Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Programme: MCA

Semester: Winter 24-25

Course: Mobile Application

Course Code: PMCA603L

Faculty: Dr. Christy Jackson J

Slot: C1 + TC1

SMART PROTECTOR APPLICATION FOR ELDERLY PEOPLE

Team Members

24MCA1031 – Gokulnath E

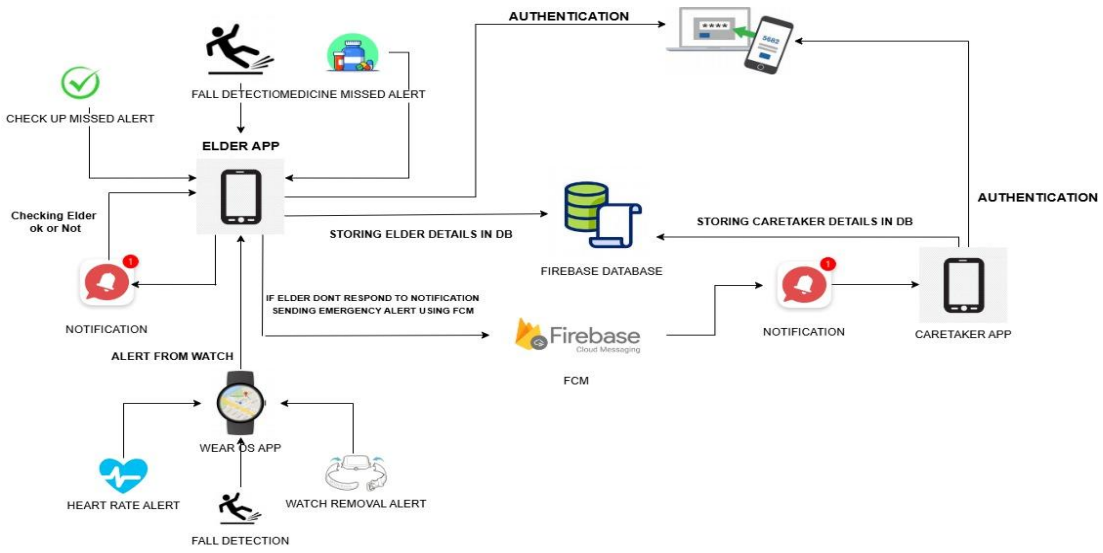
24MCA1003 – Sai Darshan B

24MCA1069 – Hari harra raju S

Abstract

In the last few years, demand for robust elderly care systems has seen a rise considerably with increasing population of seniors and a desire for real-time monitoring systems. The following is a mobile and wearable-based intelligent elderly safety and alerting system intended to support the welfare of older people particularly during key or emergency moments. The system suggested is built based on Kotlin for backend functionality and XML for the design of user interfaces, and Firebase is being used for cloud-based real-time database operations as well as for push notifications using Firebase Cloud Messaging (FCM). Shared Preferences is also used to cache local data. The system features two main user types: Caretakers and Elders. Both users register through OTP-based phone number verification. Once pairings are successfully made, Caretakers may remotely set medicine and check-up reminders for the paired Elders. The application facilitates timely drug adherence by causing reminders on the Elder's device. In the absence of acknowledgement within 30 seconds, a high-priority alert is sent to the Caretaker along with the Elder's last seen location, ID, and type of alert. To further improve safety, a Fall Detection module runs continuously in the background, analyzing data from the accelerometer, gyroscope, and magnetometer sensors. When it detects a potential fall, the system checks the Elder's state and raises an alert to the Caretaker in the event of non-responsiveness. Additionally, a Wear OS companion app is embedded using the Data Layer API, allowing for seamless communication between the Elder's smartwatch and phone. The wearable monitors physiological signals continuously like heart rate, device removal, and fall events. The system proposed combines mobile and wearable technologies in an effective manner to provide a strong, real-time elder care solution with timely intervention and ongoing monitoring. The system makes a valuable contribution to remote healthcare support and improves the quality of life and safety of elderly people.

