

- https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-practicals/content/en/
- https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-concepts/content/en/

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 1 (Research document on MAD)

Survey on Existing Mobile Operating Systems

You have to prepare a research document on current Mobile devices and their operating systems. You have to answer the following questions

1. What are the several common myths related to mobile app development
2. What are the popular mobile operating systems
3. What are the advantages with Android OS

Along with the research you have to practice the core java concepts for Android application development

From the several Mobile operating systems, students have to practice application development on Android platform hence students have the sufficient knowledge on core java concepts

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 2 (Exploring Android Studio)

Install Android Studio:

You will learn to:-

- Install and use the Android IDE.
- Understand the development process for building Android apps.
- Create an Android project from a basic app template.

Task 1. Install Android Studio

Task 2: Create "Hello World" app

Task 3: Explore the project structure and layout

Task 4: Create a virtual device

Task 5: Run your app on an emulator

Task 6. Add log statements to your app

Task 7: Explore the AndroidManifest.xml file

Task 8. Explore the build.gradle file

Task 9. Run your app on a device

Make your First Interactive Application

You will learn:

- How to create interactive user interfaces in the Layout Editor, in XML, and programmatically.
- A lot of new terminology. Check out the Vocabulary words and concepts glossary for friendly definitions.

Task 1. Create the "Hello Toast" project

Task 2: Add views to "Hello Toast" in the Layout Editor

Task 3: Edit the "Hello Toast" Layout in XML

Task 4: Add on-click handlers for the buttons

Challenge: Even a simple app like Hello Toast can be the foundation of many scoring or product ordering apps. Write one app that would be of use to you, or try one of these examples:

- Create a coffee ordering app. Add buttons to change the number of coffees ordered. Calculate and display the total price.
- Create a scoring app for your favorite team sport. Make the background an image that represents that sport. Create buttons to count the scores for each team.

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 3 (Layouts, Views and Resources)

Using Layouts:

You will learn to:

- Use the layout editor in Android Studio
- Position views within a RelativeLayout
- Position views within a ConstraintLayout
- Create variants of the layout for landscape orientation and larger displays

Task 1: Change the layout to RelativeLayout

Task 2: Change the layout to ConstraintLayout

Task 3: Create layout variants

Working with TextView Elements:

You will learn to:

- Use XML code to add multiple TextView elements.
- Use XML code to define a scrolling view.
- Display free-form text with some HTML formatting tags.
- Style the TextView background color and text color.
- Include a web link in the text.

Task 1: Add several text views

Task 2: Add active web links and a ScrollView

Task 3: Scroll multiple elements

Coding Challenge:

Add another UI element—a Button—to the LinearLayout view group that is contained within the ScrollView. Make the Button appear below the article. The

user would have to scroll to the end of the article to see the button. Use the text "Add Comment" for the Button, for users to click to add a comment to the article. For this challenge, there is no need to create a button-handling method to actually add a comment; it is sufficient to just place the Button element in the proper place in the layout.

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 4 (Activities and Intents)

Create and Start Activities:

You will learn to:

- Create a new activity in Android studio.
- Define parent and child activities for "Up" navigation.
- Start activities with explicit intents.
- Pass data between activities with intent extras.

Task 1. Create the TwoActivities project

Task 2. Create and launch the second activity

Task 3. Send data from the main activity to the second activity

Task 4. Return data back to the main activity

Coding Challenge:

Challenge: Create an app with three buttons labelled: Text One, Text Two, and Text Three. When any of those buttons are clicked, launch a second activity. That second activity should contain a ScrollView that displays one of three text passages (you can include your choice of passages). Use intents to both launch the second activity and intent extras to indicate which of the three passages to display.

Activity Lifecycle and Instance State:

You will learn to:

- Understand the activity lifecycle, and when activities are created, pause, stop, and are destroyed.
- Understand the lifecycle callback methods associated with activity changes.
- Understand the effect of actions such as configuration changes that can result in activity lifecycle events.
- Retain activity state across lifecycle events.

Task 1. Add lifecycle callbacks to TwoActivities

Task 2. Save and restore activity state

Coding Challenge:

Challenge: Create a simple shopping list builder app with two activities. The main activity contains the list itself, which is made up of ten (empty) text views. A button on the main activity labelled "Add Item" launches a second activity that contains a list of common shopping items (Cheese, Rice, Apples, and so on). Use Buttons to display the items. Choosing an item returns you to the main activity, and updates an empty TextView to include the chosen item.

