



CS364 การพัฒนาโปรแกรมประยุกต์สำหรับอุปกรณ์พกพา

Hydro Tracker

จัดทำโดย

นาย	กษิติศ	คงประพันธ์	6609650129
นาย	กฤตชญา	กลิ่นเดช	6609650160
นาย	จันทร์พงศ์	วิทย์อรุณธานี	6609650160
นาย	ศุภกฤต	ธรรมามังกร	6609650657

เสนอ

อาจารย์ ปกรณ์ ลีสุทธิพรชัย

Hydro Tracker

Hydro Tracker แอปพลิเคชันที่ช่วยให้ผู้ใช้ทราบปริมาณน้ำที่ตนเองควรดื่มในแต่ละวัน พร้อมฟังก์ชันแจ้งเตือนการดื่มน้ำที่ผู้ใช้สามารถตั้งค่าเวลาการแจ้งเตือนได้ด้วยตนเอง อีกทั้งยังมีการเก็บประวัติการดื่มน้ำรายวัน เพื่อให้ผู้ใช้สามารถติดตามการดื่มน้ำของตนเองอย่างต่อเนื่อง

Logo ของแอปพลิเคชัน



หน้าหลักของแอปพลิเคชัน

หน้าหลักจะแสดงปริมาณน้ำที่ผู้ใช้ควรดื่มในแต่ละวัน โดยผู้ใช้จำเป็นต้องกรอกข้อมูล เช่น น้ำหนัก และ ระดับกิจกรรม



- กล้องแสดงผล (กล้องใหญ่) : แสดงผลข้อมูลของปริมาณน้ำที่ผู้ใช้ควรดื่มในแต่ละวัน สามารถแสดงข้อมูลแบบเรียลไทม์ ถึงแม้ผู้ใช้จะยังไม่ได้กดปุ่ม “คำนวณปริมาณน้ำ”
- ช่องกรอกน้ำหนัก : ให้ผู้กรอกน้ำหนักของตนเองเป็นเลขจำนวนเต็มลงไป โดยใช้เป็นหน่วยกิโลกรัม
- ช่องกรอกกิจกรรม : ให้ผู้ใช้เลือกระดับกิจกรรมโดยจะมี Dialog ขึ้นมาเพื่อให้ผู้ใช้เลือกระดับกิจกรรม 4 ระดับ เช่น ไม่ออกกำลังกาย, ออกกำลังกายเล็กน้อย, ออกกำลังกายปกติ, และออกกำลังกายหนัก
- ปุ่มคำนวณปริมาณน้ำ : เมื่อกดปุ่มนี้ จะพาผู้ใช้ไปยังหน้าเป้าหมาย
- เมื่อนำทาง (ด้านล่าง) : มีเมนูที่จะพาผู้ใช้ไปยังหน้าต่างๆ เช่น หน้าหลัก, หน้าเป้าหมาย, และหน้าประวัติ

คุณสมบัติของแอปพลิเคชัน

1. หน้าหลัก : แสดงข้อมูลปริมาณน้ำที่ผู้ใช้ต้องดื่มให้ได้ในแต่ละวัน โดยใช้ข้อมูลน้ำหนักและระดับกิจกรรมในการคำนวณ
2. หน้าเป้าหมาย : แสดงข้อมูลปริมาณน้ำที่ผู้ใช้ดื่มในวันนี้ เปรียบเทียบกับ ปริมาณน้ำเป้าหมายที่ต้องดื่มและมี Progress Bar เพื่อให้ผู้ใช้สามารถเห็นได้ง่ายขึ้น สามารถเพิ่มปริมาณน้ำโดยการกดปุ่มปริมาณน้ำที่ต้องการเพิ่มหรือสามารถกรอกปริมาณน้ำที่ตนเองต้องการเพิ่มได้ สามารถตั้งค่าการแจ้งเตือนได้โดยเลือกเวลาการแจ้งเตือนให้แจ้งเตือนทุกๆนาที่
3. หน้าประวัติ : แสดงรายการการดื่มน้ำในแต่ละช่วงเวลา พร้อมทั้งปริมาณน้ำที่ดื่ม สามารถลบรายการการดื่มน้ำได้กรณีที่ผู้ใช้ใส่ข้อมูลผิด

