

# SMARTINTERNZ EXTERNSHIP PROJECT

## PROJECT REPORT

### HEALTH MEDICARE APP

#### PROJECT TEAM :

- 1) NAREN ADHARSH S      20BCE0121
- 2) SAHIL BHARTI      20BBS0059
- 3) PRIYESH KUMAR SINGH   20BBS0010
- 4) PREM KUMAR      20BBS0077

#### 1. INTRODUCTION :

##### 1.1 Overview

The Application is used to keep users aware about their Health status, maintain a Good Lifestyle in Day-to-Day Life.

##### 1.2 Purpose

Users can get a clear overall idea about their Health Status, take care of themselves without any dependency on others. The App has a lot of useful modules with a user-friendly interface.

#### 2. LITERATURE SURVEY :

##### 2.1 Existing problem

As the Country is populated, it is important to make resources available to people in an easier way. The Medical Industry is one of the critical factors of a Country. A Lot of people are unaware about how a poor Lifestyle can have impacts on their health in the Future. A lot of Medical Related Applications are available in the Market, but they target only a

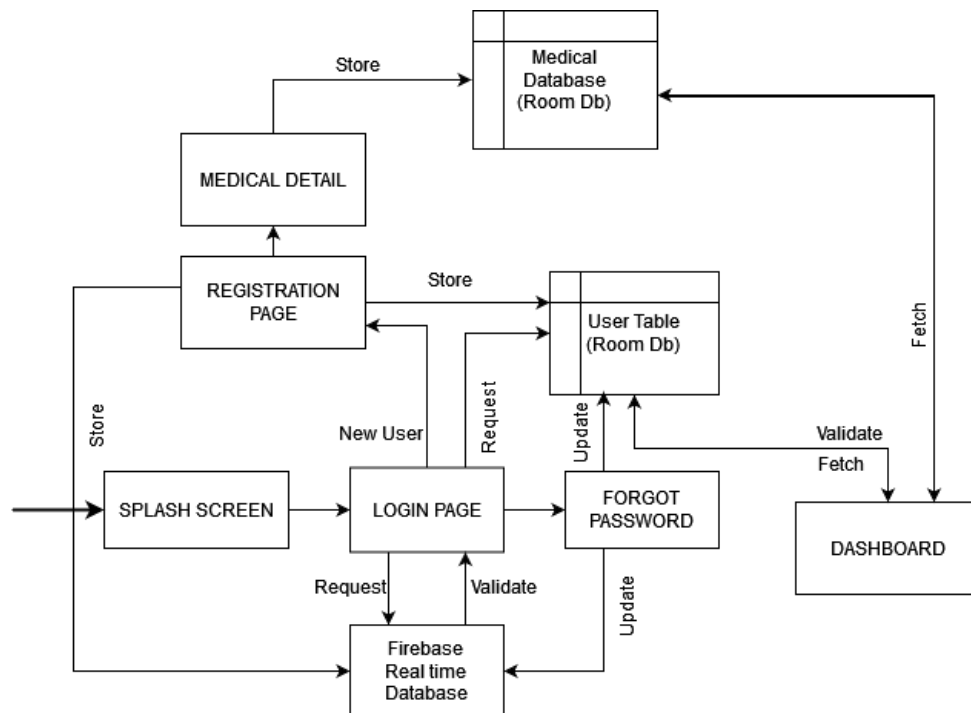
certain Department like Medicine Order, Doctor Consultation, what if we can bring all the Features within a Single Application.

## 2.2 Proposed solution

Thus, we propose **Health Medicare App**, an All-in-one Medical App, to solve all the Problems related to Health and Lifestyle. We will include all the Required features from Meditation Tracking to Doctor Consultation and Medicine Order. For now, we have done only 7 Modules including User Registration.

## 3. THEORITICAL ANALYSIS :

### 3.1 Block diagram (Database Control Flow)



### 3.2 Hardware / Software designing Hardware Requirement:

1) A PC / Laptop with more than 8Gb RAM.

#### Software Requirement:

- 1) Android Studio Flamingo.
- 2) Room DB.
- 3) Google Firebase Realtime Database.

- 4) Google Firebase Cloud Messaging.
- 5) A Virtual Android Emulator.

