

**ВИСОКА ТЕХНИЧКА ШКОЛА СТРУКОВНИХ СТУДИЈА У НОВОМ САДУ**

**РАЗВОЈ АПЛИКАЦИЈА ЗА МОБИЛНЕ СИСТЕМЕ “EDUCATION TRACKER”**

|  |  |
| --- | --- |
| **Студент:**  **Ненад Тубић, МИТ5/23** | **Ментори:**  **професор: др Лазар Копања**  **асистент: мр Нинослава Тихи** |

**Нови Сад, 17. Децембар 2024.**

**СТАНДАРД ЗА СЕМИНАРСКИ РАД ИЗ РАЗВОЈА АПЛИКАЦИЈА ЗА МОБИЛНЕ СИСТЕМЕ**

Семинарски рад подразумева израду Андроид апликације. Рад носи максимално 45 бодова. Апликација садржи већи број ставки, при чему свака ставка носи број бодова који се може видети у табели.

Студент предаје следеће:

* Комплетан пројекат Андроид апликације из Android Studio-а, укључујући ту и извршни фајл. Апликација треба да има своје име, и да је направљена тако да садржи елементе наведене у табели, а обавезно да садржи елементе који су означени у колони табеле „Обавезна барем једна употреба“. Одређена ставка из табеле може носити максимално бодова онолико колико је наведено у колони табеле „Макс. бодова“.
* Документ Андроид апликације, који се налази у оквиру овог документа, и који треба да је попуњен подацима у складу са својом апликацијом. Све наведене компоненте треба да имају сврху и да функционишу исправно. Црвеним фонтом је означен пример бодовања апликације PopupMenu где је од могућих 50 остварено 32 бодова.

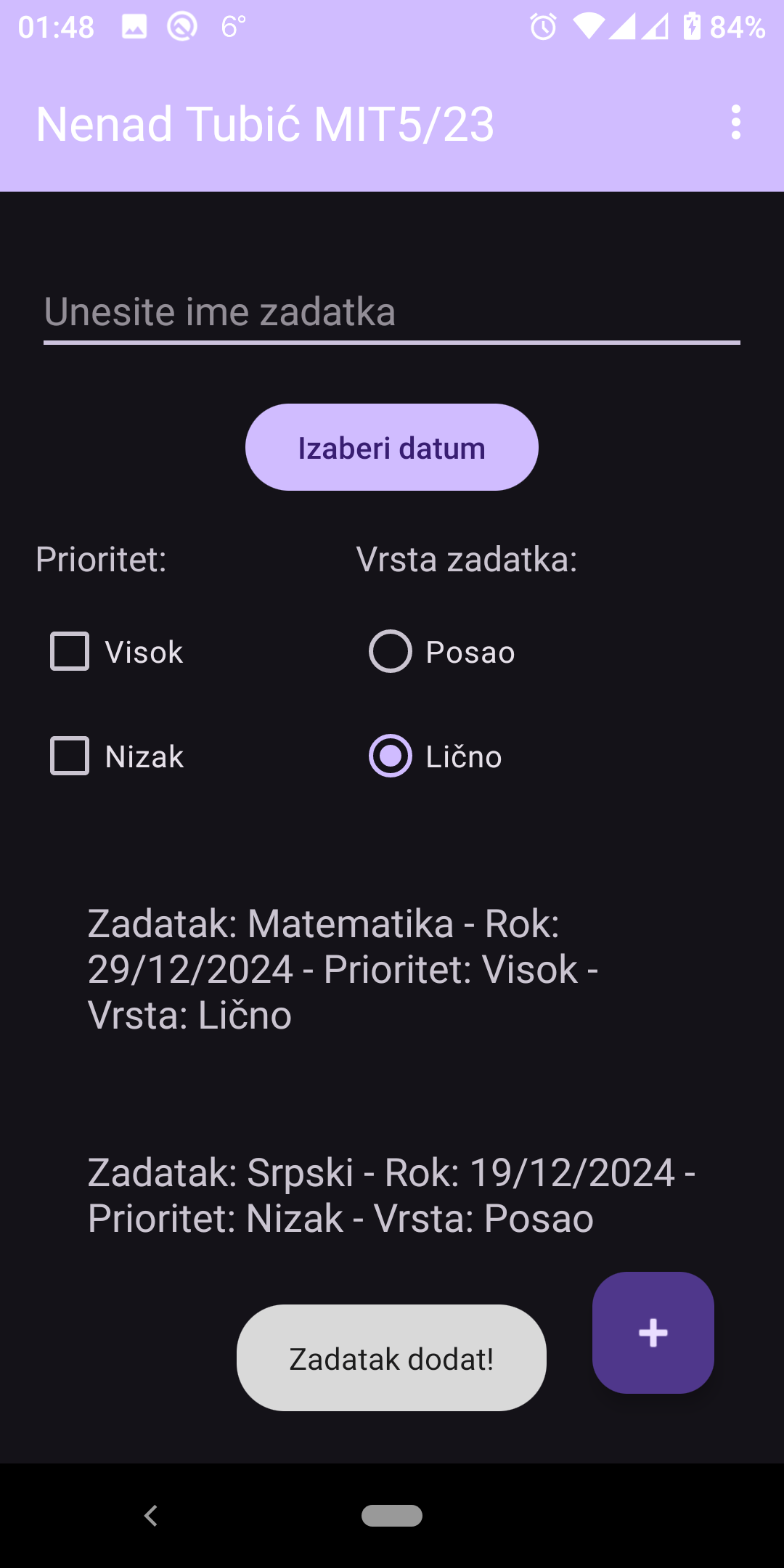
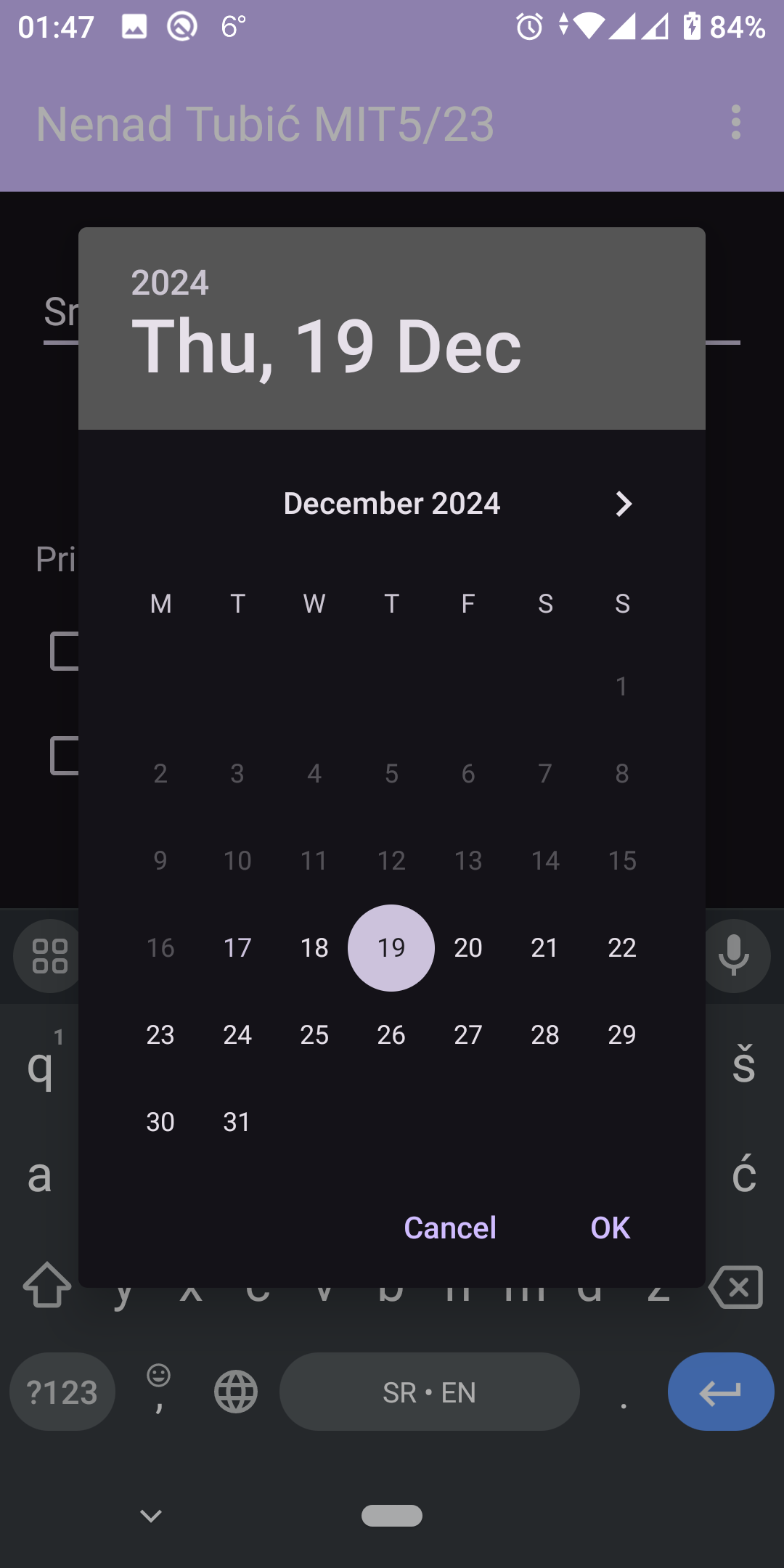
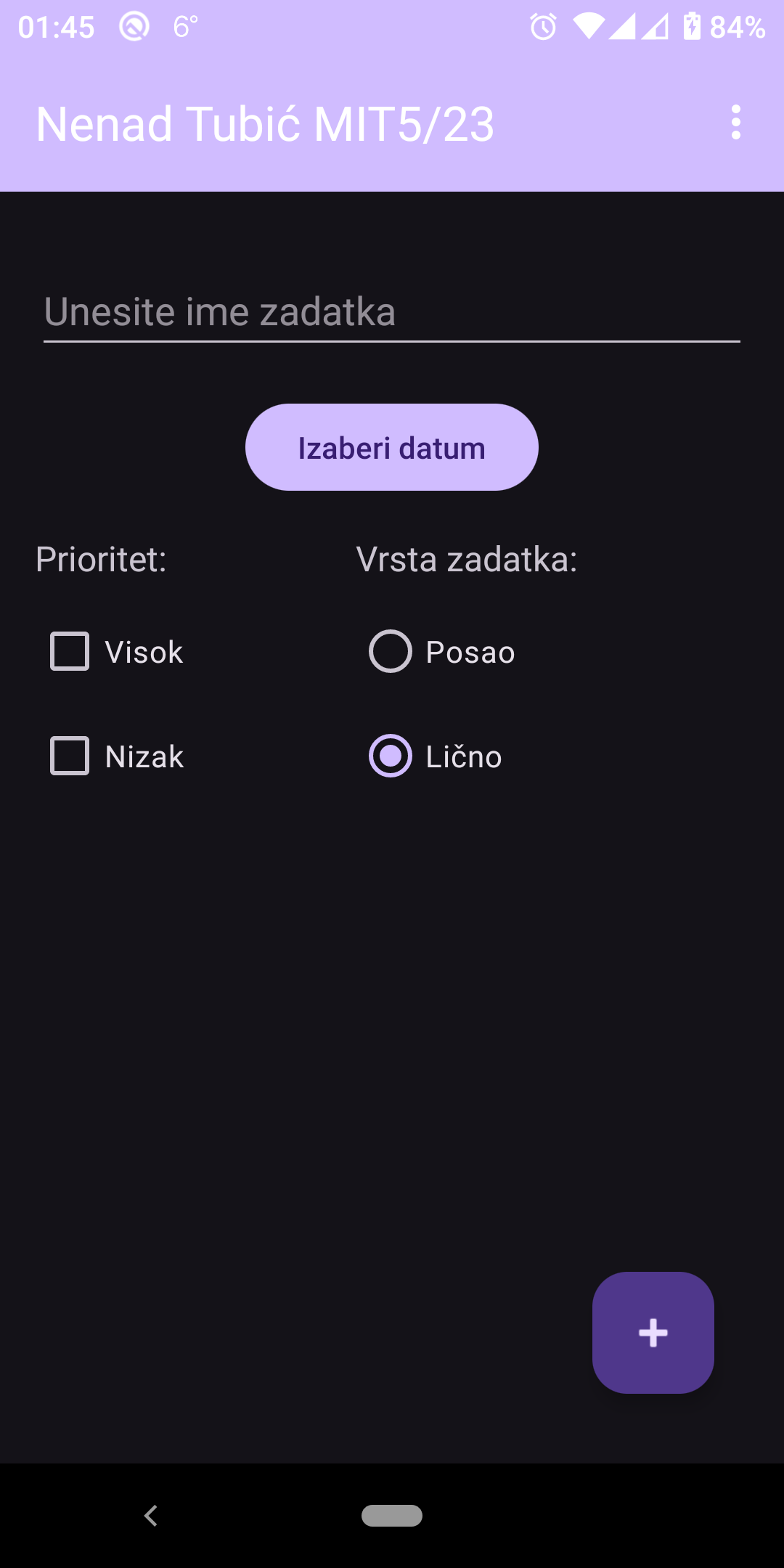
Слике екрана апликације и изворни код треба навести у наставку документа.

