A Sleep Tracking App for A Better Night's Rest

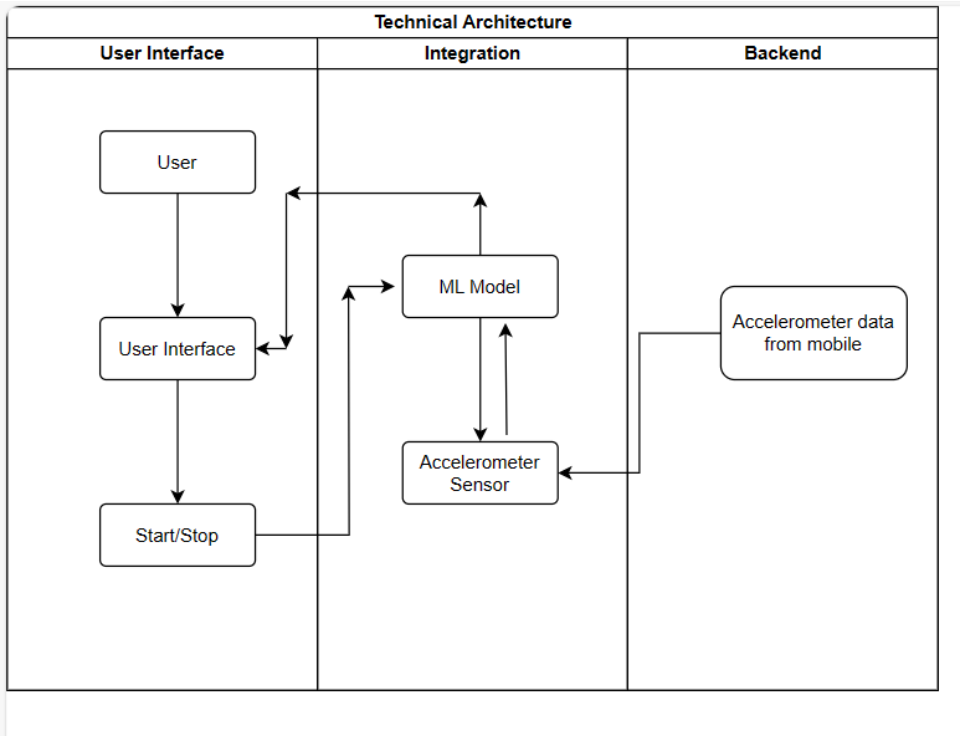
Introduction

Sleep tracking apps can be a helpful tool for improving your sleep quality. By tracking your sleep patterns over time, you can identify trends and patterns. This information can help you make changes to your sleep habits, such as going to bed and waking up at the same time each day, creating a relaxing bedtime routine, and avoiding caffeine and alcohol before bed.

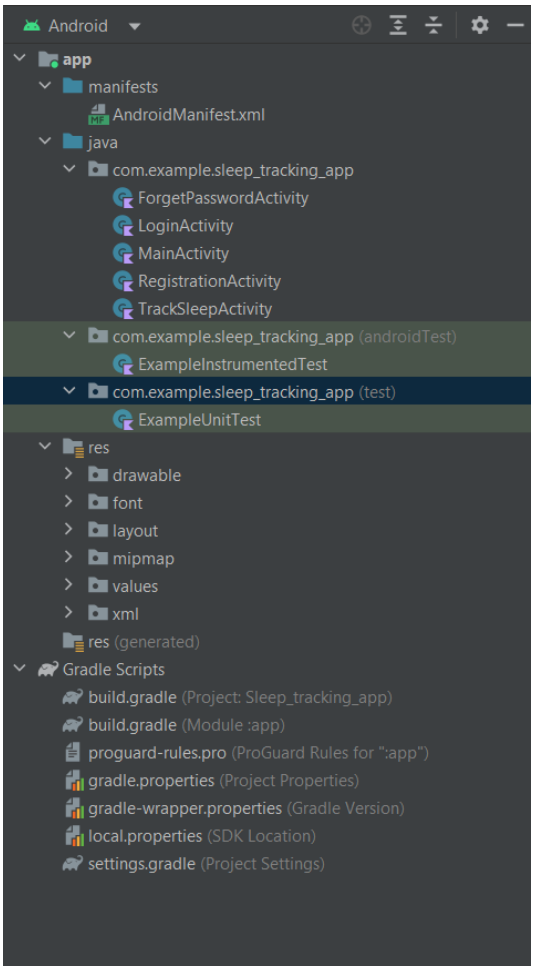
To use a sleep tracking app, simply place your smartphone near your bed while you sleep. The app will use the phone's sensors to track your movements and other physiological data. In the morning, you can open the app to view your sleep data and insights.

