TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Getting enough water every day is important for your health. Drinking water can prevent dehydration, a condition that can cause unclear thinking, result in mood change, cause your body to overheat, and lead to constipation and kidney stones. Water has no calories, so it can also help with managing body weight and reducing calorie intake when substituted for drinks with calories, such as sweet tea or regular soda.

Water helps your body:

→ Keep a normal temperature.
→ Lubricate and cushion joints.
→ Protect your spinal cord and other sensitive tissues.
→ Get rid of wastes through urination, perspiration, and bowel movements

Drink water reminder is an application with main function is to help us keep water tracker we need to replenish and water drinking reminder in time. Users only need to select a gender and enter a weight number, it will help you calculate how much water should you drink per day.

The key features of our app that makes it different from other apps designed till now are as follows:

→ We can customize the amount of water we want to consume.
→ Users can set target on their daily basis.
→ Notifications can be turned on and off and also customized with respect to the intervals and the messages.
→ The User Interface is easy to use and manage.

Android is the most widely used mobile OS motorized by Linux kernel. The current system is developed on the basis of android platform.

CHAPTER 2: SYSTEM ANALYSIS

2.1 LITERATURE SURVEY

Today we all know that the utilization and also the demand of Mobile Application Development are increasing and it makes inspiration to make user friendly and effective type of application. Android is an open-source transportable package with Linux-based platform released by Google. It consists of the package, middleware, and computer program and application software. Certainly, Android is getting ready to become the foremost widely used OS on mobile phones, but with Android comes a security vulnerability that few users take into consideration. Among all mobile platforms Android is incredibly powerful because Google has made it filled with feature and it's open source platform and it's highly scalable. The android smart phones are truly helpful with open source platform and support all types of hardware. This open source platform attracts android application developers to make fabulous mobile applications [2]. There are lots of android based mobile applications that are freely available within the play store that are accessed by the people for various purposes. Android is open source platform and provides all information and services to any or all with none license fees. it's capacity of getting compatible with the majority browsers hence you'll make it compatible. we stock smart phones everywhere with us which might remind us instantly of what we must always do next. At times, we get such a lot involve in our work and busy schedule that we actually forget important occasions and events coming in our day -to-day life. People are setting out to realize how important it's to drink enough water during the course of the day. although the majority are aware it's important to drink water, many of us still don't seem to be getting the recommend amount in line with their weight and activity level.

2.2 EXISTING SYSTEM

It is very important for a person to keep track of the amount of sleep one gets and the pattern of sleep: regular, moderate and irregular to maintain good health. Keeping track of sleep data will help to improve bedtime habits and wake up feeling energetic. A weekly graph is generated to show the amount of sleep the user gets. This graphical visualization helps the user to analyse how one’s sleep patterns look like easily with different variations. It also gives the average number of hours the user slept based on the weekly graph.

This app focuses on reminding the user every hour to drink certain amount of water. Water reminder takes input from the user to give their gender, age, weight and amount of physical activity the user does. Based on those inputs, it calculates the amount of water the user has to consume daily. Once that is calculated, it reminds the user to drink portions of the calculated amount every hour by giving reminders

2.3 PROPOSED SYSTEM

First the app collects the information from the user regarding their age, weight. Then it sets the notification message, time interval (30min, 45min, 50min) and the ringtone in which all these can be customized. It allows the user to set the targets. User can add the amount of water he or she wants to intake and this value can be customized too. It shows a diagram which visualizes the total amout of water was consumed.

2.3.1 SCOPE OF THE PROJECT

→ To prevent dehydration by creating an interactive mobile application that helps the people remind to drink the sufficient water

→ In a busy schedule, people will not concentrate on sufficient amount of drinking water which prevents us from dehydration of our body

→ Drinking water is imperative to a healthy body and a body that functions at its ideal level.

→ There are different styles of water reminders for everyone and everyone can find an option that works for their lifestyle and for their budgets. In the end, one option is not better than the other

2.3.2 AIM OF THE PROJECT

This app features a water reminder to remind the user to drink water every hour until the daily limit is reached. This is done by calculating the amount of water that should be consumed every day based on the user's details like age, weight, physical activity and gender. Apart from these, to stay fit and healthy one should do some physical activity every day which helps to activate their body. The instructions and videos of some exercises, information on health tips, and natural remedies to some common health problems that can be cured without any medication are provided in the app. This would be a very convenient application for those who are very busy with their work life and have difficult time tracking their sleep, medicine times and staying hydrated at work.

