Lab 2 Section 3 – MingleMap Software Requirements Specification

Team Iron

Old Dominion University

CS 411W

Professor Sarah Hosni

November 25, 2025

Version 2

Table of Contents

3 Product Requirements

3.1 Functional Requirements

3.1.1 User Interface (O: Nicholas, M1: Jacob)

- The system shall provide separate screens for Account Creation, Login, People Nearby, Map, Direct Messaging, and Profile management.

3.1.1.1 Account Creation (O: Taran, M1: Ahmer)

- The system shall display a registration page requesting an email address and password.
- The system shall validate the email format and ensure the email is not already associated with an existing account.
- Upon successful validation, the system shall securely hash the password and store the user's credentials in the database.

3.1.1.2 Login (O: Dustin, M1: Jacob)

- The system shall display a login screen requesting email and password.
- The system shall validate that both fields are not empty before submission.
- The system shall send login credentials to the backend API for verification.

- The system shall display an error message for invalid login attempts.
- The system shall display a descriptive error message if the email is invalid or already in use.
- The system shall store the returned JWT upon successful login.
- The system shall keep the user logged in while the JWT is valid.
- The system shall navigate the user to the main app screen after a successful login.

3.1.1.3 People Nearby Tab (O: Dustin)

- The system shall navigate the user to the main app screen after a successful login.
- The system shall show each nearby user's approximate distance.
- The system shall allow a manual refresh via pull-to-refresh.
- The system shall allow tapping a user to open their profile or start messaging.
- The system shall display a "No nearby users found" message when appropriate.

3.1.1.4 Map Tab (O: Taran)

- The system shall receive current location from the Google Maps API.

- The system shall report a change in position every 3 seconds.
- The system shall update the location based on the report.
- The system shall update the displayed location whenever the device reports a change in position.

3.1.1.5 Direct Messaging Tab (O: Geelani)

- The system shall allow users to exchange real-time text messages, with an end-to-end delivery latency not exceeding 200 milliseconds under normal network conditions.
- The system shall provide a visual confirmation to inform the sender that a message has been successfully sent to the recipient.
- The system shall hide chat sessions when a participating user leaves the location.
- The system shall prevent users from sending new messages once a chat session has closed.

3.1.1.6 Profile Tab (O: Geelani)

- The system shall display the user's profile picture, name, and interest tags.
- The system shall allow users to edit their interest tags from the profile tab.

- The system shall allow users to upload or change their profile picture. It will accept common picture formats such as JPEG, JPG, and PNG.
- The system shall display a user's current visibility and immediately update its value when changed.

3.1.1.7 Interest Tags (O: Dustin, M1: Ben, M2: Geelani, M3: Ahmer)

- The system shall allow the user to select up to 10 interest tags from a scrollable and searchable list.
- The system shall allow users to select interest tags during account creation (see Account Creation).
- The system shall allow the user to remove any previously selected interest tag from the Profile Tab after account creation.
- The system shall allow the user to add new interest tags from the Profile Tab after account creation, subject to the 10-tag limit.
- The system shall display each user's interest tags on both the Map tab and the "People Nearby" tab.

3.1.1.8 Visibility Setting (O: Ben, M1: Geelani)

- The system shall allow the user to toggle their account visibility via buttons on the "People Nearby", Map, and Profile tabs.
- The system shall **only** display visible users.

- The system shall **only** display other visible users on the "People Nearby" tab.
- The system shall **not** display the native user to themself on the "People Nearby" tab, regardless of visibility.
- The system shall **only** display the native user (if they are visible) **and** other visible users on the Map tab.

3.1.1.9 Trust Scores (O: Ben)

- The system shall display each user's Trust Score (see Trust Score Adjustment) on the Map and "People Nearby" tabs.

3.1.1.10 Reporting (O: Nicholas, M1: Ben, M2: Taran)

- The system shall let users report behavior that goes against the terms and services by pressing a button attached to the offender's profile.
- The system shall allow the user to choose a category for the reason of their report from a dropdown menu.
- The system shall post each report on the administrator board for further review.
- The system shall await an administrator's input on a report.
- If a report is approved, the system shall send a request to reduce the reported's trust score.