Overall, sleep tracking apps can be a valuable tool for improving your sleep quality. If you're interested in trying a sleep tracking app, there are many different options available. Some apps are free, while others require a subscription. Be sure to read reviews and compare features before choosing an app.

Technical Architecture



Project Structure



Dependencies

```
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

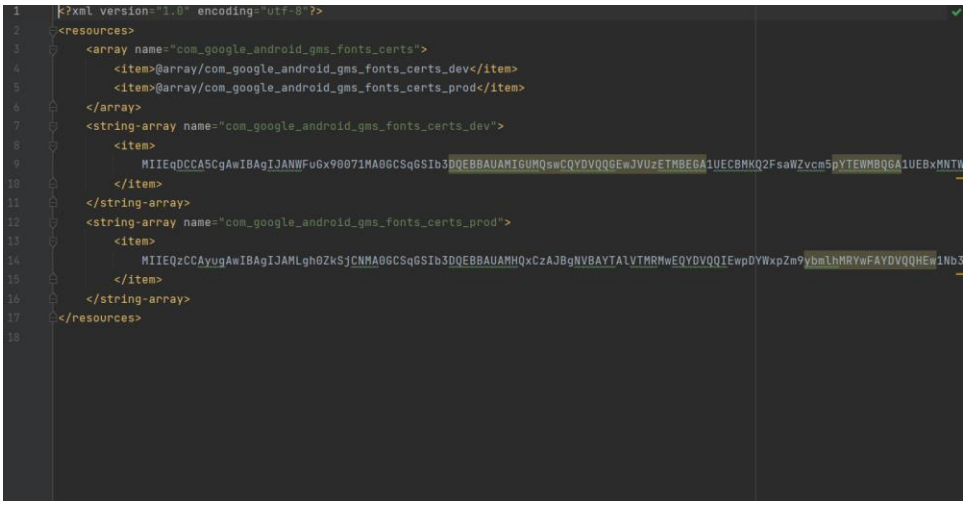
```
import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.FirebaseApp
import java.text.SimpleDateFormat
import java.util.Date
import java.util.Locale
```

```
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import java.text.SimpleDateFormat
import java.util.Date
import java.util.Locale
```

Manifest

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.sleep_tracking_app">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Sleep_tracking_app"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Sleep_tracking_app"
        tools:targetApi="31">

        <activity
            android:name=".ForgetPasswordActivity"
            android:exported="false" />

        <activity
            android:name=".TrackSleepActivity"
            android:exported="false" />

        <activity
            android:name=".LoginActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name=".RegistrationActivity"
            android:exported="false" />

        <activity
            android:name=".MainActivity"
            android:exported="false" />

        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />
    </application>

</manifest>
```

Fonts XML



Project UI

Login Page

```
package com.example.sleep_tracking_app

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class LoginActivity : AppCompatActivity() {

    private lateinit var usernameEditText: EditText
    private lateinit var passwordEditText: EditText
    private lateinit var signInButton: Button
    private lateinit var forgetPasswordTextView: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        usernameEditText = findViewById(R.id.usernameEditText)
        passwordEditText = findViewById(R.id.passwordEditText)
        signInButton = findViewById(R.id.signInButton)
        forgetPasswordTextView = findViewById(R.id.forgetPasswordTextView)

        signInButton.setOnClickListener { it: View!
            val username = usernameEditText.text.toString()
            val password = passwordEditText.text.toString()

```

```
30
31              if (username.isNotEmpty() && password.isNotEmpty()) {
32                  // Valid username and password, open RegistrationActivity
33                  val intent = Intent( packageContext: this, RegistrationActivity::class.java)
34                  startActivity(intent)
35              } else {
36                  // Invalid username or password
37                  Toast.makeText(baseContext,  text: "Could not sign in. Please try again.", Toast.LENGTH_SHORT).show()
38              }
39          }
40
41          forgetPasswordTextView.setOnClickListener {  it: View!
42              val intent = Intent( packageContext: this, ForgetPasswordActivity::class.java)
43              startActivity(intent)
44          }
45      }
46  }
47
```