Architecture Diagram



Sample Source Code

```
package com.example.spa.caretakers

import android.net.Uri
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.bumptech.glide.Glide
import com.example.spa.R
import com.example.spa.databinding.ActivityCaretakerAccountBinding
import com.example.spa.utils.helpers.Constants
import com.example.spa.utils.helpers.EncoderDecoder
import com.example.spa.utils.helpers.PreferenceManager
import com.example.spa.utils.helpers.Validation
import com.google.firebase.firestore.FirebaseFirestore

class CaretakerAccountActivity : AppCompatActivity() {
    private lateinit var img:String
    private lateinit var account : ActivityCaretakerAccountBinding
    private lateinit var preferenceManager: PreferenceManager
    private lateinit var db:FirebaseFirestore

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        account = ActivityCaretakerAccountBinding.inflate(layoutInflater)
        setContentView(account.root)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
        init()
        listeners()
    }

    private fun listeners(){
        account.caretakerUpdate.setOnClickListener {
            //check details
            val name = account.caretakerNameAcc.editText?.text.toString()
            val phone = account.caretakerAccCcp.fullNumberWithPlus.toString()

            if(checkValid(name)){
```

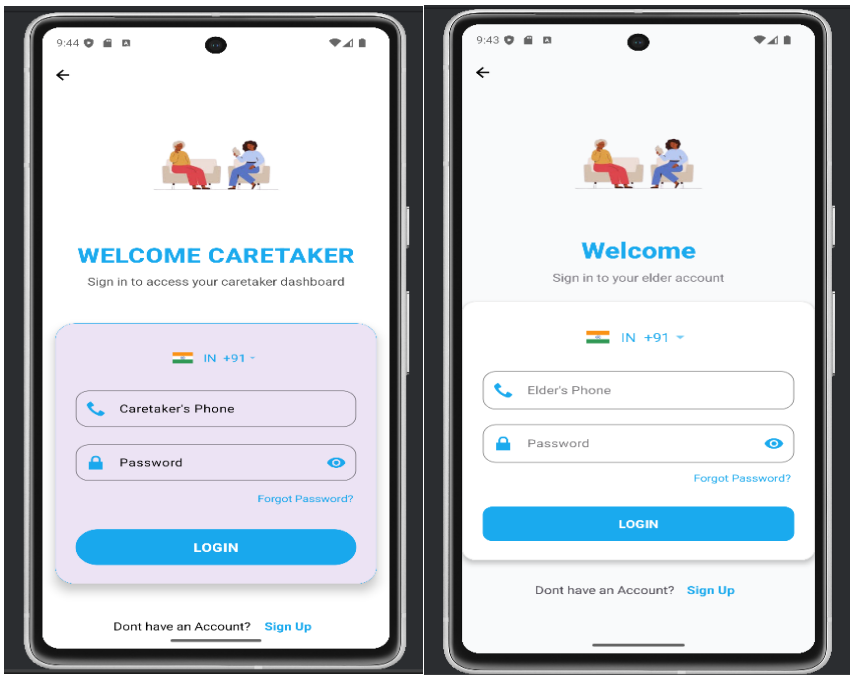
```
            if(name!= preferenceManager.getString(Constants.KEY_CARETAKER_NAME) || phone!=
preferenceManager.getString(
                    Constants.KEY_CARETAKER_PHONE) || img!= preferenceManager.getString(
                    Constants.KEY_CARETAKER_PROFILE)){
                val id = preferenceManager.getString(Constants.KEY_CARETAKER_ID)
                id?.let {
                    val doc = db.collection(Constants.KEY_CARETAKER_COLLECTION).document(id)
                    val map = mapOf(
                        Constants.KEY_CARETAKER_NAME to name,
                        Constants.KEY_CARETAKER_PHONE to phone,
                        Constants.KEY_CARETAKER_PROFILE to img
                    )

                    doc.update(map)
                        .addOnSuccessListener {
                            preferenceManager.putString(Constants.KEY_CARETAKER_NAME,name)
                            preferenceManager.putString(Constants.KEY_CARETAKER_PHONE,phone)
                            preferenceManager.putString(Constants.KEY_CARETAKER_PROFILE,img)

                            finish()
