Assignment 2

The deliverables are:

(1) This *Word* template, including your response where indicated. Name this Word doc <Last Name><First Name>\_Assignment1 as in DoeJohn\_Assignment1 and

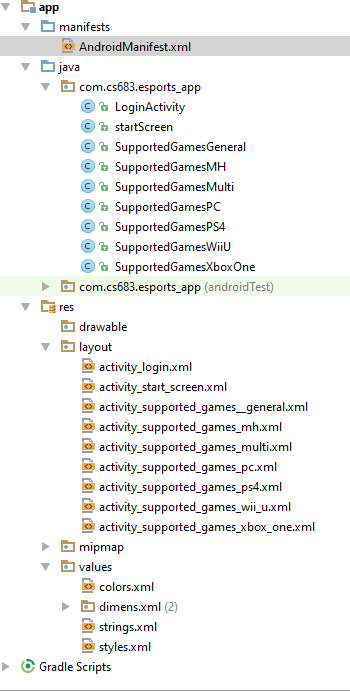
(2) Your source code, zipped.

# Part 1—Overall Project Description:

Provide a paragraph (no more) outlining the Android application you’d like to build for this course. Turn on tracking to show how and where this differed from your statement in Assignment 1.

The initial concept for my project is an application that allows people involved in eSports to find supported video games. Users are able to create an account, which requires their email address and password. Inputs will be the user’s account information (starting with desired email address and password) and device interaction. Outputs will be the specific games available for each gaming platform and a child login screen for email functionality. Once selected, the application displays the chosen gaming platform in a Toast notification. Child pages for each gaming platform were implemented.

# Part 2—Screenshot of the directory structure (project or package):

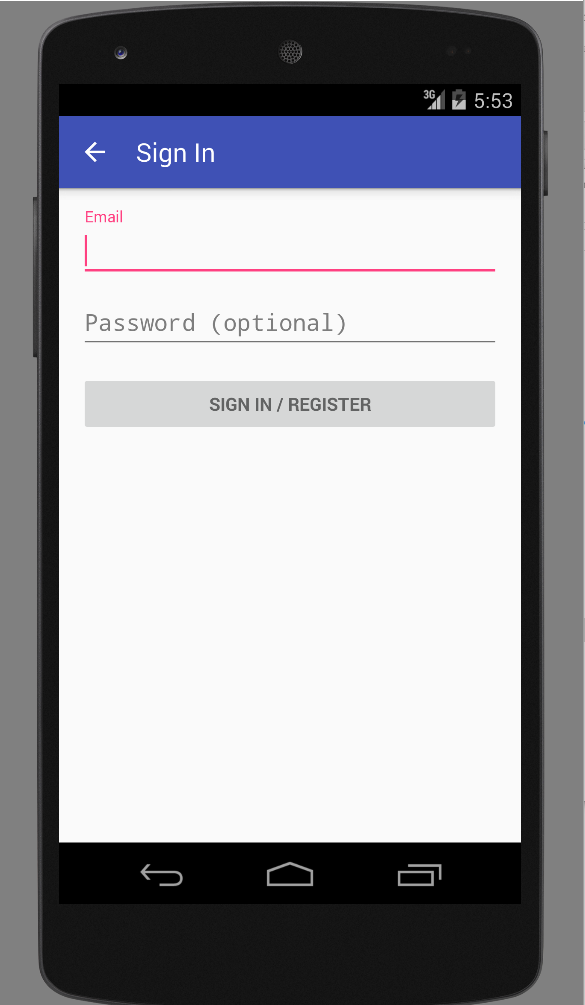


Part 3—Application Features with output and tests: *(maximum 2 pages of 12-point text, including figures)*

