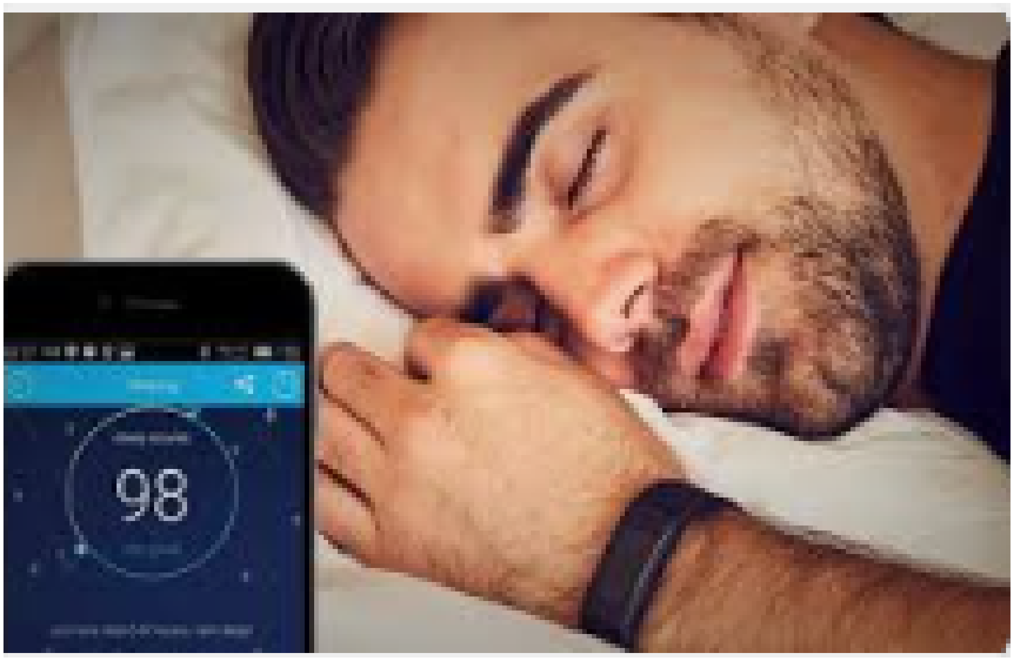
# A SLEEP TRACKING APP FOR A BETTER NIGHT'S REST

## 1. INTRODUCTION



- Sleep is a crucial biological process, and has long been recognised as an essential determinant of human health and performance.
- Whilst not all of sleep’s functions are fully understood, it is known to restore energy, promote healing, interact with the immune system and impact upon both brain function and behaviour.
- Even transient changes in sleep patterns, such as acute sleep deprivation, can impair judgement and cognitive

performance, whilst long-term aberrations have been linked to disease development.

- Global trends in sleep suggest a decrease on average sleep duration
- . Given these trends and the implications of sleep for health and well-being, better characterisation of sleep characteristics represents a public health priority.
- Sleep is known to be regulated by three main factors: circadian rhythms, sleep–wake homoeostasis and cognitive-behavioural influences
- . With regards to behavioural determinants, poor sleep quality (as defined by the National Sleep Foundation's recommendations based on total sleep time, sleep latency, wake after sleep onset, number of awakenings >5 min and sleep efficiency) has been associated with stress, anxiety, smoking, sugary drink consumption, workplace pressures, financial concerns, regularity of working hours, physical activity, sleep regularity and commuting times.
- Indeed longitudinal research has linked changes in physical activity to changes in the severity of sleep-disordered breathing and, hence, disturbed sleep
- Furthermore, dietary patterns have shown associations with sleep quality.
- It is now understood that the associations between diet, physical activity and sleep are bidirectional.

## OVERVIEW

- As new technologies are emerging in mobile apps across various industries, it makes sense that tech is being leveraged to improve and revolutionize the sleep app industry.

- While many sleep apps started off by offering similar capabilities, like soothing music, mediation, and basic sleep tracking, they've come a long way by not only improving on what already existed, but by using new technology to further sleep app innovation.

- Sleep Genius is an app that uses clinically-proven methods to help you fall asleep fast and stay asleep through the night. It includes or someone one can even mix and match the library's more than 50 sleep sounds to create the ideal sleeping environment.
- Sleep Genius also offers some features to help you get the most out of your sleep, including a personalized sleep schedule, sleep tracking, and insights into your sleep quality. It also integrates with Apple Health to track your sleep data over time.