CHAPTER 3: REQUIREMENT SPECIFICATIONS

3.1 SYSTEM REQUIREMENTS

The System Requirement Specification (SRS), the requirements specification for software system, is complete description of the functionality of the system to be implemented and may include a collection of cases describing the experiences that users will have with the software.

3.1.1 HARDWARE REQUIREMENTS

- Processor: Intel Core/ AMD Ryzen
- Processor Speed: 2.0 GHz
- RAM: 8 GB
- Storage Space: 2GB

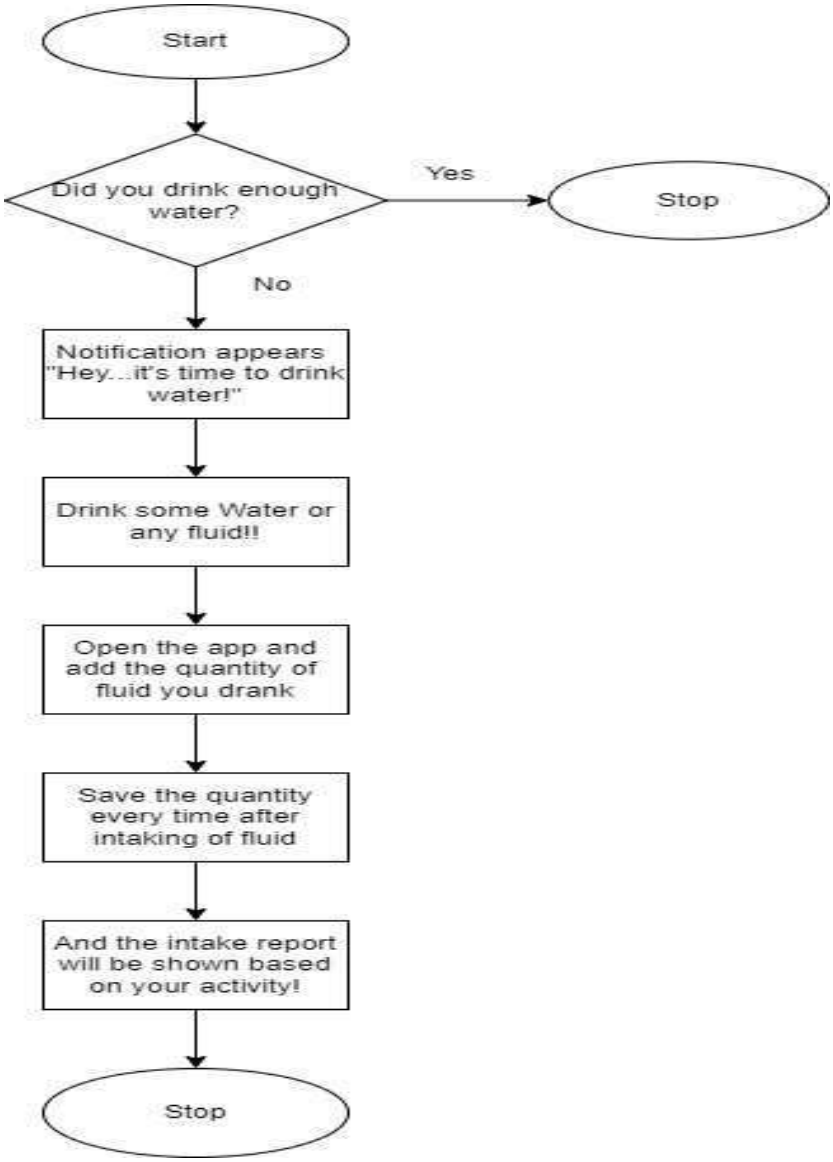
3.1.1 SOFTWARE REQUIREMENTS

(For App Development)

- Windows 10 Operating system (For Running app on phone)
- Android Version: Android Kitkat 4.4 & above.
- RAM: 2GB

CHAPTER 4: SYSTEM DESIGN

4.1 Project Flow Chart



As the application starts question is asked to the user, if the answer is no then notification is displayed asking the user to drink water. Upon this the user should add the quantity of fluid he or she wants to drinks. Quantity Is saved if with every intake. And finally the intake report is will be shown based on the activity.