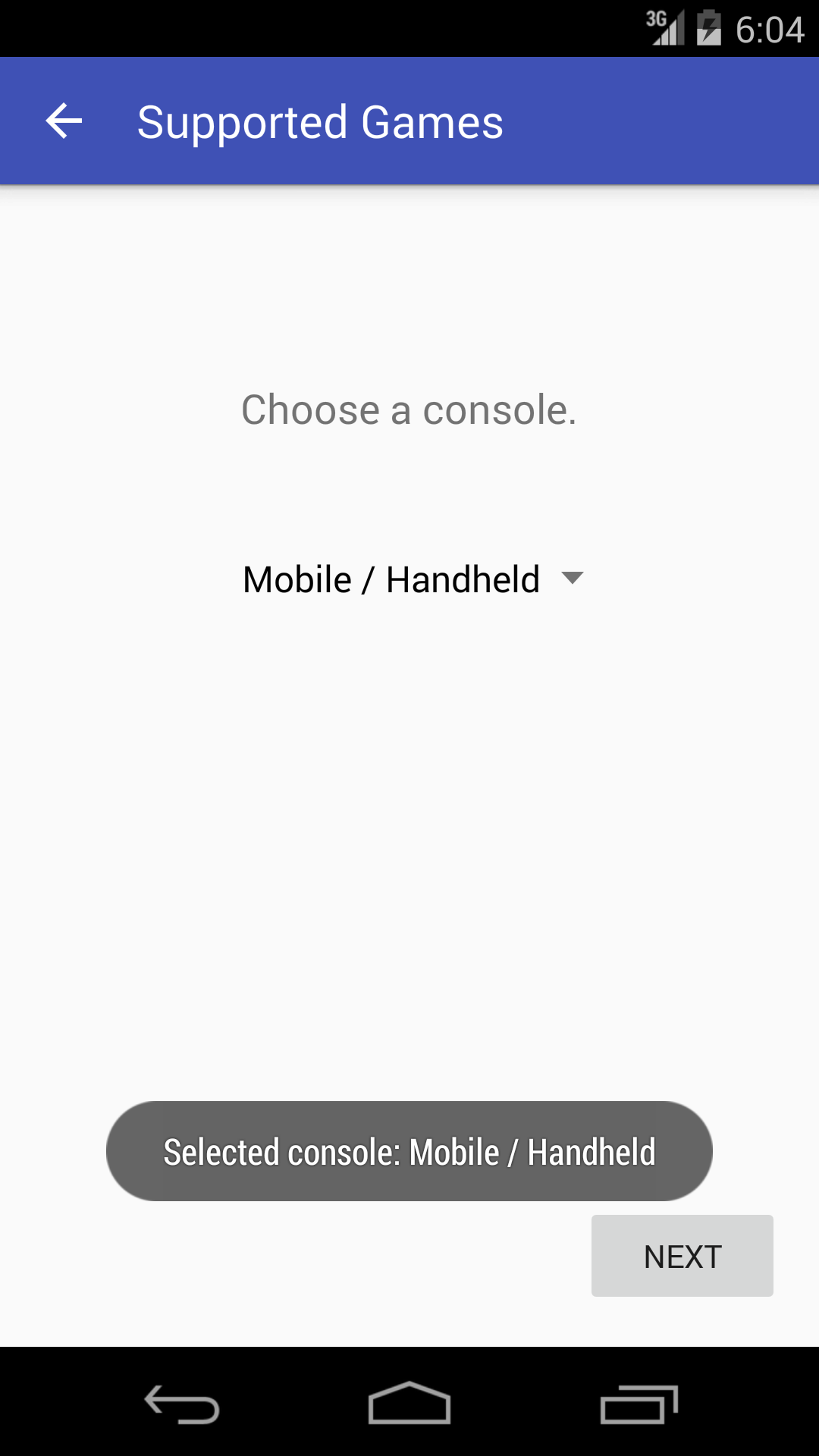
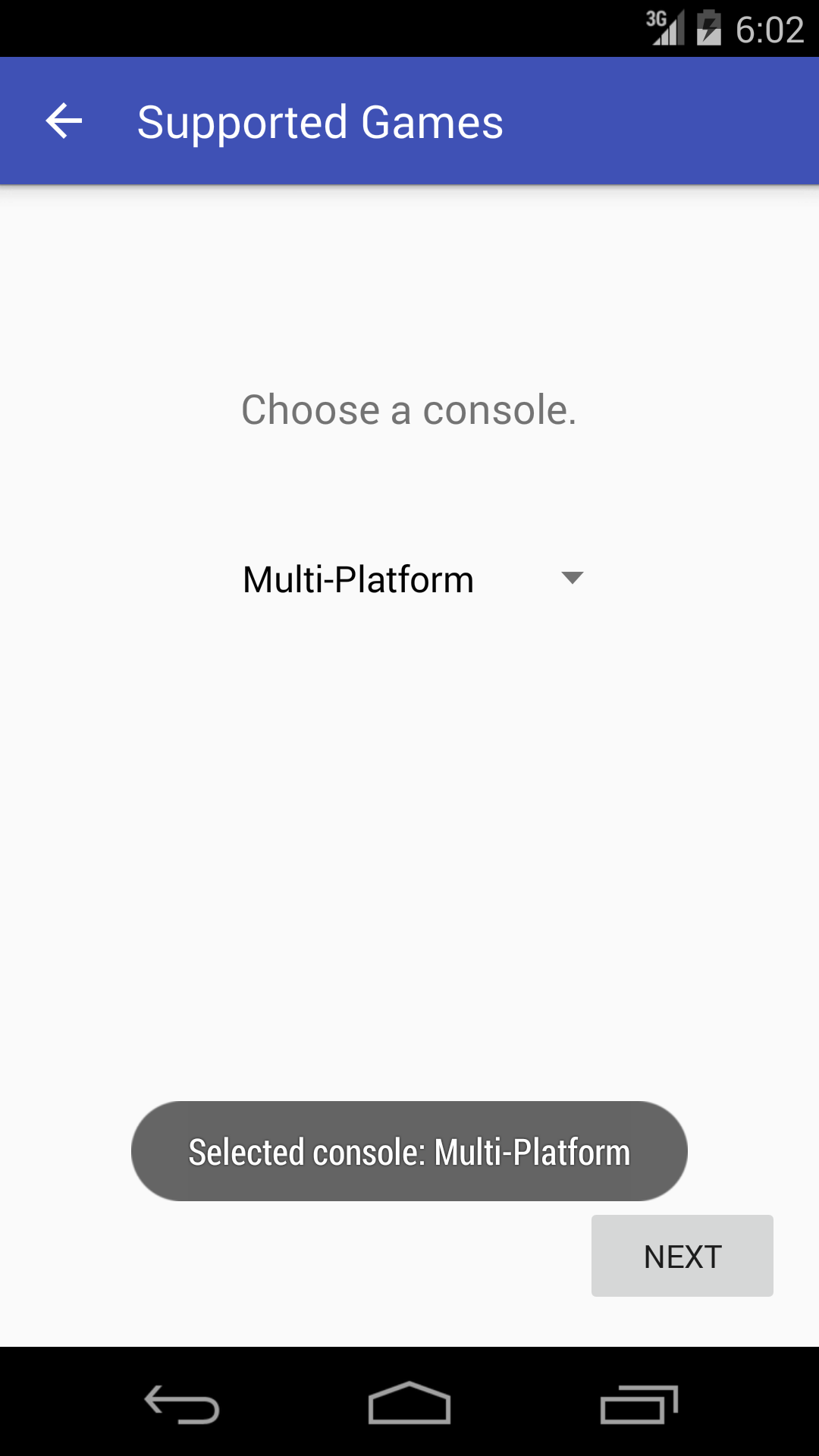
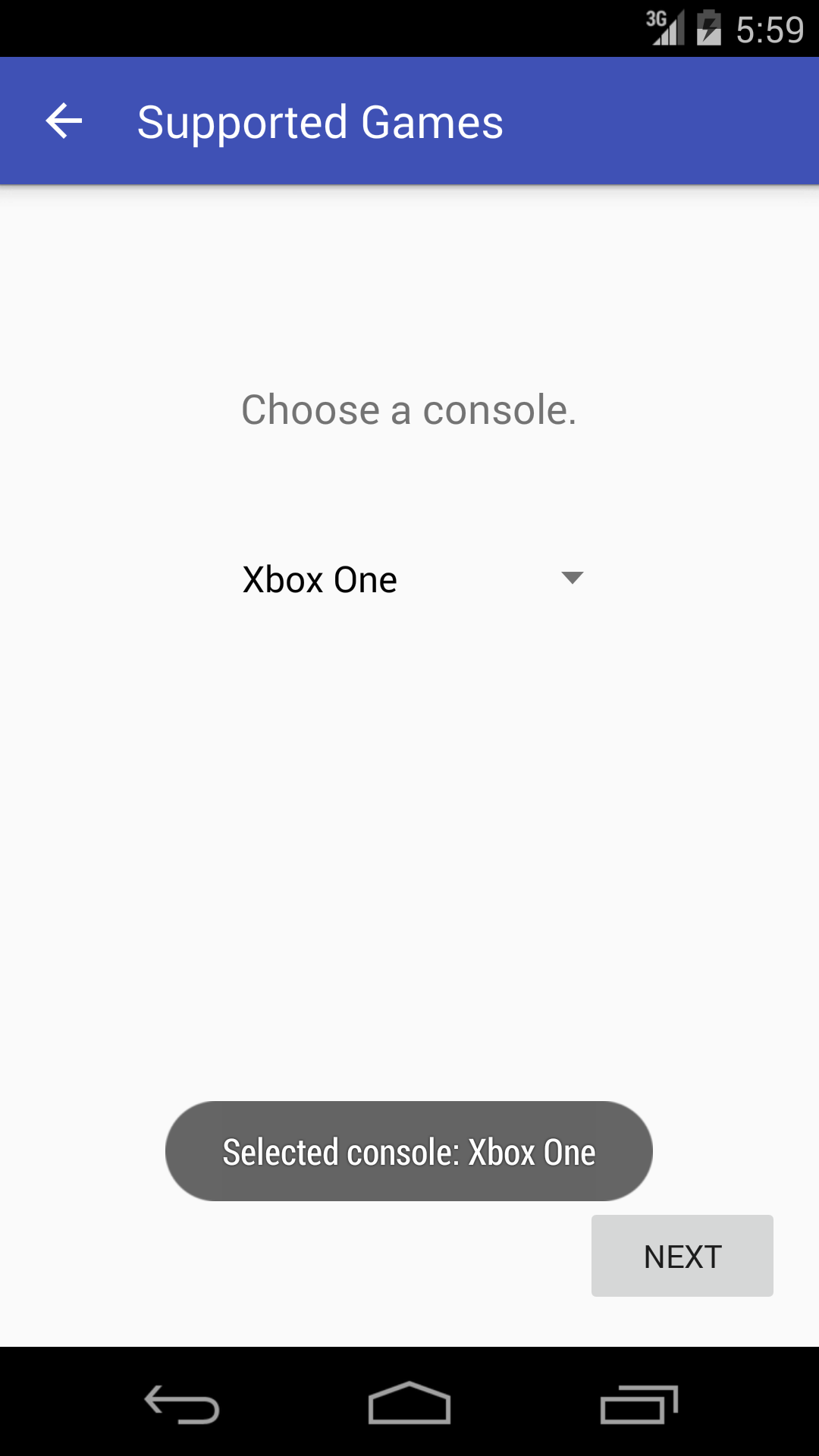
