

# A SLEEP TRACKING APP FOR A BETTER NIGHTS REST

## 1. INTRODUCTION

### 1.1 Overview

Irregular sleeping patterns are a common problem. This mobile app will fulfill the user's needs in tracking their sleeping patterns, including duration and timings. This app will track three parameters: sleep time, wake up time, and sleep duration. Users can add, edit, or remove any sleep entries.

### 1.2 Purpose

This application enables you to start the timer when they are in the bed and about to fall asleep. The timer will keep running in the background until it is stopped, whenever the user wakes up. Based on the sleep experience, you can rate your sleep quality. Finally the app will display an analysis of the kind of sleep , you had the previous night.

## 2. PROBLEM DEFINITION & DESIGN THINKING

### 2.1 Empathy Map



## 2.2 Ideation & Brainstorming Map

### Must:

- Sleep Tracker
- Suggestions for sleep changes
- Descriptions of actual benefit of good sleep
- Stats on Sleeping Habits
- Notifications of alerts
- Blockers for certain applications during sleeping hours
- Pre-load semi-boring activity for users (repetition aids in sleepiness)
- Provide a prompt for the user to vent their thoughts through text or voice memo
- After stated bed time give reminder of how much time is being used on particular apps

### Should:

- An alert that give you the vital stuff for the day when you wake up
- Save articles that are somewhat interesting that you can read in the middle of the night
- Blocker for certain applications in the morning
- An alert that give you the vital stuff for the day when you wake up
- Save articles that are somewhat interesting that you can read in the middle of the night

### Could:

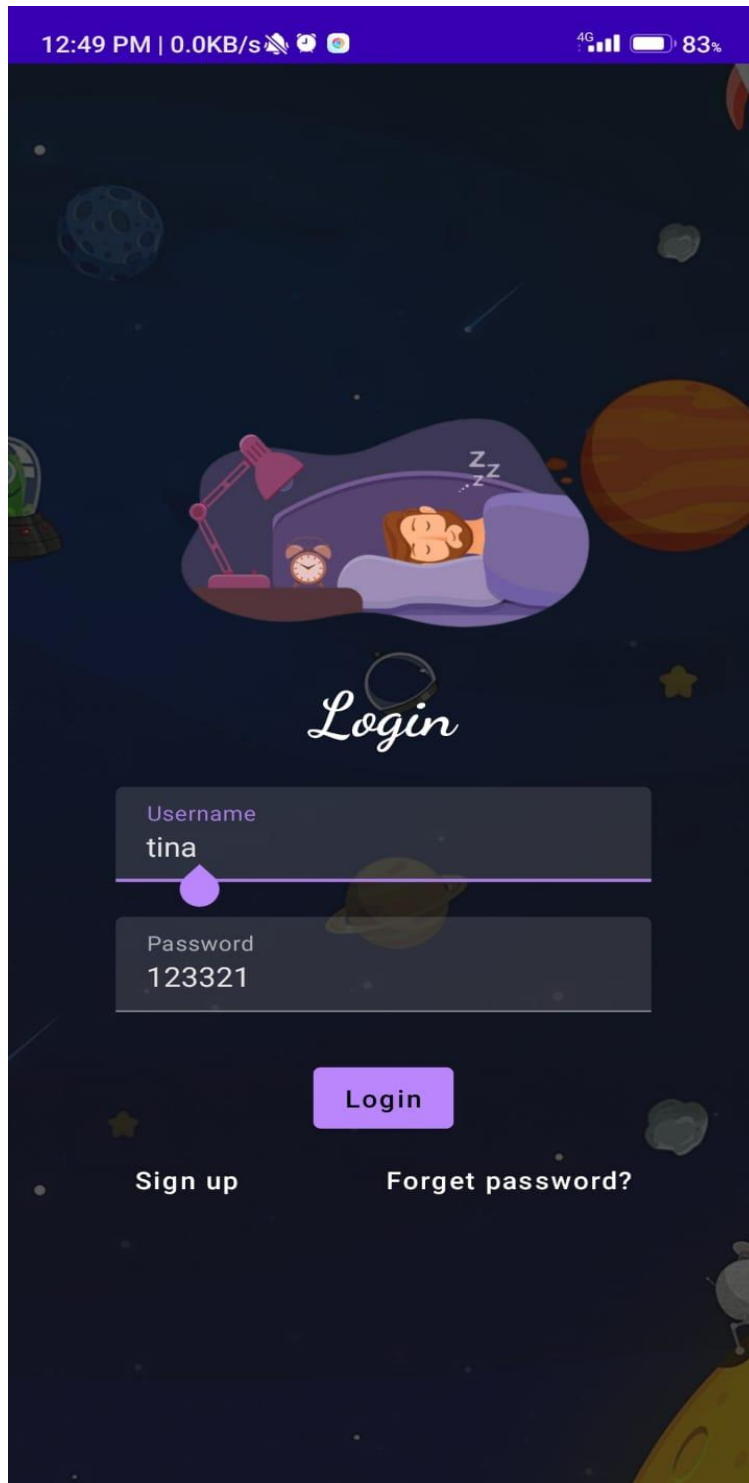
- Learning Algorithm that is applied to the user's habits
- Geo-Tracker
- Set an agenda for the night automatically for the user
- Regulate/Monitor the amount of shows user watches
- Give good thoughts/quotes for the user
- Automatically start chosen therapeutic device/sleeping aid to help user fall back to sleep
- Block time from user to remove anxiety from situation (some users)
- Learning Algorithm that is applied to the user's habits
- Geo-Tracker
- Set an agenda for the night automatically for the user so that they can not have to do guess work
- Regulate/Monitor the amount of shows user watches
- Repeat affirmations to the user to encourage them for the day to come
- Some complex action must be taken to turn alarm off
- Blocker for certain applications in the morning
- Phone only acts as an alarm for the first 30 mins in the morning