# Дикумент Андроид апликације

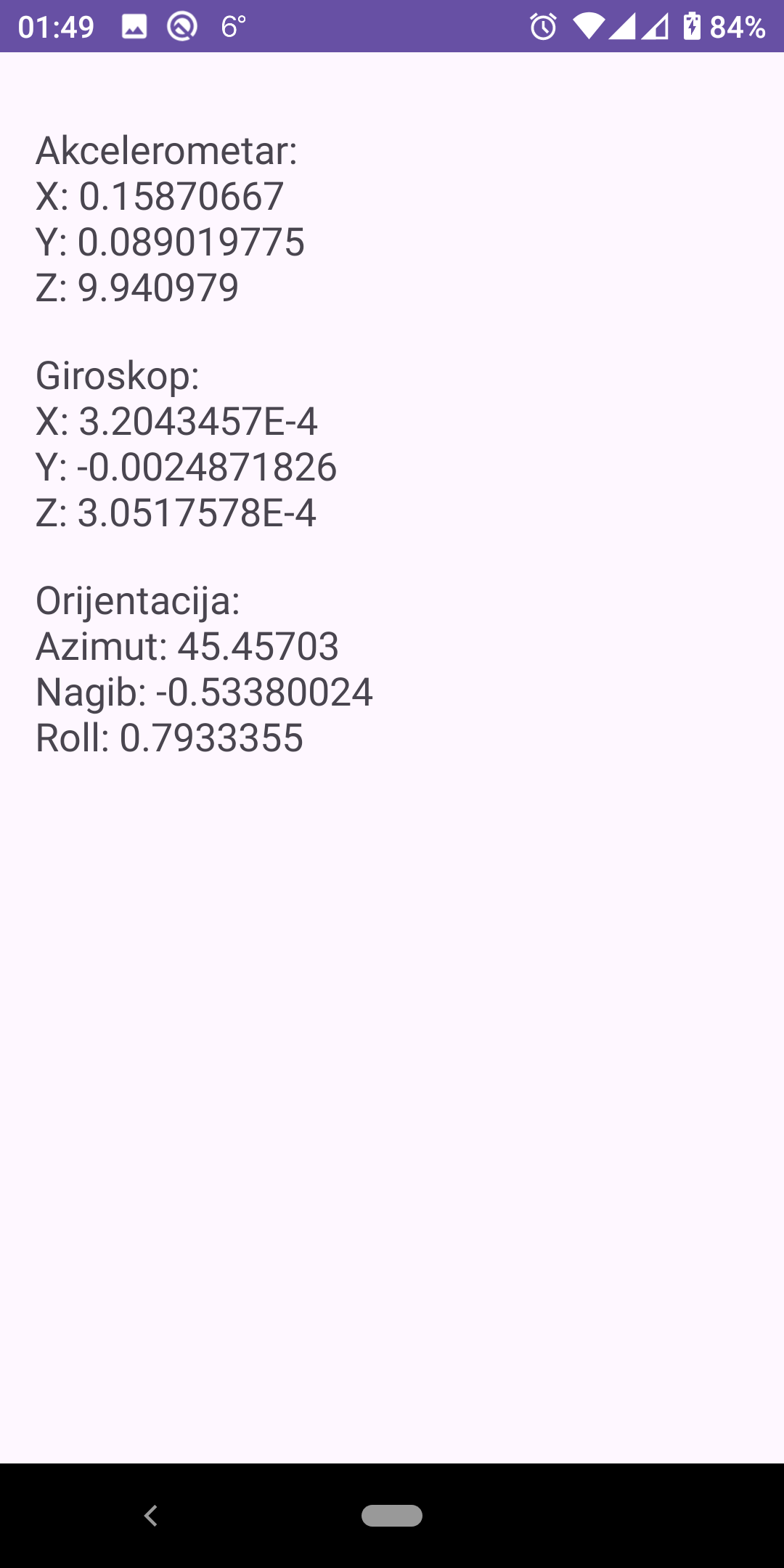
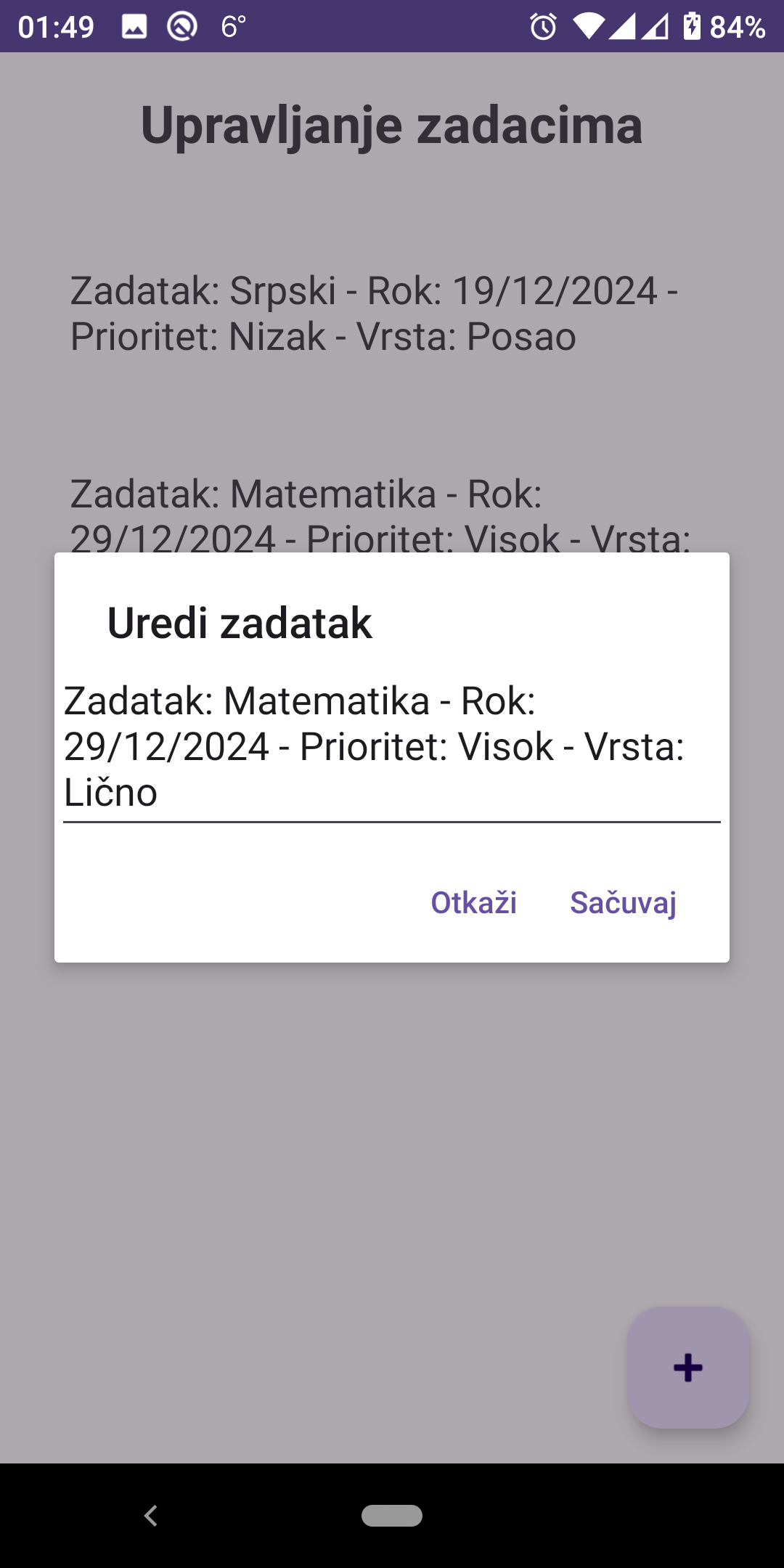
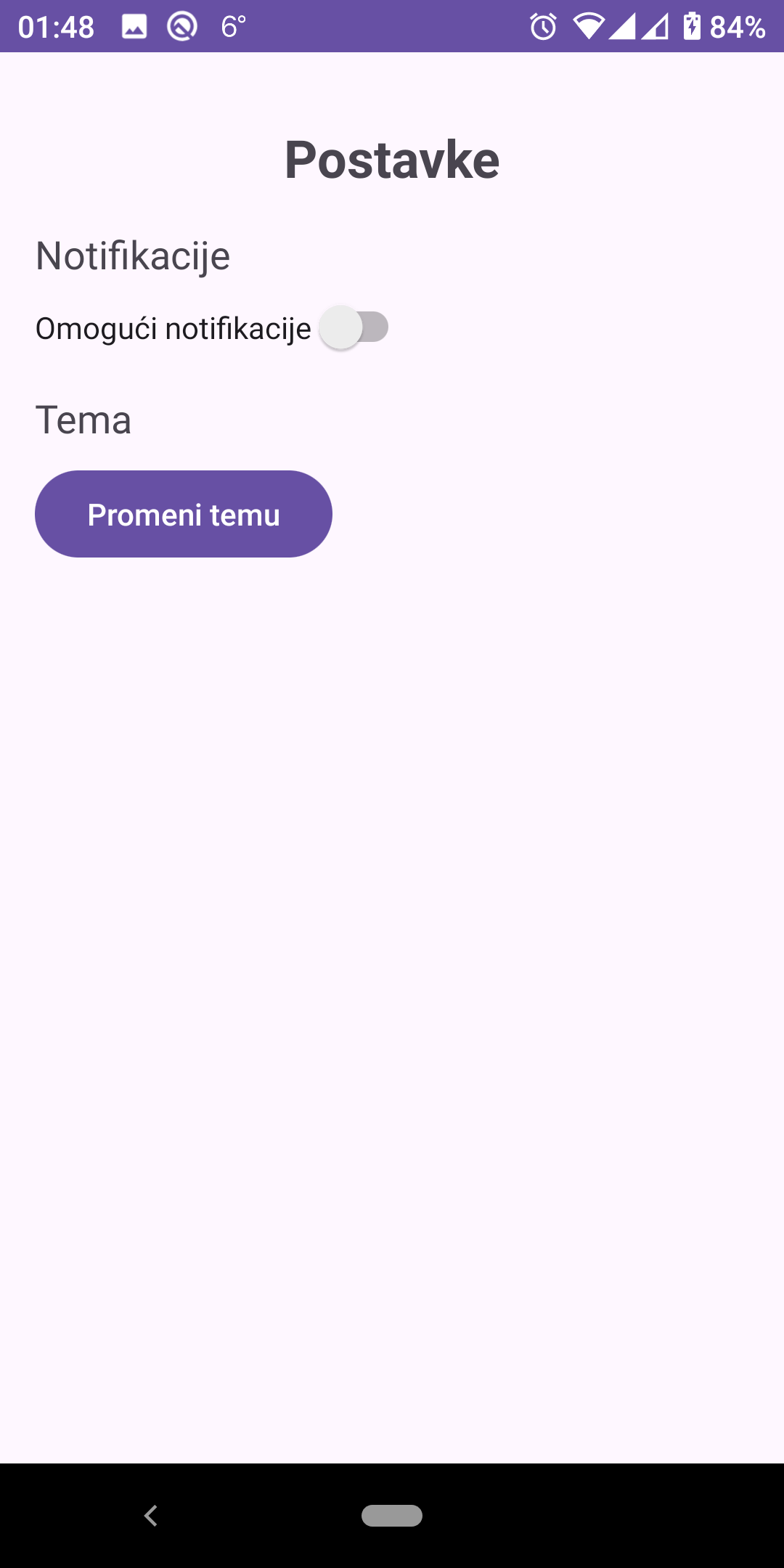
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Celina | Naziv stavke | Bodova | Max. bodova | Obavezna barem jedna upotreba | Napomena | Bodova ostvareno |
| KOMPONENTE | Title koja sarži ime, prezime i broj indeksa studenta | 2 | 2 | Da | Nenad Tubić MIT5/23 | 2 |
| EditText | 3 | 6 |  | ***et\_task\_name*** (activity\_main) | 3 |
| Button (ToggleButton, ImageButton...) | 3 | 6 |  | ***fab\_add\_task, btn\_date\_picker*** (activity\_main), ***btn\_change\_theme*** (activity\_settings)... | 6 |
| ImageView | 3 | 6 |  |  |  |
| TextView | 3 | 6 |  | activity\_main... | 6 |
| ListView | 3 | 6 |  | ***rv\_tasks*** (activity\_main), ***rv\_tasks*** (activity\_task\_management) | 6 |
| Checkbox skup od 2 ili više opcija | 3 | 6 |  | ***cb\_high\_priority, cb\_low\_priority*** activity\_main | 3 |
| Radio button skup od 2 ili više opcija | 3 | 6 |  | ***rb\_work, rb\_personal*** (activity\_main) | 3 |
| Ostale komponente | 2 | 4 |  | ***DatePicker*** (activity\_main) | 2 |
| LAYOUt | LinearLayout (Vertical) | 4 | 8 |  | activity\_main... | 8 |
| LinearLayout (Horizontal) | 4 | 8 |  | activity\_main | 4 |
| Relative Layout | 4 | 8 |  |  |  |