การทำงานของแอปพลิเคชัน

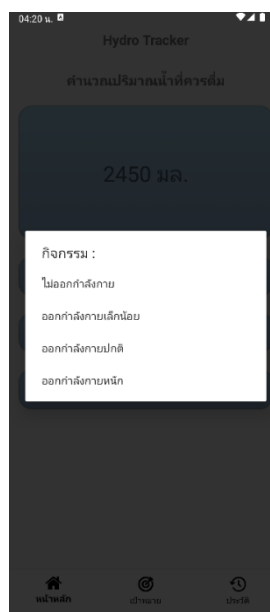
1. เริ่มต้นใช้งาน

ผู้ใช้เปิดแอปพลิเคชัน “Hydro Tracker”



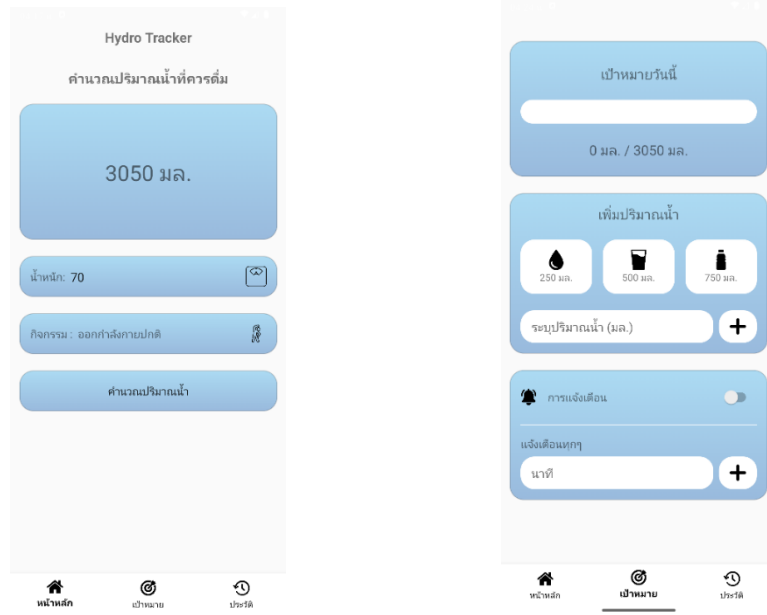
2. กรอกข้อมูลส่วนตัว และ แสดงผลปริมาณน้ำที่ควรบริโภค

ผู้ใช้งานกรอกน้ำหนักเพื่อกรอกน้ำหนักของตนเอง และ กดกล่องกิจกรรมโดยจะมี Dialog ขึ้นมาเพื่อให้เลือกระดับกิจกรรมของตนเอง จากนั้นระบบจะแสดงผลปริมาณน้ำที่ควรดื่ม



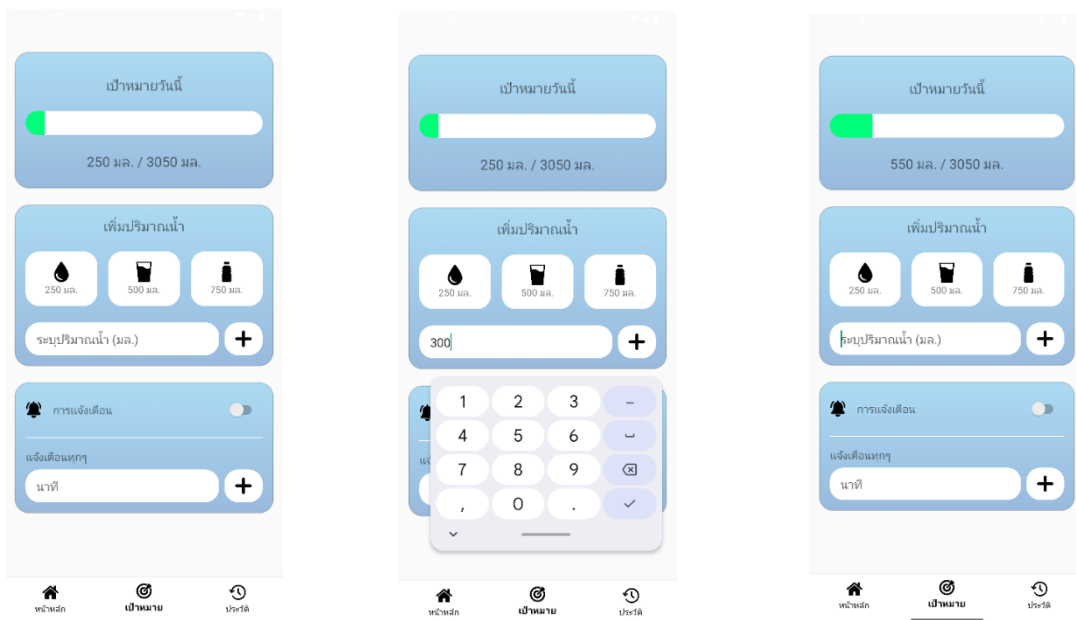
3. กดปุ่ม “คำนวณปริมาณน้ำ” และ ไปยัง “หน้าเป้าหมาย”

