NudgeMe - COMP0016 2020/21, Team 26

V2 of the CarerCare app.

NudgeMe is a privacy focused, cross-platform mobile app that helps you track and share your wellbeing & steps.

This is built on Flutter's stable branch.

Deployment

Android

1. Install flutter, and use `flutter doctor` to check if it's set up correctly.
2. Run `flutter build apk -t lib/main_production.dart` in the project directory. This builds the production version, meant for end users. It does not display the dev screen or send remote error reports. For the version used during development, simply run `flutter build apk`.
3. Install the apk on an Android device.

iOS

To build a signed ipa, you will need a paid Apple Developer account and a Mac computer.

If you have both if these and have installed flutter (type 'flutter doctor' in the terminal to check there are no issues here), do the following:

1. Open Xcode and open this project by locating the project in Finder and opening the *.xcodeproj file.
2. Select Generic iOS Device as your project's device target (Product > Destination and then choose Generic iOS Device option).
3. In the Product menu, select Clean.
4. In the Product menu, select Archive. When the archiving process completes, you will see your application listed under Archives.

5. Select your application and click Export button on the right.
6. When prompted for an export method, select iOS App Store to upload the app to the iOS App Store, Ad Hoc for internal distribution of the app, Enterprise for distribution outside the App Store, or Development for testing.
7. Set these Distribution options:
   - Set App Thinning to None.
   - If you are building for app store, select Rebuild from Bitcode. Deselect if not.
   - Select Strip Swift symbols to reduce app size. This is optional.

- Deselect Include manifest for over-the-air installation.

8. Select your Distribution Certificate and Provisioning Profile (Automatic or Manual). This will generate the .ipa file.
9. When the file generation process completes, click Export and choose where to save the .ipa file.

Configuration

Here are some aspects you may wish to configure:

- In `main.dart`, you can modify the `dsn` property of `SentryOptions` to your own DSN provided by Sentry if you wish to receive remote error logs.
- In `main_pages.dart`, you should change the BASE_URL string if you change the domain name.

Tests

In the root project directory:

- Run `flutter test` to run the unit/widget tests.
- Run `flutter drive --driver=test_driver\integration_test.dart --target=integration_test\main_test.dart` with an emulator or device connected to run the integration tests.

Integration tests run through the device, whereas the widget tests use a different (simulated) execution model.

Initial Requirements

- The system should be designed for a user between 13 and 99
- The target user is a person who wishes to share their wellbeing with others
- It should passively gather movement data from the pedometer and cross reference this against a self reported wellbeing score
- The wellbeing score will be gathered at 12 pm on a Sunday
- People will be nudged to share their wellbeing score as a pdf/jpeg is their score falls twice over any two week period. This will be facilitated.
- They will also be encouraged to do the same of there is no pedometer reading over two days. On a Monday each week they will be asked if they to share their average wellbeing score (in a locally differentially private manner) with a central wellbeing hub. This hub will request put requests in the same manner as the previous App

API Docs

Endpoint to send wellbeing data: `https://comp0016.cyberchris.xyz/add-wellbeing-record`

- postCode: string e.g. TW6
- wellbeingScore: integer

- weeklySteps: integer
- errorRate: integer, this is abs(score-userScore), where score is our estimate of their score
- supportCode: String
- date_sent: string, 'yyyy-MM-dd'

Example POST

Using curl:

```
DATA='{
    "postCode":"TW5",
    "wellbeingScore":9,
    "weeklySteps":650,
    "errorRate":9,
    "supportCode":"GP",
    "date_sent":"2021-01-02"
    }'
curl -d $DATA -H 'Content-Type: application/json;charset=UTF-8' \
    https://comp0016.cyberchris.xyz/add-wellbeing-record
```

Wellbeing Sharing

We would like to share the last 5 weeks, like with the PDF, so the JSON *data* would be something like this:

```
[
{"week": 1, "score": 8, "steps": 1005},
{"week": 2, "score": 9, "steps": 12300},
{"week": 3, "score": 7, "steps": 105},
{"week": 4, "score": 2, "steps": 200},
{"week": 5, "score": 3, "steps": 300},
]
```

(Using a dictionary instead of an array because we may want their week number.)

But we want e2e encryption, so mobile clients should convert this json to a string and encrypt this with the friend's public key, then send this as base64.

N.B. this just describes the 'data' value in the response body, see the back-end documentation for the full response format.

