

DAUIN

Department of Control and Computer Engineering

Master's Degree Thesis

Supervisor:
Prof. Morisio Maurizio

Well-being App: Designing a Secure and Scalable Backend for a Mobile Health and Wellness Solution

Candidate: Gagliardo Domenico



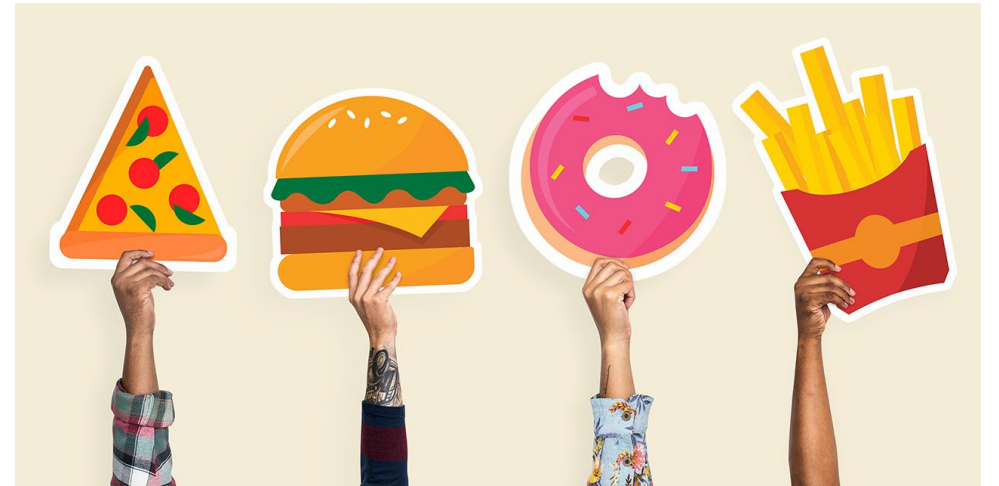
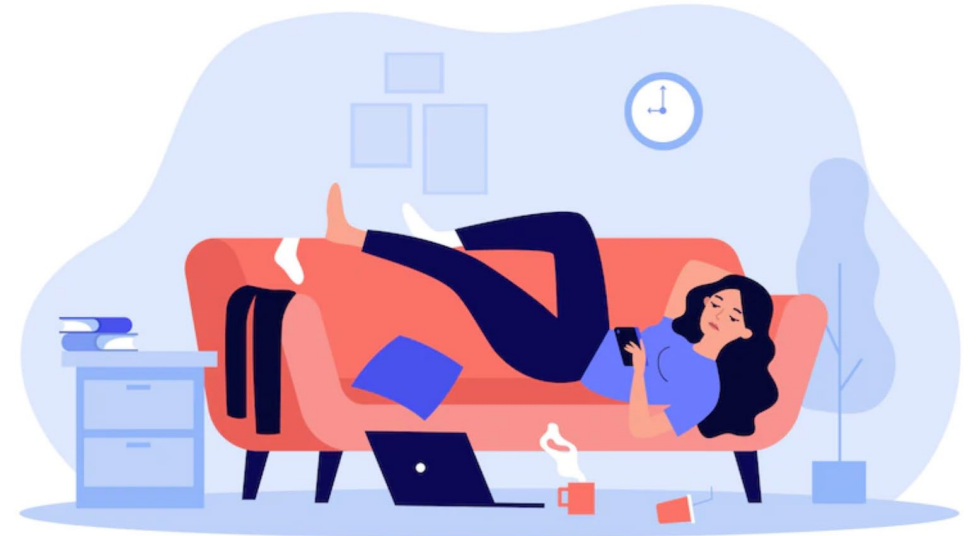
**Politecnico
di Torino**

07 April 2025, Luigi Ciminiera Conference Room

Unhealthy Habits: Poor Diet & Sedentariness

- Sedentariness leads to a whole range of problems, altering the respiratory system, cardiovascular and increasing the risk of dementia.
- Poor diet as well leads to nutrient deficiencies, weakened immune system and mental health issues.