เมื่อกดปุ่มคำนวณปริมาณน้ำแล้วมายังหน้าเป้าหมาย ที่ป्ल่องเป้าหมายวันนี้จะแสดง Progress Bar และ ปริมาณน้ำที่ดื่มไปแล้ว เปรียบเทียบกับ ปริมาณน้ำที่ควรดื่ม



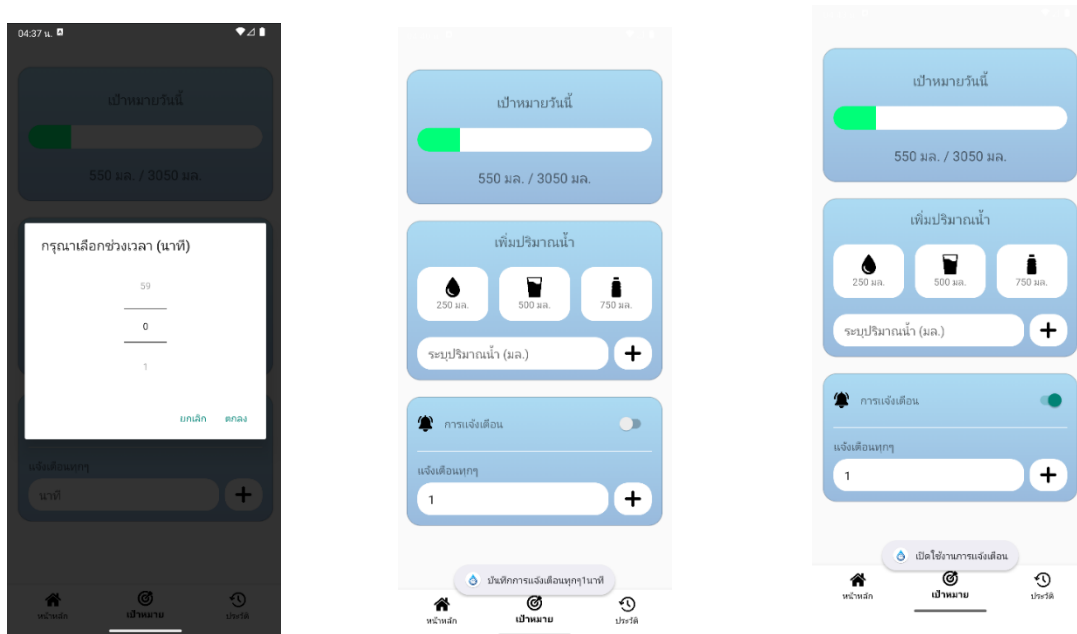
4. หน้าเป้าหมายที่ช่อง “เพิ่มปริมาณน้ำ”

ผู้ใช้งานสามารถเพิ่มปริมาณน้ำได้โดยการกดปุ่มไอคอนที่แอปพลิเคชันกำหนดไว้ให้ (250 มล. / 500 มล. / 750 มล.) หรือ กรอกปริมาณน้ำด้วยตนเองโดยกดปุ่มบวกเพื่อเพิ่มรายการ



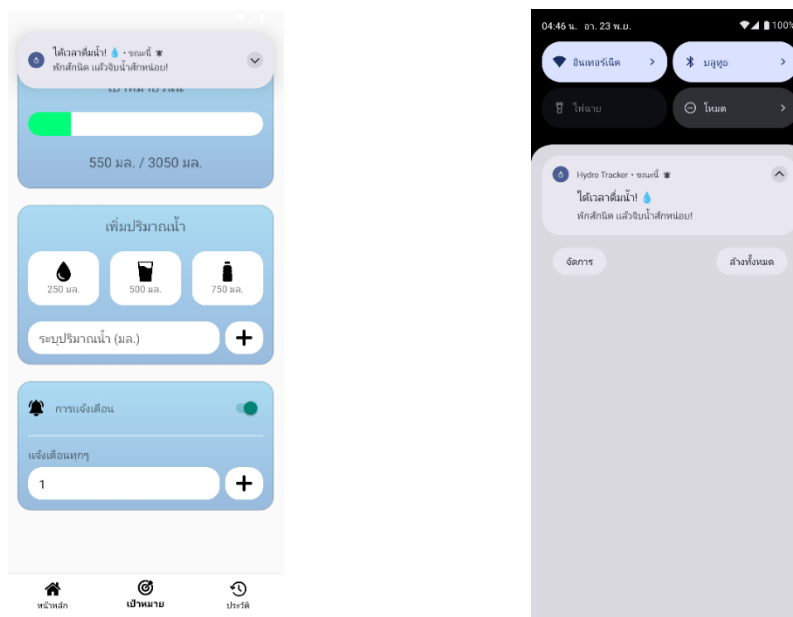
5. หน้าเป้าหมายที่ช่อง “การแจ้งเตือน”