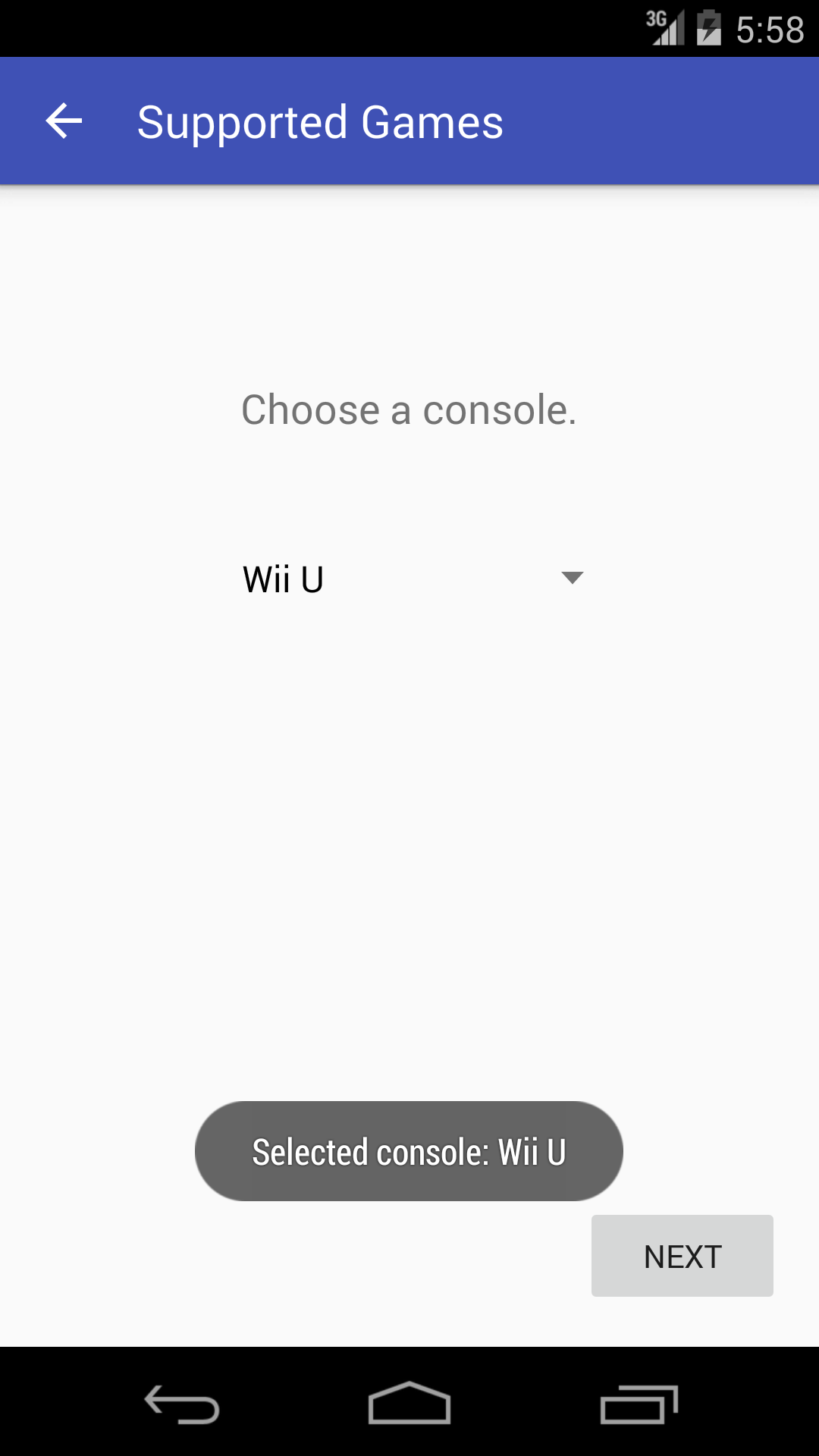
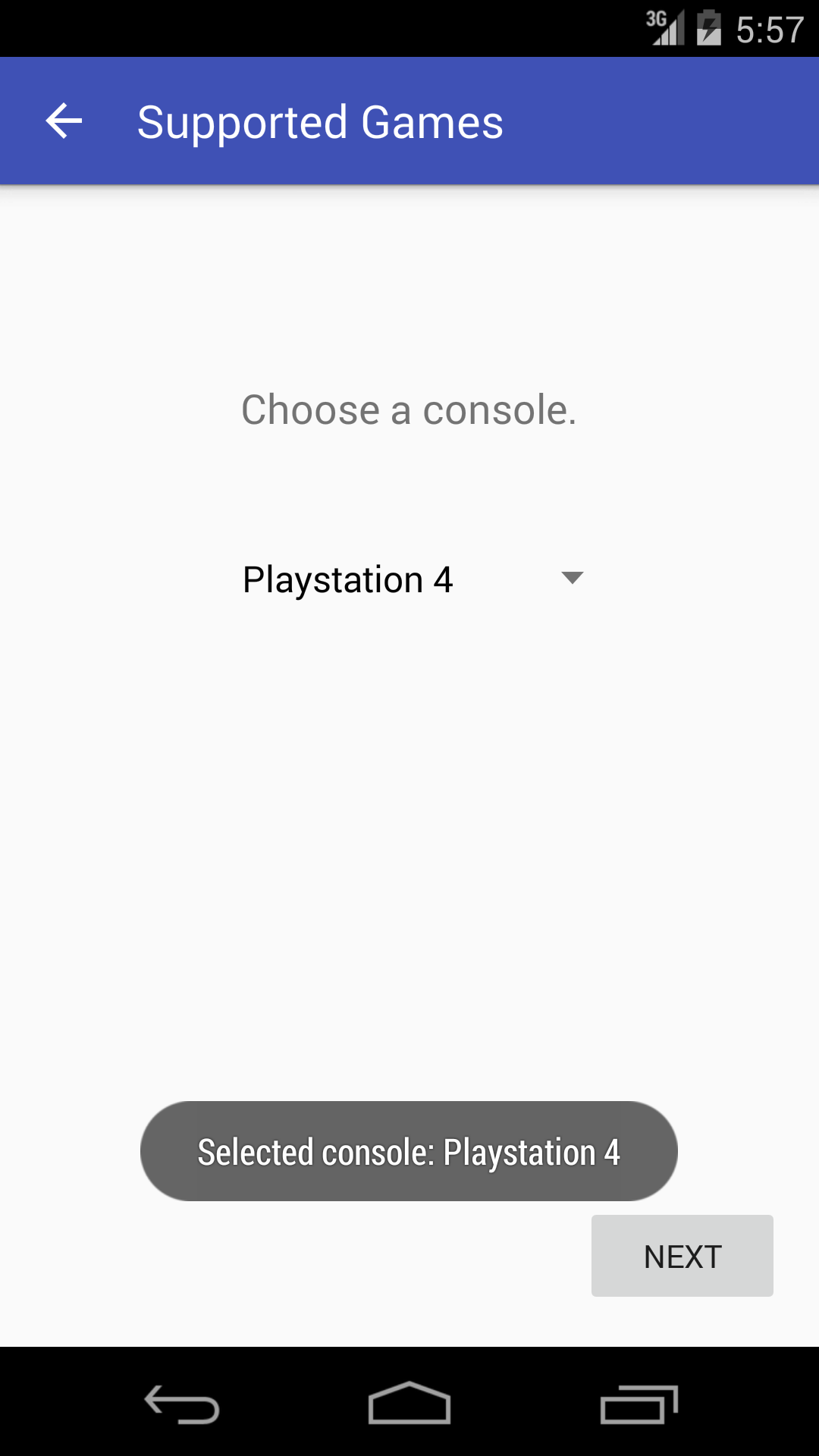
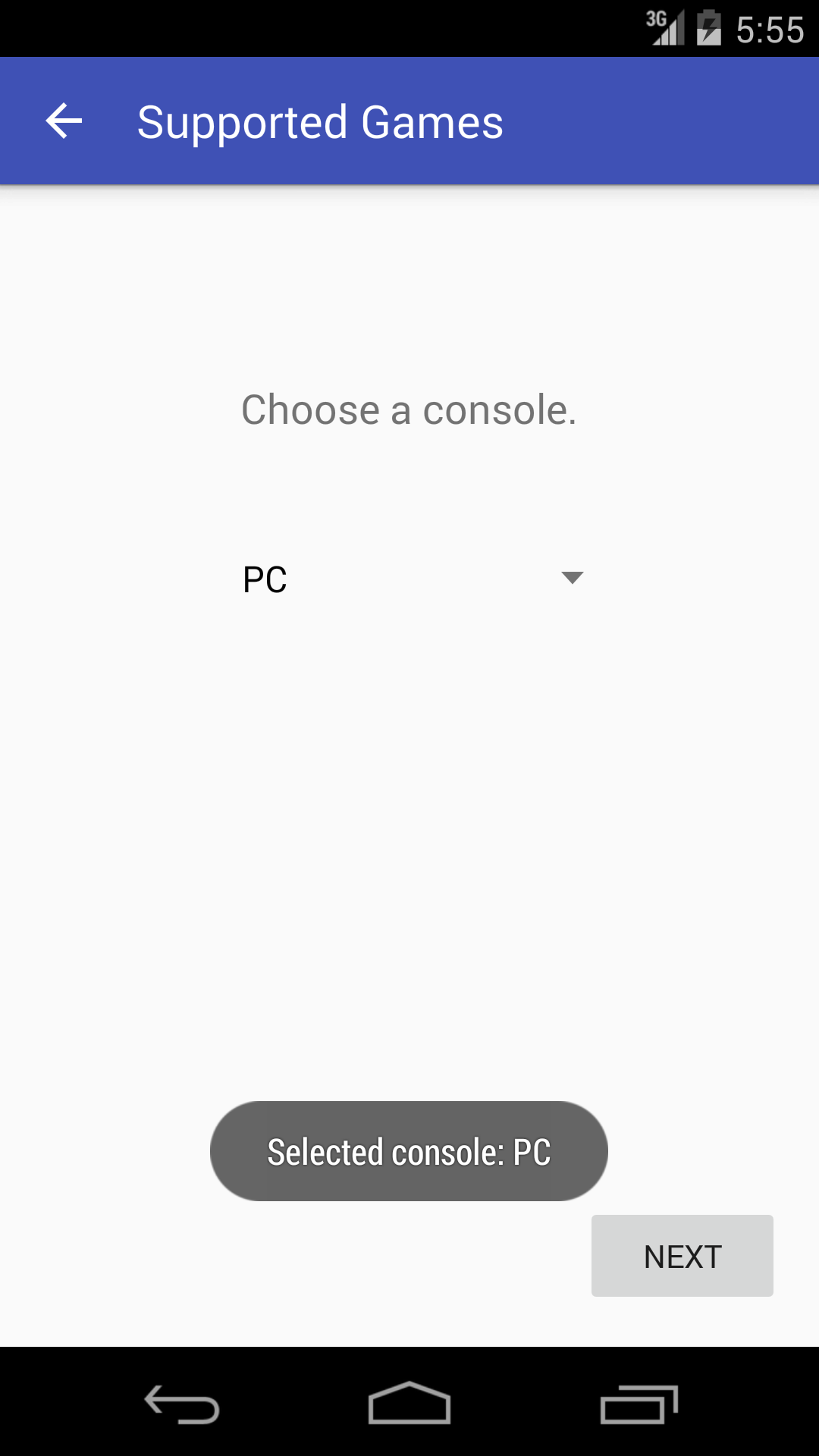
Specify the features of your application that you implemented. The priority is to get practice with as many as possible of the Android constructs covered in Module 2. Number your features 1, 2, 3, … . Each feature must accompanied by output or screen shot that show you accomplished it.