| Ostali layout-i | 4 | 8 |  | ***CoordinatorLayout, AppBarLayout.***(activity\_main) | 8 |
| MENIJI | Options meni | 4 | 8 |  | main\_menu.xml | 4 |
| Padajuci meni | 4 | 8 |  |  |  |
| Plutajuci meni | 4 | 8 |  |  |  |
| Podmeniji | 3 | 6 |  |  |  |
| Upozorenja | Toast | 2 | 4 |  | activity\_main... | 4 |
| Alert dijalog | 2 | 4 |  | TaskManagementActivity.java | 2 |
| Ostala obavestenja | 2 | 4 |  |  |  |
| UPOTREBAPODATAKA | Shared preferences | 5 | 10 |  | ***onCreate – night\_mode, notification\_enabled***  ***onSave*** – cuvanje zadatka (sve u MainActivity)... | 10 |
| SQLite baza podataka | 5 | 10 |  |  |  |
| Ostale vrste skladistenja podataka | 5 | 10 |  |  |  |
| Ostalo | Slusanje dogadjaja klika i dugog pritiska | 5 | 10 |  | TaskManagementActivity, MainActivity. | **10** |
| Citanje podataka senzora | 5 | 10 |  | ***accelerometer, gyroscope, orientationSensor*** (SensorDataActivity) | **10** |
| Citanje orijentacije uredjaja | 5 | 10 |  | ***orientationSensor*** (SensorDataActivity) | 5 |
| Reprodukcija zvucnih efekata pomocu SoundPool | 5 | 10 |  |  |  |
| Izvsavanje telefonskog poziva | 5 | 10 |  |  |  |
| Slanje SMS | 5 | 10 |  |  |  |
| Prikazivanje Web stranice u aplikaciji | 5 | 10 |  |  |  |
| Utvrdjivanje lokacije | 5 | 10 |  |  |  |
| **Bodova ukupno (max 50\*):** | | | | | | **50** |