ผู้ใช้งานสามารถเลือกเวลาที่ต้องการให้ระบบแจ้งเตือนการดื่มซ้ำได้ในรูปแบบนาฬิกา โดยเมื่อกดเข้าไปจะมี Dialog เวลาขึ้นมา ให้เลือกเวลาที่ต้องการแจ้งเตือนทุกๆ นาฬิกาแล้วกด “ตกลง” หลังจากนั้นกด “ปุ่มบวก” จะมี Toast แสดงว่า “บันทึกการแจ้งเตือนทุกๆ x นาฬิกา” ตามที่เราได้กำหนดไว้ หลังจากนั้นกดที่ปุ่มเปิดการแจ้งเตือน (จากปุ่มสืเทา -> ปุ่มสีเขียว) จะมี Toast แสดงว่า “เปิดใช้งานการแจ้งเตือน”



6. การแจ้งเตือน

เมื่อครบกำหนดเวลาที่ผู้ใช้ตั้ง จะมีการแจ้งเตือนดังขึ้นมาบนสุดของแอปพลิเคชัน โดยเมื่อสไลด์หน้าจอจะเห็นการแจ้งเตือนนั้นและเมื่อกดที่การแจ้งเตือนนั้นจะพาไปยังแอปพลิเคชัน



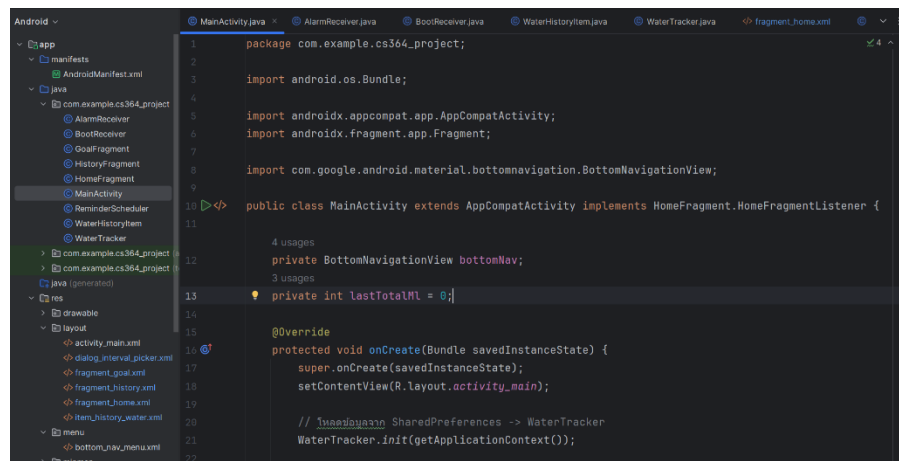
7. หน้าประวัติ

ระบบจะแสดงรายการที่ผู้ใช้น้ำดื่ม (แสดงรายการล่าสุดก่อน) โดยจะแสดงปริมาณน้ำที่ดื่ม เวลาที่ดื่ม และสามารถลบรายการนั้นได้ เมื่อลบรายการนั้น ที่ “หน้าเป้าหมาย” ข้อมูลปริมาณน้ำที่ดื่มไปแล้วจะลดลง และ Progress Bar ก็จะลดลงเช่นกัน