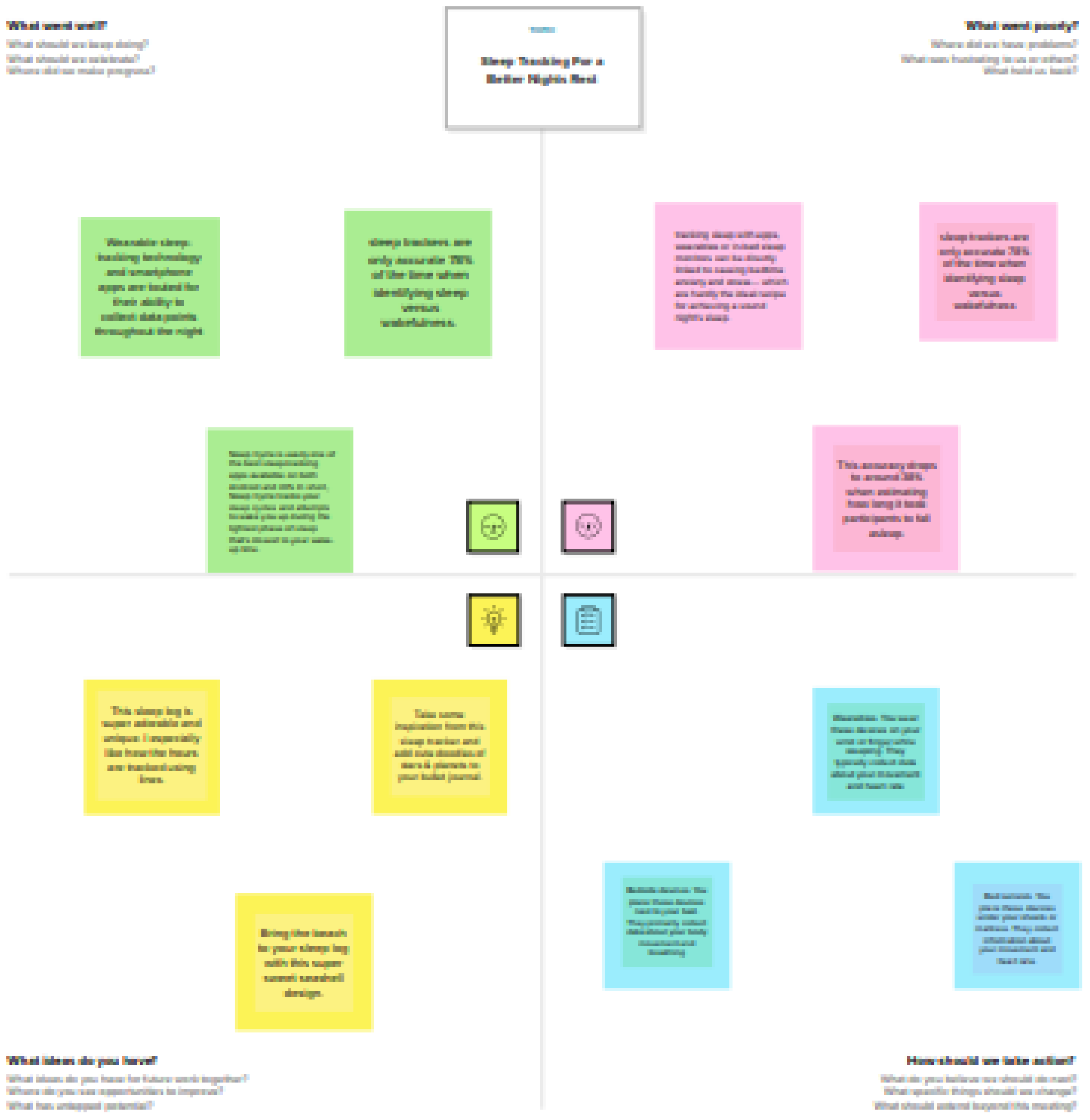
- There are some great sleep-tracking apps in the market, each with its own unique features and benefits. To find the best one for you, consider your needs and preferences and give a few of them a try. You're sure to find the perfect sleep-tracking app to help you get a good night's rest.

