CMP309 - SOFTWARE DEVELOPMENT FOR MOBILE DEVICES

FOOD EXPIRY DATE TRACKER

Michael Awoyemi 2103939

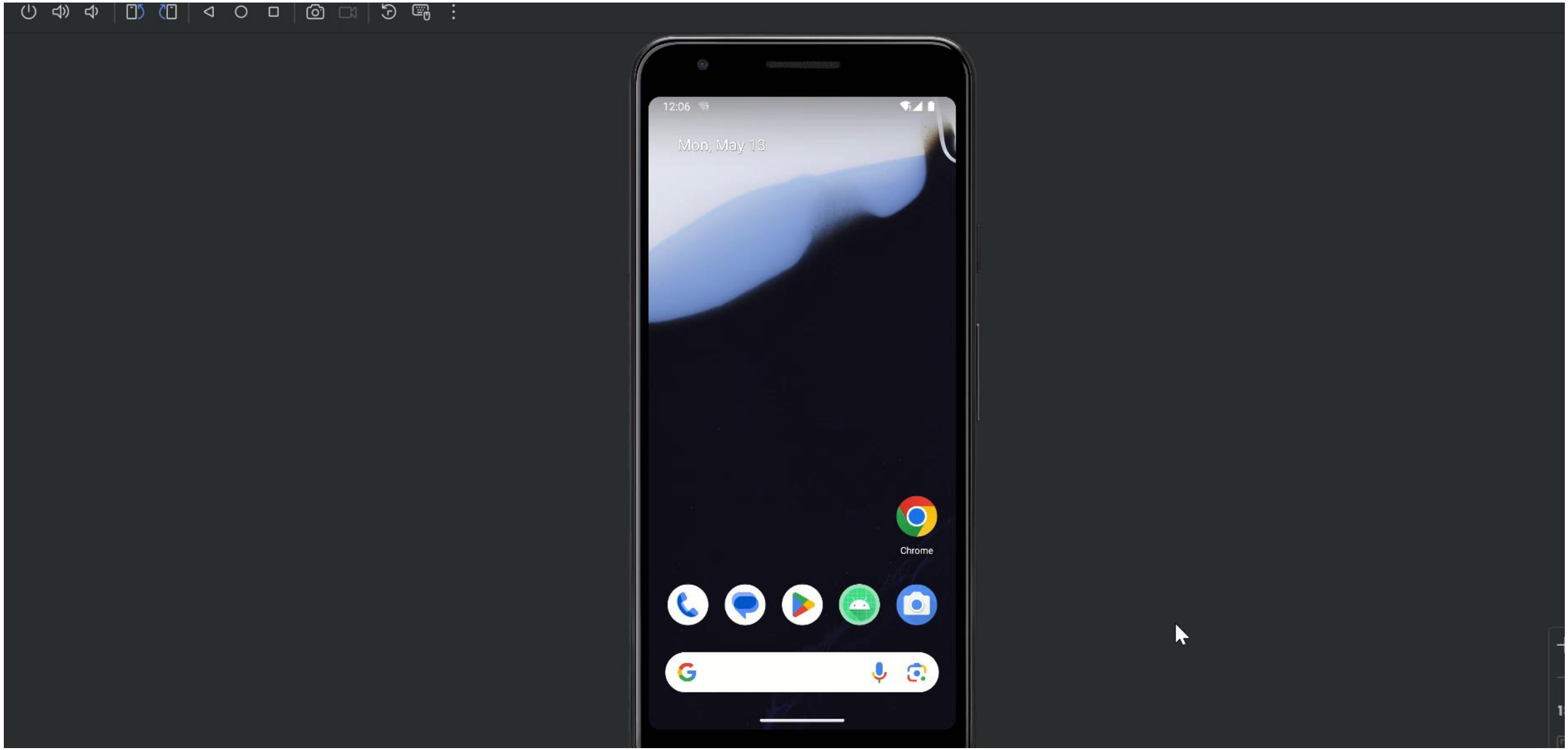
