Abstract

The Medicare app represents a groundbreaking approach to enhancing healthcare support for literate individuals. This innovative application empowers users to take proactive control of their health by providing comprehensive features and resources within a user-friendly interface. Through modules such as Calorie Manage Page, Sleep Tracker, and Emergency Dial users can efficiently manage their healthcare needs independently. By leveraging technology to streamline healthcare access and information, the Medicare app revolutionizes the way literate individuals engage with their health, fostering a proactive approach to wellness. This Application is used to keep users aware of their health status and maintain a Good Lifestyle in Day-to-Day Life. Users can get a clear overall idea about their Health Status, and take care of themselves without any dependency on others..

I. Introduction

In today's fast-paced world, where technological advancements are transforming every aspect of our lives, healthcare remains an essential domain that continues to undergo significant evolution. With the proliferation of digital technologies, the healthcare landscape is witnessing a paradigm shift towards more accessible, personalized, and efficient solutions. Recognizing the importance of harnessing the power of technology to address healthcare challenges, our project introduces the "Medicare App" – a revolutionary platform designed to provide enhanced healthcare support specifically tailored for literate individuals. Access to quality healthcare is a fundamental human right, yet it is often hindered by barriers such as geographical limitations, financial constraints, and lack of awareness. While numerous healthcare apps exist in the market, many fail to adequately address the unique needs of literate individuals who seek reliable information and support to manage their health effectively. The Medicare App seeks to fill this gap by offering a comprehensive and user-friendly platform that empowers individuals to take control of their health and well-being. At its core, the Medicare App aims to democratize healthcare by providing literate individuals with easy access to essential healthcare services and information. Through a range of innovative features and functionalities, the app caters to the diverse needs of users, whether it be managing their dietary intake, staying informed about health-related topics, tracking their sleep patterns, or locating nearby healthcare facilities.

1.1 OVERVIEW:

The "Medicare App for Enhanced Healthcare Support for Literate People" is a pioneering digital solution designed to address the specific healthcare needs of literate individuals. By leveraging technology and user-centric design principles, the app aims to empower users to proactively manage their health and well-being. Through curated content, interactive features, and streamlined access to healthcare services, the app seeks to enhance health literacy, promote engagement, and foster a sense of community and support among its users. With its focus on accessibility, convenience, and reliability, the Medicare App represents a significant step towards bridging the gap between users and healthcare services, ultimately contributing to improved health outcomes and overall well-being

1.2 OBJECTIVES OF THE PROJECT

The objectives of the "Medicare App for Enhanced Healthcare Support for Literate People" project are to develop a user-friendly platform that promotes health literacy and empowers users to make informed decisions about their well-being. This includes fostering user engagement through interactive features, streamlining healthcare management processes, and ensuring robust data security measures. Continuous improvement and innovation are central to the project's objectives, aiming to provide the most effective and up-to-date solutions for enhancing healthcare support for literate individuals.

1.3 SCOPE OF THE SYSTEM

In this project, we propose a narrative method that aims at finding significant features and functionalities aimed at providing comprehensive healthcare support to literate individuals. This includes user registration/login, dashboard interface, calorie management, health articles section, sleep tracker, healthcare facility map search, and emergency dial feature. The system facilitates easy access to essential healthcare services and information, promotes health literacy, and fosters user engagement and interaction. It also streamlines healthcare management processes and ensures robust data security measures. Additionally, the scope extends to continuous improvement and innovation, with a commitment to delivering the most effective and up-to-date solutions for enhancing healthcare support for literate people

II. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

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.

2.1.1 DRAWBACKS OF EXISTING SYSTEM

- lack of comprehensive features
- poor awareness of health impacts
- accessibility issues

2.2 PROPOSED SYSTEM:

The proposed system of this project we propose 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 Consultations and Medicine Orders. For now, we have done only 7 Modules including User Registration.

2.2.1 FEATURES OF PROPOSED SYSTEM:

- User Registration: Secure registration process allowing users to create accounts and access personalized features
- Medication Tracking: This feature enables users to track their medication intake, set reminders, and monitor adherence to treatment plans.
- Lifestyle Management Tools: Tools and resources for promoting healthy lifestyle choices, including fitness tracking, and nutrition planning.

