DEVELOPMENT PHASE / REFLECTION PHASE

Introduction

- During the development phase, I began implementing the DiaTrack habit-tracking application based on the structures and logic defined in the conception phase. This stage focused on setting up the technical environment, translating the design diagram into functional components, and enabling users to create, manage, and analyse habits related to diabetes self-care.

Frameworks, Tools, and Environment Setup

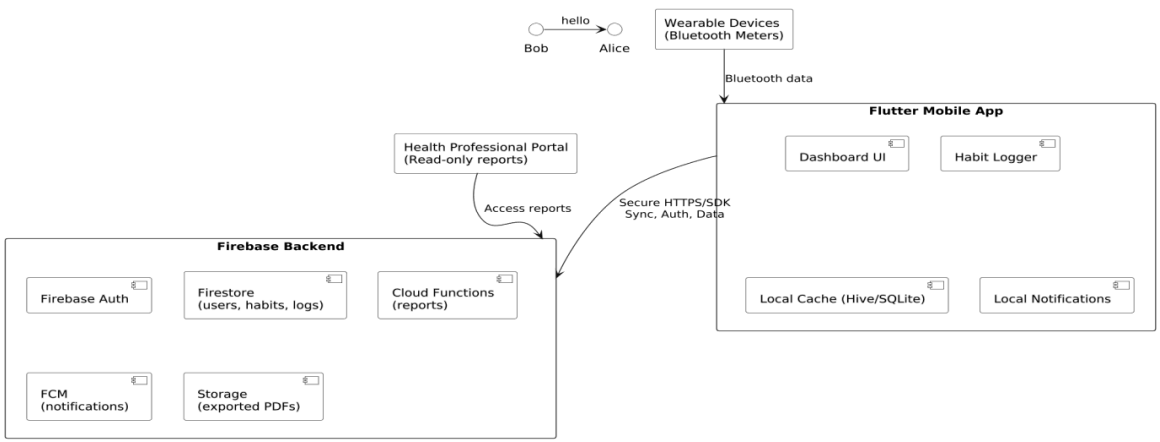
- In line with the conception phase, I set up the following tools:
- **Frontend Framework — Flutter**
- · Chosen because it allows cross-platform development for iOS and Android.
- · Provides fast UI development and supports charts for data visualisation.

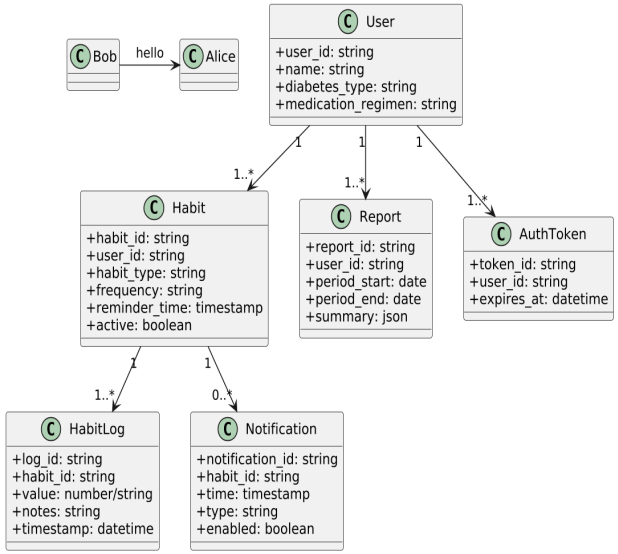
- **Backend & Cloud Services — Firebase**
- · **Firebase Authentication** for secure login
- · **Cloud Firestore** for storing habits and logs
- · **Firebase Cloud Messaging** for reminders
- · **Firebase Cloud Functions** for generating weekly reports
- **Local Tools**
- · Hive/SQLite for temporary offline storage
- · Flutter Local Notifications for habit reminders
- · GitHub for version control
- These tools fully support the requirements defined in the conception phase and ensure scalable implementation.

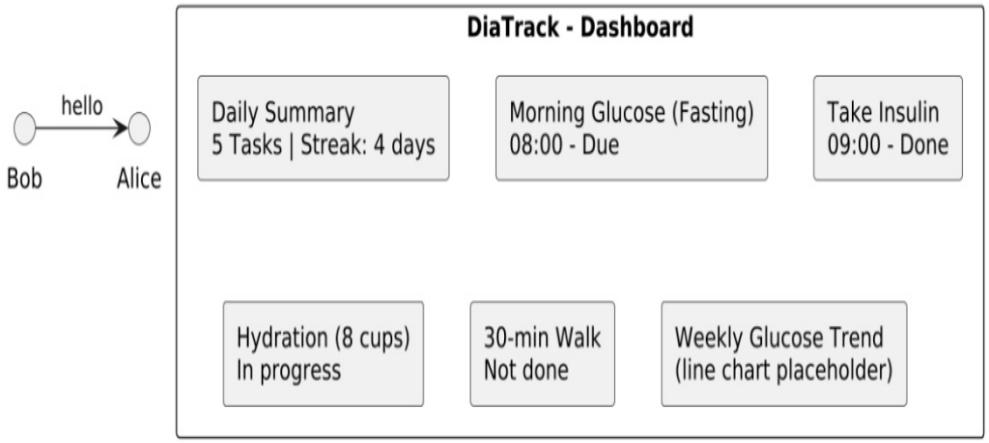
Implementation Procedure (Step-by-Step)