## PURPOSE

- Sleep is mysterious. But sleep-tracking apps promise to help you understand when you cross the threshold between waking and sleeping—and what happens in between.
- For people who want an uncomplicated interface with an intuitive design,recommend **SleepScore** (which works a lot better with iPhones than with Android phones) as well as **Sleep Cycle** (which is as compatible with Android models as it is with iPhones).
- After more than a hundred hours of research, including interviews with eight sleep scientists, and more than a month spent testing four popular sleep-tracking apps (along with a range of wearables, including the Oura Ring), we found that no app offers objectively accurate sleep analysis, and they can't replicate the experience of a traditional sleep lab.
- But you can use them to glean trends and patterns, which may help you improve your sleep over time.

# PROBLEM DEFINITION & THINKING

## EMPTHY MAP



Sleep Tracking For a Better Nights Rest

**What went well?**
What should we keep doing?
What should we celebrate?
Where did we make progress?

Wearable sleep-tracking technology and smartphone apps are touted for their ability to collect data points throughout the night

sleep trackers are only accurate 78% of the time when identifying sleep versus wakefulness.

Sleep Cycle is easily one of the best sleep-tracking apps available on both Android and iOS. In short, Sleep Cycle tracks your sleep cycles and attempts to wake you up during the lightest phase of sleep that's closest to your wake-up time.

**What went poorly?**
Where did we have problems?
What was frustrating to us or others?
What held us back?

tracking sleep with apps, wearables or in-bed sleep monitors can be directly linked to causing bedtime anxiety and stress — which are hardly the ideal recipe for achieving a sound night's sleep.

sleep trackers are only accurate 78% of the time when identifying sleep versus wakefulness.

This accuracy drops to around 38% when estimating how long it took participants to fall asleep.

**What ideas do you have?**
What ideas do you have for future work together?
Where do you see opportunities to improve?
What has untapped potential?

This sleep log is super adorable and unique. I especially like how the hours are tracked using lines.

Take some inspiration from this sleep tracker and add cute doodles of stars & planets to your bullet journal.

Bring the beach to your sleep log with this super sweet seashell design.

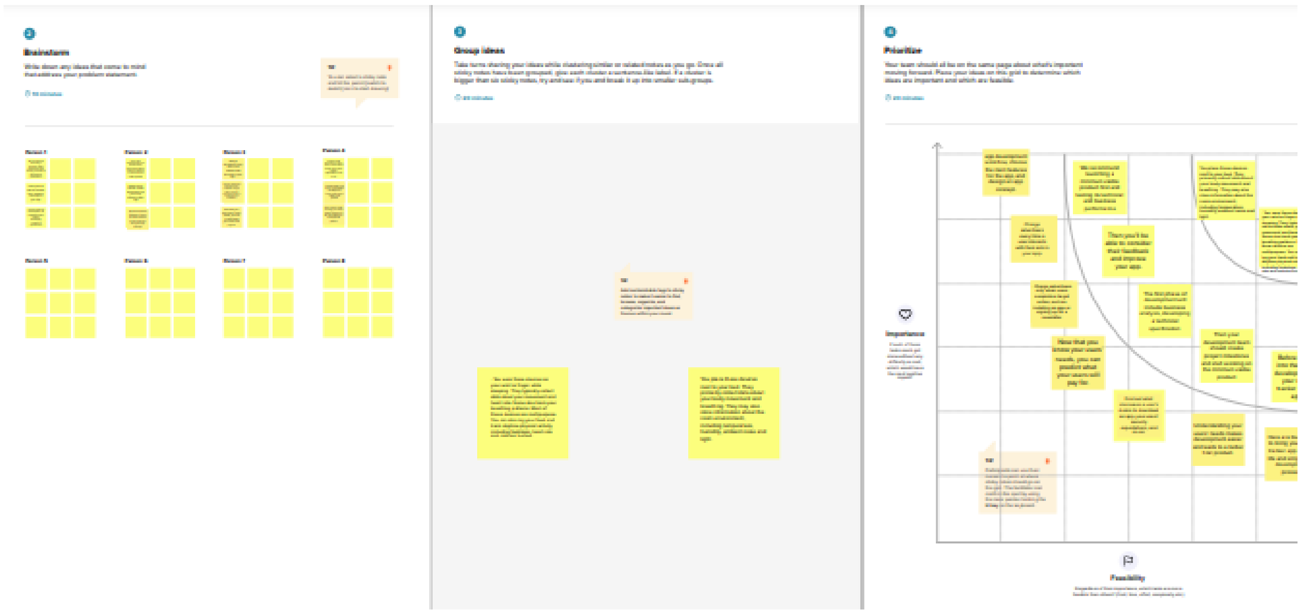
**How should we take action?**
What do you believe we should do next?
What specific things should we change?
What should extend beyond this meeting?