III. SYSTEM SPECIFICATION

3.1 SOFTWARE SPECIFICATION

Operating System	: Windows 10
Front End	: Kotlin or Java
Back End	: Room DB, Google Firebase Realtime Database, Google Firebase Cloud Messaging .
Code	: Android Studio Hedgehog

3.2 HARDWARE SPECIFICATION

Hardware	:i3 processor
Speed	:2.4 GHz
RAM	: 8GB
Hard Disk	:800 GB

3.3 SOFTWARE SPECIFICATIONS

ANDROID STUDIO

The software specifications for the development of the "Health Medicare App" entail utilizing Android Studio Hedgehog as the primary development environment on the Android platform. Programming will be carried out primarily in Java or Kotlin, with consideration for platform-specific best practices and optimizations. For local storage, SQLite will be employed, while cloud-based storage and real-time data synchronization will be facilitated through Firebase. User interface design will adhere to Material Design guidelines, ensuring an intuitive and visually appealing experience. The app will leverage Android Jetpack components for architecture guidance and simplified development, alongside Retrofit for networking and Gson for JSON serialization/deserialization. Version control will be managed using Git, enabling collaborative development and version tracking. Testing will encompass JUnit for unit testing, Espresso for

UI testing, Firebase Test Lab for automated testing on real devices, and Robolectric for local unit testing. Continuous Integration/Continuous Deployment (CI/CD) pipelines will be established using services like GitHub Actions or Bitrise, facilitating automated build, test, and deployment processes. Accessibility features will be implemented to comply with Android accessibility guidelines, ensuring usability for individuals with disabilities. Security measures will include data encryption, secure authentication, and authorization mechanisms to protect user data and privacy. Performance optimization techniques such as background task optimization, image compression, and caching will be employed to ensure smooth performance and efficient resource utilization across a variety of Android devices.

Room Database: Room is an SQLite object mapping library provided by the Android Jetpack suite of libraries. It offers an abstraction layer over SQLite, making it easier to work with databases in Android applications. Room will be used for local storage of user data, providing efficient data access and management within the app.

Google Firebase Realtime Database: Firebase Realtime Database is a cloud-hosted NoSQL database provided by Google Firebase. It offers real-time synchronization and data persistence, allowing data to be synchronized across multiple clients in real time. Firebase Realtime Database will be employed for cloud-based storage and real-time synchronization of certain user data, ensuring seamless data access and synchronization across devices.

Google Firebase Cloud Messaging (FCM): Firebase Cloud Messaging is a cross-platform messaging solution provided by Google Firebase. It enables the app to send notifications and messages to users across various platforms, including Android, iOS, and web. FCM will be utilized to send push notifications to users, providing timely updates, reminders, and notifications about important events or information within the app.

IV. SYSTEM DESIGN AND DEVELOPMENT

4.1 MODULES DESCRIPTION:

MODULES

- User Sign-up and Login
- Dashboard Interface
- Calorie Management Page
- Health Articles
- Sleep Tracker
- Healthcare Facility Map Search
- Emergency Dial

User Sign-up and Login:

Through user registration and login, users can safely utilize the app's features and build personalized accounts. Users who register can get tailored material, track their progress, and save preferences. By encouraging frequent app usage, this module gives consumers a sense of control over how their health is managed. With personalized accounts, users may customize the app's recommendations and information according to their interests, health objectives, and demographics. This increases user engagement and helps them stick to their health-related routines.

Dashboard Interface

Users can get a summary of their actions, goals, and health metrics from the dashboard, which acts as the app's main center. It provides an aesthetically pleasing and user friendly interface for switching between the many app functionalities. By providing clear and straightforward health information, the dashboard makes it easier to understand and keep track of health status. By providing personalized recommendations, regular reminders, and goal tracking, users can encourage proactive behavior modification and health management.

Calorie Management Page

It enables users to keep tabs on their food intake, keep an eye on their calorie intake, and establish nutrition objectives. It offers resources for tracking meals, figuring out caloric intake, and getting access to nutrient data for different foods. This module enables users to make healthier food choices and reach their weight management objectives by raising awareness of dietary practices and calorie intake. Monitoring one's nutritional intake promotes self-accountability and self-monitoring, which enhance eating habits and improve general health results.

Health Articles

A carefully chosen selection of evidence-based articles, advice, and resources on a range of health-related subjects may be found in the health articles section. It seeks to enlighten and educate users on disease management, healthy lifestyle choices, and preventive care. Having access to trustworthy health information improves users' health literacy and empowers them to make wise decisions regarding their care. This module encourages users to grasp and be conscious of important health concepts, which enables them to take a more active role in managing their health and adopting better habits.

Sleep Tracker:

This module helps users keep tabs on their sleeping habits, measure the quantity and quality of their sleep, and pinpoint elements that might affect their sleep health. Although many people suffer from sleep-related problems, getting enough sleep is crucial for general health and well-being. This module has the potential to enhance sleep quality and lead to improved sleep outcomes by raising awareness of sleep behaviors and supporting behavior modification.

Healthcare Facility Map Search:

The map search function helps users find local medical facilities, such as clinics, hospitals, pharmacies, and urgent care centers. For prompt health problem diagnosis, treatment, and management, access to healthcare services is essential. This module fosters timely healthcare-seeking habits and increases healthcare accessibility by making it easier to obtain healthcare services.

Emergency Dial:

Users can quickly access emergency services, such as 108 or local emergency response lines, with this tool. Timely access to medical care can save lives in emergencies. This feature makes sure that consumers can get in touch with emergency services immediately, which can help with timely action and lessen the degree of unfavorable effects.

