Salma Madoud 148586
Rim Bousta 149871
Rami Mazaoui 155665
Hassan Hankir 117263
Hadil Raad 155882

UniMates

System Requirements Document (SRD):

Functional Requirements for UniMates

1. User Profile & Purpose Setup

- The system **shall allow** users to create a personal profile containing:
    - Name, major, and year of study.
    - Interests (e.g., gym, studying, social activities, volunteering).
    - Availability (specific days/times).
    - Purpose of use (e.g., “study partner,” “gym buddy,” “event companion”).
- The system **shall allow** users to edit or update their profile information at any time.

2. Matching Algorithm

The system **shall match** users based on compatible purposes, shared interests, and overlapping availability.

The system **shall display** potential matches ranked by relevance.

The system **shall allow** users to accept or decline suggested matches.

3. Instant Notification System

The system **shall send** real-time notifications when:

- A new match is found.

- A request is received, accepted, or declined.

- An event invitation or message is received.

4. Accessibility Features (Text-to-Speech)

- The system **shall provide** a text-to-speech feature for visually impaired users across all major app screens.

- The system **shall support** voice commands for navigation and posting requests.
- The system **shall allow** users to enable or disable accessibility options in settings.

5. Safety & Verification

The system **shall require** university email verification for all users during registration.

The system **shall allow** users to report or block others for harassment or inappropriate behavior.

The system **shall automatically restrict** access to unverified or flagged accounts.

6. In-App Messaging

The system shall provide a secure chat feature between matched users before meeting.

8. Multi-Purpose Matching Options

The system **shall allow** users to select their desired goal each time they create a request (e.g., “study session,” “gym,” “coffee,” “walk”).

The system **shall update** the matching suggestions dynamically based on selected goals.

9. Feedback System

After each meeting or event, the system **shall prompt** users to rate their experience (e.g., stars or short comments).

The system **shall store** feedback securely for future reference and system improvement.

Non-Functional Requirements for the MVP:

1. Performance

- Loading Time: App screens must load within 0.5 second under normal network conditions. (at most 1 second)

- Matching Algorithm: The user-matching process should return relevant results within 0.5 second.

- Scalability: The system should support from 50 to 100 concurrent users without degradation in performance.

2. Security

- **Data Protection:** All user data (profiles, preferences, messages) must be encrypted in transit (HTTPS) and at rest.

- **Authentication:** Implement secure login using university single sign-on.

- **Privacy:** Users must have full control over what personal details are visible to others.

3. Usability

- **Intuitive Interface:** The app must be easy to navigate without requiring a tutorial.

- **Consistency:** Maintain consistent layouts, icons, and color schemes across all pages.

- **Feedback:** Provide clear feedback for every user action (e.g., sent request, matched successfully).

4. Reliability

- **Uptime:** The system should maintain 99% uptime.

- **Backup:** Data must be backed up daily to prevent loss.

- **Recovery:** System recovery from failure should occur within 10 minutes.

5. Maintainability

- **Code Structure:** Use modular, well-documented code for easier updates.

- **Testing:** Maintain unit tests with at least 80% code coverage.

6. Internationalization

- **Localization:** Support multiple languages (starting with English and French).

Layer And Technologies:

Frontend (Mobile App)
React Native, TypeScript, Redux

Backend (APIs)
Node.js (Express) + Python (FastAPI for ML logic)

Database
PostgreSQL + Firebase Firestore (hybrid)

Notifications
Firebase Cloud Messaging

Auth & Security
Firebase Auth, JWT, bcrypt

AI/Matching Logic
scikit-learn, Pandas, Python

Cloud & Hosting
AWS (EC2, S3), Firebase

Accessibility APIs
Google Cloud Speech, Expo Speech

Messaging
Socket.IO, Firebase Realtime DB