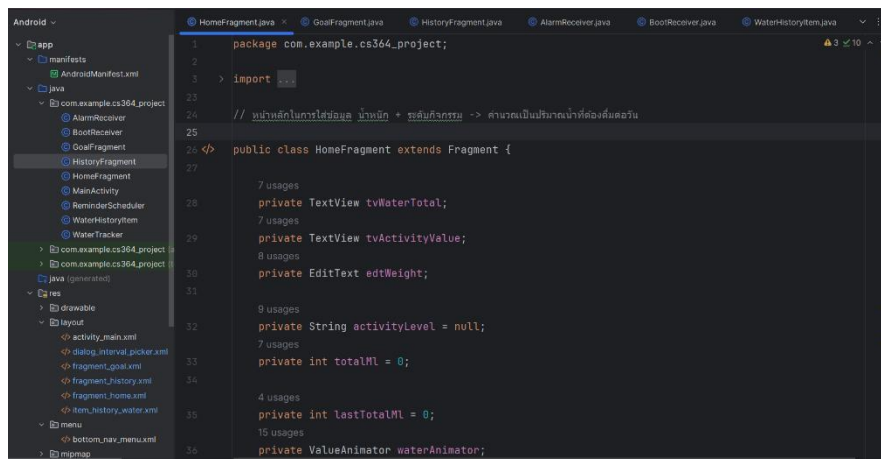
ฟังก์ชันการทำงานในส่วนต่างๆ

1. ฟังก์ชันหลัก (MainActivity)



- โหลดข้อมูลเป้าหมายน้ำที่ควรดื่มต่อวันจาก SharedPreferences เก็บไว้ในตัวแปร lastTotalML
- แสดง Fragment เมื่อเปิดแอปพลิเคชัน โดยจะแสดงหน้า HomeFragment เป็นหน้าแรก
- ควบคุมแถบ Bottom Navigation
- ส่งข้อมูลที่ใช้ป้อนแล้วคำนวณเป็นปริมาณน้ำที่ควรดื่มต่อวันเก็บไว้ใน TOTAL_ML แล้วส่งข้อมูลไปที่ GoalFragment

2. ฟังก์ชันหน้าการใช้งาน (HomeFragment, GoalFragment, HistoryFragment)



หน้า HomeFragment :

- รับค่าจากผู้ใช้งาน เช่น น้ำหนัก และ ระดับกิจกรรม
- คำนวณปริมาณน้ำที่ควรดื่ม โดยการนำน้ำหนัก (กก.) $\times 35$ แล้วบวกด้วยระดับกิจกรรม ไม่ออกกำลังกายจะบวกด้วยค่า 0, ออกกำลังกายเล็กน้อยจะบวกด้วยค่า 300, ออกกำลังกายปกติจะบวกด้วยค่า 600, ออกกำลังกายหนักจะบวกด้วยค่า 900
- ส่งผลการคำนวณไปที่ MainActivity


```

1 package com.example.cs364_project;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.fragment.app.Fragment;
5 import androidx.appcompat.widget.Toolbar;
6 import androidx.core.app.ActivityCompat;
7 import androidx.core.content.ContextCompat;
8 import androidx.core.widget.NestedScrollView;
9 import androidx.recyclerview.widget.LinearLayoutManager;
10 import androidx.recyclerview.widget.RecyclerView;
11 import androidx.appcompat.widget.Toolbar;
12 import androidx.core.app.ActivityCompat;
13 import androidx.core.content.ContextCompat;
14 import androidx.core.widget.NestedScrollView;
15 import androidx.recyclerview.widget.LinearLayoutManager;
16 import androidx.recyclerview.widget.RecyclerView;
17
18 // หน้าในการเพิ่มปริมาณน้ำ + การแจ้งเตือน + progress bar
19
20 public class GoalFragment extends Fragment {
21
22     5 usages
23     private ProgressBar progressBar;
24     3 usages
25     private TextView tvWaterTotal;
26
27     2 usages
28     private LinearLayout btnAdd250, btnAdd500, btnAdd750, btnAddCustom;
29     3 usages
30     private EditText edtCustomAmount;
31
32     5 usages
33     private Switch switchReminder;
34     14 usages
35     private EditText edtInterval;
36     3 usages
37     private LinearLayout btnSaveInterval;
38
39 }

```

หน้า GoalFragment :