Changing habits is therefore essential



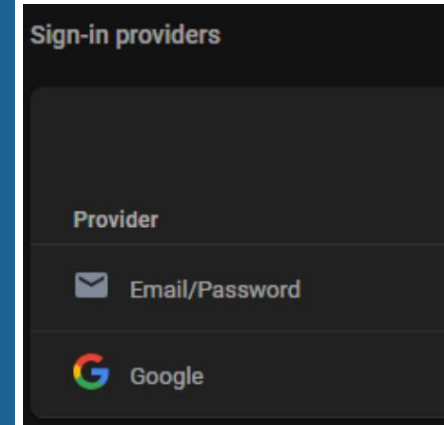
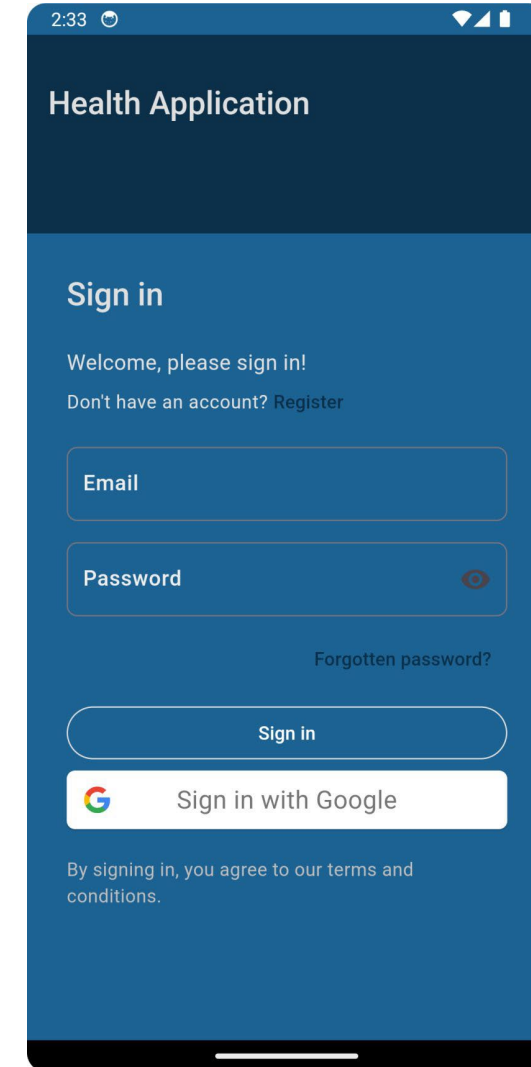
Employ Technology for Health Benefits

- Technology in this has a key role, helping people to reach a healthier lifestyle.
- This is the goal that the well being app project wants to pursue, through the usage of a smartphone application paired with a wearable device to monitor the user and collect data.



Authentication and Registration

- Firebase, chosen as the backend technology, allowed an easy authentication setup.
- Thanks to the provided APIs and graphics components, it allowed to implement a simple and reliable authentication solution.



On Boarding

- Once the account has been created, it comes the on boarding phase.
- Here all the necessary metrics are collected, and the collections that corresponds to the user are created.

```
void setUserMetrics(Map<String, dynamic> metrics) async {
    final db = FirebaseFirestore.instance;
    final user = FirebaseAuth.instance.currentUser;
    final goals = {"sleep": "400", "calories": "1500", "steps": "8000"};
    await db.collection("users").doc(user!.uid).set({"language" : Locale.languageCode}, SetOptions(merge: true));
    await db.collection("users").doc(user.uid).set({"metrics" : metrics}, SetOptions(merge: true));
    await db.collection("users").doc(user.uid).set({"goals" : goals}, SetOptions(merge: true));
    final userData = {
        "current_notification": "body_test_balance",
        "bodyTestStrengthData": [],
        "bodyTestBalanceData": [],
        "completedLessons": [],
        "completedQuizzes": [],
        "weightData": [{"weight": metrics["weight"], "date": DateFormat('yyyy-MM-dd').format(DateTime.now())}],
        "waistCircumferenceData": [{"waist circumference": metrics["waist circumference"], "date": DateFormat('yyyy-MM-dd').format(DateTime.now())}],
        "emotionalData": [],
        "lastBackupDate": "",
        "userId": user.uid
    };
    await db.collection("user_data").add(userData);
}
```

State Handling

- The best practises regarding the flutter state management ensured a fast and responsive interface.
- It also allowed to have cleaner code management with everything in one class.

