ВИСОКА ТЕХНИЧКА ШКОЛА
СТРУКОВНИХ СТУДИЈА
у Новом Саду

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.

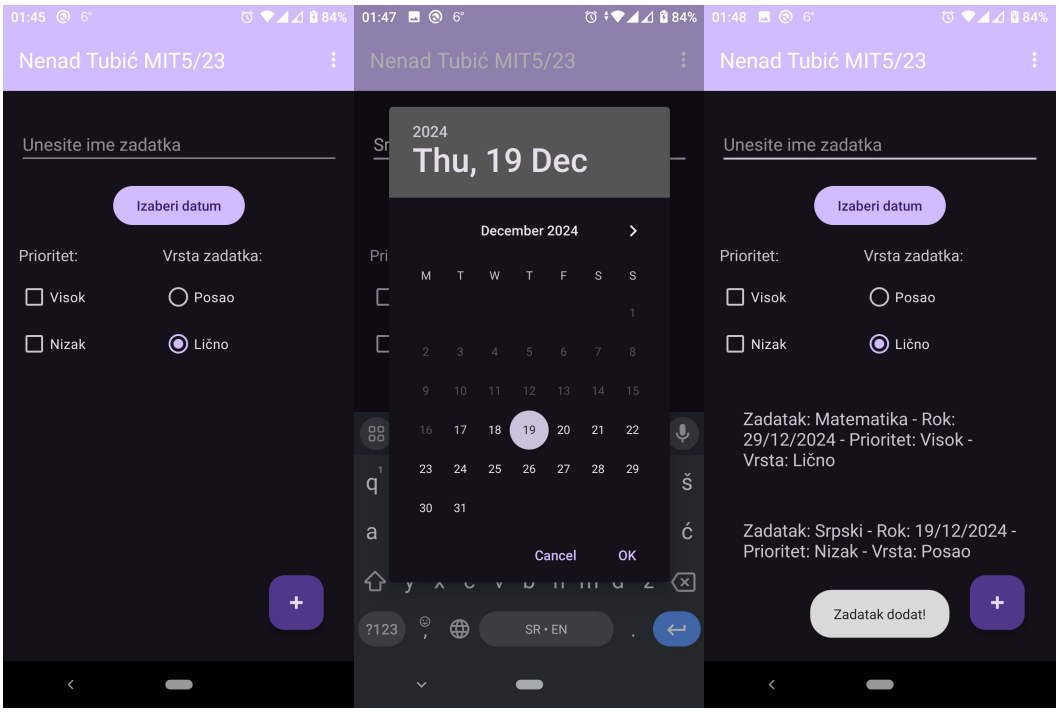
Дикумент Андроид апликације

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	3	6		***cb_high_priority, cb_low_priority***	3

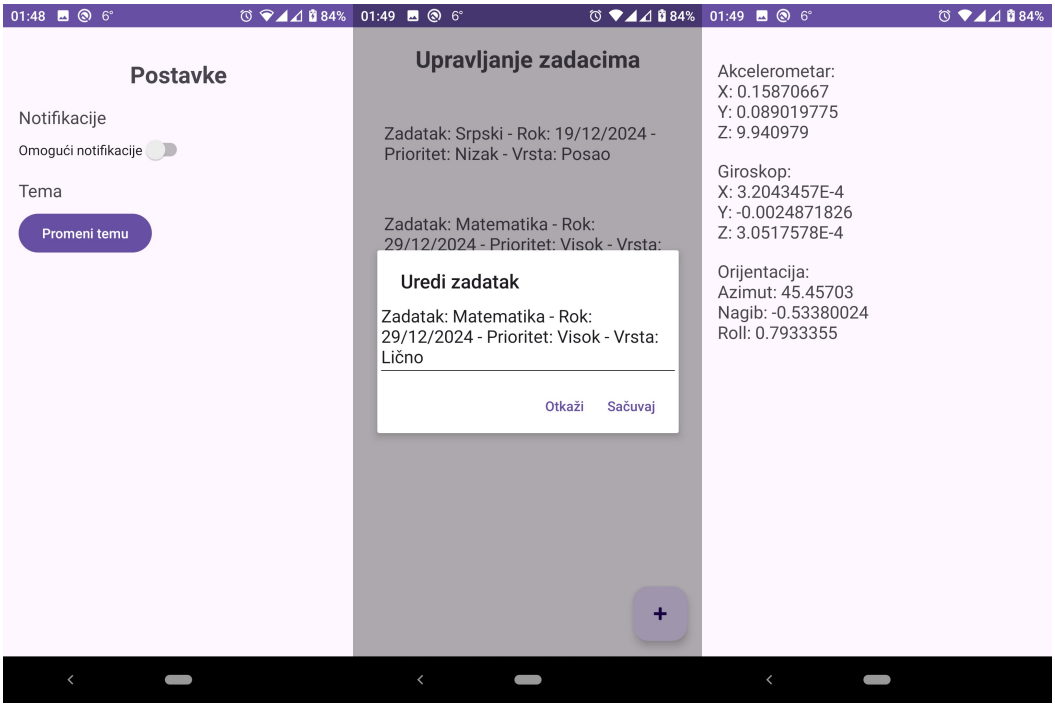
	2 ili više opcija				activity_main	
	Radio button skup od 2 ili više opcija	3	6		***rb_work, rb_personal*** (activity_main)	3
	Ostale komponente	2	4		***DatePicker*** (activity_main)	2
LAYOUt	LinearLayout (Vertical)	4	8		activity_main...	8
	LinearLayout (Horizontal)	4	8		activity_main	4
	Relative Layout	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	5	10			

	zvucnih efekata pomocu SoundPool					
	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**

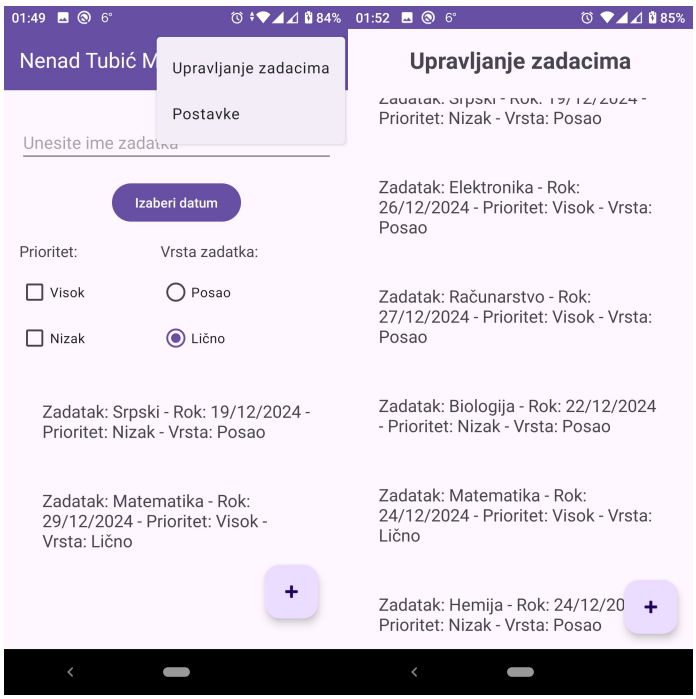
Скриншотови и изворни код апликације



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;&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;&amp;amp; gyroscope != null &amp;amp;&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;&amp;amp; gyroscope != null &amp;amp;&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;&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());
    }
}
```