- แสดง Progress Bar โดยแสดงแบบปริมาณน้ำที่ดื่มไปแล้วเปรียบเทียบกับปริมาณน้ำเป้าหมาย
- เพิ่มปริมาณน้ำที่ดื่ม เมื่อผู้ใช้กดปุ่ม 250 มล, 500 มล, 750 มล, ใส่จำนวนด้วยตนเอง
- ตั้งค่าการแจ้งเตือนการดื่มน้ำ ผู้ใช้สามารถเปิด/ปิดปุ่มการแจ้งเตือน สามารถเลือกระยะเวลาแจ้งเตือน (นาทื) บันทึกเวลาแจ้งเตือน โดยระบบจะบันทึกข้อมูลลง SharedPreferences และตั้งค่าเวลาผ่าน ReminderScheduler
- มีการขอสิทธิ์ (permission) การแจ้งเตือนที่ฟังก์ชัน requestNotificationPermissionIfNeeded()

```

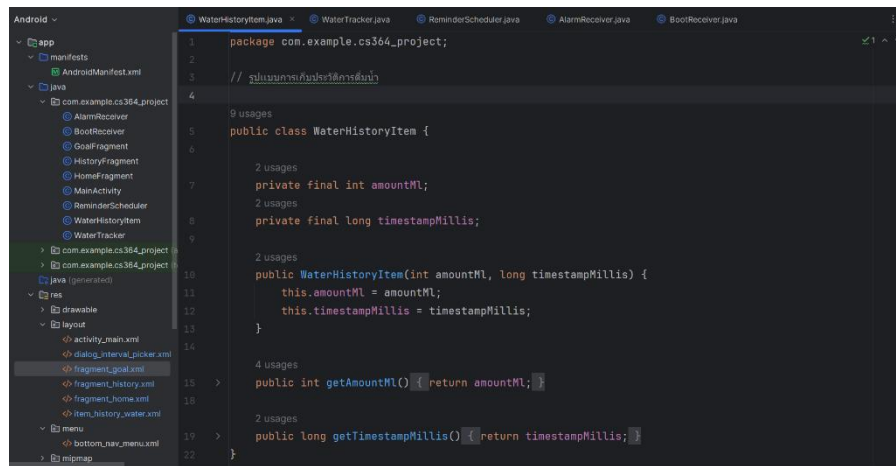
1 package com.example.cs364_project;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.fragment.app.Fragment;
5 import androidx.appcompat.widget.Toolbar;
6 import androidx.core.app.ActivityCompat;
7 import androidx.core.content.ContextCompat;
8 import androidx.core.widget.NestedScrollView;
9 import androidx.recyclerview.widget.LinearLayoutManager;
10 import androidx.recyclerview.widget.RecyclerView;
11 import androidx.appcompat.widget.Toolbar;
12 import androidx.core.app.ActivityCompat;
13 import androidx.core.content.ContextCompat;
14 import androidx.core.widget.NestedScrollView;
15 import androidx.recyclerview.widget.LinearLayoutManager;
16 import androidx.recyclerview.widget.RecyclerView;
17
18 // หน้าในการบันทึกประวัติการดื่มน้ำ + การแจ้งเตือน
19
20 public class HistoryFragment extends Fragment {
21
22     5 usages
23     private LinearLayout layoutHistoryList;
24     1 usage
25     private final SimpleDateFormat timeFormatter =
26         new SimpleDateFormat(pattern: "HH:mm", Locale.getDefault());
27
28     public HistoryFragment() {}
29
30     @Nullable
31     @Override
32     public View onCreateView(@NonNull LayoutInflater inflater,
33                             @Nullable ViewGroup container,
34                             @Nullable Bundle savedInstanceState) {
35         return inflater.inflate(R.layout.fragment_history, container, attachToRoot: false);
36     }
37
38 }

```

หน้า HistoryFragment :

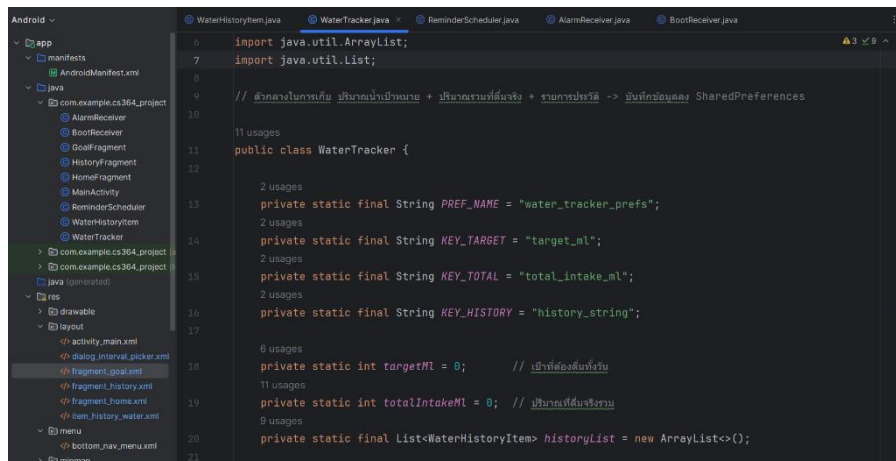
- ดึงประวัติการเติมน้ำจาก WaterTracker โดยใช้ WaterTracker.getHistory()
- แสดงรายการข้อมูลประกอบด้วย ปริมาณน้ำ, เวลา, ปุ่มลบรายการ
- ลบรายการที่ไม่ต้องการโดยเรียกใช้ btnDelete และ Progress ในหน้า GoalFragment จะลดลงด้วยเพราะใช้ WaterTracker เดียวกัน แล้วเรียกใช้ renderHistory() เพื่ออัปเดตหน้า