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    name: "health_app_mobile_client",
    options: DefaultFirebaseOptions.currentPlatform,
  );
  await FirebaseApi().initNotifications();
  // Firebase Crashlytics
  FlutterError.onError = FirebaseCrashlytics.instance.recordFlutterError;
  await FirebaseCrashlytics.instance.setCrashlyticsCollectionEnabled(true);
  // Firebase Performance
  FirebasePerformance.instance.setPerformanceCollectionEnabled(true);
  final Trace startupTrace = FirebasePerformance.instance.newTrace("app_startup");
  startupTrace.start();
  // Process initial notification if the app was launched from a notification
  ReceivedAction? initialAction = await AwesomeNotifications().getInitialNotificationAction(removeFromActionEvents: true);
  runApp(ChangeNotifierProvider(
    create: (context) => HomeDataProvider(),
    child: MyApp(initialAction: initialAction) // ChangeNotifierProvider
  ));
  startupTrace.stop();
}

final GlobalKey<NavigatorState> navigatorKey = GlobalKey<NavigatorState>();

class MyApp extends StatelessWidget {
  final ReceivedAction? initialAction;
  const MyApp({Key? key, this.initialAction});

  @override
  Widget build(BuildContext context) {
    return Consumer<HomeDataProvider>({
      builder: (context, hdp, child) {
        return MaterialApp(
          debugShowCheckedModeBanner: false,
          theme: primaryAppTheme,
          navigatorKey: navigatorKey,
          supportedLocales: L10n.all,
          locale: hdp.locale,
          localizationsDelegates: const [
            AppLocalizations.delegate,
            GlobalMaterialLocalizations.delegate,
            GlobalWidgetsLocalizations.delegate,
            GlobalCupertinoLocalizations.delegate,
          ],
          home: AuthGate(initialAction: initialAction, homeDataProvider: hdp),
        ); // MaterialApp
      }
    ); // Consumer
  }
}
```


Data Operations

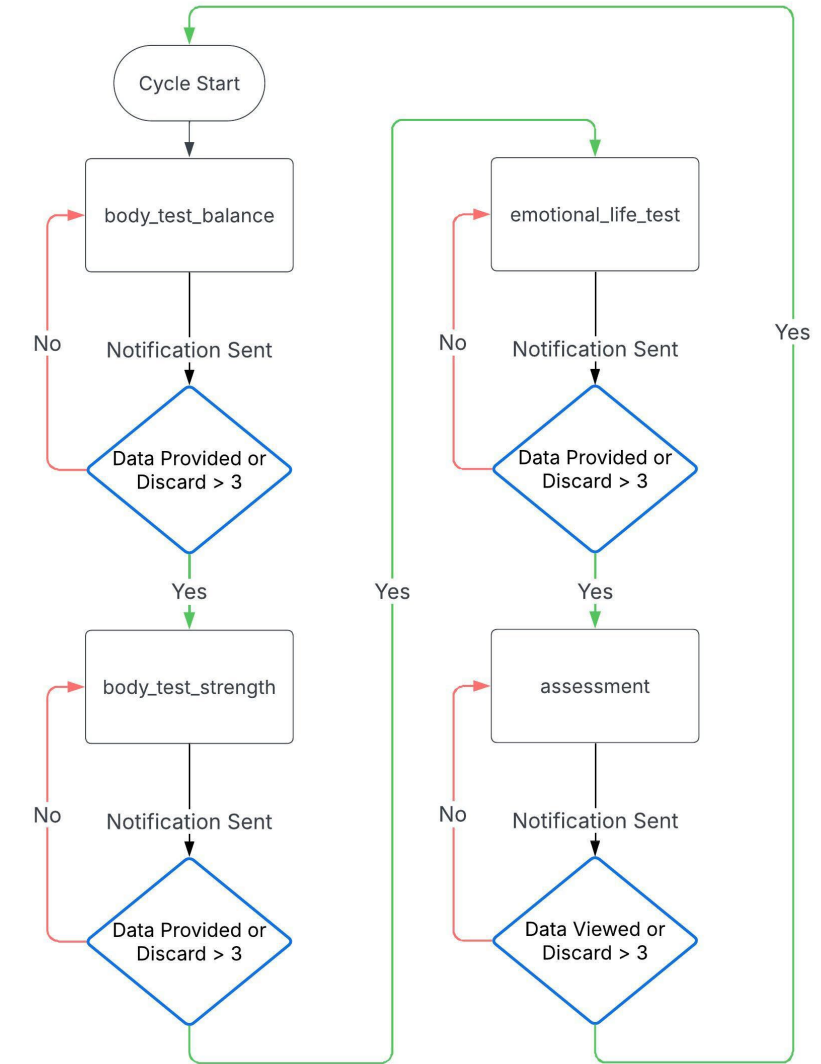
All data operations were done through either health connect or the database, allowing to use the data by showing graphs as well as personal metrics.