3.1.1.11 Profile Pictures (O: Ahmer, M1: Geelani, M2: Ahmer)

- The system shall allow users to upload a profile picture from their device and associate it with their account.
- The system shall accept JPG/JPEG/PNG formats; all others shall be rejected.
- The system shall store uploaded profile pictures securely in the backend/cloud storage.
- The system shall link the stored image file path to the corresponding user account in the database.
- The system shall display the user's current profile picture on their Profile tab.
- The system shall update the displayed profile picture when the user uploads a new image.

Acceptance Criteria

- Images upload successfully under 3 seconds on stable network conditions.

3.1.2 Algorithms (O: Nicholas)

- The system shall implement all algorithms necessary to support authentication, proximity detection, interest-based matchmaking, messaging, trust-score adjustment, and report processing for the MingleMap application.

3.1.2.1 Login Authentication (O: Daksh)

- The system shall verify that the entered email corresponds to a registered account.
- The system shall verify that the entered password matches the securely hashed password stored in the database.
- The system shall generate and return a JSON Web Token (JWT) upon successful authentication.
- The system shall deny login attempts and return an appropriate error message when verification fails.

3.1.2.2 Location & Proximity Calculation (O: Ahmer)

- The system shall calculate user-to-user distance using the Haversine formula based on GPS coordinates.
- The system shall refresh a user's stored location at least once every 60 seconds while the app is active.
- The system shall scatter nearby user markers within ±0.001° latitude/longitude to avoid revealing exact positions.
- The system shall treat users as "nearby" when they fall within a radius of 1000 feet of the current user.

3.1.2.3 Matchmaking (O: Taran, M1: Ahmer)

- The system shall retrieve all active users whose last known location is within 1000 feet of the current user's location.
- The system shall compute a match score for each nearby user based on the ratio of shared interest tags to total interest tags.
- The system shall round each match score to the nearest whole percentage.
- The system shall assign a ranked order to nearby users based on their match scores.
- The system shall update the ranked list dynamically when a user adds or removes interest tags.
- The system shall display each nearby user with their match percentage and ranked position on the "People Nearby" screen.

3.1.2.4 Messaging (O: Geelani, M1: Ahmer, M2: Geelani, M3: Ahmer)

- The system shall create a new chat session when two users begin messaging for the first time.
- The system shall store all chat sessions in the MingleMap database, including timestamps, sender IDs, receiver IDs, and message content.

- The system shall store each sent message as a separate database record, including sender ID, receiver ID, session ID, message content, and timestamp.
- The system shall store each received message as a separate database record, ensuring full preservation of conversation history for reporting and administrative review.
- The system shall provide real-time message delivery, ensuring that messages sent under normal network conditions are delivered with an end-to-end latency not exceeding 200 milliseconds.
- The system shall display a visual confirmation to inform the sender that a message has been successfully delivered to the recipient.
- The system shall hide active chat sessions when either participating user is no longer within the proximity radius (1000 ft).
- The system shall enforce closed chat sessions when one or both participating users move outside the proximity radius.
    - A closed chat session is defined as a chat where proximity rules no longer permit continued messaging.
    - Closed sessions shall prevent users from sending new messages.
    - Closed sessions shall remain viewable only until the user navigates away; upon refresh, they shall be inaccessible.

- The system shall archive closed chat sessions so that message history remains available for:
    - Trust Score Adjustment (see Trust Score Adjustment section),
    - Report Processing (see Report Processing section),
    - Admin review and moderation.
- The system shall ensure that archived chat sessions cannot be modified by either user after closure.

3.1.2.5 Trust Score Adjustment (O: Ben, M1: Ahmer)

- The system shall initialize each new account with a trust score of 99.
- The system shall increase a user's trust score by +1 after 5 consecutive chat sessions without being reported.
- The system shall reduce trust score after verified reports using a fixed penalty rule.
- The system shall restrict values to a range of 0–100.
- The system shall propagate updated trust scores to all active clients within 5 seconds.
- The system shall allow admin to adjust a user's trust score when resolving reports (see Report Processing).