CHAPTER 5: SYSTEM IMPLEMENTION

```
package keyur.diwan.project.waterReminder

import android.app.NotificationManager
import android.content.Intent
import android.content.SharedPreferences
import android.os.Bundle
import android.os.Handler
import android.text.TextUtils
import android.util.TypedValue
import android.view.LayoutInflater
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import com.daimajia.androidanimations.library.Techniques
import com.daimajia.androidanimations.library.YoYo
import com.google.android.material.snackbar.Snackbar
import com.google.android.material.textfield.TextInputLayout
import keyur.diwan.project.waterReminder.fragments.BottomSheetFragment
import keyur.diwan.project.waterReminder.helpers.AlarmHelper
import keyur.diwan.project.waterReminder.helpers.SqliteHelper
import keyur.diwan.project.waterReminder.utils.AppUtils
import kotlinx.android.synthetic.main.activity_main.*


class MainActivity : AppCompatActivity() {

private var totalIntake: Int = 0
private var inTook: Int = 0
private lateinit var sharedPref: SharedPreferences
private lateinit var sqliteHelper: SqliteHelper
private lateinit var dateNow: String
private var notificStatus: Boolean = false
private var selectedOption: Int? = null
private var snackbar: Snackbar? = null
private var doubleBackToExitPressedOnce = false


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

sharedPref = getSharedPreferences(AppUtils.USERS_SHARED_PREF,
AppUtils.PRIVATE_MODE)
sqliteHelper = SqliteHelper(this)

totalIntake = sharedPref.getInt(AppUtils.TOTAL_INTAKE, 0)

if (sharedPref.getBoolean(AppUtils.FIRST_RUN_KEY, true)) {
```

```
startActivity(Intent(this, WalkThroughActivity::class.java))
finish()
} else if (totalIntake <= 0) {
startActivity(Intent(this, InitUserInfoActivity::class.java))
finish()
}

dateNow = AppUtils.getCurrentDate()!!

}

fun updateValues() {
totalIntake = sharedPref.getInt(AppUtils.TOTAL_INTAKE, 0)

inTook = sqliteHelper.getIntook(dateNow)

setWaterLevel(inTook, totalIntake)
}

override fun onStart() {
super.onStart()

val outValue = TypedValue()
applicationContext.theme.resolveAttribute(
android.R.attr.selectableItemBackground,
outValue,
true
)

notificStatus = sharedPref.getBoolean(AppUtils.NOTIFICATION_STATUS_KEY, true)
val alarm = AlarmHelper()
if (!alarm.checkAlarm(this) && notificStatus) {
btnNotific.setImageDrawable(getDrawable(R.drawable.ic_bell))
alarm.setAlarm(
this,
sharedPref.getInt(AppUtils.NOTIFICATION_FREQUENCY_KEY, 30).toLong()
)
}

if (notificStatus) {
btnNotific.setImageDrawable(getDrawable(R.drawable.ic_bell))
} else {
btnNotific.setImageDrawable(getDrawable(R.drawable.ic_bell_disabled))
}

sqliteHelper.addAll(dateNow, 0, totalIntake)

updateValues()

btnMenu.setOnClickListener {
```

```
val bottomSheetFragment = BottomSheetFragment(this)
bottomSheetFragment.show(supportFragmentManager, bottomSheetFragment.tag)
}

fabAdd.setOnClickListener {
if (selectedOption != null) {
if ((inTook * 100 / totalIntake) <= 140) {
if (sqliteHelper.addIntook(dateNow, selectedOption!!) > 0) {
inTook += selectedOption!!
setWaterLevel(inTook, totalIntake)

Snackbar.make(it, "Your water intake was saved...!!", Snackbar.LENGTH_SHORT)
.show()

}
} else {
Snackbar.make(it, "You already achieved the goal", Snackbar.LENGTH_SHORT).show()
}
selectedOption = null
tvCustom.text = "Custom"
op50ml.background = getDrawable(outValue.resourceId)
op100ml.background = getDrawable(outValue.resourceId)
op150ml.background = getDrawable(outValue.resourceId)
op200ml.background = getDrawable(outValue.resourceId)
op250ml.background = getDrawable(outValue.resourceId)
opCustom.background = getDrawable(outValue.resourceId)

// remove pending notifications
val mNotificationManager : NotificationManager =
getSystemService(NOTIFICATION_SERVICE) as NotificationManager
mNotificationManager.cancelAll()
} else {
YoYo.with(Techniques.Shake)
.duration(700)
.playOn(cardView)
Snackbar.make(it, "Please select an option", Snackbar.LENGTH_SHORT).show()
}
}

btnNotific.setOnClickListener {
notificStatus = !notificStatus
sharedPref.edit().putBoolean(AppUtils.NOTIFICATION_STATUS_KEY, notificStatus).apply()
if (notificStatus) {
btnNotific.setImageDrawable(getDrawable(R.drawable.ic_bell))
Snackbar.make(it, "Notification Enabled..", Snackbar.LENGTH_SHORT).show()
alarm.setAlarm(
this,
sharedPref.getInt(AppUtils.NOTIFICATION_FREQUENCY_KEY, 30).toLong()
)
} else {
```