Application Feature 1: E-mail Login Screen

The “Sign In / Register” button at the Launcher screen has now been linked to a fully functional form, prompting users to enter their e-mail address and password to sign in or register for full usage of the application.

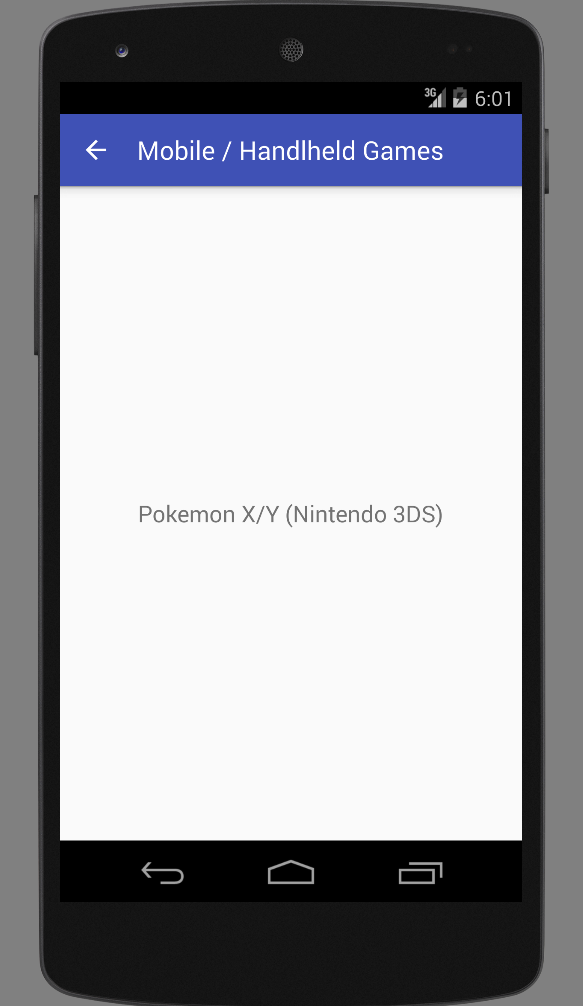
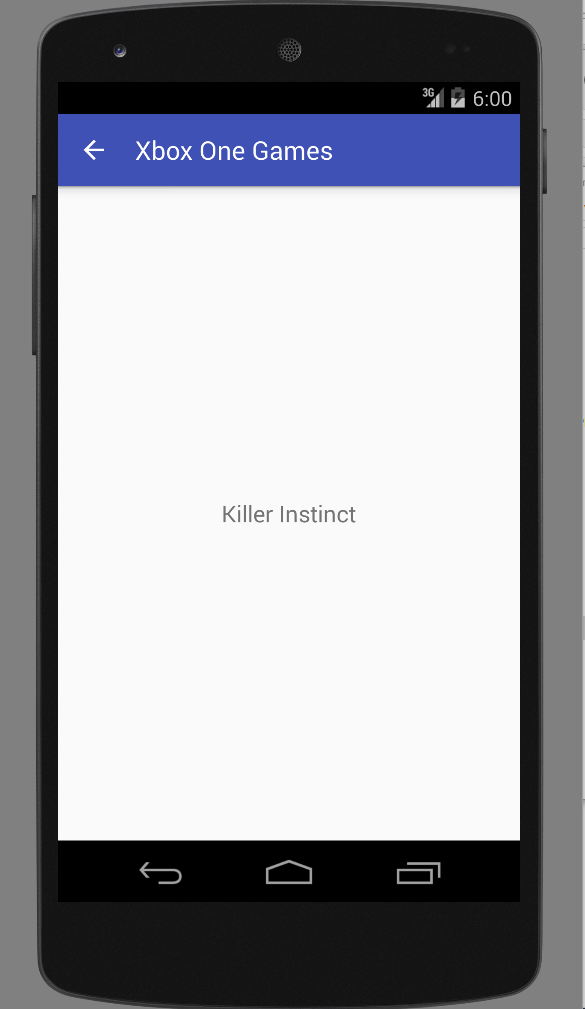
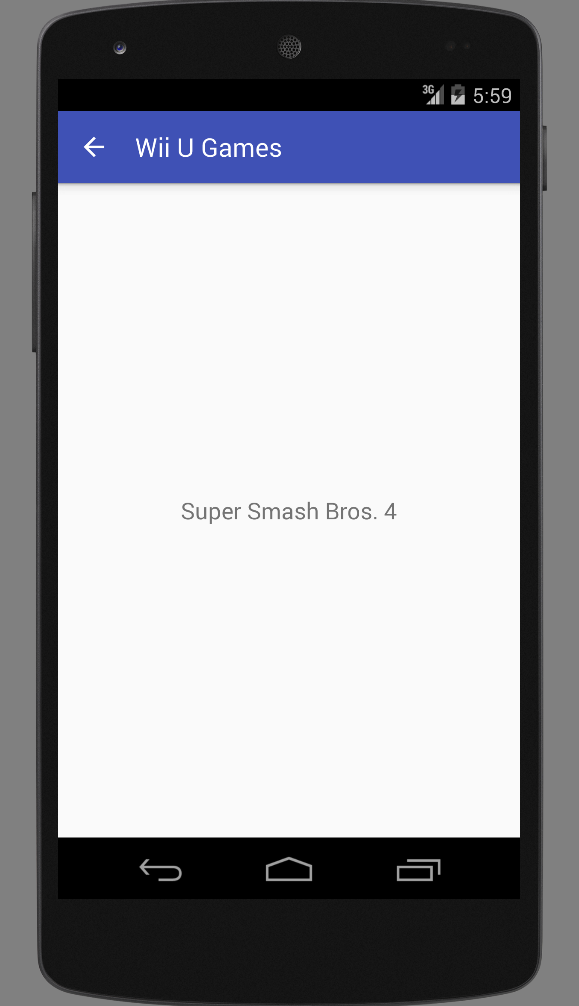
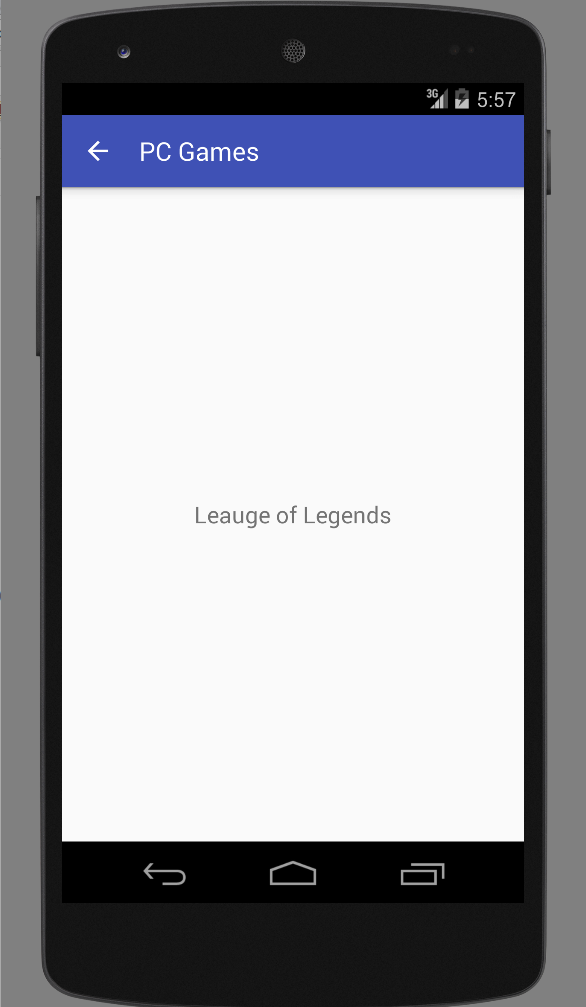


Application Feature 2: Toast notifications

When on the “Supported Games” screen, clicking on all items but “(Select)” in the drop down menu will trigger a Toast notification, specifying which console has been selected.

Application Feature 3: Child pages for gaming platforms

Child pages for all drop-down menu items except “(Select)” were created and populated with a game that aligns with the specifications determined by the menu category.