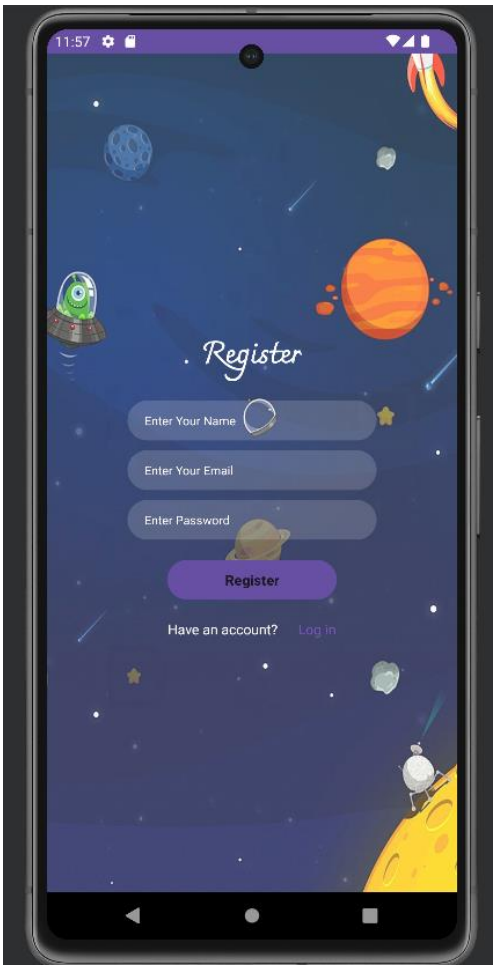
Result

Registration Page

```
package com.example.sleep_tracking_app

import ...

class RegistrationActivity : AppCompatActivity() {

    private lateinit var nameEditText: EditText
    private lateinit var emailEditText: EditText
    private lateinit var passwordEditText: EditText
    private lateinit var registerButton: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_registration)

        nameEditText = findViewById(R.id.nameEditText)
        emailEditText = findViewById(R.id.emailEditText)
        passwordEditText = findViewById(R.id.passwordEditText)
        registerButton = findViewById(R.id.registerButton)

        registerButton.setOnClickListener { it: View!
            val name = nameEditText.text.toString()
            val email = emailEditText.text.toString()
            val password = passwordEditText.text.toString()

            if (name.isNotEmpty() && email.isNotEmpty() && password.isNotEmpty()) {
                // Valid input, open MainActivity
                val intent = Intent( packageContext: this, MainActivity::class.java)
                startActivity(intent)
                finish()
```

```
            if (name.isNotEmpty() && email.isNotEmpty() && password.isNotEmpty()) {
                // Valid input, open MainActivity
                val intent = Intent( packageContext: this, MainActivity::class.java)
                startActivity(intent)
                finish()
            } else {
                // Invalid input
                Toast.makeText(baseContext,  text: "Please fill in all fields.", Toast.LENGTH_SHORT).show()
            }
        }
    }S
}
```

Result


11:57
Register
Enter Your Name
Enter Your Email
Enter Password
Register
Have an account?
Log in

