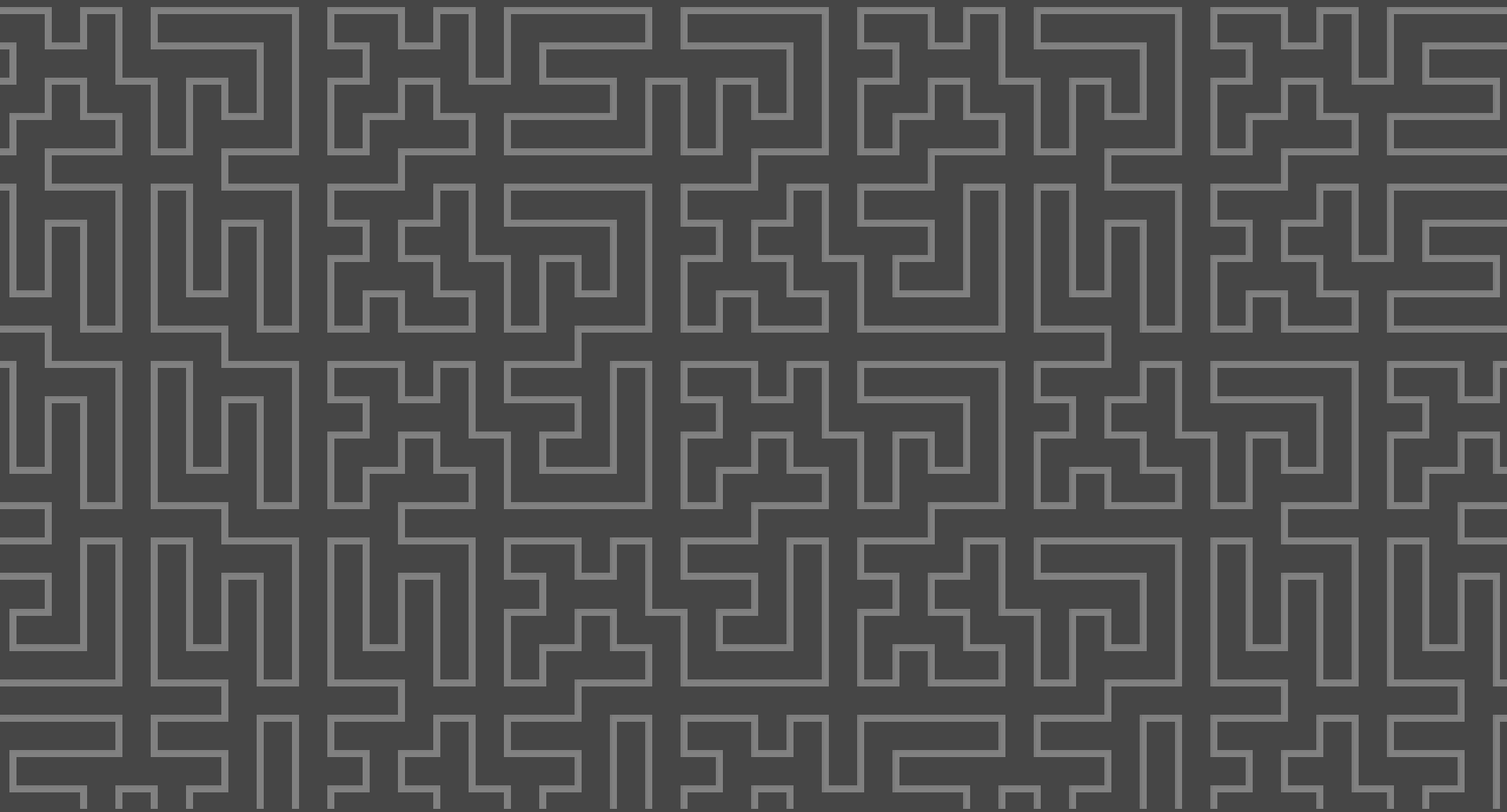


MOBILE DEVELOPMENT



File handling

Content



- ☐ Shared preferences
- ☐ Exception handling
- ☐ Write file
- ☐ Reading file

Data and Storage



- Shared Preferences
- File Storage
 - Internal
 - External
- Databases

Shared Preferences

- ❑ Save relatively small amount of data
- ❑ Key-value pairs
- ❑ You can use in multiple scenarios
- ❑ Example: Highest user score, ring tone, color scheme...