Wearables: You wear these devices on your wrist or finger while sleeping. They typically collect data about your movement and heart rate.

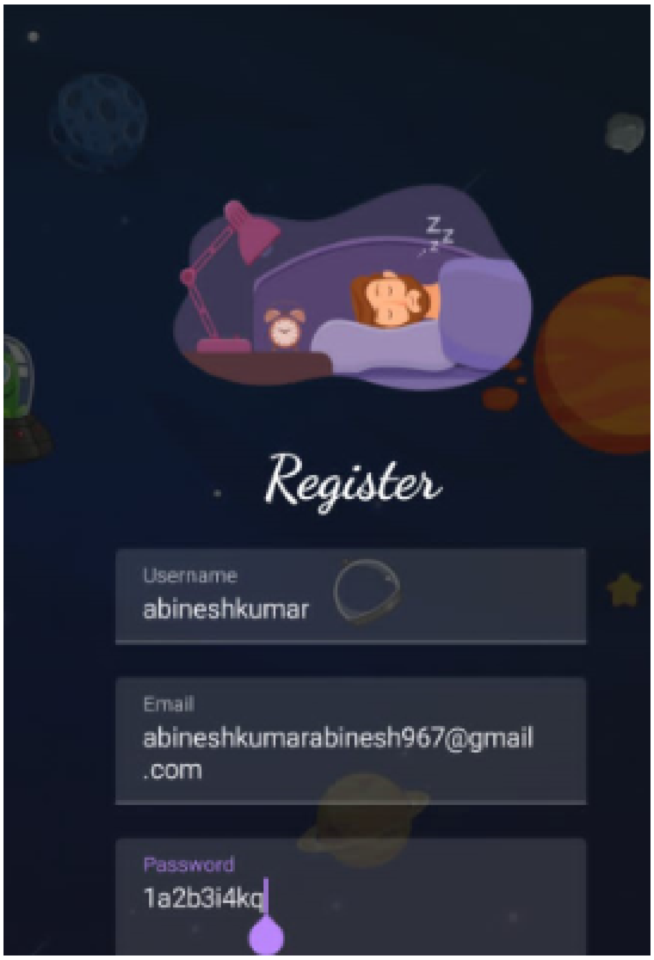
Bedside devices: You place these devices next to your bed. They primarily collect data about your body movement and breathing.

Bed sensors: You place these devices under your sheets or mattress. They collect information about your movement and heart rate.