CONTENT

BACKGROUND

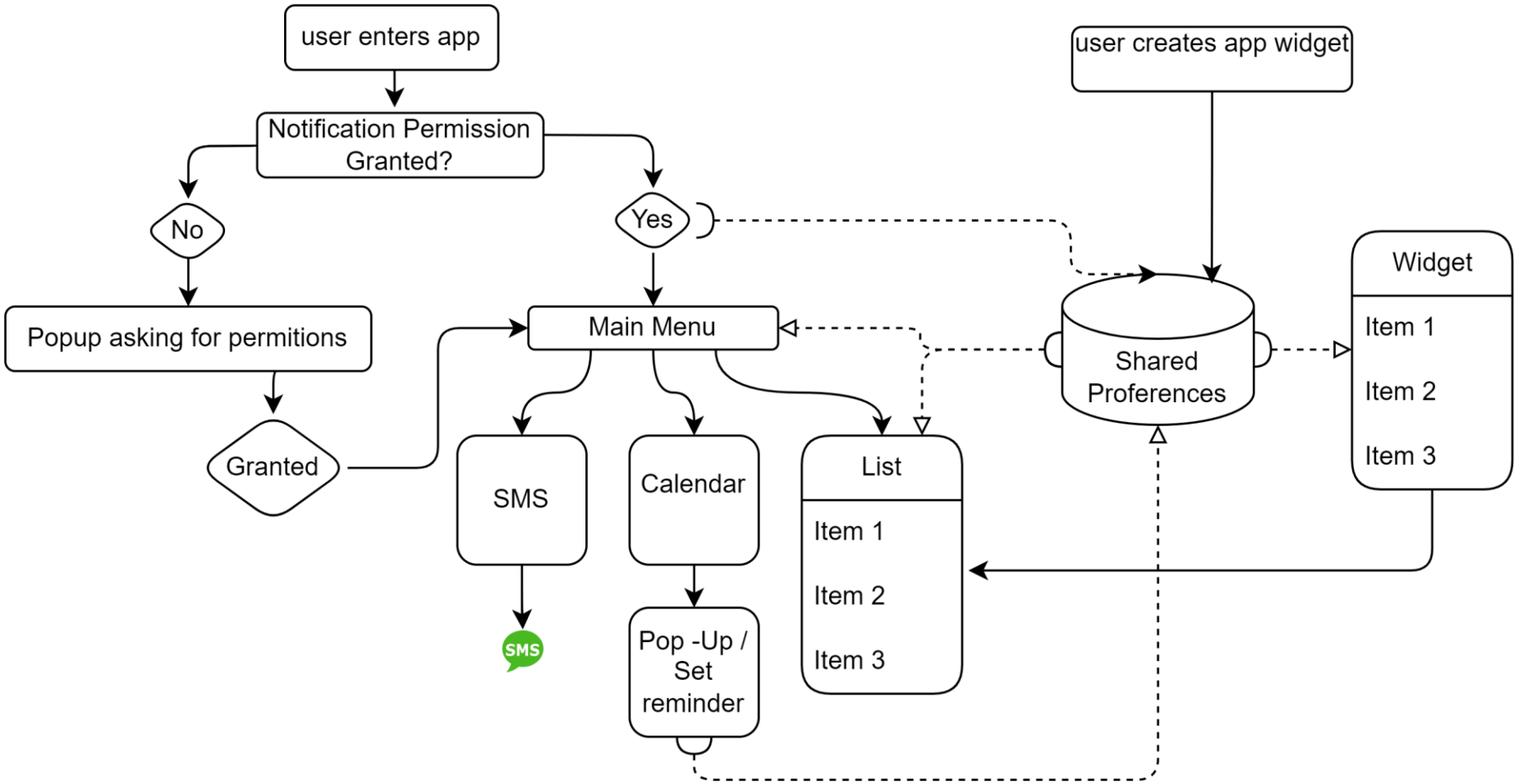
Food tracker

Schedule notification to remind users of food expiry date