# 

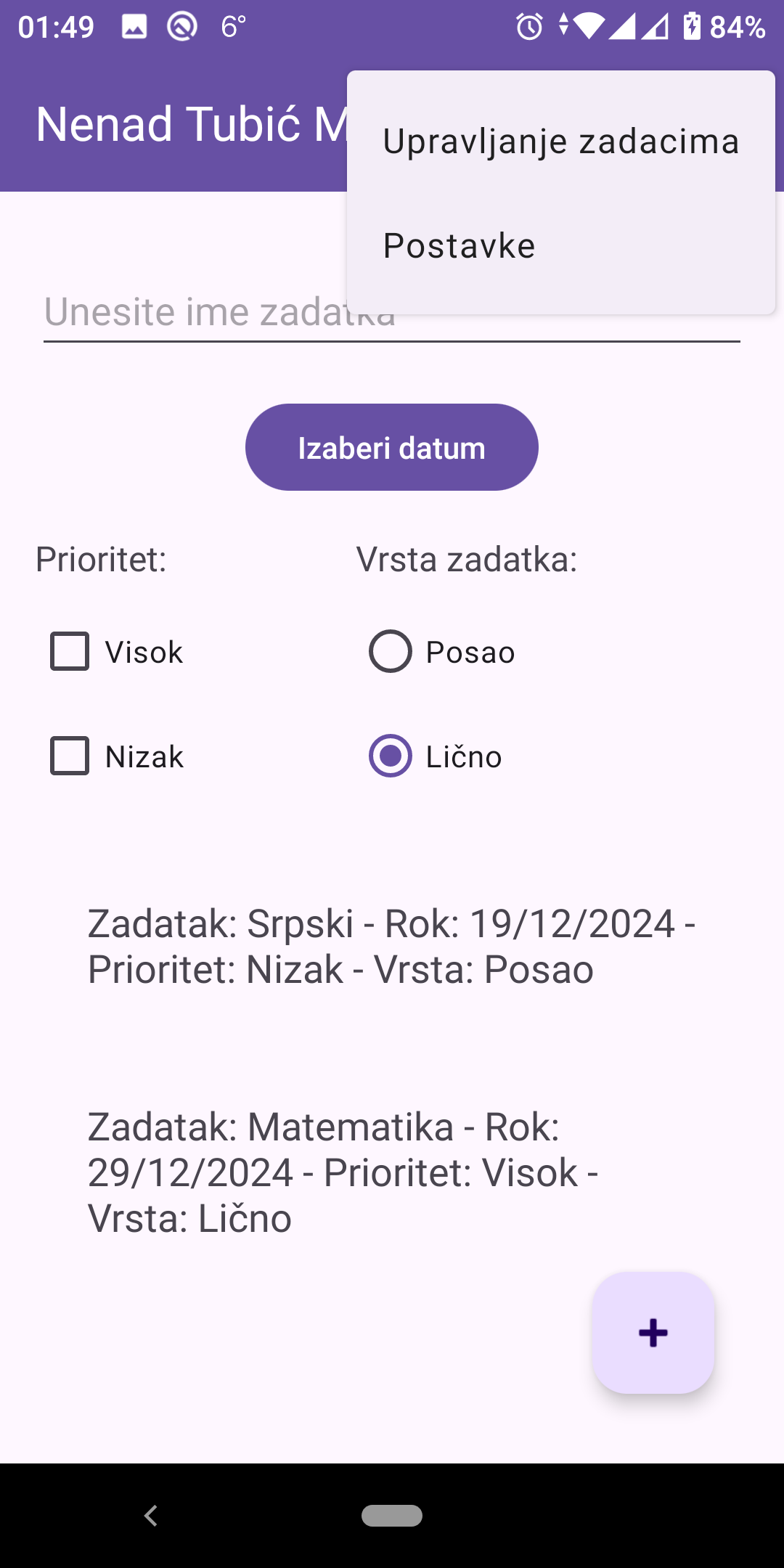
**Скриншотови и изворни код апликације**