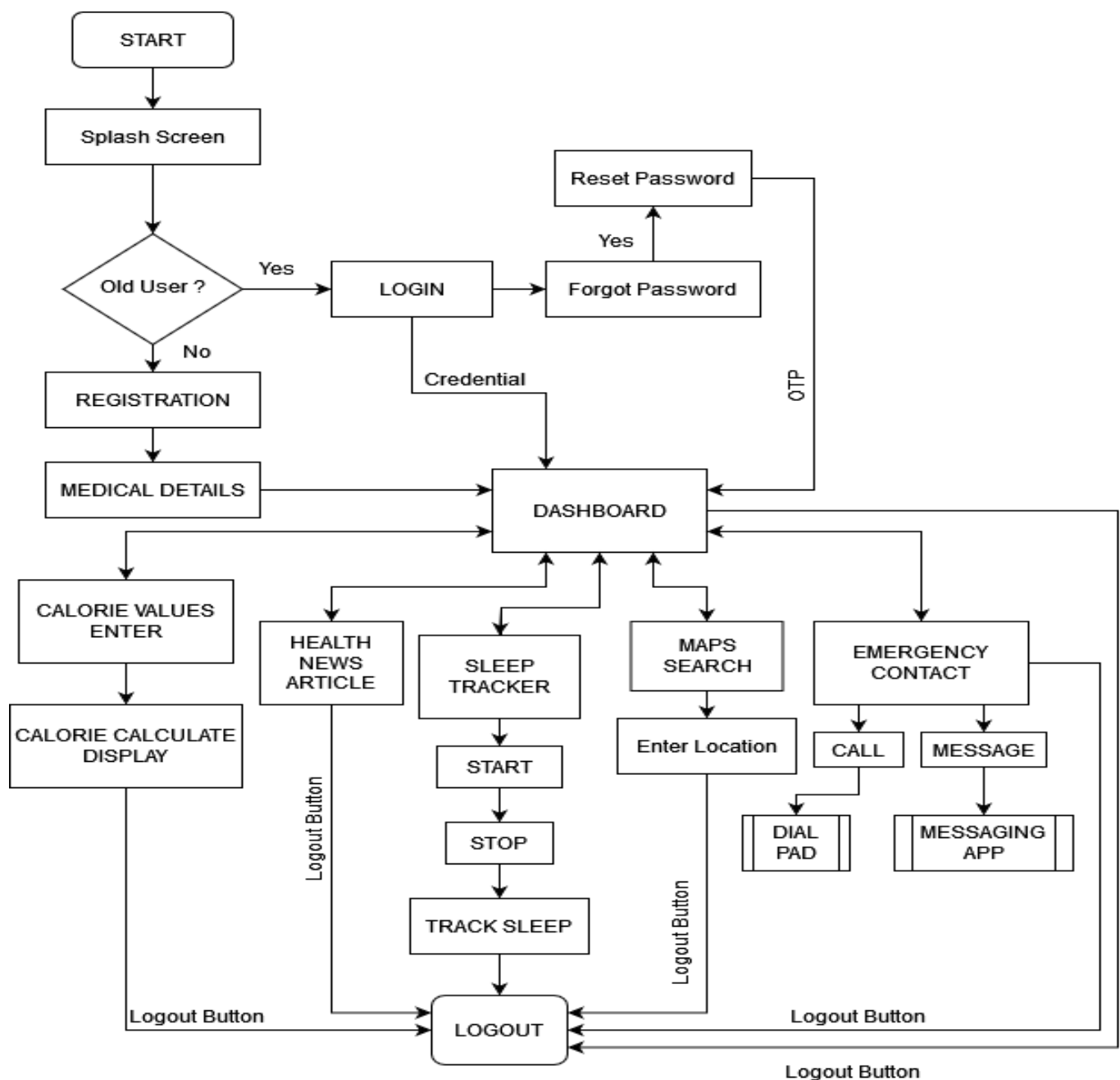
#### **4. EXPERIMENTAL INVESTIGATIONS :**

- 1) **User Registration / Forgot Password / Login**: We have to analyse which Database to use, so that it can store user's information properly. Also, it has to be stored in the Cloud as well as User's Device. So, we used both **Room Db and Firebase Realtime Database** to store Login Credentials. User Login Details in separate Table and Medical Details in another Table.
- 2) **Dashboard**: To display user's information on Successful Login in the Dashboard, we used **Room Db**. We give the User Email Address, to fetch all the user Related Details from the Database and reflect it back.
- 3) **Calorie Manage Page** : Collecting the Details from the User. Calculating all the **Calorie Data in the Background**, and displaying back to the user.
- 4) **Health Article** : We used **(An Open-source API)** to Load the Health Articles. We did research on finding the Best API to integrate within the App with proper architecture.
- 5) **Sleep Tracker** : We did some analysis on how to capture **current Timestamp in a variable**, convert them into Date, find difference, and display back to the User.
- 6) **Map Search** : We did some research on how to integrate **Google Maps** into an Android App. Also, how to display the Nearby hospitals at a given Location.
- 7) **Emergency Dial** : We did some research on finding all the important HealthCare Toll free numbers in the Country, how to integrate **Messaging and Call option** directly. We used **Intent** to successfully transfer the Mobile number.

