EPFL

Digital Epidemiology Lab

Enhancing User Engagement in Dietary Tracking Apps through Personalised Interactions and Large Language Models

Andrea Giovanni Perozziello

École Polytechnique Fédérale de Lausanne (EPFL)

School of Computer and Communication Sciences (IC)

July 2024

Supervisor: Prof. Marcel Salathé

Outlines

Introduction

- User engagement is crucial for dietary tracking apps
  - Ensures reliable data for research and clinical trials
- Existing research limitations
  - Focus on persuasive messages and notifications
  - Lack of dietary context and advanced AI integration
- Users tend to forget to log meals
  - Need for personalized, engaging interactions
- We created a framework implementing LLM for personalized interactions and reminders to test if these can increase user engagement with applications such as the MyFoodRepo app

Key Technologies

- Flask Framework
- Database Management with SQLAlchemy and SQLite
- Message Integration with Twilio
- Task Scheduling with APScheduler
- Data Manipulation with Pandas
- Chatbot Integration with OpenAI

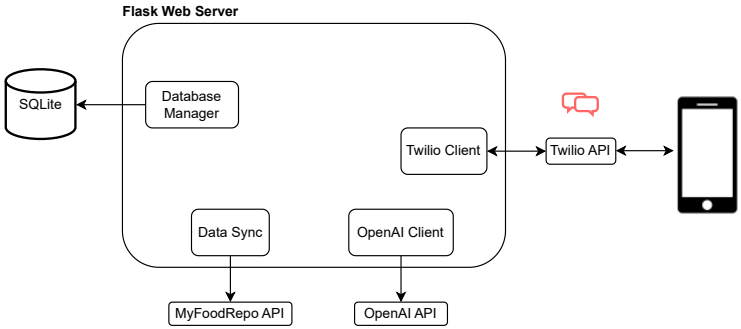
Design Overview

- Dual-component system: **web service** and **messaging service**
- Tests and enhances user engagement via personalized interactions, reminders, and chatbot functionalities
- Goal: Improve user engagement in digital health apps through LLM integration

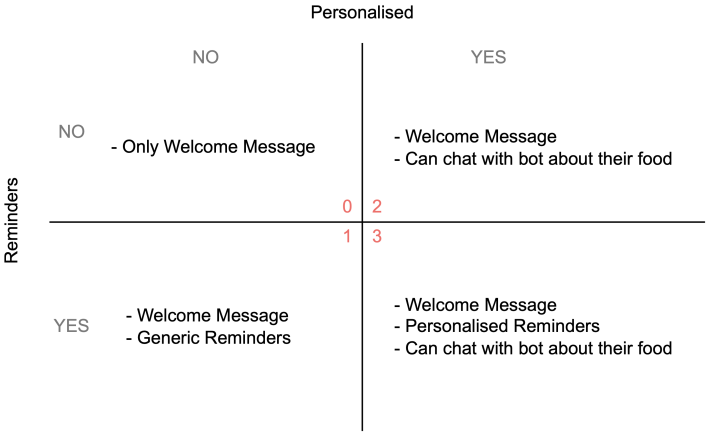
System Architecture

- Modular monolithic architecture
  - Cost-effective and suitable for experimentation
  - Simplifies design, maintenance, and scalability
- Key modules: Communication, Response, Config, Locales, Scheduler, OpenAI Client

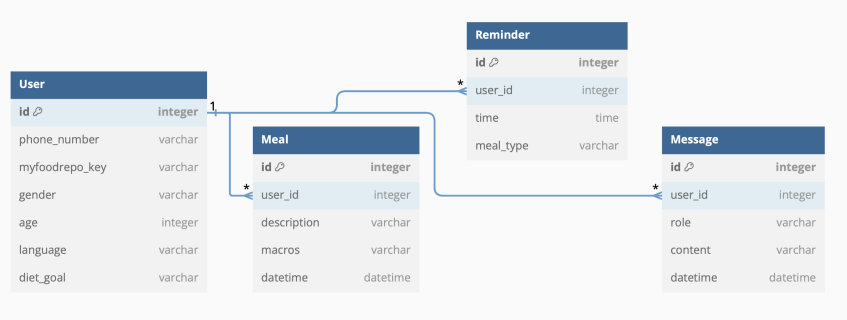
Content Diagram



User Interaction Design



Database Design



Setup and Initialization

- Performed Exploratory Data Analysis (EDA) on MyFoodRepo food logs using Python notebooks
- EDA helped understand **data structure**, **meal grouping**, **macro-nutrient aggregation**
- Created informative summaries for the LLM to provide user diet information