Shared Preferences

```
static final String SHARED_PREF = "PrefDemo";  
SharedPreferences sharedPref;  
SharedPreferences.Editor editor;
```

```
<Button
```

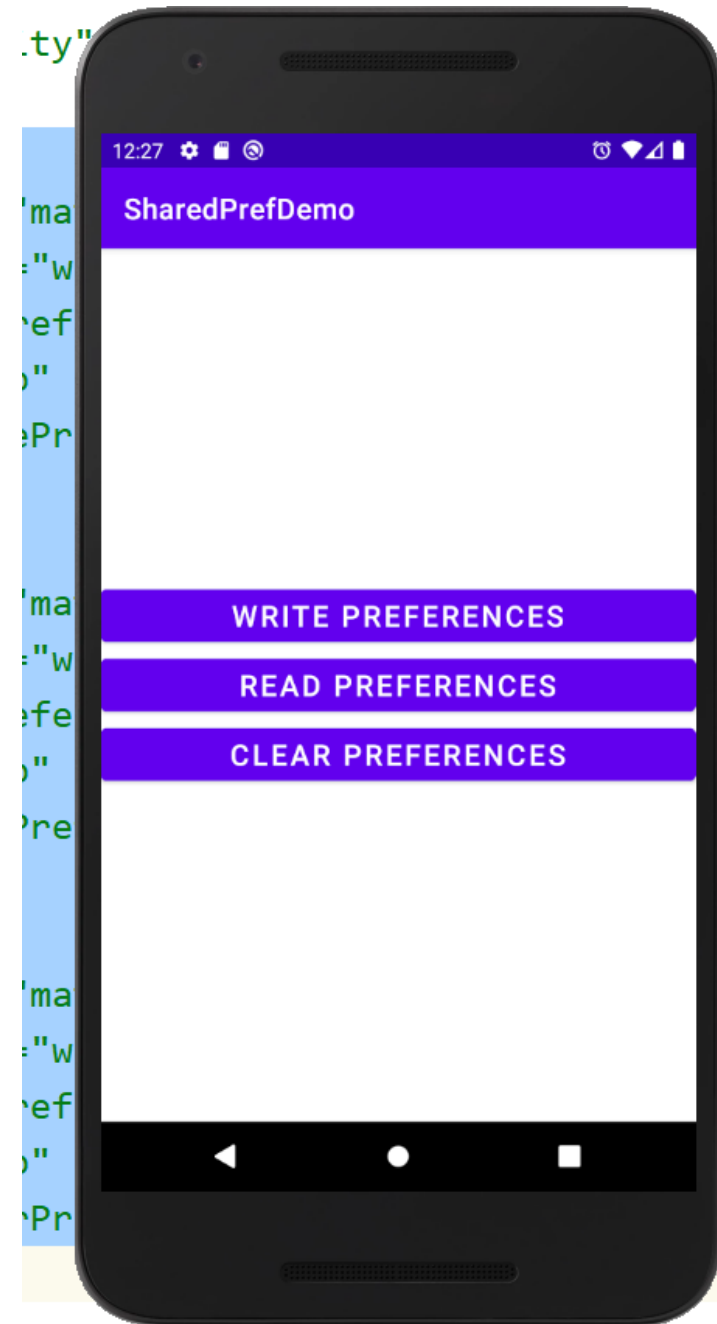
```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Write Preferences"  
    android:textSize="20sp"  
    android:onClick="writePref"  
/>
```


```
<Button
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Read Preferences"  
    android:textSize="20sp"  
    android:onClick="readPref"  
/>
```

```
<Button
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Clear Preferences"  
    android:textSize="20sp"  
    android:onClick="clearPref"  
/>
```





```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    sharedPref = getSharedPreferences(SHARED_PREF, MODE_PRIVATE);  
    editor = sharedPref.edit();  
}
```

Write Shared Preferences

```
public void writePref(View view) {  
    editor.putString("hello", "Welcome to shared Preferences");  
    editor.putInt("int_val", 2001);  
    editor.putBoolean("bool_val", false);  
    editor.apply();  
}
```


Read Shared Preferences

```
public void readPref(View view) {  
    int intVal = sharedPref.getInt("int_val",0);  
    boolean boolVal = sharedPref.getBoolean("bool_val", true);  
    String str = sharedPref.getString("hello","Default value");  
  
    Toast.makeText(getApplicationContext(),  
        str+"\n"+boolVal+"\n"+intVal,  
        Toast.LENGTH_LONG).show();  
}
```

Clear Shared Preferences

```
public void clearPref(View view) {  
    editor.clear();  
    editor.apply();  
    Toast.makeText(getApplicationContext(),  
        "All Preferences Cleared",  
        Toast.LENGTH_LONG).show();  
}
```

Exceptions-Review



- An exception is an unexpected or error condition.
- Examples:
 - Issue a command to read a file, but file does not exist.
 - Program asks for user input, but user enters wrong or invalid data.
- These errors are called exceptions. OO techniques manage such errors, with a group of methods known as exception handlers.