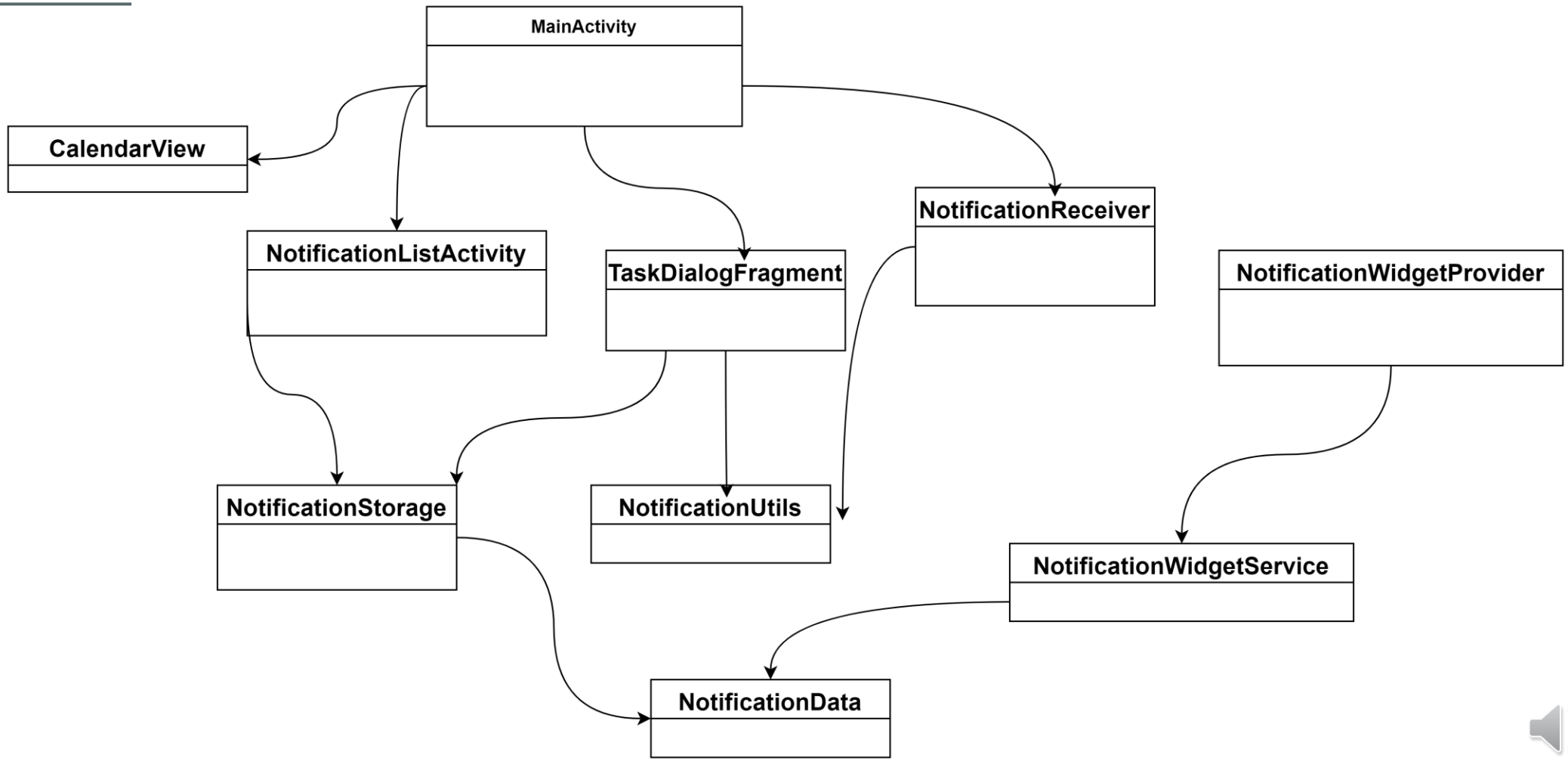
Pop-up dialogs, notifications, SMS, shared preferences, scheduling reminders.