- 8) **OTP (Forgot Password)** : We used **SMS-Manager** to directly send OTP to the user's Mobile Number.
- 9) **Daily Reminder Notifications** : We are using **Google Firebase Cloud Messaging** to send Automatic Scheduled Notifications to all the users.
- 10) **Permissions** : We ask user **SMS, Notification** Allow Permission.

## 5. FLOWCHART :

Diagram showing the Control flow of the Project



## 6. RESULT :

### Splash screen



### Registration

### Medical details

### Login

### Forget Password

### Dashboard

## Calorie Manage

15:08 41%

**CALORIE MANAGE**

Manage your Calorie Intake

Height 180 Weight 90 Age 20

Male

Lightly Active (1-3 day/ week)

Moderate Carb (50:30:20)

**CALCULATE CALORIE DATA**

Please Fill All the Details

## Calorie Manage

15:08 41%

**CALORIE MANAGE**

BODY MASS INDEX (BMI) 27

DAILY ENERGY CALORIE 2653

CALORIE (GAIN WEIGHT) 3184

CALORIE (LOSE WEIGHT) 2123

CARBO KCAL / DAY REQD 1133

CARBO GRAM / DAY REQ 331

PROTEIN KCAL/DAY REQ 680

PROTEIN GRAM/DAY REQ 199

FATS KCAL / DAY REQD 453

FATS GRAM / DAY REQD 50

## Health Article

15:08 40%

**HEALTH ARTICLE**

JUNE 29, 2023

**Insulin treatment might boost cognition in people with mild cognitive impairment or Alzheimer's disease**



Researchers report that using intranasal insulin may help some people with Alzheimer's disease and mild cognitive impairment. However, other medical...


**READ MORE**

## Sleep Track

15:09 40%

**SLEEP TRACK**

Keep Track of your Sleep



Keep the App Open in Background

'A Well-Spent Day brings Happy Sleep'  
Leonardo da Vinci, Italian Polymath

**START STOP**

**TRACK SLEEP**

Start : Fri, 30 Jun 2023 15:09:34

Stop : Fri, 30 Jun 2023 15:09:48

**Sleep Duration : 00:00:13**

## Maps Search

15:10 40%

**MAPS SEARCH**

hospital near Coimbatore



## Emergency Dial

15:10 40%

**EMERGENCY DIAL**

Important Emergency Contacts

**Police 100**

**Fire Extinguish 101**

**Ambulance 102**

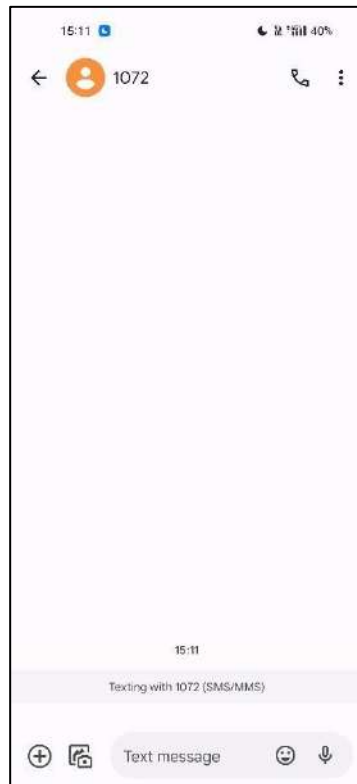
**Rail Accident 1072**

**Road Accident 1073**

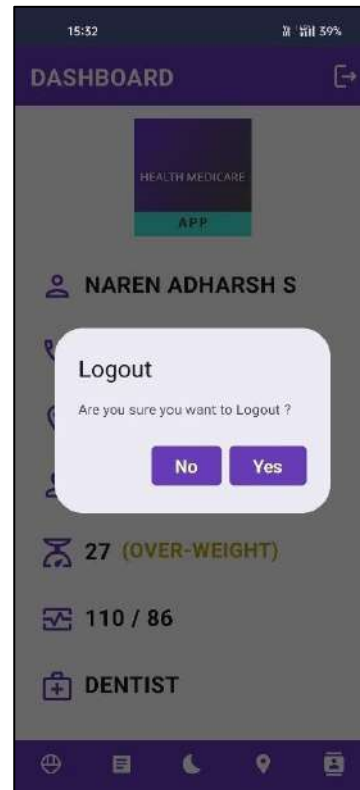
**Women Help 1091**

Contact them whenever Necessary

### Messaging



### Log out



## 7.ADVANTAGES & DISADVANTAGES

### • ADVANTAGES :

**Calorie Tracking :** User can know the BMI, amount of Calorie to be consumed based on Physical Activity and Diet Type. Carbohydrate, Protein, Fat Consumption in Calorie and Gram per day.