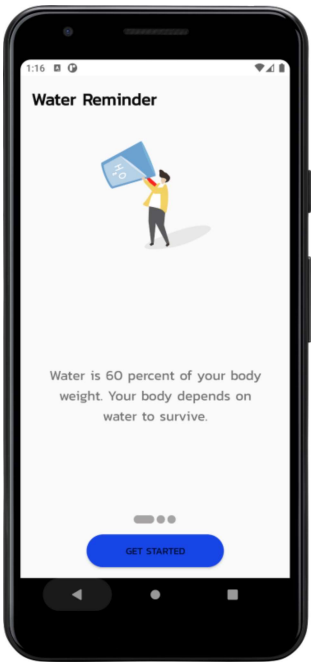
```
btnNotific.setImageDrawable(getDrawable(R.drawable.ic_bell_disabled))
Snackbar.make(it, "Notification Disabled..", Snackbar.LENGTH_SHORT).show()
alarm.cancelAlarm(this)
}
}

btnStats.setOnClickListener {
startActivity(Intent(this, StatsActivity::class.java))
}


op50ml.setOnClickListener {
if (snackbar != null) {
snackbar?.dismiss()
}
selectedOption = 50
op50ml.background = getDrawable(R.drawable.option_select_bg)
op100ml.background = getDrawable(outValue.resourceId)
op150ml.background = getDrawable(outValue.resourceId)
op200ml.background = getDrawable(outValue.resourceId)
op250ml.background = getDrawable(outValue.resourceId)
opCustom.background = getDrawable(outValue.resourceId)

}

op100ml.setOnClickListener {
if (snackbar != null) {
snackbar?.dismiss()
}
selectedOption = 100
op50ml.background = getDrawable(outValue.resourceId)
op100ml.background = getDrawable(R.drawable.option_select_bg)
op150ml.background = getDrawable(outValue.resourceId)
op200ml.background = getDrawable(outValue.resourceId)
op250ml.background = getDrawable(outValue.resourceId)
opCustom.background = getDrawable(outValue.resourceId)

}
op150ml.setOnClickListener {
if (snackbar != null) {
snackbar?.dismiss()
}
selectedOption = 150
op50ml.background = getDrawable(outValue.resourceId)
op100ml.background = getDrawable(outValue.resourceId)
op150ml.background = getDrawable(R.drawable.option_select_bg)
op200ml.background = getDrawable(outValue.resourceId)
op250ml.background = getDrawable(outValue.resourceId)
opCustom.background = getDrawable(outValue.resourceId)
```

```
}

op200ml.setOnClickListener {
if (snackbar != null) {
snackbar?.dismiss()
}
selectedOption = 200
op50ml.background = getDrawable(outValue.resourceId)
op100ml.background = getDrawable(outValue.resourceId)
op150ml.background = getDrawable(outValue.resourceId)
op200ml.background = getDrawable(R.drawable.option_select_bg)
op250ml.background = getDrawable(outValue.resourceId)
opCustom.background = getDrawable(outValue.resourceId)
}

op250ml.setOnClickListener {
if (snackbar != null) {
snackbar?.dismiss()
}
selectedOption = 250
op50ml.background = getDrawable(outValue.resourceId)
op100ml.background = getDrawable(outValue.resourceId)
op150ml.background = getDrawable(outValue.resourceId)
op200ml.background = getDrawable(outValue.resourceId)
op250ml.background = getDrawable(R.drawable.option_select_bg)
opCustom.background = getDrawable(outValue.resourceId)

}
opCustom.setOnClickListener {
if (snackbar != null) {
snackbar?.dismiss()
}

val li = LayoutInflater.from(this)
val promptsView = li.inflate(R.layout.custom_input_dialog, null)

val alertDialogBuilder = AlertDialog.Builder(this)
alertDialogBuilder.setView(promptsView)

val userInput = promptsView
.findViewById(R.id.etCustomInput) as TextInputLayout

alertDialogBuilder.setPositiveButton("OK") { dialog, id ->
val inputText = userInput.editText!!.text.toString()
if (!TextUtils.isEmpty(inputText)) {
tvCustom.text = "${inputText} ml"
selectedOption = inputText.toInt()
}
}.setNegativeButton("Cancel") { dialog, id ->
dialog.cancel()
```