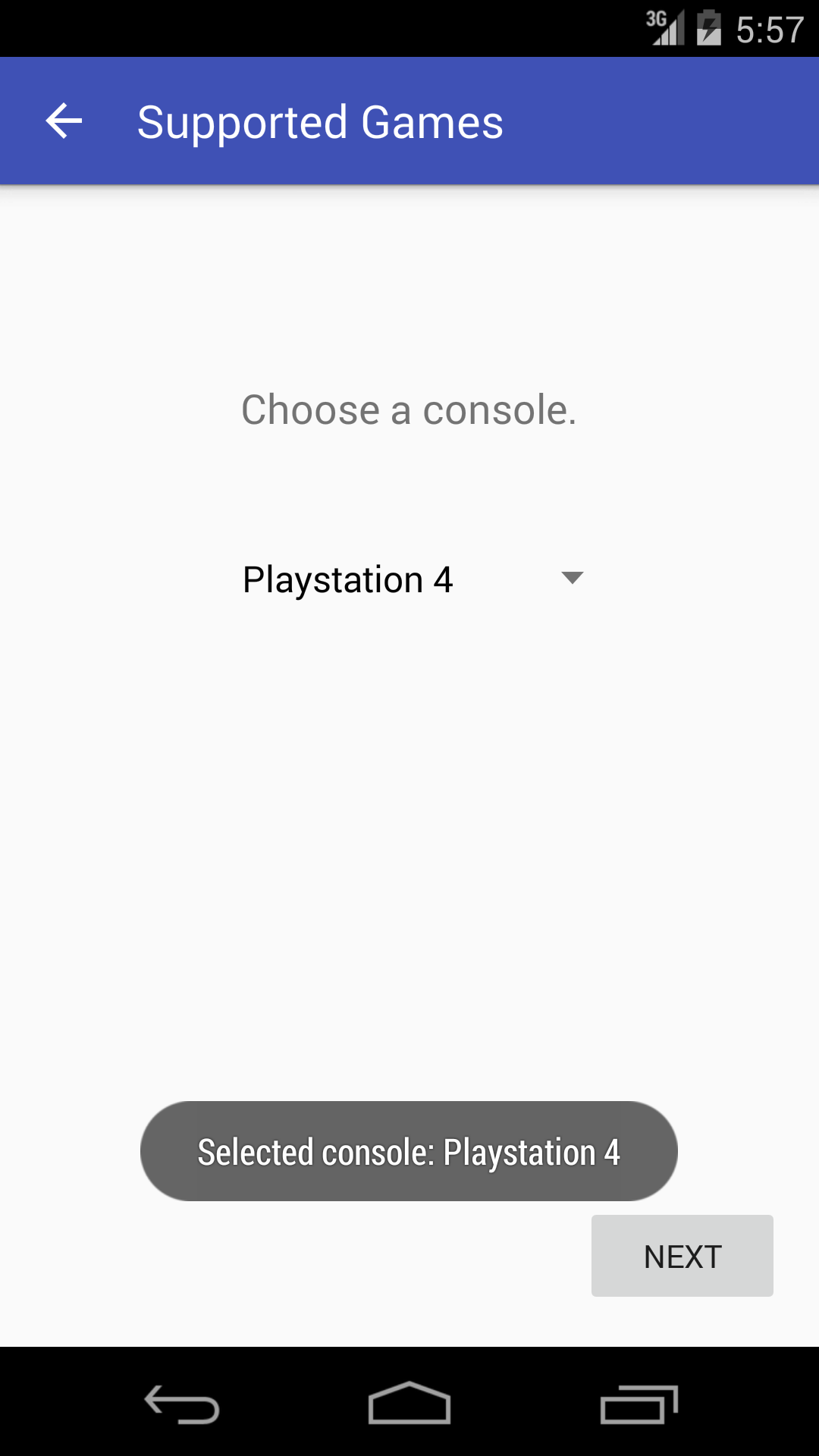
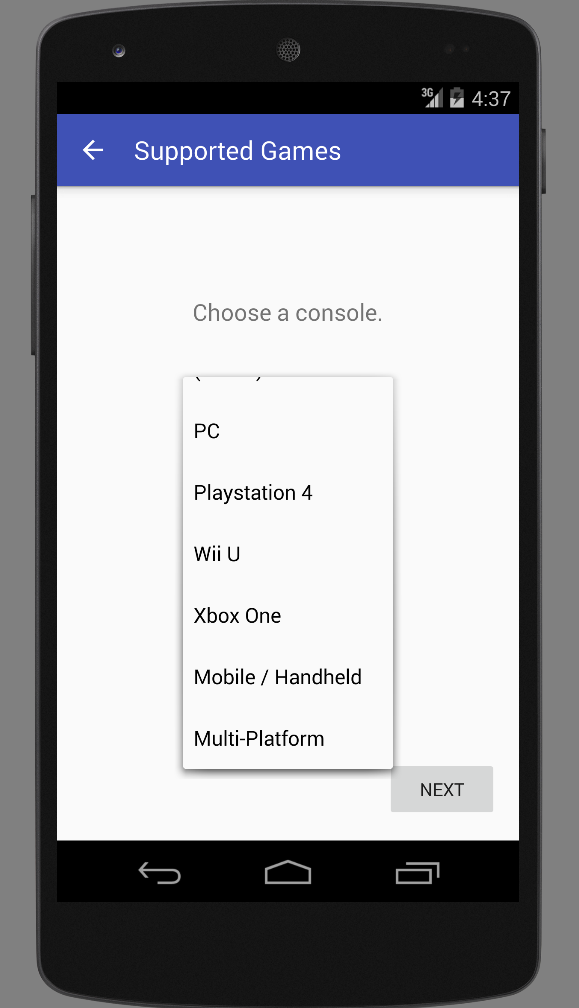
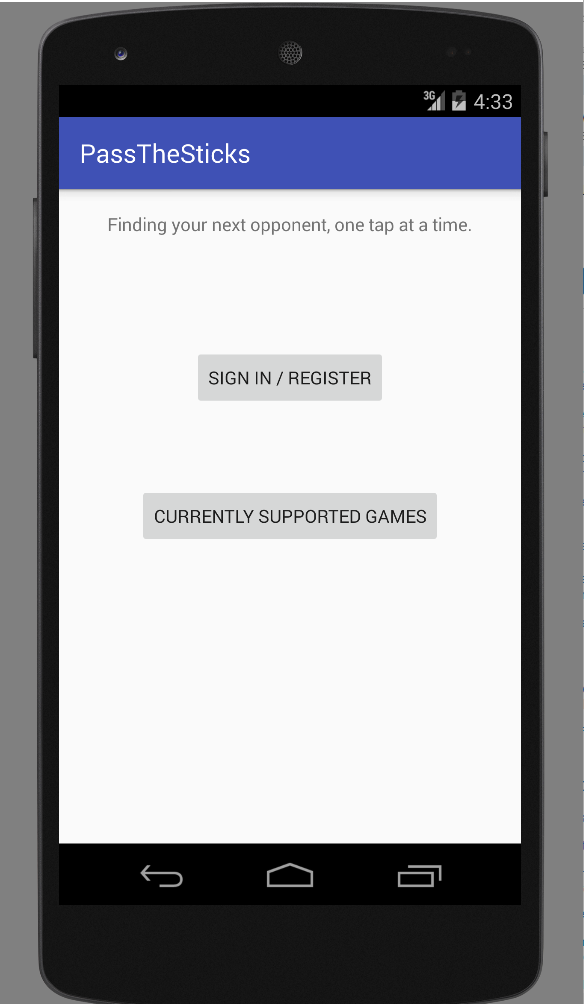
# Part 4—List of Android Elements: *(maximum 1 page of 12-point text)*