4.2. Block Diagram:

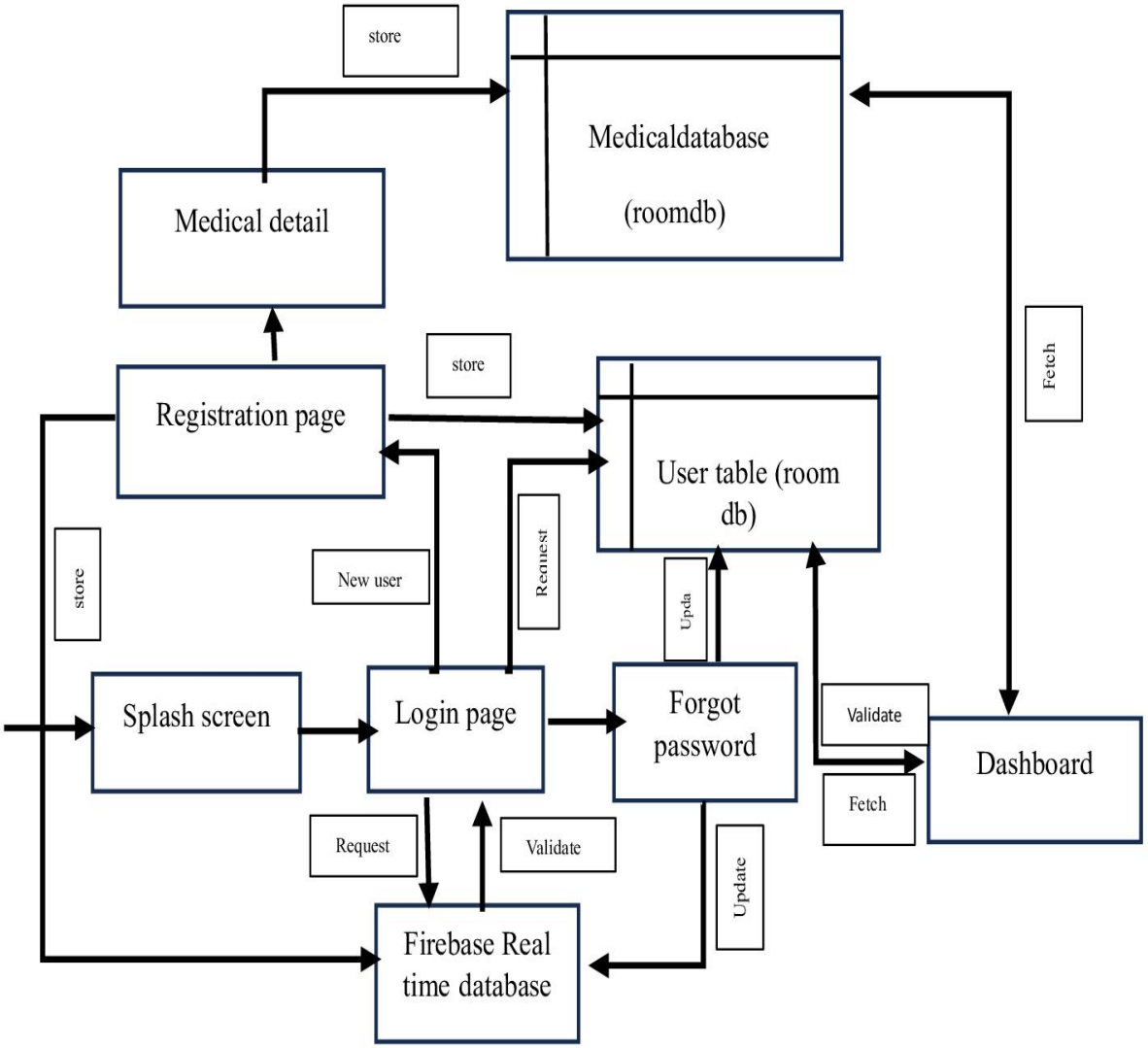

Fig 1.1 : Block diagram of Creation of Medicare App

4.1 Data Flow Diagram:

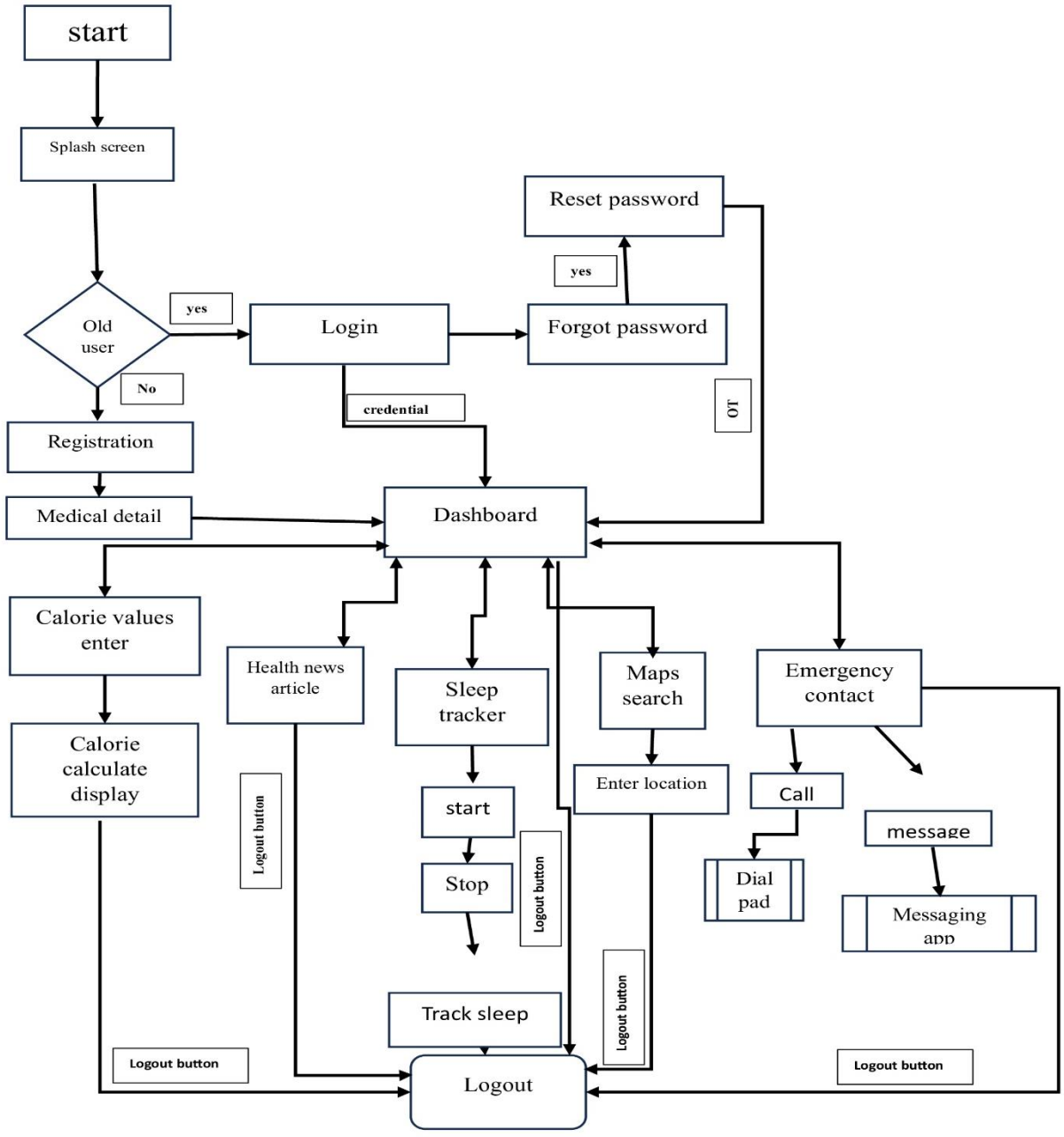

Fig 1.2 : Flow diagram of Medicare App for Healthcare support

4.2 SYSTEM DEVELOPMENT

The system development process for the Medicare App for Enhanced Healthcare Support for Literate People is a structured and iterative approach aimed at transforming project requirements into a fully functional application. It begins with thorough requirement gathering and analysis, where stakeholder needs and project objectives are identified.

Subsequently, the development team moves on to system design, defining the architecture, database schema, user interface, and APIs. Development activities encompass frontend and backend implementation using technologies such as Java/Kotlin, Firebase Realtime Database, and Spring Boot. Rigorous testing, including unit, integration, and user acceptance testing, is conducted to ensure the reliability and functionality of the application.

Deployment planning involves strategy formulation and CI/CD implementation for efficient rollout, while ongoing maintenance ensures continuous improvement and user satisfaction. This systematic approach ensures the successful development and deployment of a healthcare solution tailored to the needs of literate individuals, promoting better health outcomes and well-being.

4.3 DESIGN PROCESS

In the design process for the Medicare App for Enhanced Healthcare Support for Literate People, an in-depth understanding of user requirements guides the creation of detailed user personas, encompassing diverse demographics and needs. Information architecture is meticulously crafted to organize content and features in a logical hierarchy, ensuring seamless navigation and intuitive user flow. Wireframing then translates conceptual ideas into tangible screen layouts, mapping out the structure of each interface element. Visual design elevates the user experience by incorporating aesthetically pleasing color palettes, typography, and iconography while adhering to accessibility standards for inclusivity. Prototypes are developed to simulate user interactions and gather valuable feedback, facilitating iterative refinement of the design based on user insights. Through rigorous usability testing, the design undergoes scrutiny to ensure it effectively meets user needs and preferences, with continuous iterations made until optimal usability and user satisfaction are achieved. This iterative and user-centric approach ensures that the design of the Medicare App prioritizes usability, accessibility, and effectiveness in addressing the healthcare needs of literate individuals, ultimately enhancing their overall healthcare experience.