Nudging Other Users

There are two types of messages related to nudges: a new nudge, or an update indicating that the goal of the nudge has been met.

```
{"type": "nudge-new", "goal": 7000},
```

```
{"type": "nudge-completed", "goal": 8400},
```

For now, clients might not need to encrypt, as we are only sending step goals, and whether they were met. So the worst case scenario where our server is malicious: our server could track an IP addresses and see that address 1 is setting a goal of x amount of steps for address 2, and that they are hitting that goal y days later.

(In contrary, with wellbeing sharing unencrypted, a malicious server could get precise data on numbers of steps and their current mental health, courtesy of the wellbeing score.)

Architecture Diagrams

Main Diagram

See the corresponding Figma here

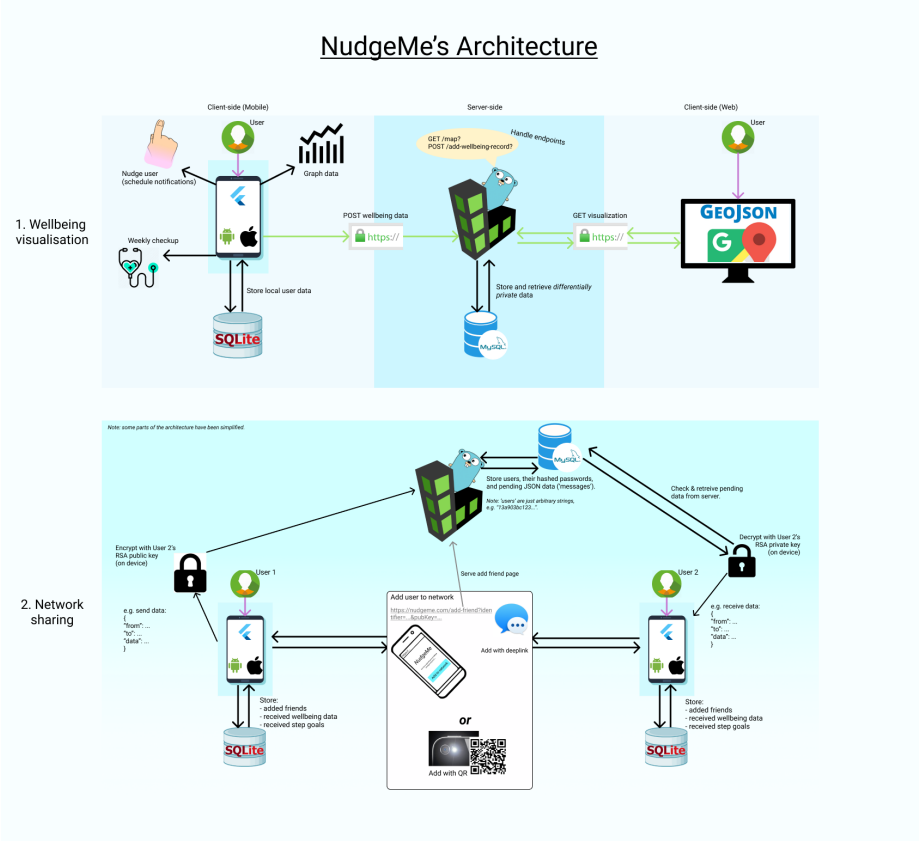

Figure 1: image

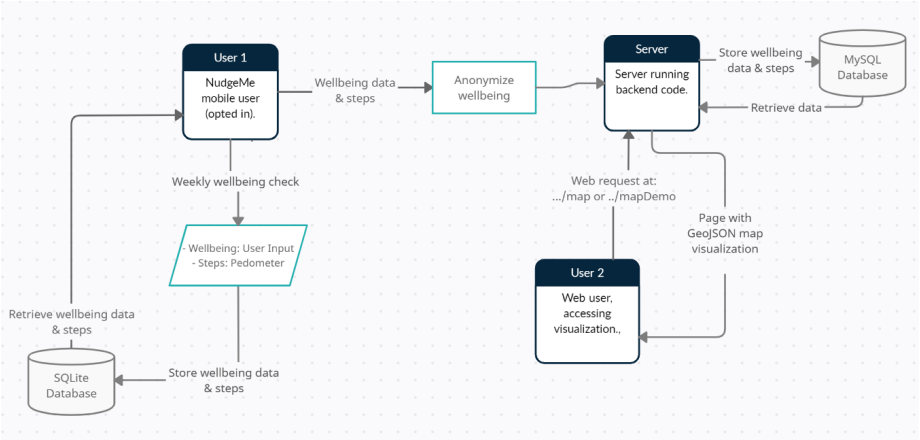

Figure 2: image

Dataflow Diagram - Wellbeing Visualization

Sequence Diagram - User Wellbeing Sharing

Figure 3: image