Use intents to pass information between the two activities. Make sure that the current state of the shopping list is saved when you rotate the device.

Activities and Implicit Intents:

You will learn to:

- Create implicit intents, and use their actions and categories.
- Use the ShareCompat.IntentBuilder helper class to easily create implicit intents for sharing data.
- Advertise that your app can accept implicit intents by declaring intent filters in the Andriod manifest

Task 1. Create new project and layout

Task 2. Implement open website

Task 3. Implement open location

Task 4. Implement share this text

Task 5. Receive implicit intents

Coding Challenge:

Challenge: In the last section's challenge you created a shopping list app builder with two activities: one to display the list, and one to pick an item. Add an EditText and a Button to the shopping list activity to locate a particular store on a map.

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 5 (Debugging and Testing your App)

Using the Debugger:

You will learn to:

- Run your app in debug mode in an emulator or on a device.
- Step through the execution of your app.
- Set and organize breakpoints.
- Examine and modify variables in the debugger.

Task 1. Create the SimpleCalc Project and App

Task 2. Run SimpleCalc in the Debugger

Task 3. Explore Debugger Features

Testing your App:

You will LEARN

- How organizing and running tests works in Android Studio
- What a unit test is, and how to write unit tests for your code.
- How to create and run local unit tests in Android Studio.

Task 1. Explore and run SimpleCalc in Android Studio

Task 2. Add more unit tests to CalculatorTest

Coding Challenge:

Challenge 1: Dividing by zero is always worth testing for, because it a special case in arithmetic. If you try to divide by zero in the current version of the SimpleCalc app, it behaves the way Java defined: Dividing a number by returns the "Infinity" constant (Double.POSITIVE_INFINITY). Dividing 0 by 0 returns the not a number constant (Double.NaN). Although these values are correct for Java, they're not necessarily useful values for the user in the app itself. How might you

change the app to more gracefully handle divide by zero? To accomplish this challenge, start with the test first -- consider what the right behavior is, and then write the tests as if that behavior already existed. Then change or add to the code so that it makes the tests come up green.

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 6 (User Input Controls)

Using Keyboards, Input Controls, Alerts, and Pickers:

You will Learn:

- Change the input methods to enable spelling suggestions, auto-capitalization, and password obfuscation.
- Change the generic on-screen keyboard to a phone keypad or other specialized keyboards.
- Add a spinner input control to show a dropdown menu with values, from which the user can select one.
- Add an alert with OK and Cancel for a user decision.
- Use date and time pickers and recording the selections.
- Use images as buttons to launch an activity.
- Add radio buttons for the user to select one item from a set of items

What you will DO

- Create new Android Studio projects to show keyboards, a spinner, an alert, and time and date pickers.
- Provide spelling suggestions when a user enters text, and automatically capitalize new sentences, by experimenting with the input method.
- Experiment with the input type attribute to change the on-screen keyboard to a special keyboard for entering email addresses, and then to a numeric keypad to force numeric entry.
- Add a spinner input control for the phone number field for selecting one value from a set of values.
- Create a new project with an alert dialog to notify the user to make a decision, such as OK or Cancel.
- Add the date picker and time picker to the new project, and use listeners to record the user's selection.
- Create a new project to use images as buttons.
- Create a second activity and add radio buttons for selecting an option.
- Set onClick handlers for the images used as buttons to launch a second activity

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 7 (Menus)

Using an Options Menu

What you will LEARN

- Adding menu items to the options menu.
- Adding icons for items in the options menu.
- Setting menu items to show in the action bar.
- Adding the event handlers for menu item clicks.

What you will DO

- Continue adding features to the Droid Cafe project from the previous practical.
- Add menu items to the options menu.
- Add icons for menu items to appear in the action bar.
- Connect menu item clicks to event handlers that process the click events.

Task 1: Add items to the options menu

Task 2: Add icons for menu items

Task 3: Handle the selected menu item

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 8 (Screen Navigation)

Using the App Bar and Tabs for Navigation

What you will LEARN

In this practical, you will learn to:

- Add the Up button to the app bar.
- Set up an app with tab navigation and swipe views.