```
I/flutter (15228): ----- FETCHING STEPS... -----
I/FLUTTER_HEALTH(15228): Getting data for STEPS between 2025-03-31T22:00:00Z and 2025-04-03T21:59:59Z, filtering by []
D/OplusScrollToTopManager(15228): com.example.health_app_mobile_client/com.example.health_app_mobile_client.MainActivity,This com.android.internal.policy.DecorView{32db185 V.E..... R..... 0,0-1080,2376 aid=0 alpha=1.0
viewInfo = }[MainActivity] change focus to true
I/flutter (15228): ----- FETCHED STEPS -----
I/flutter (15228): {uuid: 2185420c-5a05-42f6-bfcc-2ad21ffc6e27, value: {__type: NumericHealthValue, numericValue: 10}, type: STEPS, unit: COUNT, dateFrom: 2025-04-01T00:05:20.745, dateTo: 2025-04-01T00:06:20.746,
sourcePlatform: googleHealthConnect, sourceDeviceId: TP1A.220905.001, sourceId: , sourceName: com.google.android.apps.fitness, recordingMethod: unknown}
I/flutter (15228): {uuid: 3e9e7482-0583-4311-a7f0-5a76b94689e1, value: {__type: NumericHealthValue, numericValue: 14}, type: STEPS, unit: COUNT, dateFrom: 2025-04-01T09:31:29.415, dateTo: 2025-04-01T09:32:29.416,
sourcePlatform: googleHealthConnect, sourceDeviceId: TP1A.220905.001, sourceId: , sourceName: com.google.android.apps.fitness, recordingMethod: unknown}
I/flutter (15228): {uuid: 85b32cf5-c78e-4267-bf4c-cfe3683f4e01, value: {__type: NumericHealthValue, numericValue: 10}, type: STEPS, unit: COUNT, dateFrom: 2025-04-01T10:21:32.973, dateTo: 2025-04-01T10:22:32.974,
sourcePlatform: googleHealthConnect, sourceDeviceId: TP1A.220905.001, sourceId: , sourceName: com.google.android.apps.fitness, recordingMethod: unknown}
I/flutter (15228): {uuid: 0d9b2d72-d1fa-409c-9256-dfcb67b61d25, value: {__type: NumericHealthValue, numericValue: 5}, type: STEPS, unit: COUNT, dateFrom: 2025-04-01T21:41:57.175, dateTo: 2025-04-01T21:42:57.176,
sourcePlatform: googleHealthConnect, sourceDeviceId: TP1A.220905.001, sourceId: , sourceName: com.google.android.apps.fitness, recordingMethod: unknown}
```