### Won't:

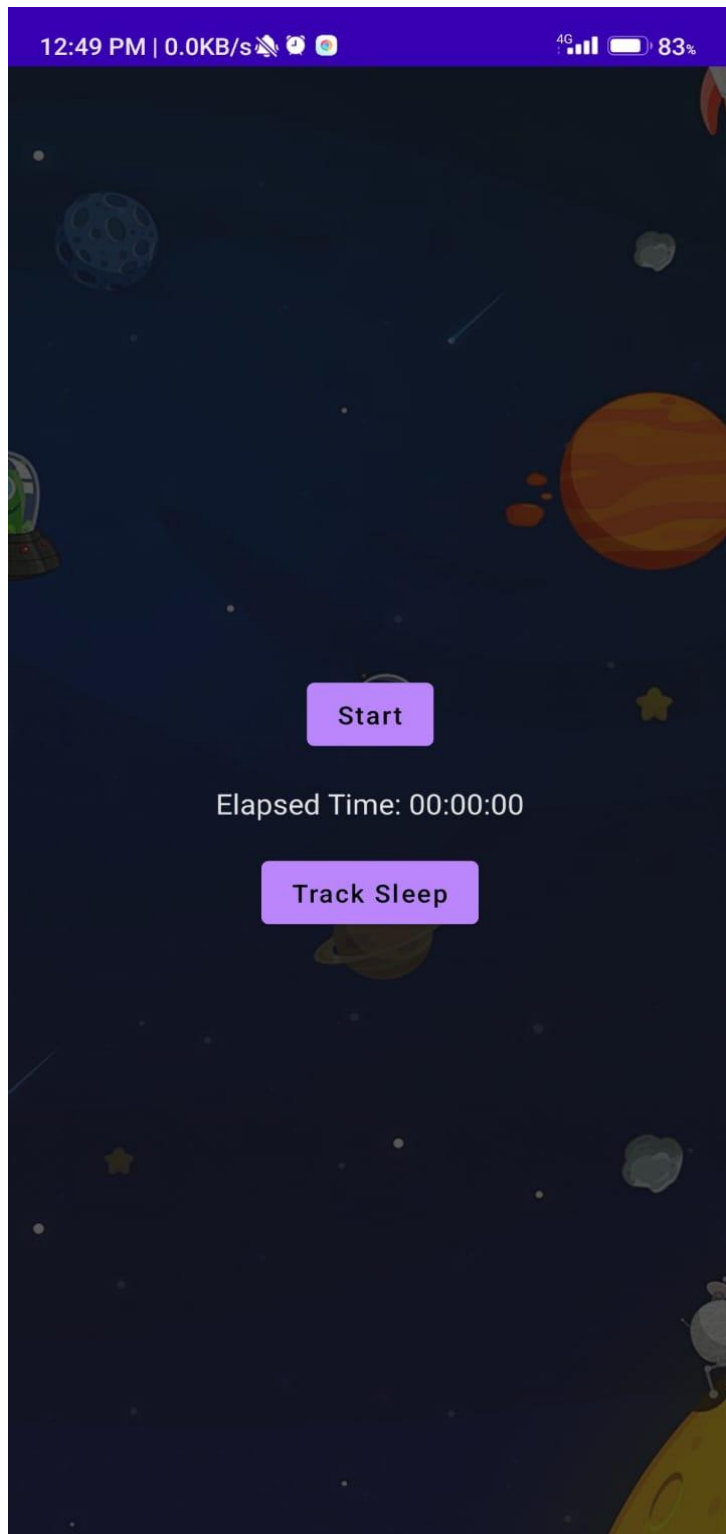
- Create a way to save the user's place or quickly return to the information during a break the next day
- Automatically opens a different app upon opening the phone in the morning
- Suggest quick no sweat movements to get blood flowing