In addition to user personas, the design process involves conducting user research and usability studies to gain insights into user behaviors, preferences, and pain points. This qualitative data informs design decisions, ensuring that the app resonates with the target audience. Information architecture extends beyond the content organization to encompass task flows and user journeys, mapping out the sequence of interactions users will have with the app. Wireframes evolve into high-fidelity prototypes, offering a realistic representation of the app's functionality and visual design. User testing sessions provide valuable feedback on usability and user experience, allowing designers to iterate and refine the design iteratively. Collaborative design workshops and design sprints foster creativity and alignment among cross-functional teams, promoting innovation and problem-solving. By embracing a holistic approach to design, the Medicare App is not only functional and aesthetically pleasing but also deeply resonates with its users, ultimately enhancing their engagement and satisfaction with the healthcare support it provides

V. TESTING AND IMPLEMENTATION

5.1 SYSTEM TESTING

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

5.2 SYSTEM IMPLEMENTATION

System Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence in the new system for the user that it will work efficiently and effectively. The existing system was a long time process.

This process includes several steps, starting with finalizing the development of all components, including frontend interfaces, backend systems, and database integration. Once development is complete, the system is thoroughly tested to ensure its functionality, reliability, and security. Upon successful testing, the app is prepared for deployment, which may involve packaging the app for distribution through app stores or other distribution channels. Deployment includes activities such as configuring servers, setting up databases, and configuring networking components to ensure the app operates smoothly in the production environment. Post-deployment, the system is continuously monitored to detect and address any issues that may arise, ensuring that the Medicare App provides uninterrupted healthcare.