3. ฟังก์ชันการเก็บข้อมูล (WaterHistory, WaterTracker)



หน้า WaterHistoryItem :

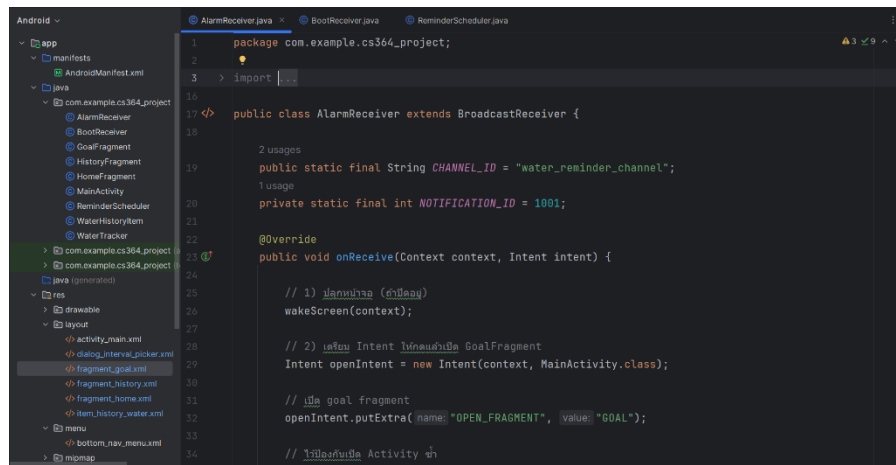
- Object ที่เก็บปริมาณน้ำและเวลา



หน้า WaterTracker :

- โหลดข้อมูลเป้าหมาย, ปริมาณน้ำที่ดื่มไปแล้ว, ประวัติการดื่มน้ำจาก SharedPreferences
- เก็บเป้าหมายน้ำที่ต้องดื่ม โดยใช้ setTarget(), getTarget()
- บันทึกปริมาณน้ำที่ดื่ม addEntry(amountML) และลบประวัติ removeEntry()

4. ฟังก์ชันการแจ้งเตือน (AlarmReceiver, BootReceiver, ReminderScheduler)



หน้า AlarmReceiver :

- แสดง Notification แจ้งเตือนการดื่มน้ำโดยการสร้าง Notification พร้อมข้อความแจ้งเตือนและเมื่อกดที่ Notification จะพาไปแอปพลิเคชัน
- ปลุกหน้าจอโดยใช้ wakeScreen(Context)

```

1 package com.example.cs364_project;
2
3 import android.content.BroadcastReceiver;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.content.SharedPreferences;
7
8 </> public class BootReceiver extends BroadcastReceiver {
9
10     @Override
11     public void onReceive(Context context, Intent intent) {
12         if (Intent.ACTION_BOOT_COMPLETED.equals(intent.getAction())) {
13
14             SharedPreferences prefs = context.getSharedPreferences("reminder_prefs", Context.MODE_PRIVATE);
15             boolean isOn = prefs.getBoolean("reminder_on", false);
16             long intervalMillis = prefs.getLong("interval_millis", 0);
17
18             if (isOn && intervalMillis > 0) {
19                 ReminderScheduler.scheduleRepeatingAlarm(context, intervalMillis);
20             }
21         }
22     }
23 }

```

หน้า BootReceiver :

- ทำงานทันทีหลังเปิดเครื่อง (BOOT เสร็จ) โดยเช็คว่าคุณใช้เปิดแจ้งเตือนคืนน้ำไว้หรือไม่ ถ้าเปิดก็ตั้งแจ้งเตือนใหม่อีกครั้ง

```

1 package com.example.cs364_project;
2
3 > import ...
4
5 4 usages
6
7 public class ReminderScheduler {
8
9     1 usage
10     public static final int REQUEST_CODE_ALARM = 2001;
11
12     3 usages
13     public static void scheduleRepeatingAlarm(Context context, long intervalMillis) {
14
15         AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
16         if (alarmManager == null) return;
17
18         PendingIntent pendingIntent = buildAlarmPendingIntent(context);
19
20         long triggerAtMillis = SystemClock.elapsedRealtime() + intervalMillis;
21
22         // ELAPSED_REALTIME_WAKEUP + Repeating
23         alarmManager.setRepeating(
24             AlarmManager.ELAPSED_REALTIME_WAKEUP,
25             triggerAtMillis, SystemClock.elapsedRealtime() + intervalMillis,