*Од лева ка десно приказан је почетни екран апликације у тамној теми без унетих података, потом у средини је приказан процес уноса датума приликом уноса података, те скроз десно је приказана иста почетна страница са унета два задатка.*



*Од лева ка десно приказан је екран за поставке, потом управљање задацима где се види процес едитовања и на крају се види скривени екран очитаних сензора уређаја.*



*Од лева ка десно приказан је почетни екран у светлој теми са приказаним менијем за приступ поставкама и опцијом за управљање задацима, док је поред приказана опција управљање задацима где се може видети како изгледа са више уноса, наравно у светлој теми.*

**“strings.xml” датотека са дефинисаним статичким текстом**

<resources>

<string name="app\_name">Nenad Tubić MIT5/23</string>

<string name="task\_name\_hint">Unesite ime zadatka</string>

<string name="add\_task\_button">Dodaj zadatak</string>

<string name="task\_added">Zadatak dodat!</string>

<string name="task\_name\_empty">Unesite ime zadatka!</string>

</resources>

**“AndroidManifest.xml”датотека са кодом за поставку**

<?xml version="1.0" encoding="utf-8"?>

<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.EducationTracker"

tools:targetApi="31">

<activity android:name=".SensorDataActivity"></activity>

<activity android:name=".SettingsActivity"></activity>

<activity android:name=".TaskManagementActivity"></activity>

<activity android:name=".MainActivity" android:exported="true">

<intent-filter>

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

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

</intent-filter>

</activity>

<receiver

android:name=".TaskNotificationReceiver"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.BOOT\_COMPLETED" />

</intent-filter>

</receiver>

</application>

<uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />

<uses-permission android:name="android.permission.BODY\_SENSORS" />

<uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />

</manifest>

**“build.gradle”датотека са кодом за конфигурацију билда апликације**

plugins {

alias(libs.plugins.android.application)

}

android {

namespace 'com.example.educationtracker'

compileSdk 34

defaultConfig {

applicationId "com.example.educationtracker"

minSdk 24

targetSdk 34

versionCode 1

versionName "1.0"

testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'

}

}

compileOptions {

sourceCompatibility JavaVersion.VERSION\_11

targetCompatibility JavaVersion.VERSION\_11

}

}

dependencies {

implementation libs.appcompat

implementation libs.material

implementation libs.activity

implementation libs.constraintlayout

testImplementation libs.junit

androidTestImplementation libs.ext.junit

androidTestImplementation libs.espresso.core

}

**“main\_menu.xml”датотека са кодом за мену за опције**

<?xml version="1.0" encoding="utf-8"?>

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

xmlns:app="http://schemas.android.com/apk/res-auto">

<item

android:id="@+id/action\_manage\_tasks"

android:title="Upravljanje zadacima"

app:showAsAction="never"/>

<item

android:id="@+id/action\_settings"

android:title="Postavke"

app:showAsAction="never"/>

</menu>

**“activity\_main.xml”датотека са кодом за почетну страницу апликације**

<androidx.coordinatorlayout.widget.CoordinatorLayout

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

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

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<com.google.android.material.appbar.AppBarLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

<androidx.appcompat.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="?attr/colorPrimary"

app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>

</com.google.android.material.appbar.AppBarLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp"

android:layout\_marginTop="?attr/actionBarSize">

<EditText

android:id="@+id/et\_task\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/task\_name\_hint"

android:layout\_marginTop="16dp" />

<Button

android:id="@+id/btn\_date\_picker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Izaberi datum"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="16dp" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="16dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_weight="1">

<TextView

android:id="@+id/tv\_priority"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Prioritet:"

android:textSize="16sp"

android:layout\_marginBottom="8dp" />

<CheckBox

android:id="@+id/cb\_high\_priority"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Visok" />

<CheckBox

android:id="@+id/cb\_low\_priority"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Nizak" />

</LinearLayout>

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_weight="1">

<TextView

android:id="@+id/tv\_task\_type"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Vrsta zadatka:"

android:textSize="16sp"

android:layout\_marginBottom="8dp" />

<RadioGroup

android:id="@+id/rg\_task\_type"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_weight="1">

<RadioButton

android:id="@+id/rb\_work"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Posao" />

<RadioButton

android:id="@+id/rb\_personal"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Lično"

android:checked="true" />

</RadioGroup>

</LinearLayout>

</LinearLayout>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rv\_tasks"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1"

android:layout\_marginTop="16dp" />

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/fab\_add\_task"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="end|bottom"

android:layout\_margin="16dp"

android:src="@android:drawable/ic\_input\_add" />

</LinearLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

**“activity\_sensor\_data.xml”датотека са кодом за страницу за приказ података сензора**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp"

tools:context=".SensorDataActivity">

<TextView

android:id="@+id/tv\_accelerometer"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Akcelerometar: -"

android:textSize="18sp"

android:layout\_marginTop="16dp" />

<TextView

android:id="@+id/tv\_gyroscope"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Giroskop: -"

android:textSize="18sp"

android:layout\_marginTop="16dp" />

<TextView

android:id="@+id/tv\_orientation"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Orijentacija: -"

android:textSize="18sp"

android:layout\_marginTop="16dp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

**“activity\_settings.xml”датотека са кодом за страницу са поставкама**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp"

tools:context=".SettingsActivity">

<TextView

android:id="@+id/tv\_settings\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Postavke"

android:textSize="24sp"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="16dp"

android:textStyle="bold" />

<TextView

android:id="@+id/tv\_notification\_settings"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Notifikacije"

android:textSize="18sp"

android:layout\_marginTop="16dp" />

<Switch

android:id="@+id/sw\_notifications"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Omogući notifikacije"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/tv\_theme\_settings"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Tema"

android:textSize="18sp"

android:layout\_marginTop="16dp" />

<Button

android:id="@+id/btn\_change\_theme"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Promeni temu"

android:layout\_marginTop="8dp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