                        }
                        .addOnFailureListener {
                            Toast.makeText(this@CaretakerAccountActivity,"Issue
:${it.localizedMessage}",Toast.LENGTH_SHORT).show()
                        }
                }?:Toast.makeText(this@CaretakerAccountActivity,"ID null",Toast.LENGTH_SHORT).show()
            }else{
                Toast.makeText(this@CaretakerAccountActivity,"Update details
before!",Toast.LENGTH_SHORT).show()
            }
        }
    }

    account.editCaretakerImg.setOnClickListener {
        launcher.launch("image/*")
    }

    account.caretakerLoginBackBtn.setOnClickListener {
        finish()
    }
}

private val launcher = registerForActivityResult(
    ActivityResultContracts.GetContent()
){uri:Uri?->
    if(uri!=null){
        Glide.with(this@CaretakerAccountActivity)
            .load(uri)
            .error(R.drawable.error)
```

```
            .placeholder(R.drawable.place_holder)
            .into(account.caretakerAccImg)

        img = EncoderDecoder.encodeImage(uri,this@CaretakerAccountActivity)
      }else{
        Toast.makeText(this@CaretakerAccountActivity,"Null ",Toast.LENGTH_SHORT).show()
      }
    }

    private fun init(){
      account.caretakerAccCcp.registerCarrierNumberEditText(account.caretakerPhnAcc.editText)
      preferenceManager = PreferenceManager(this@CaretakerAccountActivity)
      db = FirebaseFirestore.getInstance()
      setDetails()
    }

    private fun setDetails(){
      loading()
      account.caretakerAccImg.setImageBitmap(
        EncoderDecoder.decodeImage(
          preferenceManager.getString(Constants.KEY_CARETAKER_PROFILE)!!
        )
      )


account.caretakerNameAcc.editText!!.setText(preferenceManager.getString(Constants.KEY_CARETAK
ER_NAME))

account.caretakerPhnAcc.editText!!.setText(preferenceManager.getString(Constants.KEY_CARETAKER
_PHONE)!!.substring(3))
      account.caretakerAccCcp.setCountryForPhoneCode(
        preferenceManager.getString(Constants.KEY_CARETAKER_PHONE)!!.substring(1,4).toInt()
      )
      unloading()
    }

    private fun loading(){
      account.caretakerAccProgress.visibility = View.VISIBLE
      account.caretakerUpdate.visibility = View.GONE
    }

    private fun unloading(){
      account.caretakerUpdate.visibility = View.VISIBLE
      account.caretakerAccProgress.visibility = View.GONE
    }

    private fun checkValid(name:String):Boolean{
      if(!Validation.isNameValid(name)){
        account.caretakerNameAcc.error = "Invalid name"
```