```

หน้า ReminderScheduler :

- จัดการการตั้ง Alarm แจ้งเตือนคืนน้ำและยกเลิก Alarm
- มีการใช้ ELAPSED_REALTIME_WAKEUP เพื่อปลุกหน้าจอแม้เครื่องจะปิดอยู่
- ตั้ง Alarm ให้มีการแจ้งเตือนซ้ำๆ โดยใช้ฟังก์ชัน scheduleRepeatingAlarm()
- ยกเลิกการตั้ง Alarm โดยใช้ฟังก์ชัน cancelAlarm()

ประเด็นความท้าทายทางเทคนิค

1. การออกแบบแถบ Bottom Navigation ร่วมกับ Fragment

ในตอนแรกของการออกแบบได้มีการออกแบบแต่ละส่วนแบบใช้ LinearLayout แล้วนำ BottomNavigationView ไปใส่ส่วนล่างใน Layout ต่างๆ ปรากฏว่าเมื่อทำในรูปแบบนี้จะเกิดปัญหาที่เวลาในการเปลี่ยนหน้า Bottom Navigation จะมีการเปลี่ยนแปลงไปด้วย ซึ่งเราแก้ไขโดยการออกแบบให้หน้า activity_main.xml เป็นโครงที่มีส่วนของ FrameLayout แล้วก็ BottomNavigation แล้วใช้ MainActivity.java ในการเรียกเปิดหน้า Fragment ต่างๆ เช่น HomeFragment.java, GoalFragment.java, HistoryFragment.java

2. การตั้งค่าการแจ้งเตือนผู้ใช้งาน

ในส่วนนี้จะมีความยากในการตั้งค่าขอสิทธิ์การอนุญาตต่างๆ (permission) โดยใน Android เวอร์ชันใหม่ต้องมีการขอ POST_NOTIFICATIONS จากผู้ใช้อีก การตั้งค่าจะไม่สามารถได้รับการแจ้งเตือนได้แม้ขาดส่วนใดส่วนหนึ่งทำให้ยากในการตรวจสอบ ซึ่งเราแก้ไขโดยการต้องตั้งค่าในทั้ง AndroidManifest.xml และใน GoalFragment.java ในส่วนของฟังก์ชัน requestNotificationPermissionIfNeeded()

3. การจัดการข้อมูลที่ต้องใช้ร่วมกันระหว่างหลาย Fragment ผ่าน WaterTracker

ถ้าไม่มีข้อมูลกลางที่ใช้ในการเก็บข้อมูลจะทำให้ข้อมูลในแต่ละ Fragment ไม่ตรงกัน เช่น หน้า GoalFragment กับ HistoryFragment เราแก้ปัญหาโดยการสร้าง Class WaterTracker เพื่อเป็นตัวกลางในการเก็บข้อมูล แล้วบันทึกและโหลดข้อมูลจาก SharedPreferences

4. การเชื่อมต่อข้อมูลของ Progress Bar กับ ประวัติการดื่มน้ำ

ในตอนแรก UI ของ Progress Bar ไม่อัปเดตค่าถึงแม้ผู้ใช้จะเพิ่ม/ลบ รายการการดื่มน้ำโดยเราแก้ปัญหาให้เรียกใช้ข้อมูลจาก WaterTracker แล้วทำการรีเฟรช UI โดยใช้ฟังก์ชัน refreshProgressUi()

รายชื่อสมาชิกและหน้าที่

นาย กษิตศ คงประพันธ์

- ออกแบบ UI ของแอปพลิเคชัน, จัดการโฟลเดอร์ drawable

นาย กฤตชญา กลิ่นเดช

- ออกแบบ UI ของแอปพลิเคชัน และ จัดการโฟลเดอร์ values, นำเสนอ

นาย จันทรพงศ์ วิทยอรุณธานี

- ทดสอบระบบ และ ตรวจสอบให้ตรงตามเงื่อนไข

นาย ศุภกฤต ธรรมางกูร

- จัดการโฟลเดอร์ layout และ menu, เชื่อมต่อกับไฟล์ Java ต่างๆ, นำเสนอ

Link : Github ของโครงการ

<https://github.com/SupakitThammagoon/CS364-Project.git>

Link : Presentation

<https://youtu.be/bp7rpgmFVRw>