**“activity\_task\_management.xml”датотека са кодом за страницу за уређивање задатака**

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".TaskManagementActivity">

<TextView

android:id="@+id/tv\_task\_management\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Upravljanje zadacima"

android:textSize="24sp"

android:layout\_marginTop="16dp"

android:layout\_centerHorizontal="true"

android:textStyle="bold" />

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rv\_tasks"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@id/tv\_task\_management\_title"

android:layout\_marginTop="16dp"

android:padding="8dp" />

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/fab\_add\_task"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_margin="16dp"

android:src="@android:drawable/ic\_input\_add"

android:contentDescription="Dodaj novi zadatak" />

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

**“task\_list\_item.xml”датотека са кодом за приказ једног елемента са листе (задатак)**

<?xml version="1.0" encoding="utf-8"?>

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

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/task\_item"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="18sp"

android:text="Zadatak"

android:padding="8dp" />

</LinearLayout>

**“MainActivity.java”датотека са кодом за логику почетне странице**

public class MainActivity extends AppCompatActivity {

private EditText etTaskName;

private FloatingActionButton fabAddTask;

private RecyclerView rvTasks;

private TaskAdapter taskAdapter;

private ArrayList<String> taskList;

private int selectedDay = -1, selectedMonth = -1, selectedYear = -1;

private CheckBox cbHighPriority, cbLowPriority;

private RadioGroup rgTaskType;

private RadioButton rbWork, rbPersonal;

private SharedPreferences sharedPreferences;

private static final String PREFERENCES\_NAME = "TaskPreferences";

private static final String TASKS\_KEY = "Tasks";

@Override

protected void onCreate(Bundle savedInstanceState) {

sharedPreferences = getSharedPreferences("AppSettings", MODE\_PRIVATE);

int nightMode = sharedPreferences.getInt("night\_mode", AppCompatDelegate.MODE\_NIGHT\_NO);

AppCompatDelegate.setDefaultNightMode(nightMode);

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toolbar toolbar = findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

toolbar.setOnLongClickListener(view -> {

Intent intent = new Intent(MainActivity.this, SensorDataActivity.class);

startActivity(intent);

return true;

});

etTaskName = findViewById(R.id.et\_task\_name);

fabAddTask = findViewById(R.id.fab\_add\_task);

rvTasks = findViewById(R.id.rv\_tasks);

cbHighPriority = findViewById(R.id.cb\_high\_priority);

cbLowPriority = findViewById(R.id.cb\_low\_priority);

rgTaskType = findViewById(R.id.rg\_task\_type);

rbWork = findViewById(R.id.rb\_work);

rbPersonal = findViewById(R.id.rb\_personal);

SharedPreferences sharedPreferences = getSharedPreferences("AppSettings", MODE\_PRIVATE);

boolean notificationsEnabled = sharedPreferences.getBoolean("notifications\_enabled", true);

taskList = new ArrayList<>();

taskAdapter = new TaskAdapter(

this,

taskList,

position -> {

String task = taskList.get(position);

Toast.makeText(MainActivity.this, "Kliknuli ste na: " + task, Toast.LENGTH\_SHORT).show();

},

position -> {

}

);

rvTasks.setAdapter(taskAdapter);

rvTasks.setLayoutManager(new LinearLayoutManager(this));

rvTasks.setAdapter(taskAdapter);

rbPersonal.setChecked(true);

loadTasks();

findViewById(R.id.btn\_date\_picker).setOnClickListener(view -> {

Calendar calendar = Calendar.getInstance();

int year = calendar.get(Calendar.YEAR);

int month = calendar.get(Calendar.MONTH);

int day = calendar.get(Calendar.DAY\_OF\_MONTH);

DatePickerDialog datePickerDialog = new DatePickerDialog(

MainActivity.this,

(datePicker, selectedYear, selectedMonth, selectedDay) -> {

this.selectedDay = selectedDay;

this.selectedMonth = selectedMonth;

this.selectedYear = selectedYear;

},

year, month, day);

datePickerDialog.getDatePicker().setMinDate(calendar.getTimeInMillis());

datePickerDialog.show();

});

cbHighPriority.setOnCheckedChangeListener((buttonView, isChecked) -> {

if (isChecked) {

cbLowPriority.setChecked(false);

}

});

cbLowPriority.setOnCheckedChangeListener((buttonView, isChecked) -> {

if (isChecked) {

cbHighPriority.setChecked(false);

}

});

fabAddTask.setOnClickListener(view -> addTask());

rvTasks.addOnItemTouchListener(

new RecyclerItemClickListener(this, rvTasks, new RecyclerItemClickListener.OnItemClickListener() {

@Override

public void onItemClick(View view, int position) {

String task = taskList.get(position);

Toast.makeText(MainActivity.this, "Kliknuli ste na: " + task, Toast.LENGTH\_SHORT).show();

}

})

);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.main\_menu, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

if (id == R.id.action\_manage\_tasks) {

Intent manageTasksIntent = new Intent(MainActivity.this, TaskManagementActivity.class);

startActivity(manageTasksIntent);

return true;

} else if (id == R.id.action\_settings) {

Intent settingsIntent = new Intent(MainActivity.this, SettingsActivity.class);

startActivity(settingsIntent);

return true;

} else {

return super.onOptionsItemSelected(item);

}

}

private void addTask() {

String taskName = etTaskName.getText().toString().trim();

if (taskName.isEmpty()) {

Toast.makeText(MainActivity.this, "Unesite ime zadatka!", Toast.LENGTH\_SHORT).show();

return;

}

if (selectedDay == -1 || selectedMonth == -1 || selectedYear == -1) {

Toast.makeText(MainActivity.this, "Izaberite datum!", Toast.LENGTH\_SHORT).show();

return;

}

if (!cbHighPriority.isChecked() &amp;&amp; !cbLowPriority.isChecked()) {

Toast.makeText(MainActivity.this, "Izaberite prioritet!", Toast.LENGTH\_SHORT).show();

return;

}

String taskDate = String.format("%02d/%02d/%d", selectedDay, selectedMonth + 1, selectedYear);

String priority = cbHighPriority.isChecked() ? "Visok" : "Nizak";

int selectedTaskTypeId = rgTaskType.getCheckedRadioButtonId();

String taskType = selectedTaskTypeId == rbWork.getId() ? "Posao" : "Lično";

String task = "Zadatak: " + taskName + " - Rok: " + taskDate + " - Prioritet: " + priority + " - Vrsta: " + taskType;

taskList.add(task);

taskAdapter.notifyDataSetChanged();

saveTasks();

etTaskName.setText("");

selectedDay = -1;

selectedMonth = -1;

selectedYear = -1;

cbHighPriority.setChecked(false);

cbLowPriority.setChecked(false);

rgTaskType.clearCheck();

rbPersonal.setChecked(true);

Toast.makeText(MainActivity.this, "Zadatak dodat!", Toast.LENGTH\_SHORT).show();

SharedPreferences sharedPreferences = getSharedPreferences("AppSettings", MODE\_PRIVATE);

boolean notificationsEnabled = sharedPreferences.getBoolean("notifications\_enabled", true);

if (notificationsEnabled) {

Intent intent = new Intent(MainActivity.this, TaskNotificationReceiver.class);

intent.putExtra("taskName", task);

intent.putExtra("notificationId", taskList.size());

sendBroadcast(intent);

}

}

private void saveTasks() {

SharedPreferences sharedPreferences = getSharedPreferences(PREFERENCES\_NAME, Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit();

Set<String> taskSet = new HashSet<>(taskList);

editor.putStringSet(TASKS\_KEY, taskSet);

editor.apply();