Use as many of the Android elements (e.g., built-in keywords, data types or functions) as possible covered in module 2. Provide lines from your code and where they were used once.

Android Feature 1: UI Development Process (Relative Layout + Corresponding Features)

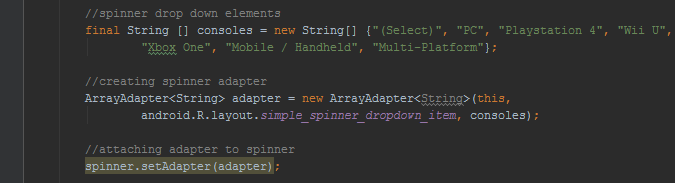


Android Feature 2: Event Handling (Buttons)



<----------- Replace this with UI or test showing this Android feature implemented ----------->

Android Feature 3: Creating Instances on the Fly

<----------- Replace this with UI or test showing this Android feature implemented ----------->

# Part 5—Example Code

Provide a page—in 10-point Ariel Narrow font—of your best (clearly documented)[[1]](#footnote-1) code. You may precede it with a paragraph (not counted in the page limit) explaining where and how it fits with the rest of your project.

The code below contains the majority of the Java code in the “SupportedGamesGeneral” activity. This page contains links to the pertinent child pages for each console, functionality for the spinner, adapter for the spinner, and the included “NextPage” button.

public class SupportedGamesGeneral extends AppCompatActivity {

Button buttonNextPage;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_supported\_games\_\_general*);

buttonNextPage = (Button) findViewById(R.id.*buttonNextPage*);

//declare spinner

Spinner spinner = (Spinner) findViewById(R.id.*consoleDynamicSpinner*);

//spinner drop down elements

final String [] consoles = new String[] {"(Select)", "PC", "Playstation 4", "Wii U",

"Xbox One", "Mobile / Handheld", "Multi-Platform"};

//creating spinner adapter

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,

android.R.layout.*simple\_spinner\_dropdown\_item*, consoles);

//attaching adapter to spinner

spinner.setAdapter(adapter);

//spinner on click + toast settings

spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parent, View view, final int position, long id) {

parent.getItemAtPosition(position);

if (position > 0) {

TextView myText = (TextView) view;

Toast.*makeText*(parent.getContext(), "Selected console: " + myText.getText(), Toast.*LENGTH\_SHORT*).show();

}

buttonNextPage.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//linking to PC activity list

if (position == 1) {

Intent gamesPC = new Intent(SupportedGamesGeneral.this, SupportedGamesPC.class);

startActivity(gamesPC);

}

//linking to Playstation 4 activity list

else if (position == 2) {

Intent gamesPS4 = new Intent(SupportedGamesGeneral.this, SupportedGamesPS4.class);

startActivity(gamesPS4);

}

//linking to Wii U activity list

else if (position == 3) {

Intent gamesWiiU = new Intent(SupportedGamesGeneral.this, SupportedGamesWiiU.class);

startActivity(gamesWiiU);

}

//linking to Xbox One activity list

else if (position ==4){

Intent gamesXboxOne = new Intent(SupportedGamesGeneral.this,

SupportedGamesXboxOne.class);

startActivity(gamesXboxOne);

}

//linking to Mobile / Handheld activity list

else if (position == 5) {

Intent gamesMH = new Intent (SupportedGamesGeneral.this, SupportedGamesMH.class);

startActivity(gamesMH);

}

//linking to multi-platform games

else {

Intent gamesMulti = new Intent (SupportedGamesGeneral.this, SupportedGamesMulti.class);

startActivity(gamesMulti);

}

}

});

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

}

});

}

Don’t forget to attach your source code, zipped.

# Below is the matrix that will be used to evaluate your response:



A regular A translates as 95, A-=93, B+=87, B=85, B-=83, C+=77, C=75, C-=73, D+=67, F=0 etc. To get an A grade for the course, your weighted average should be >93. A-:>=90. B+:>=87. B:>83. B-:>=80. C+:>=77. C:>73. C-:>=70 etc.

1. Document intentions—don’t paraphrase code. Nontrivial functions should have (an informal) *Intent* statement*,* (precise) *Preconditions* (if any), *Returns* (if any),and *Postconditions* (always). Each block of code should be preceded by its intended objectives. [↑](#footnote-ref-1)