**Health News :** Users are aware about Latest Daily Health News in and around the World easily.

**Sleep Track :** Users can track their Nap / Sleep easily and maintain a healthy Sleep Schedule.

**Maps Search :** User can easily Locate Hospitals in and around their location with a minimal time in case of Emergency.

**Emergency Dial :** In case of Emergency, the user can directly call/message the important helpline from the Application.

**Health Notifications :** Notifications include Wake Up (7.30 AM Daily), Health News (8.00 AM Daily), Drink Water Reminder (12.30 PM Daily), Calorie Manage (12.00 PM every Sunday). These notifications are helpful to the user.

• **DISADVANTAGES :**

**Unable to Consult a doctor:** Users cannot Book an Appointment / Consult a doctor.

**Unable to Order Medicine :** Users cannot Order Medicine that are prescribed by the Doctor.

**Unable to make Payments :** Users cannot make Payments via the App. A user cannot pay the Doctor Consultation Fee / Medicine Purchase Payment.

**Login Every time :** The user has to login Every time with his credentials.

**8. APPLICATIONS :**

The App when deployed along with the Above Disadvantages can be launched in **Play store**.

Any Person can download the App and can make full use of it. Any user who wants to maintain a **Healthy Lifestyle** with all the Facilities can make use of this App.

**9. CONCLUSION :**

Summarizing Entire Work, we have developed a Medical Care Application, which can act as **one Stop Application** for All health-related issues. For Now, we have done only **7 modules (Login, Dashboard, Calorie Manage, Health News Article, Sleep Tracker, Maps Search, Emergency Dial)**. Along with the Reminder Notifications, the App can be useful in Dayto-Day Life.

**10. FUTURE SCOPE :**



In the Future, we are planning to overcome current Disadvantages by including Doctor Search according to Specific Location, Direct Consultation with Doctor (Call/Video), Book an Appointment with Doctor, Order Medicine According to the Doctor Prescription, Cart, Track Medicine Order, Payment Portal, Health Insurance. With this the App becomes a Complete Package, thus can be launched in Play store.

## **11 BIBILOGRAPHY :**

We took our inspiration from,

- 1) **Practo** – An Online Doctor Consulting Website / App.
- 2) **Pharmeasy** – An Online Medicine Order Website / App.

We referred multiple websites, 1)

Macronutrient Calculation Websites.

2) MedicalNewsToday.com API / Website.

3) Google Maps.

## **APPENDIX**

Source Code

**MainActivity.kt** File Code :

```
package com.example.health_medicare_application

class MainActivity : ComponentActivity() {
    private lateinit var databaseHelper1: UserDatabaseHelper
    private lateinit var databaseHelper2: MedicalDatabaseHelper
    @RequiresApi (Build.VERSION_CODES.TIRAMISU)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper1 = UserDatabaseHelper(this)
        databaseHelper2 = MedicalDatabaseHelper(this)
        setContentView {
            Health_MediCare_ApplicationTheme {
                Surface(
```

```

        modifier = Modifier.fillMaxSize(),
color = MaterialTheme.colorScheme.background
    ) {
        val firebaseDb = FirebaseDatabase.getInstance();
val reference = firebaseDb.getReference("User");
get_permission = rememberLauncherForActivityResult(
    ActivityResultContracts.RequestPermission()
    ) { isGranted ->
if (isGranted) {
    // Permission accepted Do Something
} else {
    // Permission not accepted show message
}
    }
    SideEffect {
get_permission.launch(POST_NOTIFICATIONS)
    }

App(applicationContext, reference, databaseHelper1, databaseHelper2)
    }
}
}

@Composable fun App(context: Context, databaseReference:
DatabaseReference, databaseHelper1: UserDatabaseHelper, databaseHelper2:
MedicalDatabaseHelper) {
    val navController = rememberNavController()
NavHost(
    navController = navController,
startDestination = "splashscreen"
    ) {
        composable("splashscreen") {
            SplashScreen(navController)
        }
        composable("reg") {
            RegistrationPage(context, navController, databaseReference,
databaseHelper1)
        }
        composable("medusreg") {
            MedicalRegPage(context, navController, databaseHelper2)
        }
        composable("login") {
            LoginPage(context, navController, databaseHelper1)
        }
        composable("forgotpw") {
            ForgotPasswordPage(context, navController, databaseReference,
databaseHelper1)
        }
        composable("dashboard/{email}") { backStackEntry ->
email = backStackEntry.arguments?.getString("email")
DashboardPage(navController, email.toString(), databaseHelper1,
databaseHelper2)
        }
        composable("caloriemgt") {
            CaloriePage(navController)
        }
        composable("article") {
            HealthArticlePage(navController)
        }
    }
}