}

private void loadTasks() {

SharedPreferences sharedPreferences = getSharedPreferences(PREFERENCES\_NAME, Context.MODE\_PRIVATE);

Set<String> taskSet = sharedPreferences.getStringSet(TASKS\_KEY, new HashSet<>());

if (taskSet != null) {

taskList.clear();

taskList.addAll(taskSet);

taskAdapter.notifyDataSetChanged();

}

}

@Override

protected void onResume() {

super.onResume();

loadTasks();

}

}

**“RecyclerItemClickListener.java”датотека са кодом за слушање догађаја за клик**

public class RecyclerItemClickListener implements RecyclerView.OnItemTouchListener {

private OnItemClickListener mListener;

private GestureDetector mGestureDetector;

public interface OnItemClickListener {

void onItemClick(View view, int position);

default void onLongItemClick(View view, int position){

}

}

public RecyclerItemClickListener(Context context, final RecyclerView recyclerView, OnItemClickListener listener) {

mListener = listener;

mGestureDetector = new GestureDetector(context, new GestureDetector.SimpleOnGestureListener() {

@Override

public boolean onSingleTapUp(MotionEvent e) {

return true;

}

@Override

public void onLongPress(MotionEvent e) {

View child = recyclerView.findChildViewUnder(e.getX(), e.getY());

if (child != null &amp;&amp; mListener != null) {

mListener.onLongItemClick(child, recyclerView.getChildAdapterPosition(child));

}

}

});

}

@Override

public boolean onInterceptTouchEvent(@NonNull RecyclerView view, @NonNull MotionEvent e) {

View child = view.findChildViewUnder(e.getX(), e.getY());

if (child != null &amp;&amp; mListener != null &amp;&amp; mGestureDetector.onTouchEvent(e)) {

mListener.onItemClick(child, view.getChildAdapterPosition(child));

}

return false;

}

@Override

public void onTouchEvent(@NonNull RecyclerView view, @NonNull MotionEvent motionEvent) {

}

@Override

public void onRequestDisallowInterceptTouchEvent(boolean disallowIntercept) {

}

}

**“SensorDataActivity.java”датотека са кодом за логику приказа података сензора**

public class SensorDataActivity extends AppCompatActivity implements SensorEventListener {

private SensorManager sensorManager;

private Sensor accelerometer, gyroscope, orientationSensor;

private TextView tvAccelerometer, tvGyroscope, tvOrientation;

private static final String TAG = "SensorDataActivity";

@Override

protected void onCreate(Bundle savedInstanceState) {

SharedPreferences sharedPreferences = getSharedPreferences("AppSettings", MODE\_PRIVATE);

int nightMode = sharedPreferences.getInt("night\_mode", AppCompatDelegate.MODE\_NIGHT\_NO);

AppCompatDelegate.setDefaultNightMode(nightMode);

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_sensor\_data);

Log.d(TAG, "onCreate: Activity Created");

tvAccelerometer = findViewById(R.id.tv\_accelerometer);

tvGyroscope = findViewById(R.id.tv\_gyroscope);

tvOrientation = findViewById(R.id.tv\_orientation);

sensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

if (sensorManager != null) {

accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);

gyroscope = sensorManager.getDefaultSensor(Sensor.TYPE\_GYROSCOPE);

orientationSensor = sensorManager.getDefaultSensor(Sensor.TYPE\_ORIENTATION);

}

if (accelerometer != null &amp;&amp; gyroscope != null &amp;&amp; orientationSensor != null) {

sensorManager.registerListener(this, accelerometer, SensorManager.SENSOR\_DELAY\_NORMAL);

sensorManager.registerListener(this, gyroscope, SensorManager.SENSOR\_DELAY\_NORMAL);

sensorManager.registerListener(this, orientationSensor, SensorManager.SENSOR\_DELAY\_NORMAL);

} else {

Log.e(TAG, "Senzori nisu dostupni");

tvAccelerometer.setText("Akcelerometar nije dostupan");

tvGyroscope.setText("Giroskop nije dostupan");

tvOrientation.setText("Orijentacija nije dostupna");

}

}

@Override

public void onSensorChanged(SensorEvent event) {

if (event.sensor.getType() == Sensor.TYPE\_ACCELEROMETER) {

float x = event.values[0];

float y = event.values[1];

float z = event.values[2];

tvAccelerometer.setText("Akcelerometar: \nX: " + x + "\nY: " + y + "\nZ: " + z);

} else if (event.sensor.getType() == Sensor.TYPE\_GYROSCOPE) {

float x = event.values[0];

float y = event.values[1];

float z = event.values[2];

tvGyroscope.setText("Giroskop: \nX: " + x + "\nY: " + y + "\nZ: " + z);

} else if (event.sensor.getType() == Sensor.TYPE\_ORIENTATION) {

float azimuth = event.values[0];

float pitch = event.values[1];

float roll = event.values[2];

tvOrientation.setText("Orijentacija: \nAzimut: " + azimuth + "\nNagib: " + pitch + "\nRoll: " + roll);

}

}

@Override

public void onAccuracyChanged(Sensor sensor, int accuracy) {

}

@Override

protected void onPause() {

super.onPause();

sensorManager.unregisterListener(this);

Log.d(TAG, "onPause: Listener Unregistered");

}

@Override

protected void onResume() {

super.onResume();

if (accelerometer != null &amp;&amp; gyroscope != null &amp;&amp; orientationSensor != null) {

sensorManager.registerListener(this, accelerometer, SensorManager.SENSOR\_DELAY\_NORMAL);

sensorManager.registerListener(this, gyroscope, SensorManager.SENSOR\_DELAY\_NORMAL);

sensorManager.registerListener(this, orientationSensor, SensorManager.SENSOR\_DELAY\_NORMAL);

}

Log.d(TAG, "onResume: Listener Registered");

}

}

**“SettingsActivity.java”датотека са кодом за логику дела за поставке апликације**

public class SettingsActivity extends AppCompatActivity {

private static final int REQUEST\_NOTIFICATION\_PERMISSION = 1;

private Switch swNotifications;

private Button btnChangeTheme;

private SharedPreferences sharedPreferences; // Dodata promenljiva klase

@Override

protected void onCreate(Bundle savedInstanceState) {

sharedPreferences = getSharedPreferences("AppSettings", MODE\_PRIVATE);

int nightMode = sharedPreferences.getInt("night\_mode", AppCompatDelegate.MODE\_NIGHT\_NO);

AppCompatDelegate.setDefaultNightMode(nightMode);

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_settings);

swNotifications = findViewById(R.id.sw\_notifications);

btnChangeTheme = findViewById(R.id.btn\_change\_theme);

swNotifications.setChecked(sharedPreferences.getBoolean("notifications\_enabled", true));

nightMode = sharedPreferences.getInt("night\_mode", AppCompatDelegate.MODE\_NIGHT\_NO); // Izmena vrednosti

AppCompatDelegate.setDefaultNightMode(nightMode);

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.TIRAMISU) {

if (ContextCompat.checkSelfPermission(this, Manifest.permission.POST\_NOTIFICATIONS)

!= PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(this,

new String[]{Manifest.permission.POST\_NOTIFICATIONS},

REQUEST\_NOTIFICATION\_PERMISSION);

}

}

btnChangeTheme.setOnClickListener(view -> {

int currentNightMode = AppCompatDelegate.getDefaultNightMode();

int newNightMode = currentNightMode == AppCompatDelegate.MODE\_NIGHT\_YES

? AppCompatDelegate.MODE\_NIGHT\_NO

: AppCompatDelegate.MODE\_NIGHT\_YES;

AppCompatDelegate.setDefaultNightMode(newNightMode);

sharedPreferences.edit().putInt("night\_mode", newNightMode).apply();

});

swNotifications.setOnCheckedChangeListener((buttonView, isChecked) -> {

sharedPreferences.edit().putBoolean("notifications\_enabled", isChecked).apply();

if (isChecked) {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

CharSequence name = "TaskNotificationChannel";

String description = "Channel for Task Notifications";

int importance = NotificationManager.IMPORTANCE\_DEFAULT;

NotificationChannel channel = new NotificationChannel("TASK\_NOTIFICATION\_CHANNEL", name, importance);

channel.setDescription(description);

NotificationManager notificationManager = getSystemService(NotificationManager.class);

notificationManager.createNotificationChannel(channel);

}

NotificationCompat.Builder builder = new NotificationCompat.Builder(SettingsActivity.this, "TASK\_NOTIFICATION\_CHANNEL")

.setSmallIcon(android.R.drawable.ic\_dialog\_info)

.setContentTitle("Notifikacije omogućene")

.setContentText("Sada ćeš primati notifikacije za zadatke.")