```
I/flutter (15228): ----- FETCHING USER GOALS... -----
I/flutter (15228): ----- FETCHED USER GOALS -----
I/flutter (15228): {sleep: 500, calories: 1800, steps: 8000}
I/flutter (15228): ----- SAVING USER GOALS... -----
I/flutter (15228): ----- SAVED USER GOALS -----
I/flutter (15228): {sleep: 520, calories: 1850, steps: 8300}
```

Notifications

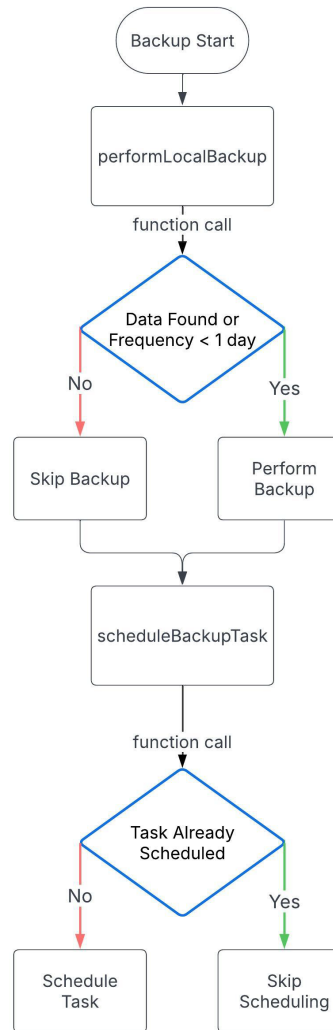
- A notification system has been set up to encourage the user to participate actively.
- The system, which is composed of several states, has the goal of notifying the user according to the `assessment_period` parameter and `current_notification` field in the database.



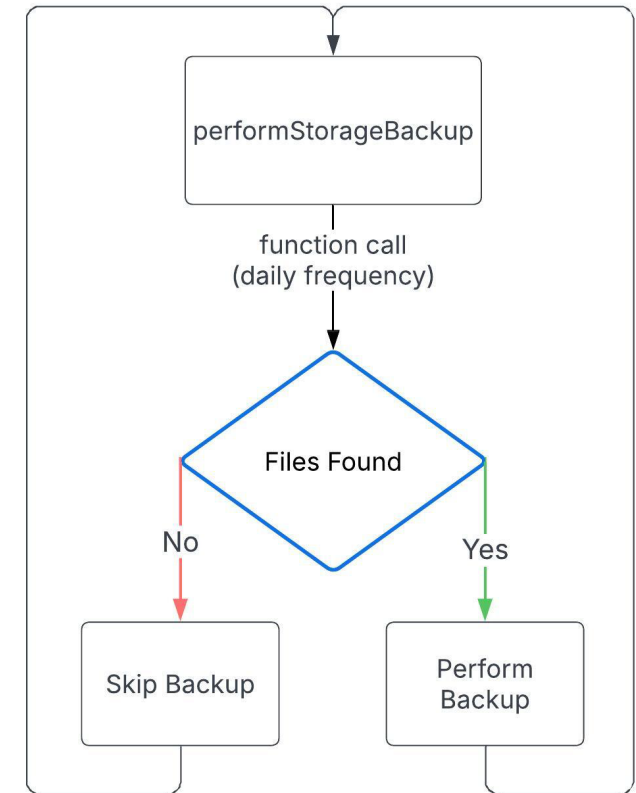
Health Data Backup