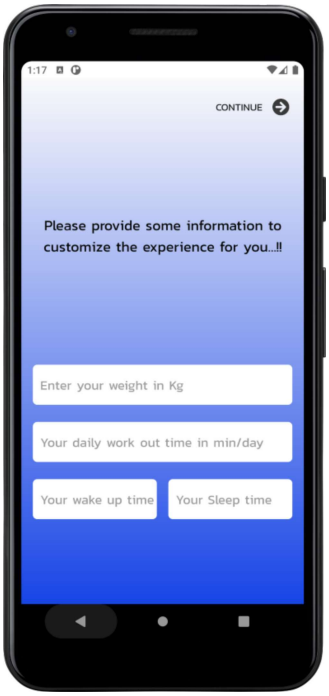
```
}

val alertDialog = alertDialogBuilder.create()
alertDialog.show()

op50ml.background = getDrawable(outValue.resourceId)
op100ml.background = getDrawable(outValue.resourceId)
op150ml.background = getDrawable(outValue.resourceId)
op200ml.background = getDrawable(outValue.resourceId)
op250ml.background = getDrawable(outValue.resourceId)
opCustom.background = getDrawable(R.drawable.option_select_bg)

}
}

private fun setWaterLevel(inTook: Int, totalIntake: Int) {

YoYo.with(Techniques.SlideInDown)
.duration(500)
.playOn(tvIntook)
tvIntook.text = "$inTook"
tvTotalIntake.text = "/$totalIntake ml"
val progress = ((inTook / totalIntake.toFloat()) * 100).toInt()
YoYo.with(Techniques.Pulse)
.duration(500)
.playOn(intakeProgress)
intakeProgress.currentProgress = progress
if ((inTook * 100 / totalIntake) > 140) {
Snackbar.make(main_activity_parent, "You achieved the goal", Snackbar.LENGTH_SHORT)
.show()
}
}

override fun onBackPressed() {
if (doubleBackToExitPressedOnce) {
super.onBackPressed()
return
}

this.doubleBackToExitPressedOnce = true
Snackbar.make(
this.window.decorView.findViewById(android.R.id.content),
"Please click BACK again to exit",
Snackbar.LENGTH_SHORT
).show()

Handler().postDelayed({ doubleBackToExitPressedOnce = false }, 1000)
}

}
```