.setPriority(NotificationCompat.PRIORITY\_DEFAULT);

NotificationManagerCompat notificationManager = NotificationManagerCompat.from(SettingsActivity.this);

notificationManager.notify(1, builder.build());

} else {

NotificationManagerCompat notificationManager = NotificationManagerCompat.from(SettingsActivity.this);

notificationManager.cancelAll();

}

});

}

}

**“TaskAdapter.java”датотека са кодом за логику која управља приказом листе задатака у RecycleView**

public class TaskAdapter extends RecyclerView.Adapter<TaskAdapter.TaskViewHolder> {

private List<String> taskList;

private LayoutInflater mInflater;

private OnItemClickListener onItemClickListener;

private OnItemLongClickListener onItemLongClickListener;

public TaskAdapter(Context context, List<String> taskList,

OnItemClickListener clickListener,

OnItemLongClickListener longClickListener) {

this.taskList = taskList;

this.mInflater = LayoutInflater.from(context);

this.onItemClickListener = clickListener;

this.onItemLongClickListener = longClickListener;

}

@Override

public TaskViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

View itemView = mInflater.inflate(R.layout.task\_list\_item, parent, false);

return new TaskViewHolder(itemView);

}

@Override

public void onBindViewHolder(TaskViewHolder holder, int position) {

String task = taskList.get(position);

holder.taskItemView.setText(task);

holder.itemView.setOnClickListener(v -> onItemClickListener.onItemClick(position));

holder.itemView.setOnLongClickListener(v -> {

onItemLongClickListener.onItemLongClick(position);

return true;

});

}

@Override

public int getItemCount() {

return taskList.size();

}

public interface OnItemClickListener {

void onItemClick(int position);

}

public interface OnItemLongClickListener {

void onItemLongClick(int position);

}

class TaskViewHolder extends RecyclerView.ViewHolder {

public final TextView taskItemView;

public TaskViewHolder(View itemView) {

super(itemView);

taskItemView = itemView.findViewById(R.id.task\_item);

}

}

}

**“TaskManagementActivity.java”датотека са кодом за логику која је задужена за део апликације за уређивање задатака**

public class TaskManagementActivity extends AppCompatActivity {

private RecyclerView rvTasks;

private TaskAdapter taskAdapter;

private ArrayList<String> taskList;

private FloatingActionButton fabAddTask;

@Override

protected void onCreate(Bundle savedInstanceState) {

SharedPreferences sharedPreferences = getSharedPreferences("AppSettings", MODE\_PRIVATE);

int nightMode = sharedPreferences.getInt("night\_mode", AppCompatDelegate.MODE\_NIGHT\_NO);

AppCompatDelegate.setDefaultNightMode(nightMode);

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_task\_management);

rvTasks = findViewById(R.id.rv\_tasks);

taskList = new ArrayList<>();

taskAdapter = new TaskAdapter(

this,

taskList,

position -> editTask(position),

position -> deleteTask(position)

);

rvTasks.setLayoutManager(new LinearLayoutManager(this));

rvTasks.setAdapter(taskAdapter);

fabAddTask = findViewById(R.id.fab\_add\_task);

fabAddTask.setOnClickListener(view -> {

Intent intent = new Intent(TaskManagementActivity.this, MainActivity.class);

startActivity(intent);

});

loadTasks();

}

private void loadTasks() {

SharedPreferences sharedPreferences = getSharedPreferences("TaskPreferences", Context.MODE\_PRIVATE);

Set<String> taskSet = sharedPreferences.getStringSet("Tasks", new HashSet<>());

taskList.clear();

if (taskSet != null) {

taskList.addAll(taskSet);

}

taskAdapter.notifyDataSetChanged();

}

private void deleteTask(int position) {

if (position >= 0 &amp;&amp; position < taskList.size()) {

taskList.remove(position);

taskAdapter.notifyItemRemoved(position);

taskAdapter.notifyItemRangeChanged(position, taskList.size());

saveTasks();

}

}

private void editTask(int position) {

String task = taskList.get(position);

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Uredi zadatak");

final EditText input = new EditText(this);

input.setText(task);

builder.setView(input);

builder.setPositiveButton("Sačuvaj", (dialog, which) -> {

String editedTask = input.getText().toString();

if (!editedTask.trim().isEmpty()) {

taskList.set(position, editedTask);

taskAdapter.notifyItemChanged(position);

saveTasks();

} else {

Toast.makeText(this, "Zadatak ne može biti prazan!", Toast.LENGTH\_SHORT).show();

}

});

builder.setNegativeButton("Otkaži", (dialog, which) -> dialog.cancel());

builder.show();

}

private void saveTasks() {

SharedPreferences sharedPreferences = getSharedPreferences("TaskPreferences", Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit();

Set<String> taskSet = new HashSet<>(taskList);

editor.putStringSet("Tasks", taskSet);

editor.apply();

}

private void addNewTask() {

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Dodaj novi zadatak");

final EditText input = new EditText(this);

builder.setView(input);

builder.setPositiveButton("Dodaj", (dialog, which) -> {

String newTask = input.getText().toString();

taskList.add(newTask);

taskAdapter.notifyItemInserted(taskList.size() - 1);

saveTasks();

Intent refreshIntent = new Intent(TaskManagementActivity.this, MainActivity.class);

startActivity(refreshIntent);

});

builder.setNegativeButton("Otkaži", (dialog, which) -> dialog.cancel());

builder.show();

}

}

**“TaskNotificationReceiver.java”датотека са кодом за логику која је задужена за приказивање нотификација задатака**

public class TaskNotificationReceiver extends BroadcastReceiver {

private static final String CHANNEL\_ID = "task\_channel";

@Override

public void onReceive(Context context, Intent intent) {

String taskName = intent.getStringExtra("taskName");

int notificationId = intent.getIntExtra("notificationId", 0);

NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

NotificationChannel channel = new NotificationChannel(CHANNEL\_ID, "Task Notifications", NotificationManager.IMPORTANCE\_HIGH);

notificationManager.createNotificationChannel(channel);

}

Intent notificationIntent = new Intent(context, MainActivity.class);

PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, notificationIntent, PendingIntent.FLAG\_UPDATE\_CURRENT | PendingIntent.FLAG\_IMMUTABLE);

NotificationCompat.Builder builder = new NotificationCompat.Builder(context, CHANNEL\_ID)

.setSmallIcon(R.drawable.ic\_launcher\_foreground)

.setContentTitle("Podsetnik za zadatak")

.setContentText(taskName)

.setPriority(NotificationCompat.PRIORITY\_HIGH)

.setContentIntent(pendingIntent)

.setAutoCancel(true);

notificationManager.notify(notificationId, builder.build());

}

}