mPulse Mobile Coding Challenge

Background

Thank you for applying to mPulse! To better evaluate your technical skills, we would like you to complete this coding assignment as part of your interview. We hope that by providing a coding challenge, you can find your flow, work in an environment you're comfortable in, and show us your best code. If at any point you have questions, please feel free to reach out to us.

Task

At mPulse Mobile a central entity to our application is members. Using our system, Healthcare providers interact with their subscribers (members) to convey important information about their health. As you'd imagine, Healthcare providers have a lot of members and as an integral point in our system, adding and retrieving members has to be a fast operation for us. As part of this assignment, we'd like for you to take on a smaller version of this problem.

We would like you to write an application in Python and Django (the majority of our tech stack) which serves an API to fetch and create members. As part of this challenge, a CSV file is provided with the following columns:

- ID
- First Name
- Last Name
- Phone Number
- Client Member ID
- Account ID

These columns should help make up how a member is represented in your application.

Requirements

Heads up! These are hard requirements and will be the portions we focus the most on when reviewing your code.

In addition to the API that needs to be exposed above, there are a number of technical and business requirements that we would like you to adhere to:

- Certain members can belong to multiple Healthcare providers. These members may subscribe to different providers using the same Phone Number or Client Member ID. We want to ensure that our phone numbers or Client Member IDs are unique per Account.
    - **Example**
        - Assume my phone number is 818-845-1059 and I have health insurance with a provider like Kaiser and vision insurance with a provider like Guardian.

        - We need to be able to store my information so that I can appear in the system multiple times with my same phone number (818-845-1059) but that I only exist once for Guardian and once for Kaiser.
- The application should allow users to upload large files while still servicing other requests, i.e. uploading should happen in the background.
    - Python is a single threaded language. If we only have one instance of the application deployed and a user uploads a large file, we want to make sure that our application can still service other requests.
- When batch inserting members, your application should provide a best-effort approach and attempt to insert as many members as possible.
- In a real-world situation, we receive files with up to 5 million rows. Consider how your upload API scales and handles large files.
- This should be production-ready code that you would feel comfortable deploying.

Given this information, your API should provide endpoints which allow its clients to:

- Get a member for a given Account ID
- Get a member by their ID
- Get a member by their Phone Number
- Get a member by their Client Member ID
- Create a new Member
- Batch insert members by uploading a CSV (similar to the one provided)

Notes

In addition to this PDF, two additional files have been added which should help with testing.

- member_data_with_duplicate_records.csv
    - This is a CSV file which contains members with the same client member id or phone numbers for the same account.
    - You can use this file for testing batch uploads and making sure that those users with valid records are still inserted.
- member_data.csv
    - This is a “large” csv file of 1,000 rows.
    - In production we get much larger CSV files, but this should help with testing.

When creating your application, please take into consideration that a developer will need to not only evaluate your code but also run your application to validate that it works correctly. It would be very helpful if you could provide instructions on how to setup and run the application including things like running migrations, installing dependencies, etc. In light of this, please take some time to consider how to create an easy setup would be for a new developer for your project.