3.1.2.6 Report Processing (O: Jacob, M1: Geelani, M2: Ahmer)

- The system shall allow administrators to access a report-management dashboard listing all submitted user reports, including reporter ID, reported user ID, timestamp, and report reason.
- The system shall retrieve report details from the MingleMap database within 500 milliseconds under normal server load.
- The system shall display the retrieved report details on the administrator dashboard immediately after the retrieval operation completes.
- The system shall allow administrators to update a report's status to **reviewed**, **in progress**, or **resolved**, and shall save the updated status in real time.
- The system shall restrict report visibility to authorized administrator accounts only, ensuring compliance with MingleMap's privacy and data-access policies.

3.2 Non-Functional Requirements

3.2.1 Performance (O: Ahmer)

- The system shall retrieve nearby users and render location markers within **2 seconds** after the map view loads on a stable network connection.

- The system shall process login and registration requests within **3 seconds** under normal server load.
- The system shall support **simultaneous connections from at least 100 concurrent users** without degraded response times beyond 5 seconds.
- The backend shall maintain an **uptime of 99% or higher** during testing and deployment phases.

3.2.2 Security (O: Ahmer)

- The system shall **encrypt all user passwords** using bcrypt or an equivalent one-way hashing algorithm.
- The system shall **transmit all data over HTTPS** using TLS 1.3 or higher.
- The system shall prevent unauthorized data access through JWT-based authentication and session expiration after 24 hours of inactivity.
- The system shall restrict profile visibility when a user disables "Visibility" mode, ensuring that hidden profiles are excluded from the Map and "Nearby" lists.

3.2.3 Usability (O: Ahmer, M1: Jacob)

- The system shall ensure that all primary user actions (viewing the map, messaging other users, toggling visibility, and reporting another user) are accessible within at most two taps from the main navigation.
- The user interface shall maintain consistent visual themes across Android and iOS platforms, achieving at least 90% WCAG AA compliance for text and icon contrast ratios.
- The system shall display confirmation prompts before completing any action defined as sensitive. Sensitive actions shall include:
    - **Deleting a message**
    - **Reporting a user**
    - **Changing visibility status**
    - **Replacing a profile picture**

3.2.4 Reliability and Availability (O: Ahmer, M1: Jacob)

- The system shall automatically reconnect to the API when a network interruption shorter than 30 seconds occurs.
- The system shall persist cached session data on the device so that authenticated users can re-enter the app without logging in again, unless their token has expired.

- The system shall maintain data integrity by verifying all API responses with proper status codes and error handling.

3.2.5 Maintainability and Scalability (O: Ahmer)

- Backend shall follow modular Node.js + Express + Prisma architecture.
- Codebase shall conform to ESLint + Prettier.
- The system shall support migration from Railway's PostgreSQL to any cloud-hosted PostgreSQL instance with minimal configuration changes.

3.3 Design Constraints

3.3.1 Platform and Frameworks (O: Ahmer)

- The system shall be developed using **React Native (Expo SDK 54)** for cross-platform mobile compatibility.
- The backend shall use **Node.js v18 or higher** with Express and Prisma ORM connected to **PostgreSQL 15 or later**.
- Cloud storage for profile pictures shall be managed via **Cloudinary API**, using secure environment variables.

3.3.2 Hardware and OS (O: Ahmer)

- The mobile application shall support Android 11+ and iOS 15+.

- The emulator testing environment shall mirror a 1080×2340 resolution and 6 GB RAM configuration.
- GPS accuracy shall meet a ±15 m margin when tested under open-sky conditions.

3.3.3 External Interface and Compatibility (O: Ahmer)

- The system shall conform to REST API standards using JSON for request and response bodies.
- All environment variables (API URL, Cloudinary key) shall be stored securely in `.env` files excluded from version control.
- The frontend shall be compatible with **Expo Go** and **Android emulator** environments without requiring native builds.

3.3.4 Regulatory and Ethical Constraints (O: Ahmer)

- The system shall comply with **FERPA-aligned privacy expectations** for student data at Old Dominion University.
- The system shall ensure that all stored data may be deleted upon user request, fulfilling data-erasure compliance.
- The system shall display a privacy disclaimer informing users about location sharing and data use before enabling map visibility.