12:06
Mon, May 13
Chrome

FLOW CHART DIAGRAM



STRUCTURE


MainActivity
CalendarView
NotificationListActivity
TaskDialogFragment
NotificationReceiver
NotificationWidgetProvider
NotificationStorage
NotificationUtils
NotificationWidgetService
NotificationData

KEY FEATURES

- Pop-up dialogs
- Notification Permissions
- Scheduling Notification
- Saving notifications (shared preferences)
- SMS
- Widget

NOTIFICATION PERMISSIONS

```
import android.app.NotificationManager;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
 <uses-permission
android:name="android.permission.POST_
NOTIFICATIONS" />
```

```
private val notificationPermissionInitializer =
   registerForActivityResult(ActivityResultContracts.RequestPermission()) { isGranted ->
      hasNotificationPermissionGranted = isGranted
      if (!isGranted) {
         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if (Build.VERSION.SDK_INT >= 33) {
               if
(shouldShowRequestPermissionRationale(android.Manifest.permission.POST_NOTIFICATIO
NS)) {
                  showNotificationPermissionRationale()
               } else {
                  showSettingDialog()
               }
            }
         }
      } else {
         Toast.makeText(
            applicationContext,
            "notification permission granted",
            Toast.LENGTH_SHORT
         )
            .show()
      }
```

SCHEDULING NOTIFICATIONS

```
import android.app.AlarmManager;
import android.content.Context;
import android.content.Intent;
import android.app.PendingIntent;
import android.os.Build;
```

```
fun scheduleNotification(context: Context, notificationData: NotificationData) {
    // Create an intent for the NotificationReceiver
    val intent = Intent(context, NotificationReceiver::class.java)

    // Add title and message as extras to the intent
    intent.putExtra("notificationTitle", notificationData.title)
    intent.putExtra("notificationContent", notificationData.content)

    val pendingIntent = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        PendingIntent.getBroadcast(
            context,
            notificationData.id.toInt(),
            intent,
            PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
        )
    } else {
        PendingIntent.getBroadcast(
            context,
            notificationData.id.toInt(),
            intent,
            PendingIntent.FLAG_UPDATE_CURRENT
        )
    }

    // Get the AlarmManager service
    val alarmManager = context.getSystemService(Context.ALARM_SERVICE) as AlarmManager

    // schedule the notification
    alarmManager.setExactAndAllowWhileIdle(
        AlarmManager.RTC_WAKEUP,
        notificationData.dateTime,
        pendingIntent
    )
}
```

SMS

```
<uses-permission
android:name="android.permission.SEND_SMS" />
import android.telephony.SmsManager
```

```
val smsButton = findViewById<Button>(R.id.smsbutton)
smsButton.setOnClickListener {
    val builder = AlertDialog.Builder(this)
    val inflater = layoutInflater
    val dialogLayout = inflater.inflate(R.layout.sms, null)
    val message = dialogLayout.findViewById<EditText>(R.id.smsedit)
    var number = " "
    val smsMessage = "Hi, I am using food tracker by Michael Awoyemi. Join me!!! "
    val smsManager: SmsManager
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        smsManager = this.getSystemService(SmsManager::class.java)
    } else {
        smsManager = SmsManager.getDefault()
    }
    with(builder) {
        setTitle("Phone number")
        setPositiveButton("<Send>") { dialog, which ->
            number = message.text.toString()
            try {
                smsManager.sendTextMessage(number, null, smsMessage, null, null)
                Toast.makeText(applicationContext, "SMS Message Sent!", Toast.LENGTH_LONG).show()
            } catch (e: Exception) {
                Toast.makeText(applicationContext, "SMS failed to send!!", Toast.LENGTH_LONG).show()
            }
        }
        setNegativeButton("cancel") { dialog, which ->
            Log.d("Main", "SMS Cacncelled")
        }
        setView(dialogLayout)
        show()
    }
```