### 3. RESULTS

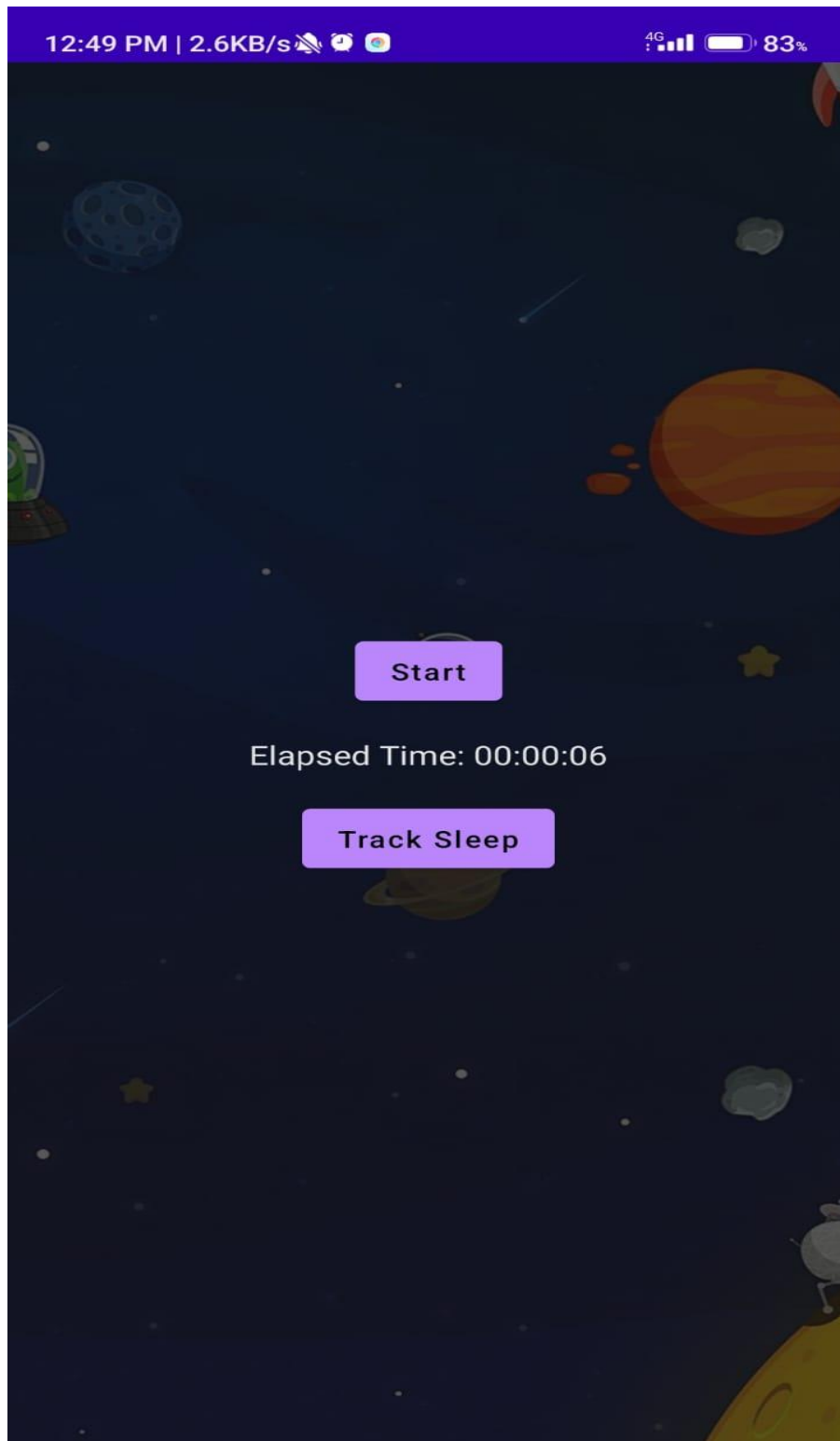
#### LOGIN SCREEN



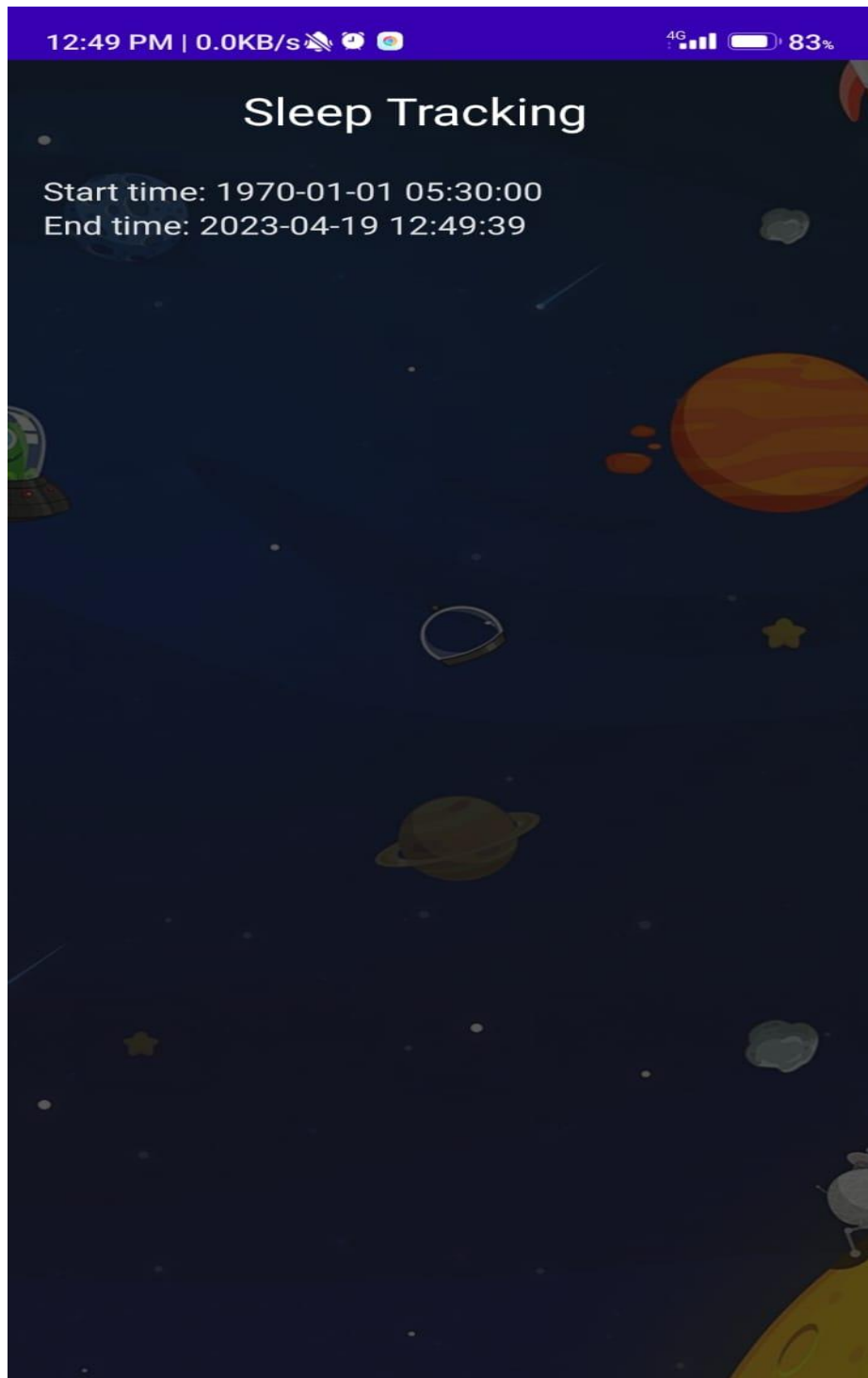
## SLEEP TRACK START SCREEN



## SLEEP TRACK STOPPED SCREEN



## SLEEP TRACKING SCREEN



## **4. ADVANTAGES & DISADVANTAGE**

### **4.1 ADVANTAGES OF SLEEP TRACKER**

Sleep trackers offer a variety of benefits for those looking to improve their quality and duration of sleep.

sleep trackers offer an invaluable benefit: heightened awareness. By diligently monitoring your sleep, these devices empower you to better understand your nocturnal habits.

Sleep trackers can prove to be incredibly insightful when it comes to detecting potential sleep issues. By closely monitoring your nightly rest, you can identify irregularities, such as sleep apnea or insomnia, which may have gone unnoticed otherwise.