Exceptions Types



- Unchecked Exception
- Checked Exception

Unchecked Exception



- The exceptions that are not checked at compile time are called unchecked exceptions
- Example
 - `ArithmeticException`
 - `ArrayIndexOutOfBoundsException`
 - `NullPointerException`

Checked Exception



- Exceptions that are checked at compile-time are called checked exceptions
- Force the programmer to write a Exception handler at compile time
- Examples
 - IOException
 - FileNotFoundException
 - ParseException

Exception Handling in JAVA



- In java exception handling mechanism is based on following keywords
 - try
 - catch
 - finally
 - throws

File Handling in Java

- **java.io** library
- File
- Reading or writing large amounts of data
- File Location vary across devices
- File.getAbsolutePath() - give absolute path of file -> Device File Explorer

File handling-Save Steps

```
FileOutputStream fos;
```

```
fos =openFileOutput(FILE_NAME,MODE_PRIVATE|MODE_APPEND);
```

```
fos.write(str.getBytes());
```

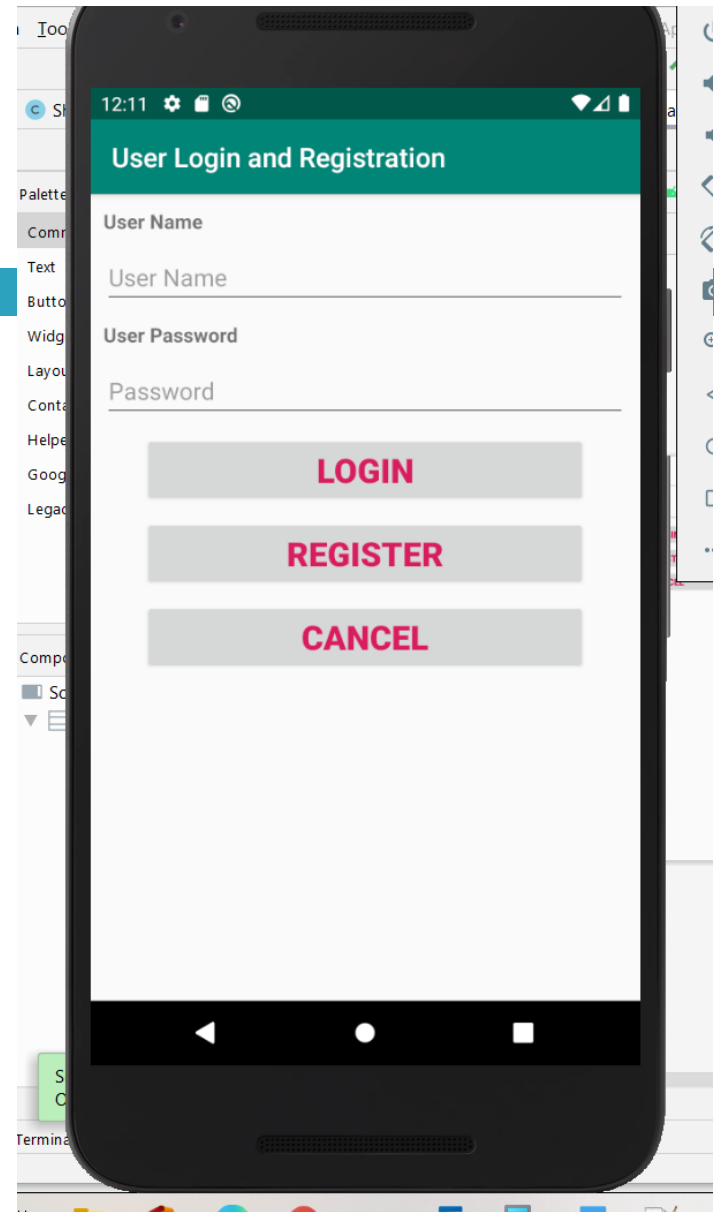
```
fos.close();
```

File handling-Read Steps

```
FileInputStream fis  
  
fis = openFileInput(FILE_NAME);  
  
BufferedReader reader = new BufferedReader(new InputStreamReader(fis));  
  
String str = reader.readLine();
```