What you will DO

- Continue adding features to the Droid Cafe project from the previous practical.
- Provide the Up button in the app bar to navigate to the previous screen within an activity.
- Create a new app with tabs for navigating activity screens that can also be swiped.

Task 1: Add an Up button for ancestral navigation

Task 2: Use tab navigation with swipe views

Coding Challenge:

Challenge 1: When you created the layout for tab navigation in the first step of the previous lesson, you established a Toolbar for the app bar in the activity_main.xml layout file. Add an options menu to the app bar as a challenge. To start, you will want to create the menu_main.xml file, and add menu items for the options menu. You must add at least one menu item, such as Settings.

You can inflate the options menu in the Toolbar by adding the onCreateOptionsMenu() method, as you did in a previous lesson on using the options menu.
Finally you can detect which options menu item is checked by using the onOptionsItemSelected()method.

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 9 (Recycler View)

Create a Recycler View:

What you will LEARN

In this practical, you will learn to:

- Use the RecyclerView class to display items in a scrollable list.
- Dynamically add items to the RecyclerView as they become visible through scrolling.
- Perform an action when the user taps a specific item.
- Show a floating action button and perform an action when the user taps it.

What you will DO

- Create a new application that uses a RecyclerView to display a list of items as a scrollable list and associate click behavior with the list items. Use a floating action button to let the user add items to the RecyclerView.

Task 1. Create and configure a WordList project

Task 2: Create a dataset

Task 3: Create a RecyclerView

Task 4: Add onClick to list items

Task 5: Add a FAB to insert items

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 10 (Async Task and Loaders)

Create an AsyncTask

What you will LEARN

During this practical, you will learn to:

- Add an AsyncTask to your app in order to run a task in the background, off of the UI thread.
- Identify and understand the benefits and drawbacks of using AsyncTask for background tasks.

What you will DO

During this practical, you will:

- Create a simple application that executes a background task using an AsyncTask.
- Run the app and see what happens when you rotate the screen.

Task 1: Set up the SimpleAsyncTask project

Task 2: Create the AsyncTask subclass

Task 3: Implement the final steps

Coding Challenge:

- Challenge: AsyncTask provides another very useful override method: `onProgressUpdate()`, which allows you to update the UI while the AsyncTask is running. Use this method to update the UI with the current sleep time. Look to the AsyncTask documentation to see how `onProgressUpdate()` is properly implemented. Remember that in the class definition of your AsyncTask, you will need to specify the data type to be used in the `onProgressUpdate()` method.

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 11 (Transferring Data)

SQLite Database:

In this practical you will learn to:

- Create and manage a SQLite database with an SQLiteOpenHelper.
- Implement insert, delete, update, and query functionality through your open helper.
- Use an adapter and custom click handler to let users interact with the database from the user interface.

What you will DO

- You start with an app that is the same as the RecyclerView word list app you created previously, with additional user interface elements already added for you, so that you can focus on the database code.

You will extend and modify the base app to:

- Implement a custom class to model your data.
- Create a subclass of SQLiteOpenHelper that creates and manages your app's database.
- Display data from the database in the RecyclerView.
- Implement functionality to add, modify, and delete data in the UI, and store the changes in the database.

Task 1. Create a data model for word list data
Task 2: Extend SQLiteOpenHelper to create and populate the database
Task 3: Display the data in the RecyclerView
Task 4: Edit words in the UI and store changes in the database
Task 5: Create UI Elements
Task 6: Handle Clicks

IIIT RK Valley,
Rajiv Gandhi University of Knowledge Technologies – A.P 516330

MOBILE APPLICATION DEVELOPMENT

Instructor: Mr. SATYANANDARAM N Lab Assistant: B SUDHAKAR
AY2018-19, SEM2, E2 LAB: 12 (Shared Preferences)

Shared Preferences

You will learn to:

- Identify what shared preferences are.
- Create a shared preferences file for your app.
- Save data to shared preferences, and read those preferences back again.
- Clear the data in the shared preferences.

What you will DO

- Add the ability to save, retrieve, and reset shared preferences to an app.

Task 1. Explore HelloSharedPrefs

Task 2. Save and restore data to shared preferences