Sleep trackers are designed to collect and analyze information over extended periods. They help you monitor trends and observe the effectiveness of interventions, like waking up consistently at the same time every day, reading a book before bed, or screening off two hours before bedtime.

### **4.2 DISADVANTAGES OF SLEEP TRACKER**

In the realm of sleep trackers, accuracy can be a hot topic since not all devices and apps are created equal, leading to varying degrees of precision. Of course, if your sleep tracker is inaccurate, that means the information you collect might cause you to make assumptions about your sleep habits that may not hold.

In the pursuit of better sleep, you may become overly fixated on numbers and metrics, inadvertently neglecting to listen to your body and address the underlying issues affecting your sleep.

## **5. APPLICATIONS**

A sleep tracker can make an educated guess about your sleep stages. But the only way to accurately identify what stage of sleep you are in.

Sleep tracking app track your sleep patterns and provide feedback on how to get better sleep.

## **6. CONCLUSION**

Sleep trackers may be useful in improving user's self-management, and increasing sleep hygiene awareness, knowledge and behaviours. Thus, apps may present valuable tools for improving sleep quality. It is recommended to assess behavioral changes associated with sleep trackers in different populations, such as elders, and people with sleep disorders and major illnesses.

Apps available for sleep tracking may be valuable tools to improving sleep quality, yet they require improvement in terms of quality and content, highlighting the need for further validity studies.

## **7. FUTURE SCOPE**

Sleep apps must undergo rigorous validation studies to ensure that their claims are evidence-based. These validation studies should compare the sleep data generated by the app with sleep data from a relevant gold-standard. Finally, the developers of sleep apps should listen to users and make the changes they recommend.



## 8. APPENDIX

### A. Source Code

```
package com.example.projectone

import android.content.Context
import android.content.Intent
import android.icu.text.SimpleDateFormat
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.material.Button
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.alpha
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.core.content.ContextCompatCompat
import com.example.projectone.ui.theme.ProjectOneTheme
import java.util.*

class MainActivity : ComponentActivity() {

    private lateinit var databaseHelper: TimeLogDatabaseHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = TimeLogDatabaseHelper(this)
        databaseHelper.deleteAllData()
        setContent {
            ProjectOneTheme {
                // A surface container using the 'background' color from the
theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    MyScreen(this, databaseHelper)
                }
            }
        }
    }
}

@Composable
fun MyScreen(context: Context, databaseHelper: TimeLogDatabaseHelper) {
    var startTime by remember { mutableStateOf(0L) }
    var elapsedTime by remember { mutableStateOf(0L) }
```

```

var isRunning by remember { mutableStateOf(false) }
val imageModifier = Modifier
Image(
    painterResource(id = R.drawable.sleeptracking),
    contentScale = ContentScale.FillHeight,
    contentDescription = "",
    modifier = imageModifier
        .alpha(0.3F),
)

Column(
    modifier = Modifier.fillMaxSize(),
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Center
) {
    if (!isRunning) {
        Button(onClick = {
            startTime = System.currentTimeMillis()
            isRunning = true
        }) {
            Text("Start")
            //databaseHelper.addTimeLog(startTime)
        }
    } else {
        Button(onClick = {
            elapsedTime = System.currentTimeMillis()
            isRunning = false
        }) {
            Text("Stop")
            databaseHelper.addTimeLog(elapsedTime, startTime)
        }
    }
    Spacer(modifier = Modifier.height(16.dp))
    Text(text = "Elapsed Time: ${formatTime(elapsedTime - startTime)}")

    Spacer(modifier = Modifier.height(16.dp))
    Button(onClick = { context.startActivity(
        Intent(
            context,
            TrackActivity::class.java
        )
    ) }) {
        Text(text = "Track Sleep")
    }
}

}

private fun startTrackActivity(context: Context) {
    val intent = Intent(context, TrackActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}

fun getCurrentDateTime(): String {
    val dateFormat = SimpleDateFormat("yyyy-MM-dd HH:mm:ss",
        Locale.getDefault())

```

```
    val currentTime = System.currentTimeMillis()
    return dateFormat.format(Date(currentTime))
}

fun formatTime(timeInMillis: Long): String {
    val hours = (timeInMillis / (1000 * 60 * 60)) % 24
    val minutes = (timeInMillis / (1000 * 60)) % 60
    val seconds = (timeInMillis / 1000) % 60
    return String.format("%02d:%02d:%02d", hours, minutes, seconds)
}
```