## BRAINSTORMING



## 3.RESULT



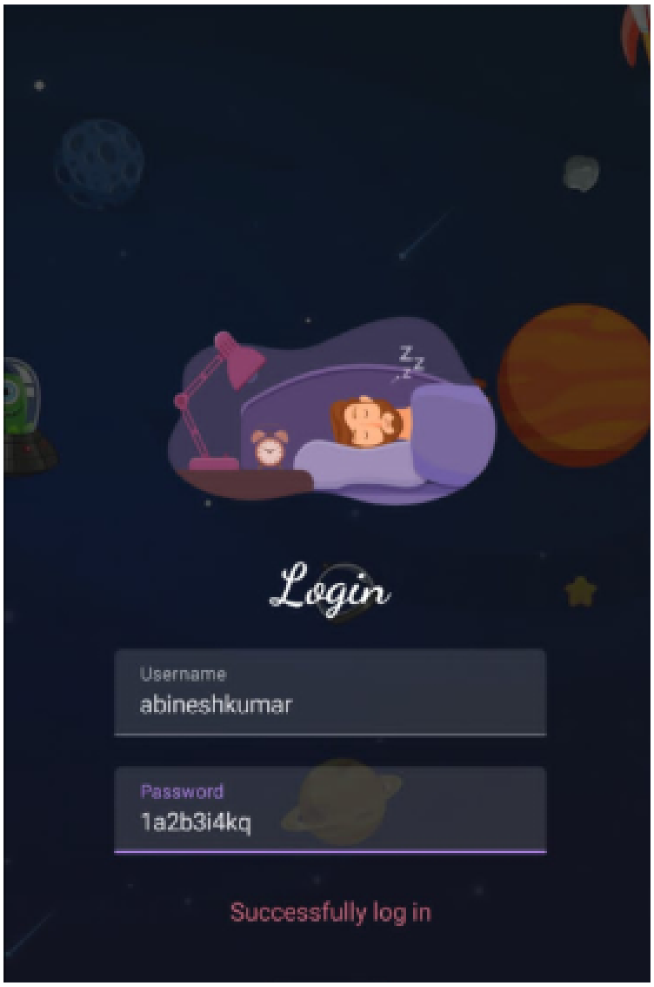
Register

Username
abineshkumar

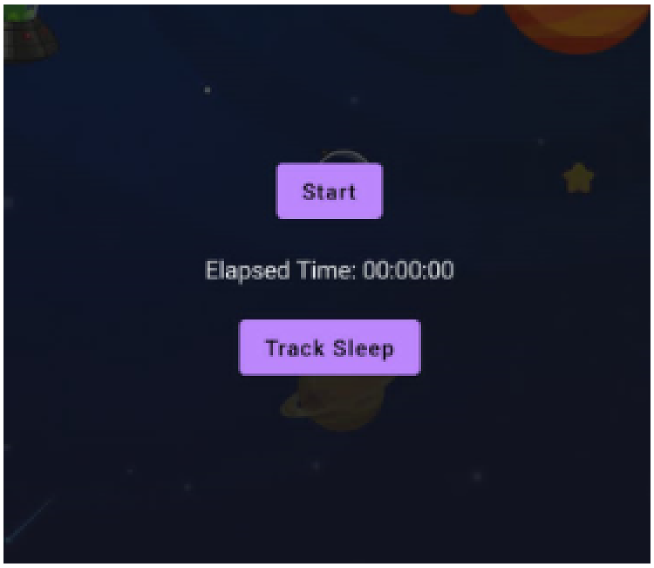
Email
abineshkumarabinesh967@gmail
.com

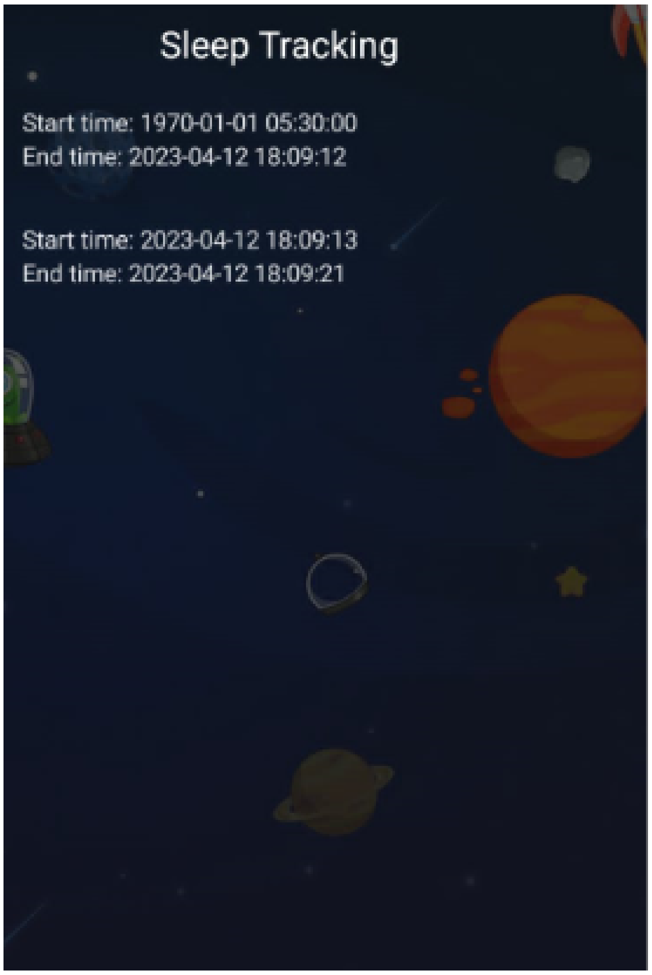
Password
1a2b3i4kq

User registered successfully



Username
abineshkumar
Password
1a2b3i4kq
Successfully log in





Start time: 1970-01-01 05:30:00
End time: 2023-04-12 18:09:12

Start time: 2023-04-12 18:09:13
End time: 2023-04-12 18:09:21

## 4.ADVANTAGES & DISADVANTAGES

### ADVANTAGES

- Trackers can detect interrupted sleep, letting you know when you're tossing and turning or waking during the night.
- Sleep phases: Some tracking systems track the phases of your sleep and time your alarm to go off during a period when you're sleeping less deeply.

