Vidzemes augstskola

Inženierzinātņu fakultāte

**Mobilās programmatūras inženierija 1**

grupa 3

Praktiskais darbs #3

Valmiera, 2025

**Saturs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kontakti un autors** | | | |
| Vārds, Uzvārds | Grupa | Pozīcija | Kontakti (e-mail) |
| Linda Brante | Grupa #3 | Koordinators | [linda.brante@va.lv](mailto:linda.brante@va.lv) |
| Arturs Siliņš | Grupa #3 | Izstrādātājs | [arturs.silins@va.lv](mailto:arturs.silins@va.lv) |
| Jānis Petrovs | Grupa #3 | Izstrādātājs | [janis.petrovs@va.lv](mailto:janis.petrovs@va.lv) |
|  |  |  |  |
|  |  |  |  |

**Ievads**

Šis praktiskais darbs ir daļa no Mobilās programmatūras inženierijas kursa, kurā tiek izstrādāta Android lietotne ar divām aktivitātēm, izmantojot DataStore datu saglabāšanai un tēmu mainīšanu. Lietotne ir izveidota, izmantojot Kotlin valodu un Android Studio vidi.

**Mērķis**

Izveidot lietotni “Repository” ar divām saistītām aktivitātēm: Pirmā aktivitāte sastāv no teksta ievades lauka, pogas “Save”, pogas “Go to 2-nd” un nolaižamā saraksta ar trīs vērtībām (Default, Dark theme, Light theme). Poga “Go to 2-nd” aizved lietotāju uz otro aktivitāti; poga “Save” saglabā ievadīto vērtību DataStore; ja DataStore ir pieejama vērtība lietotnes palaišanas laikā, tā tiek parādīta ievades laukā; izvēloties “Dark Theme”, lietotne maina ierīces tēmu uz tumšu, izvēloties “Light theme”, maina uz gaišu. Otrā aktivitāte sastāv no teksta lauka, pogas “Read storage”, kas nolasa un parāda saglabāto vērtību, vai parāda Toast “Nothing found”, ja DataStore ir tukšs. Poga “Back” atgriež uz pirmo aktivitāti un notīra backstack. Lietotne atbalsta Dark mode ar brīvi izvēlētām krāsām un Material Design principiem.

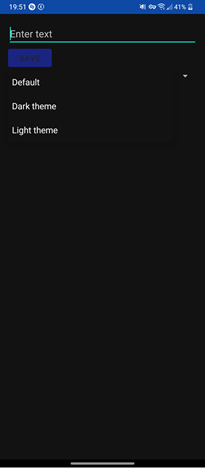
**Lietotnes Ekrānšāviņš**



Pirmais ekrānšāviņš rāda pirmo aktivitāti: Virsū ir teksta ievades lauks ar hint "Enter text", zem tā poga "Save", tad spinner ar izvēlētām tēmām, un poga "Go to 2-nd". Fons ir atbilstošs izvēlētajai tēmai (piemēram, gaišs ar violetām krāsām light mode).



Otrais ekrānšāviņš rāda otro aktivitāti: Virsū teksta lauks ar "No data", zem tā poga "Read storage", un poga "Back". Ja nolasīts teksts, tas parādās teksta laukā; ja tukšs, Toast "Nothing found".



Trešais ekrānšāviņš rāda tēmas maiņu.

**Kods**

**MainActivity.kt**

package com.example.repositoryg\_group3

import android.content.Intent

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.ArrayAdapter

import android.widget.Button

import android.widget.EditText

import android.widget.Spinner

import androidx.appcompat.app.AppCompatDelegate

import kotlinx.coroutines.CoroutineScope

import kotlinx.coroutines.Dispatchers

import kotlinx.coroutines.launch

import com.example.repositoryg\_group3.DataStoreManager

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val editText = findViewById<EditText>(R.id.editText)

val saveButton = findViewById<Button>(R.id.saveButton)

val goToSecondButton = findViewById<Button>(R.id.goToSecondButton)

val themeSpinner = findViewById<Spinner>(R.id.themeSpinner)