- A backup system has been set up to handle daily backs up of each user's health data.
- The system, composed of several phases and two different isolates, enables the backup logic through the usage of the lastBackupDate field for each user.

MAIN ISOLATE

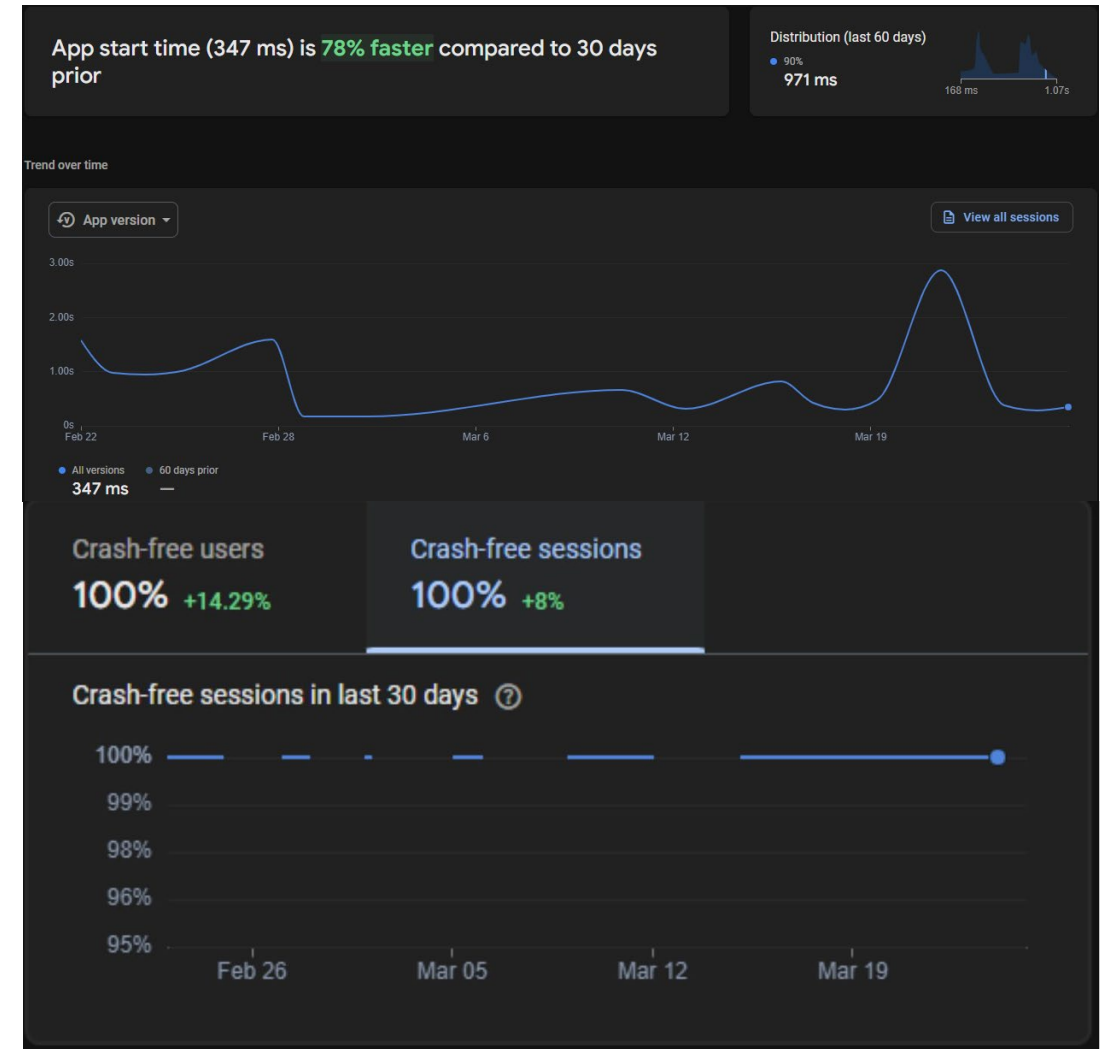


BACKGROND ISOLATE



Achieved Results

- The resulting application achieved an average start-up time below 1s and an average API response time of 176 ms.
- Also crash sessions, initially present, have been reduced to zero.



Future works

In addition to the current features, the system still has room for improvement:

- IOS Integration
- LLM acting as personal coach and consequent integration into the system



Thank you
for your attention