Database Configuration

- Used SQLAlchemy for smooth database operations
- Database auto-creation and initialization at server startup
- Implemented scoped session manager with *sessionmaker* for efficient and safe transactions

API Endpoints

- Developed several endpoints for system communication:
  - **sms_reply**: Handles all incoming messages
  - **add_all_reminders**: Adds reminders for users in Groups 1 and 3
  - **init**: Populates the database from a CSV file
  - **start**: Sends welcome messages to all participants
- Facilitates interactions between web service and messaging components
- Processes user inputs and generates appropriate responses
- Initializes components during project setup

Integration of OpenAI

- Created a dedicated module for OpenAI interactions
- Functions: **initialize conversations**, **manage message roles**, **create chat completions**
- Each message can have one of the following roles:
  1. *system*: Used to prompt before requesting a Chat Completions
  2. *assistant*: The actual response returned by the Chat Completions
  3. *user*: The incoming message received from the user
- Used Chat Completions for chatbot messages and personalized reminders
- Enhanced user experience with timely, relevant, and engaging interactions

System Prompt Example

```
REMINDER_PROMPT_TEMPLATE = """
You are a helpful assistant tasked with generating personalized reminders for users of a food diary app.
Below is the information about the user and their dietary habits.
Use this information to create a personalized reminder message.
This {meal_type} reminder should encourage the user to log their meals, considering their dietary goals, and meal history.
User Information:
- Gender: {gender}
- Age: {age}
- Dietary Goal: {dietary_goal}
- Recent Meals: {recent_meals}

Use these examples to guide your responses, but feel free to be creative and adjust based on the users recent meals and specific situation. Generate one personalized reminder message for the user."
"""
```

Recent Meals Example

```
Food Item: Wholemeal pasta egg-free, cooked in salted water (uniodised), Green pea, steamed (without addition of salt), Tomato sauce
Time: 2024-06-15 14:04
Macros:
- Calories: 357.45
- Water_ml: 162.87ml
- Protein_g: 15.19g
- Carbs_g: 60.68g
- Fat_g: 2.90g
- Sat_fat_g: 0.48g
- Fiber_g: 14.20g
```

Message Handling

- Implemented using Twilio for SMS and WhatsApp communication
- Abstraction for messaging services in the `communication` module
  - Simplifies code by avoiding low-level message sending details
- Developed **three main functions**:
  1. Handle incoming messages and generate AI-based replies
  2. Send messages (like reminders) to users
  3. Send a welcome message to all users
- Uses user's phone number as recipient and identification mechanism

Scheduling Tasks

- Designed using APScheduler for timely task execution
- Scheduler initiates upon server startup
- Performs three main tasks:
  1. **Fetches reminders** for users in groups 1 and 3 from the database
  2. **Schedules reminders** as jobs using `'cron'`
  3. **Schedules database synchronization** jobs multiple times a day using `'intervals'`

Data Synchronization

- Sync jobs **fetch data for each user** from the
- Sync jobs scheduled at regular intervals (every 2 hours)
- Process data using Pandas to extract relevant information
- Store processed data in the database
- Maintains up-to-date information on user meals logged in the MyFoodRepo app

Evaluation Introduction

- Purpose: Confirm design effectiveness
- Hypothesis: Virtual conversational agent enhances user engagement in the MyFoodRepo app

Evaluation Setup and Experiment Design

- Participants: Four groups, 50 participants each
- Group Assignments:
  - **Group 0**: Control, existing features only
  - **Group 1**: Static reminders (breakfast, lunch, dinner) using Cialdini's principles
  - **Group 2**: Chat with bot about logged food for nutritional guidance
  - **Group 3**: Chat with bot + personalised reminders based on user data
- Setting: Real-world usage with MyFoodRepo app on smartphones, with chatbot and reminders features by SMS

Evaluation Metrics

- **User Engagement**: Frequency of app use, chatbot interactions, meal logging consistency
- **User Retention**: Number of meals logged, time gaps, calorie logging consistency
- **User Satisfaction**: 5-point Likert scale feedback form

Expected Results

- Chatbot and personalised reminders to enhance engagement, retention, satisfaction
- Higher engagement for Groups 2 and 3
- Better retention with personalised reminders (Group 3)
- Higher satisfaction with tailored experiences (Group 3)

Discussion

- Potential Performance: Expected to improve engagement and retention
- Strengths: Detailed analysis, link to MyFoodRepo app
- Limitations: LLM capabilities, SMS vs. push notifications, need for complex interactions
- Lessons Learned: Diversify LLM-generated reminders, integrate user feedback
- Insights for Future Work: Conduct experiments, enhance personalisation and adaptability

Thank you!