// Load saved text from DataStore

CoroutineScope(Dispatchers.IO).launch {

val savedText = DataStoreManager.getText(this@MainActivity)

if (savedText != null) {

editText.setText(savedText)

}

}

// Setup theme spinner

val themes = arrayOf("Default", "Dark theme", "Light theme")

val adapter = ArrayAdapter(this, android.R.layout.simple\_spinner\_item, themes)

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item)

themeSpinner.adapter = adapter

themeSpinner.onItemSelectedListener = object : android.widget.AdapterView.OnItemSelectedListener {

override fun onItemSelected(parent: android.widget.AdapterView<>?, view: android.view.View?, position: Int, id: Long) {

when (themes[position]) {

"Dark theme" -> AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_YES)

"Light theme" -> AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO)

else -> AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_FOLLOW\_SYSTEM)

}

delegate.applyDayNight()

}

override fun onNothingSelected(parent: android.widget.AdapterView<>?) {}

}

// Save button functionality

saveButton.setOnClickListener {

val text = editText.text.toString()

CoroutineScope(Dispatchers.IO).launch {

DataStoreManager.saveText(this@MainActivity, text)

}

}

goToSecondButton.setOnClickListener {

val intent = Intent(this, SecondActivity::class.java)

startActivity(intent)

}

}

}

**activity\_main.xml**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<EditText

android:id="@+id/editText"

android:layout\_width="match\_parent"

android:layout\_height="48dp"

android:hint="@string/edit\_text\_hint"

android:inputType="text"

android:autofillHints="@string/edit\_text\_hint" />

<Button

android:id="@+id/saveButton"

android:layout\_width="wrap\_content"

android:layout\_height="48dp"

android:text="@string/save\_button\_text" />

<Spinner

android:id="@+id/themeSpinner"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<Button

android:id="@+id/goToSecondButton"

android:layout\_width="wrap\_content"

android:layout\_height="48dp"

android:text="@string/go\_to\_second\_button\_text" />

</LinearLayout>

**SecondActivity.kt**

package com.example.repositoryg\_group3

import android.content.Intent

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.Button

import android.widget.TextView

import android.widget.Toast

import kotlinx.coroutines.CoroutineScope

import kotlinx.coroutines.Dispatchers

import kotlinx.coroutines.launch

import com.example.repositoryg\_group3.DataStoreManager

class SecondActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_second)

val textView = findViewById<TextView>(R.id.textView)

val readButton = findViewById<Button>(R.id.readButton)

val backButton = findViewById<Button>(R.id.backButton)

// Read button functionality

readButton.setOnClickListener {

CoroutineScope(Dispatchers.IO).launch {

val savedText = DataStoreManager.getText(this@SecondActivity)

runOnUiThread {

if (savedText != null) {

textView.text = savedText

} else {

Toast.makeText(this@SecondActivity, "Nothing found", Toast.LENGTH\_SHORT).show()

}

}

}

}

// Back button functionality

backButton.setOnClickListener {

val intent = Intent(this, MainActivity::class.java)

intent.flags = Intent.FLAG\_ACTIVITY\_NEW\_TASK or Intent.FLAG\_ACTIVITY\_CLEAR\_TASK

startActivity(intent)

finish()

}

}

}

**activity\_second.xml**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="No data" />

<Button

android:id="@+id/readButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Read storage" />

<Button

android:id="@+id/backButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Back" />

</LinearLayout>

DataStoreManager.kt

(package and code as above)

**AndroidManifest.xml**

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools">

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

[android:theme="@style/Theme.RepositoryG\_group3](mailto:android:theme="@style/Theme.RepositoryG_group3)">

<activity

android:name=".MainActivity"

android:exported="true"

android:label="@string/app\_name"

[android:theme="@style/Theme.RepositoryG\_group3](mailto:android:theme="@style/Theme.RepositoryG_group3)">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name=".SecondActivity"

android:exported="false" />

</application>

</manifest>

* **Pielikums**

Github liks uz “repository” <https://github.com/MrAlphaF/ViA_mobile3.git>