### DISADVANTAGES

**Cons of Using a Sleep Tracker**

- Sleep trackers introduce poor sleep hygiene. ...
- Sleep trackers may be inaccurate. ...
- Sleep trackers can worsen insomnia. ...
- Sleep trackers make some people resistant to treatment. ...
- Sleep trackers are tied to a sleep disorder.

## 5.APPLIACTION

- Sleep Cycle
- Sleep as Android

- AutoSleep Track Sleep on

- If you are waking up tired or are concerned about the quality of your sleep, there are many apps to help. Sleep tracker apps analyze your sounds, movement, and behaviors as you sleep to give you a clear snapshot of the duration and quality of your sleep.
- These apps can also help you determine how much time you spend in Rapid Eye Movement (REM) sleep and how many times you are disturbed throughout the night.
- The best sleep tracker apps use input such as sound, heart rate, bedtime, or wake time to give you an accurate picture of your night.
- Many apps use data from wearable devices such as an Apple Watch to provide you with a sleep score and create graphs that show changes over time. Some apps allow you to export your data so you can share it with your healthcare professional to get assistance with your sleep issues.
- We looked at several sleep tracker apps to find the best ones to help you identify issues and improve your sleep quality.

## Best Sleep Apps of 2023

- **Best Overall:** SleepScore
- **Best for Apple Watch:** SleepWatch
- **Best Free:** Sleep++
- **Best for Extra Features:** Pillow
- **Best Versatile App:** Sleep Cycle
- **Best for Android:** PrimeNap

## 6.CONCLUSION

- Mobile apps have become a popular way to support sleep, and many such apps exist in the market.

- The most frequently provided functions by the apps are sleep monitoring, measuring sleep, providing alarms, and recording sleep using a sleep diary.
- There is a lack of apps that support active medical therapy. Therefore, the role of sleep apps in supporting sleep disorder treatments still needs further investigation.
- Easy-to-use, low-cost, simple device, mobility and flexibility features make the sleep apps an excellent choice to support sleep.
- However, current sleep apps have some limitations, such as accuracy, privacy and security issues, short battery life and information quality.
- Frequently reported design guidelines for developing sleep apps are user-centered, evidence-based, theory-based, engagement, feedback, accessible format and social connectedness. We did not find much information about the user requirements of sleep apps for different populations.
- Future research is needed to determine the user requirements for diverse populations.

- It was very interesting to discover that when users are using a health app they want to feel that it is more personal and not like they are using a cold automated device.
- For them, it was crucial to feel that their sleep is automatically tracked but that they also have the opportunity to provide their own input. Only then they would feel that their data has value for their health.
- As designers, it was a great reminder to discover that users with sleep issues need clean and organized info as this would cause them less stress.

- The fast pace routines usually affect them not only during the day but mainly right before their night sleep so it was crucial for them to have an app that would ease this disturbance.
- The covid pandemic and the cultural differences were two factors that determined the type of communication the users have with their health providers.
- Users prefer to have asynchronous/distant contact with their doctor for sleep issues.
- "Zzzloth" sleep tracking app seemed to the users a convenient way to communicate with their doctor efficiently and the feeling that they have someone taking care of them when needed.
- This was a very insightful project, during which I feel I grew a lot as a designer. It was obvious that having a thorough and solid User Research process, helped us to understand the user from all perspectives.
- This knowledge gave us the power to compose realistic assumptions and to visually design the app in the most efficient way, which our tester users also confirmed in the end.