VI. RESULT

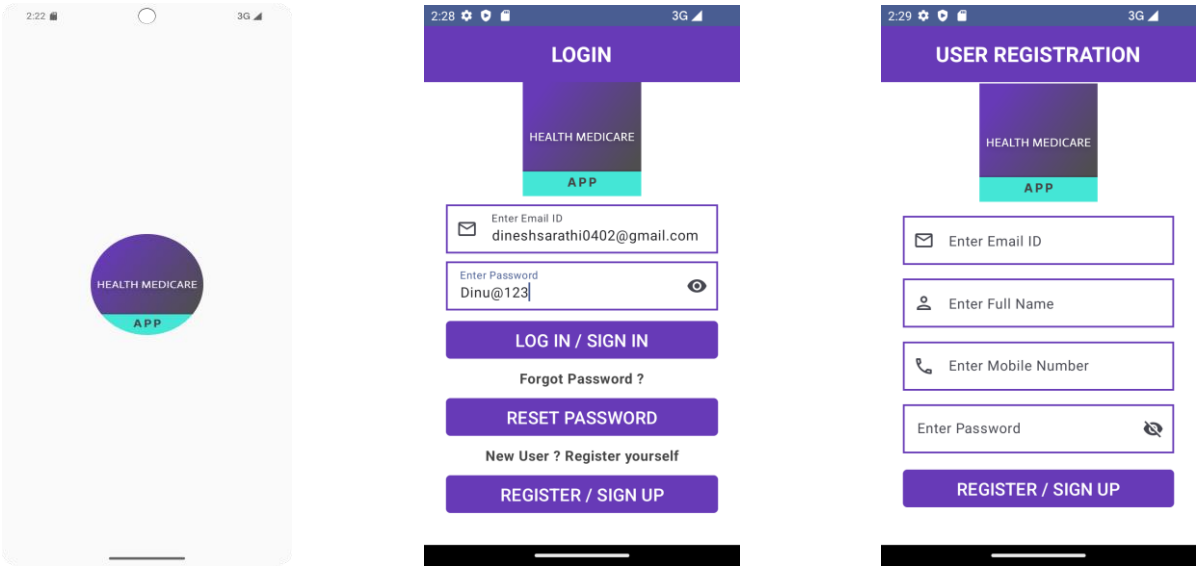

Fig 3 : App logo, Login screen, User Registration screen

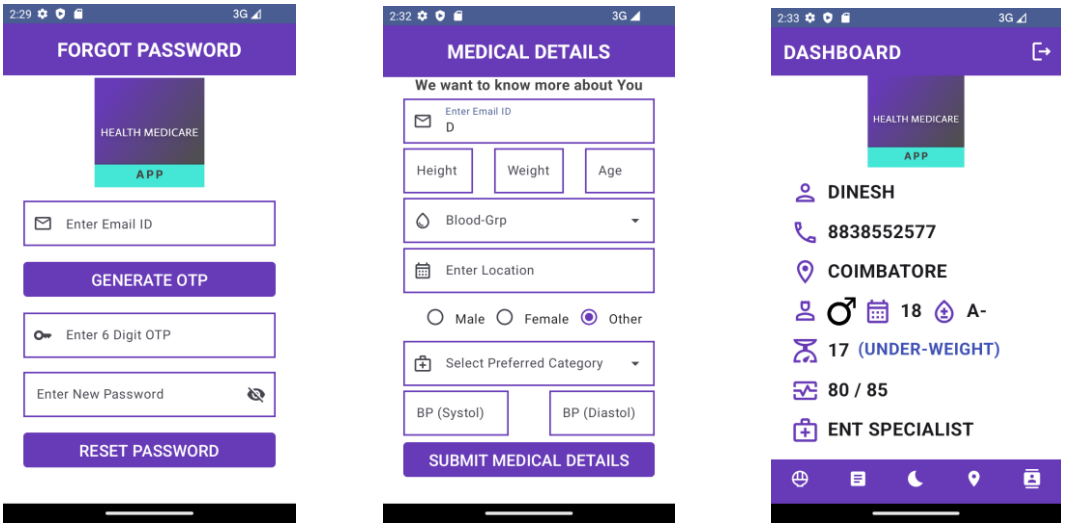

Fig 4 : Forgot Password screen , Medical Details screen , Dashboard screen

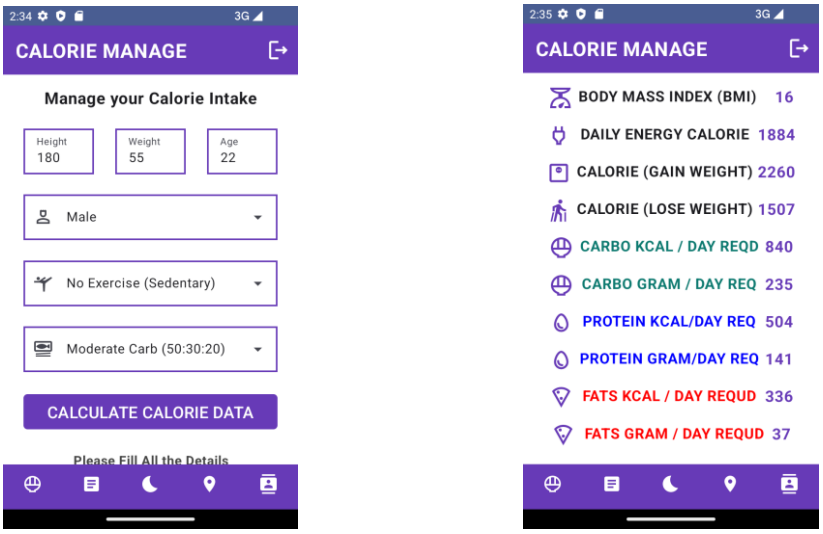

Fig 5: Calorie Manage screen

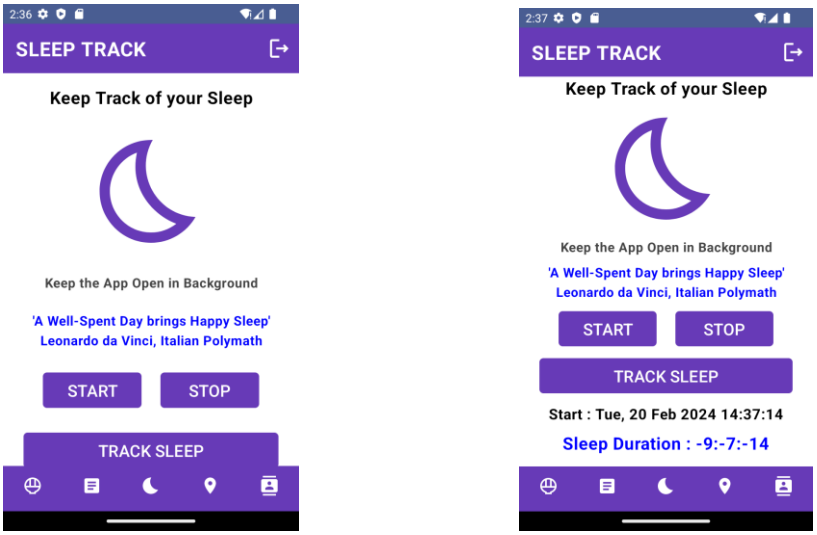

Fig 6: Sleep Tracker screen

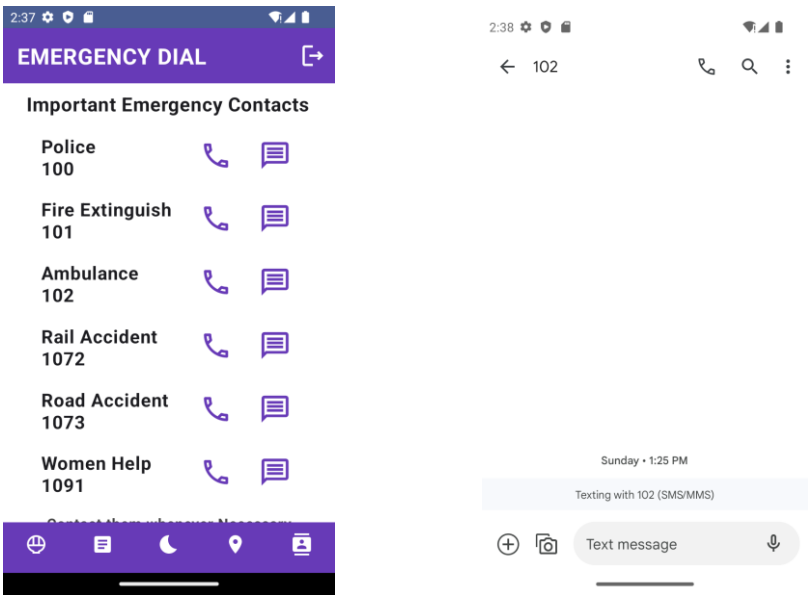

Fig 7: Emergency Dial screen

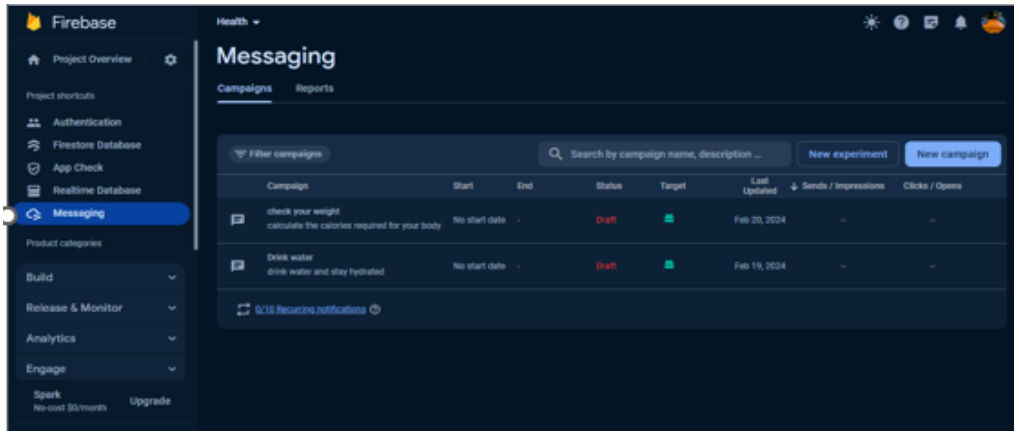

Fig 8 : Firebase Cloud Messaging

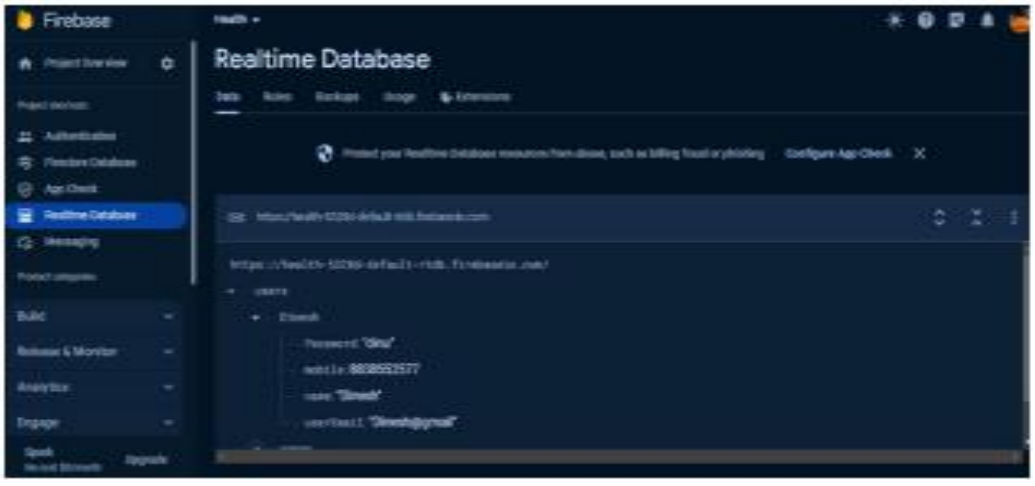

Fig 9: Firebase RealTime Database

VII. CONCLUSION

the development of the Medicare App for Enhanced Healthcare Support represents a significant step towards addressing the healthcare needs of individuals in our society. By providing a comprehensive and user-friendly platform that integrates various healthcare services and resources, the app aims to empower users to take proactive steps toward managing their health and well-being effectively. Through thorough system development, design, testing, and implementation processes, the app has been meticulously crafted to meet the specific needs and preferences of its users. With features such as user registration, calorie management, health articles, sleep tracking, healthcare facility map search, and emergency dial feature, the app offers a holistic approach to healthcare management and support. Moving forward, continuous monitoring, maintenance, and user feedback will be crucial in ensuring the app's ongoing success and effectiveness in enhancing healthcare support for literate individuals. Ultimately, the Medicare App represents a promising solution for promoting better health outcomes and improving the overall quality of life for its users.