CHAPTER 6: SNAPSHOTS

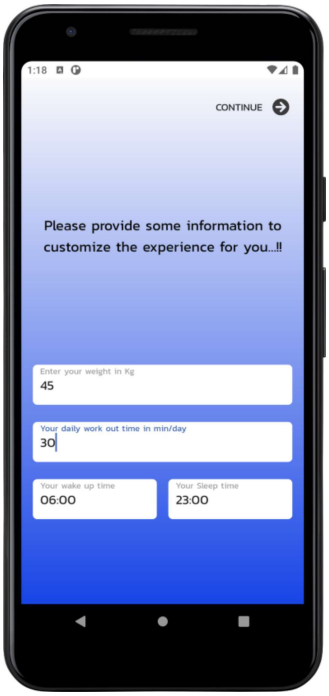
START PAGE



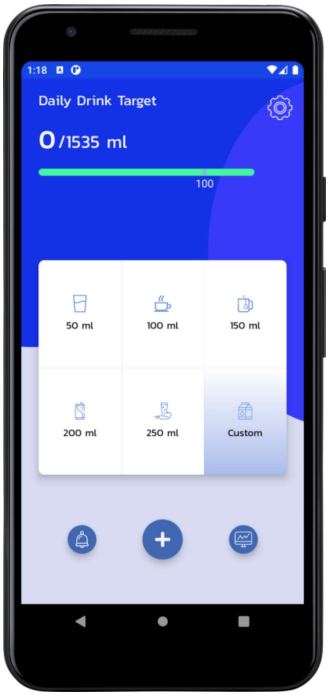
ENTER YOUR DETAILS



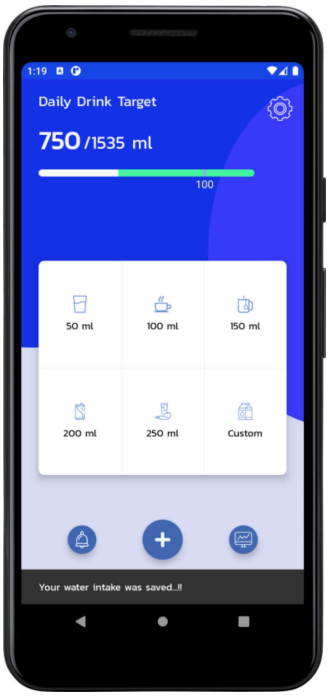
ENTERED DETAILS



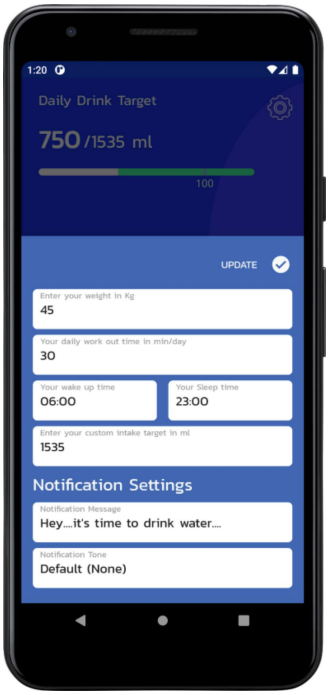
HOME PAGE



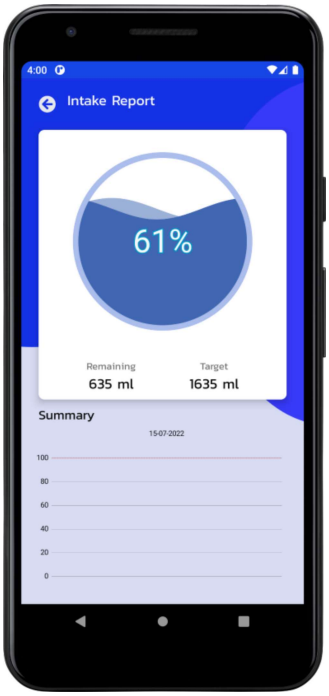
SAVED ACTIVITY



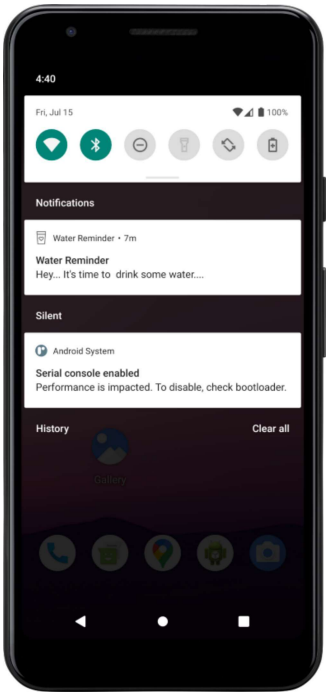
SETTINGS



REPORT



NOTIFICATION



CONCLUSION

This application is very convenient for users who want to focus on the overall health improvement. This application combines different factors like sleep, exercises, medicine & water reminders, health tips and natural remedies that contribute to the betterment of the user's health. In today's busy world, it is difficult for a lot of people to keep track of these things. This application focuses on improving the sleep patterns of the user by giving analysis of his sleep cycle, reminds the user to stay hydrated and take medicines on time, and provided them with health tips, natural remedies and exercises which will improve the health of the user.

FUTURE ENHANCEMENT

In light of these limitations, this section provides suggestions for future research studies on developing efficient and usable reminder apps. Future research can examine what content and functions need to feature on the ‘Add Task’ page. Studies could also involve different users testing each function and content and selecting which should be displayed, and which ‘Task title’ and date/time picker needs to be followed.

Moreover, usability testing should be conducted on two types of ‘Add Task’ page, either displaying the extra contents/functions on the same page or by using different popups.Developing or implementing sharing functions in reminder apps can also be examined. Developing the voice recognition on reminder app can also be explored. This technology is quite new and users face problems with using this technology to add their tasks in the reminder apps.Further research studies can investigate the comparative use of reminder apps among different groups of people. For example, a study can compare the use of reminder apps by groups of professionals, such as doctors, lawyers or businessmen, and compare how they use the same app for different types of tasks.Future studies can also incorporate reminder app developers as study participants, in order to better assist users of the reminder apps and better understand the problems faced by reminder app users.

REFERENCES

1. https://github.com/
2. https://developer.android.com/
3. https://www.google.com/
4. https://developer.android.com/design
5. https://stackoverflow.com/
6. Head First Android Development: A brain-friendly guide book by Bill Phillips and Brian Hardy
7. https://www.geeksforgeeks.org/