WIDGET

```
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.Context;
import android.content.Intent;
import android.widget.RemoteViews;
```

```
class NotificationWidgetProvider : AppWidgetProvider() {

   override fun onUpdate(context: Context, appWidgetManager:
AppWidgetManager, appWidgetIds: IntArray) {
      for (appWidgetId in appWidgetIds) {
         val views = RemoteViews(context.packageName,
R.layout.notification_widget_provider)

         val intent = Intent(context, NotificationWidgetService::class.java)
         views.setRemoteAdapter(R.id.appwidget_list,              intent)

         appWidgetManager.updateAppWidget(appWidgetId, views)
      }
   }
}
```

SHARED PREFERENCES

```
import android.content.Context
import com.google.gson.Gson
import com.google.gson.reflect.TypeToken
```

```
object NotificationStorage {
    // Constants for shared preferences
    private const val PREFERENCES_NAME = "notification_preferences"
    private const val KEY_NOTIFICATIONS = "notifications"
    fun saveNotification(context: Context, notificationData: NotificationData) {
        // Get SharedPreferences instance
        val sharedPreferences = context.getSharedPreferences(PREFERENCES_NAME, Context.MODE_PRIVATE)
        val editor = sharedPreferences.edit()
        val gson = Gson()
        // Retrieve existing notifications from SharedPreferences
        val notifications = getNotifications(context).toMutableList()
        // Add new notification to the list
        notifications.add(notificationData)
        // Convert notifications list to JSON
        val json = gson.toJson(notifications)
        // Save JSON string to SharedPreferences
        editor.putString(KEY_NOTIFICATIONS, json)
        editor.apply()
    }
    fun getNotifications(context: Context): List<NotificationData> {
        // Get SharedPreferences instance
        val sharedPreferences = context.getSharedPreferences(PREFERENCES_NAME, Context.MODE_PRIVATE)
        val gson = Gson()
        // Retrieve JSON string of notifications from SharedPreferences
        val json = sharedPreferences.getString(KEY_NOTIFICATIONS, null)
        val type = object : TypeToken<List<NotificationData>>() {}.type
        return gson.fromJson(json, type) ?: emptyList()
    }
}
```

OVERVIEW

Simple food expiry date reminder

Runs smoothly without interruptions

Compatibility with API 29 and above (75%)

Appropriate permissions request for specific API

User Interface is simple and understandable

FUTURE WORK

Back up reminder list

Edit reminder or delete in case of user mistake

REFERENCES

Android Developers. (2024). Build a notification. Retrieved from https://developer.android.com/develop/ui/views/notifications/build-notification?hl=es-419#kts (accessed May 8, 2024).

Android Developers. (2024). Request app permissions. Retrieved from https://developer.android.com/training/permissions/requesting?hl=es-419 (accessed May 12, 2024).

Stack Overflow. (2018). Ask permission for push notification. Retrieved from https://stackoverflow.com/questions/44305206/ask-permission-for-push-notification (accessed May 9, 2024).

Android Developers. (2024). Notification permission. Retrieved from https://developer.android.com/develop/ui/views/notifications/notification-permission?hl=es-419 (accessed May 9, 2024).

Android Developers. (2024). Constraint Layout. Retrieved from https://developer.android.com/develop/ui/views/layout/constraint-layout?hl=es-419 (accessed May 9, 2024).

YouTube. (2023). Notification Permission in Android 13 | Runtime Notification Permission | POST_NOTIFICATIONS [Video]. Aaviskar Infotech. Retrieved May 9, 2023, from https://www.youtube.com/watch?v=_UubmZ4qJII

Android Developers. (2024). Schedule exact alarms. Retrieved from https://developer.android.com/develop/ui/views/notifications/notification-permission?hl=es-419 (accessed May 110, 2024).

Stack Overflow. (2020). How to set an alarm to be scheduled at an exact time after all the newest restrictions. Retrieved from https://stackoverflow.com/questions/60079472/how-to-set-an-alarm-to-be-scheduled-at-an-exact-time-after-all-the-newest-restri (accessed May 11, 2024).