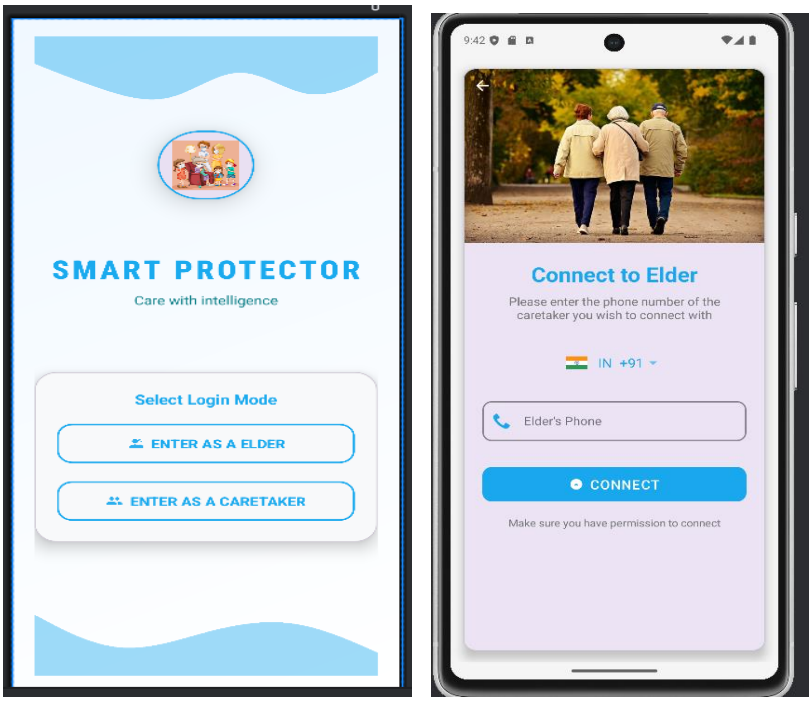
```
            return false
        }
        else{
            account.caretakerNameAcc.error = null
        }

        if (!account.caretakerAccCcp.isValidFullNumber){
            account.caretakerPhnAcc.error = "Invalid phone"
            return false;
        }
        else{
            account.caretakerPhnAcc.error = null
        }

        return true;
    }
}

package com.example.spa.home

import android.annotation.SuppressLint
import android.app.ActivityOptions
import android.content.Intent
import android.os.Bundle
import android.view.animation.Animation
import android.view.animation.AnimationUtils
import android.widget.LinearLayout
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import androidx.lifecycle.lifecycleScope
import com.example.spa.R
import com.example.spa.caretakers.CaretakerHomeActivity
import com.example.spa.elders.ElderHomeActivity
import com.example.spa.utils.helpers.Constants
import com.example.spa.utils.helpers.PreferenceManager
import kotlinx.coroutines.*

@SuppressLint("CustomSplashScreen")
class SplashActivity : AppCompatActivity() {
    private lateinit var main:LinearLayout
    private lateinit var fadeOut:Animation
    private lateinit var preferenceManager: PreferenceManager
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_splash)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
```

```
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        main = findViewById(R.id.main)
        preferenceManager = PreferenceManager(this@SplashActivity)
        fadeOut = AnimationUtils.loadAnimation(this, R.anim.fade_out)
        lifecycleScope.launch {
            delay(1000)
            startAnim()
        }
    }

    private fun startAnim(){
        main.startAnimation(fadeOut)
        fadeOut.setAnimationListener(object : Animation.AnimationListener {
            override fun onAnimationStart(animation: Animation?) {

            }

            override fun onAnimationEnd(animation: Animation?) {
                if(preferenceManager.getBoolean(Constants.KEY_IS_ELDER_SIGNED_IN)){
                    val intent = Intent(this@SplashActivity,ElderHomeActivity::class.java)
                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_NEW_TASK)
                    startActivity(intent)
                    finish()
                }else if(preferenceManager.getBoolean(Constants.KEY_IS_CARETAKER_SIGNED_IN)){
                    val intent = Intent(this@SplashActivity,CaretakerHomeActivity::class.java)
                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_NEW_TASK)
                    startActivity(intent)
                    finish()
                }else{
                    val intent = Intent(this@SplashActivity, EntryActivity::class.java)
                    val options = ActivityOptions.makeCustomAnimation(
                        this@SplashActivity,
                        android.R.anim.fade_in,
                        android.R.anim.fade_out
                    )
                    startActivity(intent, options.toBundle())
                    finish()
                }
            }

            override fun onAnimationRepeat(animation: Animation?) {
            }
        })
    }
}
```