## 7.FUTURE SCOPE.

- The design should be focused on an MVP, as the app was not supposed to solve all of the wellness issues people have.
- Must-Have:
  Users need to be able to set up their profile to include important information relevant to their goals
  Users need to be able to set goals and track their progress

Users need to be able to share their stats with their wellness coaches

- Nice to Have:
  Educational component:
- find a way for users to stay informed throughout the process and understand why they are doing these things, and how it will affect their well-being.
- Initially, we needed to do some market research so that we can identify which would be our direct and indirect competitors.
- We collected the most used sleep tracking apps like Sleep Cycle, Sleep++, Sleep Score, and Snore Lab, but also apps that track multiple health data like MiFit and Fitbit.
- After conducting Feature and Brand Analysis, we were ready to do the market positioning of our app.
- It seems that most apps related to sleep habits offer just one feature and that is mainly tracking sleep.
- Our goal would be to design an app that would provide help to the users so that they can analyze their sleep and improve their sleep habits
- Sleep is an essential biological need for human beings that will support us in getting resting, healing, and being ready for the next day.
- It is widely accepted that disturbed sleep is an influential factor leading to many mental health disorders. According to the Sleep Health Survey of Australian Adults (1), inadequate sleep has affected 33–45% of adults in Australia.
- Sleep disorders include short sleep duration, insomnia, snoring, sleep apnea, parasomnias, and restless leg syndrome. The treatment of sleep disorders varies, and some can be delivered online, such as Cognitive Behavioral

Therapy for insomnia. With the development of technology, there are fewer barriers to accessing mobile phones, and we can establish mobile apps with many useful functions.

- Thus, mobile apps have become a popular tool for delivering sleep treatments. For example, with the microphone and sound sensor of the mobile phone, we can monitor people's breath while sleeping.
- It is crucial to analyse the utilization of mobile apps to support sleep and further improve the quality of daily life.

# 8.APPENDIX

## ▢ SOURCE CODE

### • UESER CODE

```
package com.example.projectone


import androidx.room.ColumnInfo

import androidx.room.Entity

import androidx.room.PrimaryKey


@Entity(tableName = "user_table")
```

```
data class User(
    @PrimaryKey(autoGenerate = true) val id: Int?,
    @ColumnInfo(name = "first_name") val firstName: String?,
    @ColumnInfo(name = "last_name") val lastName: String?,
    @ColumnInfo(name = "email") val email: String?,
    @ColumnInfo(name = "password") val password: String?,

    )
```

## UserDao interface code

```
package com.example.projectone

import androidx.room.*

@Dao
interface UserDao {

    @Query("SELECT * FROM user_table WHERE email = :email")
    suspend fun getUserByEmail(email: String): User?

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insertUser(user: User)
    @Update
    suspend fun updateUser(user: User)

    @Delete
    suspend fun deleteUser(user: User)
}
```

## UserDatabase class code :

```
package com.example.projectone

import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {

    abstract fun userDao(): UserDao

    companion object {

        @Volatile
        private var instance: UserDatabase? = null


        fun getDatabase(context: Context): UserDatabase {
            return instance ?: synchronized(this) {
                val newInstance = Room.databaseBuilder(
                    context.applicationContext,
                    UserDatabase::class.java,
                    "user_database"
                ).build()
                instance = newInstance
                newInstance
            }
        }
    }
}
```

## UserDatabaseHelper class code :

```
package com.example.projectone


import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper


class UserDatabaseHelper(context: Context) :
    SQLiteOpenHelper(context, DATABASE_NAME, null,
DATABASE_VERSION) {


    companion object {
        private const val DATABASE_VERSION = 1
        private const val DATABASE_NAME = "UserDatabase.db"


        private const val TABLE_NAME = "user_table"
        private const val COLUMN_ID = "id"
        private const val COLUMN_FIRST_NAME = "first_name"
        private const val COLUMN_LAST_NAME = "last_name"
        private const val COLUMN_EMAIL = "email"
        private const val COLUMN_PASSWORD = "password"
    }


    override fun onCreate(db: SQLiteDatabase?) {
        val createTable = "CREATE TABLE $TABLE_NAME (" +
            "$COLUMN_ID INTEGER PRIMARY KEY
AUTOINCREMENT, " +
            "$COLUMN_FIRST_NAME TEXT, " +
            "$COLUMN_LAST_NAME TEXT, " +
            "$COLUMN_EMAIL TEXT, " +
            "$COLUMN_PASSWORD TEXT" +
            ")"

      db?.execSQL(createTable)
   }
```

## TimeLog data class code:

```
package com.example.projectone

import androidx.room.Entity
import androidx.room.PrimaryKey
import java.sql.Date

@Entity(tableName = "TimeLog")
```

```
data class TimeLog(
    @PrimaryKey(autoGenerate = true)
    val id: Int = 0,
    val startTime: Date,
    val stopTime: Date
)
```