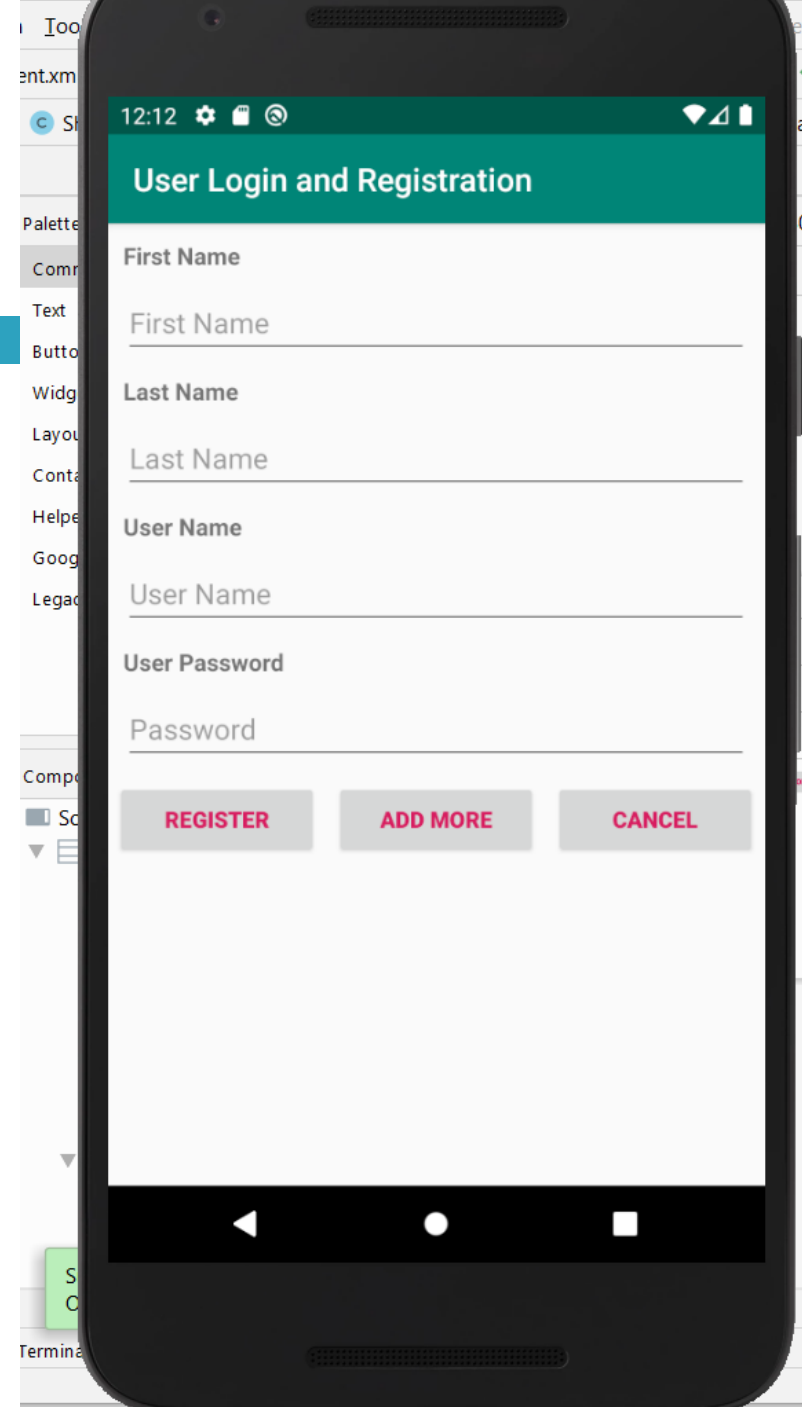
Exercise

- Use interface which you designed during week 1 and implement save data to local file.



Exercise

- Design interfaces for login and registration app then the app should read and Write data to file.



References

- ❑ <https://www.geeksforgeeks.org/shared-preferences-in-android-with-examples/>
- ❑ <https://developer.android.com/guide/topics/data>
- ❑ <https://developer.android.com/training/data-storage/shared-preferences>