```

```

        composable("maps") {
            MapPage(navController)
        }
        composable("sleep") {
            SleepTrackerPage(navController)
        }
        composable("emergency") {
            EmergencyContactPage(navController)
        }
    }
}
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun TopBar(abc:String) {
    CenterAlignedTopAppBar(
        colors = TopAppBarDefaults.smallTopAppBarColors(containerColor =
purple673),
        title = {
            Text(
                abc, color = purewhite,
                fontSize = fnt24, fontWeight = txtbold
            )
        }
    )
}
@Composable
fun TopApplicationBar(abc:String,navController: NavController) {
    Row(
        horizontalArrangement = Arrangement.SpaceBetween,
        verticalAlignment = Alignment.CenterVertically,
        modifier = Modifier
            .fillMaxWidth()
            .height(60.dp)
            .background(purple673)
    ) {
        val openDialog = remember { mutableStateOf(false) }
        val context = LocalContext.current
        Text(
            abc, color = purewhite,
            fontSize = fnt24, fontWeight = txtbold,
            modifier = Modifier.padding(start = 15.dp)
        )
        IconButton(
            onClick = { openDialog.value = true; }) {
            Icon(
                imageVector = Icons.Outlined.Logout,
                contentDescription = "Logout",
                tint = purewhite,
                modifier = icon,
            )
            if (openDialog.value) {
                AlertDialog(
                    onDismissRequest = { openDialog.value = false },
                    title = { Text(text = "Logout") },
                    text = { Text("Are you sure you want to Logout ?") },
                    confirmButton = {
                        Button(
                            openDialog.value = false;
                            Toast.makeText(
                                context, "Logout Successful ",
                                Toast.LENGTH_SHORT
                            ).show()
                        )
                    }
                )
            }
        }
    }
}

```

```

        navController.navigate("login")
    }, colors = ButtonDefaults.buttonColors(purple673),
    shape = rcshape
    )
    {
        Text(
            "Yes", color = purewhite,
            fontSize = fnt18, fontWeight = txtbold,
        )
    }
    },
    dismissButton = {
        Button(
            onClick = {
                openDialog.value = false
            }, colors = ButtonDefaults.buttonColors(purple673),
            shape = rcshape
        )
        {
            Text(
                "No", color = purewhite,
                fontSize = fnt18, fontWeight = txtbold,
            )
        }
    },
    )
    }
}

@Composable
fun BottomBar(navController: NavController) {
    Row(
        modifier = Modifier
            .fillMaxWidth()
            .height(55.dp)
            .background(purple673),
        horizontalArrangement = horzspacear
    ) {
        val butcolor = ButtonDefaults.buttonColors(purple673)
        val size24 = Modifier.size(24.dp)
        Button(
            onClick = { navController.navigate("caloriemgt") },
            colors = butcolor,
        ) {
            Icon(
                imageVector = Icons.Outlined.RiceBowl,
                contentDescription = "Calorie",
                modifier = size24
            )
        }
        Button(
            onClick = { navController.navigate("article") },
            colors = butcolor,
        ) {
            Icon(

```

```

        imageVector = Icons.Filled.Article,
        contentDescription = "Article",
        modifier = size24
    )
}
Button(
    onClick = { navController.navigate("sleep") },
    colors = butcolor,
) {
    Icon(
        imageVector = Icons.Filled.Bedtime,
        contentDescription = "Doctor",
        modifier = size24
    )
    Button(
        onClick = { navController.navigate("maps") },
        colors = butcolor,
    ) {
        Icon(
            imageVector = Icons.Filled.LocationOn,
            contentDescription = "Map",
            modifier =
            size24
        )
    }
    Button(
        onClick = { navController.navigate("emergency") },
        colors = butcolor,
    ) {
        Icon(
            imageVector = Icons.Filled.Contacts,
            contentDescription = "Emergency",
            modifier = size24
        )
    }
}
}

```