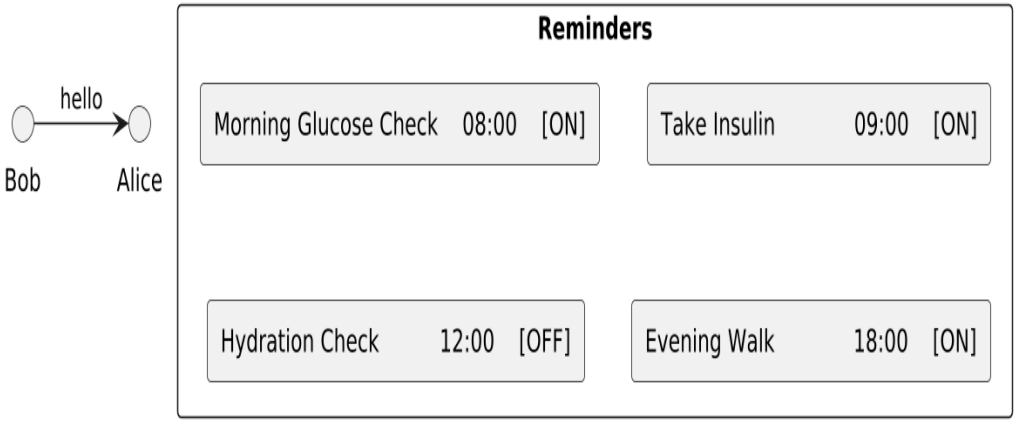
- **Core Components Implemented**
- ✓ **Habit Class**
- I translated the original UML into code by implementing a **Habit** class containing:
- · habit_id
- · habit_type
- · frequency
- · reminder_time
- · completion_status
- · log_data
- This modular class design allows new habits to be added easily.
- ✓ **User Interface Screens**
- · **Registration & Login:** FirebaseAuth setup with input validation
- · **Dashboard:** Displays today's habits and completion status
- · **Habit Logger:** Allows users to log glucose values, medication, hydration, steps, sleep, and foot care
- · **Analysis Screen:** Charts for weekly glucose trends and habit completion rates

- ✓ **Database Integration**
- · Created Firestore collections: users, habits, habit_logs
- · Implemented read/write operations with timestamping
- · Set up real-time syncing for instant data updates
- ✓ **Notifications**
- · Local reminders for medication, glucose checks, hydration
- · Notification scheduling updates when user edits habit times
- ✓ **Initial Data Analysis Features**
- · Glucose trend line graph
- · Weekly activity/habit completion charts
- · Highlighting missing logs or irregularities
-


hello
Bob
Alice
Wearable Devices
(Bluetooth Meters)
Bluetooth data
Flutter Mobile App
Dashboard UI
Habit Logger
Local Cache (Hive/SQLite)
Local Notifications
Health Professional Portal
(Read-only reports)
Access reports
Secure HTTPS/SDK
Sync, Auth, Data
Firebase Backend
Firebase Auth
Firestore
(users, habits, logs)
Cloud Functions
(reports)
FCM
(notifications)
Storage
(exported PDFs)


Bob
hello
Alice
User
+user_id: string
+name: string
+diabetes_type: string
+medication_regimen: string
Habit
+habit_id: string
+user_id: string
+habit_type: string
+frequency: string
+reminder_time: timestamp
+active: boolean
Report
+report_id: string
+user_id: string
+period_start: date
+period_end: date
+summary: json
AuthToken
+token_id: string
+user_id: string
+expires_at: datetime
HabitLog
+log_id: string
+habit_id: string
+value: number/string
+notes: string
+timestamp: datetime
Notification
+notification_id: string
+habit_id: string
+time: timestamp
+type: string
+enabled: boolean


hello
Bob
Alice
DiaTrack - Dashboard
Daily Summary
5 Tasks | Streak: 4 days
Morning Glucose (Fasting)
08:00 - Due
Take Insulin
09:00 - Done
Hydration (8 cups)
In progress
30-min Walk
Not done
Weekly Glucose Trend
(line chart placeholder)


hello
Bob
Alice
Reminders
Morning Glucose Check 08:00 [ON]
Take Insulin 09:00 [ON]
Hydration Check 12:00 [OFF]
Evening Walk 18:00 [ON]

Tools Justification

- **Flutter**
- · Fast development
- · Beautiful, responsive UI
- · Ideal for mobile apps used by patients
- **Firebase**
- · Secure storage for health-related data
- · Real-time sync helps users track habits across devices
- · Cloud Functions automate report generation
- **Local Notifications**
- · Essential for reminding diabetic patients about time-sensitive habits
- By selecting these tools, development became faster, more reliable, and aligned with the needs of chronic disease management apps.

Reflection on Development Process

- During implementation, I ensured alignment with the original goals:
- ✓ **Realization of the task**
- I implemented all core functions from the conception phase, including habit creation, logging, reminders, and visual analysis.
- ✓ **Orientation to goals**
- The priority was usability for diabetic patients:
- · Simple UI
- · Clear daily tasks
- · Direct feedback through charts
- ✓ **Resource considerations**
- · Flutter allowed rapid progress within the project timeline
- · Firebase reduced backend workload and cost

- ✓ **Challenges and Risks**
- · Handling health data requires strict security (mitigated using Firebase rules)
- · Ensuring notifications run reliably across devices
- · Designing charts understandable for non-technical users
- ✓ **Opportunities for improvement**
- · Add AI-based predictions for glucose trends
- · Integrate wearable devices (glucose meters, step counters)
- · Expand habit library based on patient feedback
- ✓ **Monitoring progress**
- · Weekly commits to GitHub
- · Testing after each module
- · Feedback from test users helped improve UI clarity