VIII. SCOPE FOR FUTURE ENHANCEMENT

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

BIBLIOGRAPHY

[1] Smith, A. (2021). The Role of Mobile Health Applications in Improving Healthcare Accessibility and Outcomes: A Literature Review. Journal of Mobile Health, 5(2), 45-63.

[2] Medicare.gov. (n.d.). What's Medicare? Retrieved from https://www.medicare.gov/what-medicare-covers .

[3] Bandura, A. (1977). Social Learning Theory. Prentice-Hall.

[4] Rosenstock, I. M. (1966). Why People Use Health Services. The Milbank Memorial Fund Quarterly, 44(3), 94-127.

[5] Prochaska, J. O., & DiClemente, C. C. (1983). Stages and Processes of Self-Change of Smoking: Toward an Integrative Model of Change. Journal of Consulting and Clinical Psychology, 51(3), 390-395.

[6] Patel, M. S., Asch, D. A., & Volpp, K. G. (2015). Wearable Devices as Facilitators, Not Drivers, of Health Behavior Change. JAMA, 313(5), 459-460.

[7] Centers for Medicare & Medicaid Services. (2020). Medicare & You: 2021. Retrieved from https://www.medicare.gov/pubs/pdf/10050- Medicare-and-You.pdf .

[8] World Health Organization. (2019). Promoting Health Literacy to Encourage Prevention and Healthy Behaviours. Retrieved from https://www.who.int/health-topics/health-literacy#tab=tab_1 .

[9] Lee, J. A., & Kim, J. (2017). Healthcare Applications for Smart Watches: A Review. Healthcare Informatics Research, 23(1), 1-8.

[10] Tricco, A. C., et al. (2016). Mobile Health Apps to Facilitate Self Care: A Systematic Review of Effectiveness. Journal of Medical Internet Research, 18(5), e97.

APPENDICES

(A) SOURCE CODE

```
package com.example.health_medicare_application

import android.Manifest.permission.POST_NOTIFICATIONS
import android.content.Context
import android.os.Build
import android.os.Bundle
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.rememberLauncherForActivityResult
import androidx.activity.compose.setContent
import androidx.activity.result.contract.ActivityResultContracts
import androidx.annotation.RequiresApi
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Article
import androidx.compose.material.icons.filled.Bedtime
import androidx.compose.material.icons.filled.Contacts
import androidx.compose.material.icons.filled.LocationOn
import androidx.compose.material.icons.outlined.Logout
import androidx.compose.material.icons.outlined.RiceBowl
import androidx.compose.material3.AlertDialog
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.CenterAlignedTopAppBar
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
import androidx.compose.material3.IconButton
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TopAppBarDefaults
import androidx.compose.runtime.Composable
import androidx.compose.runtime.SideEffect
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.unit.dp
import androidx.navigation.NavController
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import com.example.health_medicare_application.model.MedicalDatabaseHelper
import com.example.health_medicare_application.model.UserDatabaseHelper
import
com.example.health_medicare_application.ui.theme.Health_MediCare_ApplicationT
heme
import com.example.health_medicare_application.ui.theme.fnt18
```

```
import com.example.health_medicare_application.ui.theme.fnt24
import com.example.health_medicare_application.ui.theme.horzspacear
import com.example.health_medicare_application.ui.theme.icon
import com.example.health_medicare_application.ui.theme.purewhite
import com.example.health_medicare_application.ui.theme.purple673
import com.example.health_medicare_application.ui.theme.rcshape
import com.example.health_medicare_application.ui.theme.txtbold
import com.example.health_medicare_application.uiactivity.CaloriePage
import com.example.health_medicare_application.uiactivity.DashboardPage
import
com.example.health_medicare_application.uiactivity.EmergencyContactPage
import com.example.health_medicare_application.uiactivity.ForgotPasswordPage
import com.example.health_medicare_application.uiactivity.HealthArticlePage
import com.example.health_medicare_application.uiactivity.LoginPage
import com.example.health_medicare_application.uiactivity.MapPage
import com.example.health_medicare_application.uiactivity.MedicalRegPage
import com.example.health_medicare_application.uiactivity.RegistrationPage
import com.example.health_medicare_application.uiactivity.SleepTrackerPage
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase

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)
        setContent {
            Health_MediCare_ApplicationTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    val firebasedb = FirebaseDatabase.getInstance();
                    val reference = firebasedb.getReference("User");
                    val 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 ->
            val 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(
                        onClick = {
                            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
            )
        }
```