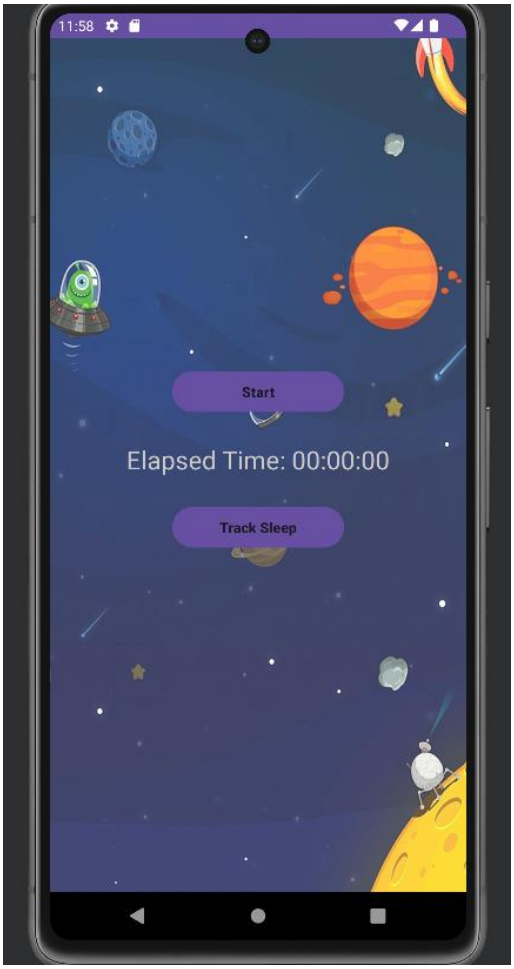
Dashboard

```
package com.example.sleep_tracking_app

import ...

class MainActivity : AppCompatActivity() {

    private lateinit var startSleepTrackingButton: Button
    private lateinit var elapsedTimeTextView: TextView
    private lateinit var trackSleepButton: Button

    private val handler = Handler(Looper.getMainLooper())
    private var startTimeMillis = 0L

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        FirebaseApp.initializeApp( context: this)

        setContentView(R.layout.activity_main)

        startSleepTrackingButton = findViewById(R.id.startSleepTrackingButton)
        elapsedTimeTextView = findViewById(R.id.elapsedTimeTextView)
        trackSleepButton = findViewById(R.id.trackSleepButton)

        startSleepTrackingButton.setOnClickListener { it: View!
            startTimeMillis = System.currentTimeMillis()
            handler.postDelayed(updateElapsedTime,  delayMillis: 1000)
        }

        trackSleepButton.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, TrackSleepActivity::class.java)
```

```
            // Set start and end times as extras
            val currentTimeMillis = System.currentTimeMillis()
            val endTimeMillis = currentTimeMillis + (8 * 60 * 60 * 1000) // Assuming 8 hours of sleep

            intent.putExtra( name: "startTime", currentTimeMillis)
            intent.putExtra( name: "endTime", endTimeMillis)

            startActivity(intent)
        }
    }

    private val updateElapsedTime = object : Runnable {
        override fun run() {
            val currentTimeMillis = System.currentTimeMillis()
            val elapsedTimeMillis = currentTimeMillis - startTimeMillis

            val hours = (elapsedTimeMillis / (1000 * 60 * 60)) % 24
            val minutes = (elapsedTimeMillis / (1000 * 60)) % 60
            val seconds = (elapsedTimeMillis / 1000) % 60

            val formattedTime = String.format("%02d:%02d:%02d", hours, minutes, seconds)

            elapsedTimeTextView.text = "Elapsed Time: $formattedTime"

            handler.postDelayed( r: this,  delayMillis: 1000)
        }
    }
```

```
71      override fun onDestroy() {
72          super.onDestroy()
73          handler.removeCallbacks(updateElapsedTime)
74      }
75  }
76
```

Result



Sleep Track

```kotlin
package com.example.sleep_tracking_app

import ...

class TrackSleepActivity : AppCompatActivity() {

    private lateinit var startTimeTextView: TextView
    private lateinit var endTimeTextView: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_track_sleep)

        startTimeTextView = findViewById(R.id.startTimeTextView)
        endTimeTextView = findViewById(R.id.endTimeTextView)

        val startTimeMillis = intent.getLongExtra( name: "startTime",  defaultValue: 0)
        val endTimeMillis = intent.getLongExtra( name: "endTime",  defaultValue: 0)

        val startTime = formatTime(startTimeMillis)
        val endTime = formatTime(endTimeMillis)

        startTimeTextView.text = "Start Time: $startTime"
        endTimeTextView.text = "End Time: $endTime"
    }

    private fun formatTime(timeMillis: Long): String {
        val dateFormat = SimpleDateFormat( pattern: "dd-MM-yyyy", Locale.getDefault())
        val timeFormat = SimpleDateFormat( pattern: "HH:mm:ss", Locale.getDefault())
        val date = Date(timeMillis)
        val formattedDate = dateFormat.format(date)
        val formattedTime = timeFormat.format(date)

        return "$formattedDate $formattedTime"
    }
}
```

Result

Summary

Sleep tracking apps can be a valuable tool for improving your sleep quality. By tracking your sleep patterns over time, you can identify trends and patterns. This information can help you make changes to your sleep habits, such as going to bed and waking up at the same time each day, creating a relaxing bedtime routine, and avoiding caffeine and alcohol before bed.

Sleep tracking apps can also help you identify and address potential sleep problems, such as insomnia, sleep apnea, and restless leg syndrome. This information can be helpful for discussing your sleep concerns with your doctor.

Overall, sleep tracking apps can be a helpful tool for improving your sleep quality and overall health and well-being.

Here are some tips for using a sleep tracking app effectively:

- Place your smartphone near your bed while you sleep.
- Use the app to set sleep goals.
- Track your sleep data over time.
- Identify trends and patterns in your sleep data.
- Make changes to your sleep habits based on your sleep data.
- Discuss your sleep data with your doctor if you have any concerns.