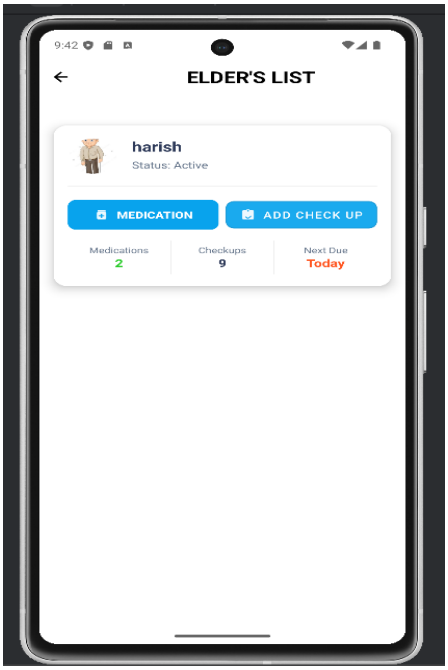
Output Screenshots



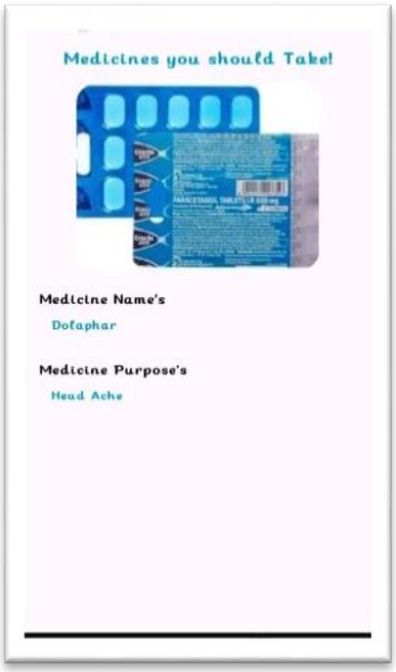
Login Page of App for Both caretaker and elder



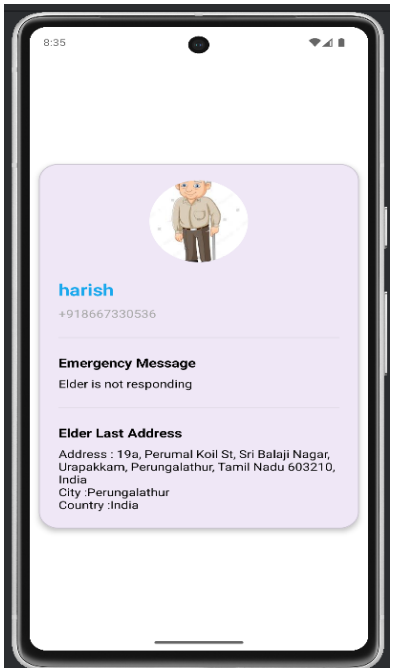
Entry Screen and Connect to Elder for Caretaker screen



Caretaker Dashboard for setting reminders to elder

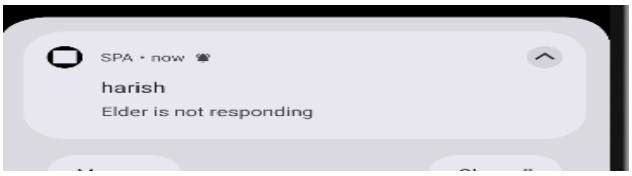


Medicine reminder for elder



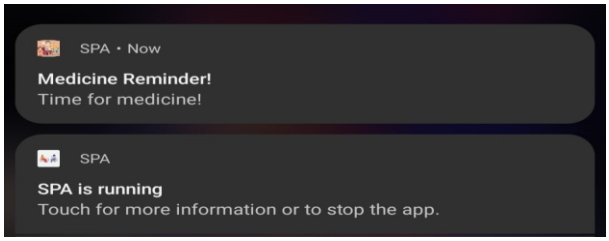
Emergency details of elder shown to caretaker screen

NOTIFICATIONS SNAPSHOT



Emergency notification for caretaker

Fall detection notification check up call for elder



Medicine reminder notification for elder