TimeLogDao interface code:

```
package com.example.projectone


import androidx.room.Dao
import androidx.room.Insert


@Dao
interface TimeLogDao {
    @Insert
    suspend fun insert(timeLog: TimeLog)


}
```

## TimeLogDao interface code:

```
package com.example.projectone


import androidx.room.Dao
import androidx.room.Insert


@Dao
interface TimeLogDao {
    @Insert
    suspend fun insert(timeLog: TimeLog)


}
```

## AppDatabase class code:

```
package com.example.projectone

import android.content.Context
import androidx.room.Database

import androidx.room.Room

import androidx.room.RoomDatabase


@Database(entities = [TimeLog::class], version = 1,
exportSchema = false)

abstract class AppDatabase : RoomDatabase() {


    abstract fun timeLogDao(): TimeLogDao


    companion object {

        private var INSTANCE: AppDatabase? = null


        fun getDatabase(context: Context): AppDatabase {

            val tempInstance = INSTANCE

            if (tempInstance != null) {

                return tempInstance

            }

            synchronized(this) {

                val instance = Room.databaseBuilder(

                    context.applicationContext,

                    AppDatabase::class.java,
            "app_database"
          ).build()
          INSTANCE = instance
          return instance
        }
      }
    }
}
```

TimeDatabaseHelper class code:

```
package com.example.projectone

import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import
android.database.sqlite.SQLiteOpenHelper
```

```
import java.util.*


class TimeLogDatabaseHelper(context: Context)
: SQLiteOpenHelper(context, DATABASE_NAME,
null, DATABASE_VERSION) {
    companion object {
        private const val DATABASE_NAME =
"timelog.db"
        private const val DATABASE_VERSION = 1
        const val TABLE_NAME = "time_logs"
        private const val COLUMN_ID = "id"
        const val COLUMN_START_TIME =
"start_time"
        const val COLUMN_END_TIME = "end_time"


        // Database creation SQL statement
        private const val DATABASE_CREATE =
            "create table $TABLE_NAME
($COLUMN_ID integer primary key
autoincrement, " +
                "$COLUMN_START_TIME integer not
null, $COLUMN_END_TIME integer);"
    }

    override fun onCreate(db: SQLiteDatabase?) {
        db?.execSQL(DATABASE_CREATE)
    }

    override fun onUpgrade(db: SQLiteDatabase?,
oldVersion: Int, newVersion: Int) {
        db?.execSQL("DROP TABLE IF EXISTS
$TABLE_NAME")
        onCreate(db)
    }

    // function to add a new time log to the
database
    fun addTimeLog(startTime: Long, endTime:
Long) {
        val values = ContentValues()
        values.put(COLUMN_START_TIME, startTime)
        values.put(COLUMN_END_TIME, endTime)
        writableDatabase.insert(TABLE_NAME, null,
values)
    }


    // function to get all time logs from the
database
    @SuppressLint("Range")
    fun getTimeLogs(): List<TimeLog> {
        val timeLogs = mutableListOf<TimeLog>()
        val cursor =
readableDatabase.rawQuery("select * from
$TABLE_NAME", null)
        cursor.moveToFirst()
        while (!cursor.isAfterLast) {
            val id =
cursor.getInt(cursor.getColumnIndex(COLUMN_I
D))
```

```
            val startTime =
cursor.getLong(cursor.getColumnIndex(COLUMN
_START_TIME))
            val endTime =
cursor.getLong(cursor.getColumnIndex(COLUMN
_END_TIME))
            timeLogs.add(TimeLog(id, startTime,
endTime))
            cursor.moveToNext()
        }
        cursor.close()
        return timeLogs
    }

    fun deleteAllData() {
        writableDatabase.execSQL("DELETE FROM
$TABLE_NAME")
    }

    fun getAllData(): Cursor? {
        val db = this.writableDatabase
        return db.rawQuery("select * from $TABLE_NAME", null)
    }

    data class TimeLog(val id: Int, val startTime: Long, val endTime: Long?) {
        fun getFormattedStartTime(): String {
            return Date(startTime).toString()
        }

        fun getFormattedEndTime(): String {
            return endTime?.let { Date(it).toString() } ?: "not ended"
        }
    }
}
```