UNLMTD - UI Mockup

Joyce Fu
Anisah Chowdhury
Nadir Ali
Luai Almaznai
Matthew Martinez

TABLE OF CONTENTS

1. UNLMTD - Data Definitions V2

1.1 User Account Data

Field Name	Type	Description
user_id	UUID/String	Unique identifier for each user
full_name	String	Full name for display and communication
email	String	Login and contact address
phone_number	String	For verification and optional notifications
role	Enum	[Athlete, Coach, Trainer, Therapist]
profile_settings	JSON	Includes preferences for notifications, privacy, language
age	Integer	Used for training thresholds and baseline metrics
height_cm	Float	User's height (for HR/VO2 estimations)
weight_kg	Float	Used in calorie calculations and workload estimates
fitness_level	Enum	[Beginner, Intermediate, Advanced, Pro]

1.2 Wearable Biometric Data

Field Name	Type	Description
timestamp	DateTime	Time of sensor recording
heart_rate_bpm	Integer	Beats per minute
hrv_ms	Float	Heart rate variability
hydration_level	Float	Sensor-derived or manually entered; % of recommended intake
muscle_fatigue_index	Float	AI-generated fatigue score from EMG or motion pattern analysis
step_count	Integer	Activity tracker value
body_temp_c	Float	Skin/body temperature (where available)
movement_pattern	Enum	[linear, erratic, repetitive, explosive]
device_id	UUID	Linked wearable device
session_id	UUID	Links data to a specific workout or event

1.3 Workout Session

Field Name	Type	Description
session_id	UUID	Unique ID for workout
start_time	DateTime	Workout start
end_time	DateTime	Workout end
workout_type	Enum	[Cardio, Strength, HIIT, Sport, Rehab, Custom]
intensity_level	Enum	[Low, Moderate, High, Max] or 1–10 user scale
calories_burned	Float	Estimated from HR + body weight
notes	Text	Optional trainer/athlete observations

1.4 AI Recommendations & Recovery Plans

Field Name	Type	Description
plan_id	UUID	Unique identifier for each plan
recommended_actions	JSON	Array of action strings like rest, hydrate, train light
plan_type	Enum	[Recovery, Training Boost, Rehab, Maintenance]
priority_level	Enum	[Low, Moderate, Critical]
duration_days	Integer	Number of days covered in plan
customized_by	Enum	["AI", "Coach", "Hybrid"]
hydration_target_ml	Float	Recommended water intake per day
sleep_target_hours	Float	Ideal hours of rest

1.5 Alerts & Notifications

Field Name	Type	Description
alert_id	UUID	Unique system-generated alert
alert_type	Enum	[Overtraining Risk, High Fatigue, Dehydration, HR Spike]
severity_level	Enum	[Low, Medium, High]
trigger_data_id	UUID	Reference to triggering session or metric
timestamp	DateTime	When alert was generated
resolved	Boolean	True if alert acknowledged by user

1.6 Progress Logs & Reports

Field Name	Type	Description
log_id	UUID	Unique progress entry
date_range	DateTime	Daily, weekly, or monthly reporting window
avg_fatigue	Float	Rolling average of fatigue readings
hydration_score	Float	% of hydration target met
adherence_score	Float	Recovery plan compliance score
coach_notes	Text	Optional human feedback
ai_summary	Text	Automated pattern/insight generation

2. Functional Requirements

2.1 AI-Driven Recommendations (Priority: 1)

- The AI engine must generate real-time recommendations for:
    - Recovery protocols
    - Workout modifications
    - Hydration reminders
- The AI must learn over time using past performance and user feedback.
- Users must be able to provide subjective input (e.g., perceived fatigue, mood) to refine AI suggestions.
- The system must personalize recommendations based on the user's goals and biometrics.

2.2 Wearable Device Integration (Priority: 1)

- The system must establish a secure Bluetooth connection with the UNLMTD wearable device.
- The system must ingest and process the following real-time biometric and motion data:
    - Muscle fatigue via EMG or motion sensors
    - Heart rate
    - Hydration levels (via sensor or manual input)
    - Movement patterns (e.g., acceleration, deceleration, direction changes)
- The system must normalize and store this data for use by the AI engine and analytics algorithm.

2.3 Real-Time Recovery Monitoring (Priority: 1)

- The system must analyze incoming biometric data in real time.
- The system must display a recovery score or status calculated from multiple signals.
- The system must issue alerts when fatigue, hydration, or exertion thresholds are exceeded.
- The system must adapt thresholds based on historical performance and personalized metrics.
- The system must personalize recommendations based on the user's goals and biometrics.

2.5 User Registration & Authentication (Priority: 2)

- The system must allow users to register using an email and/or phone number.

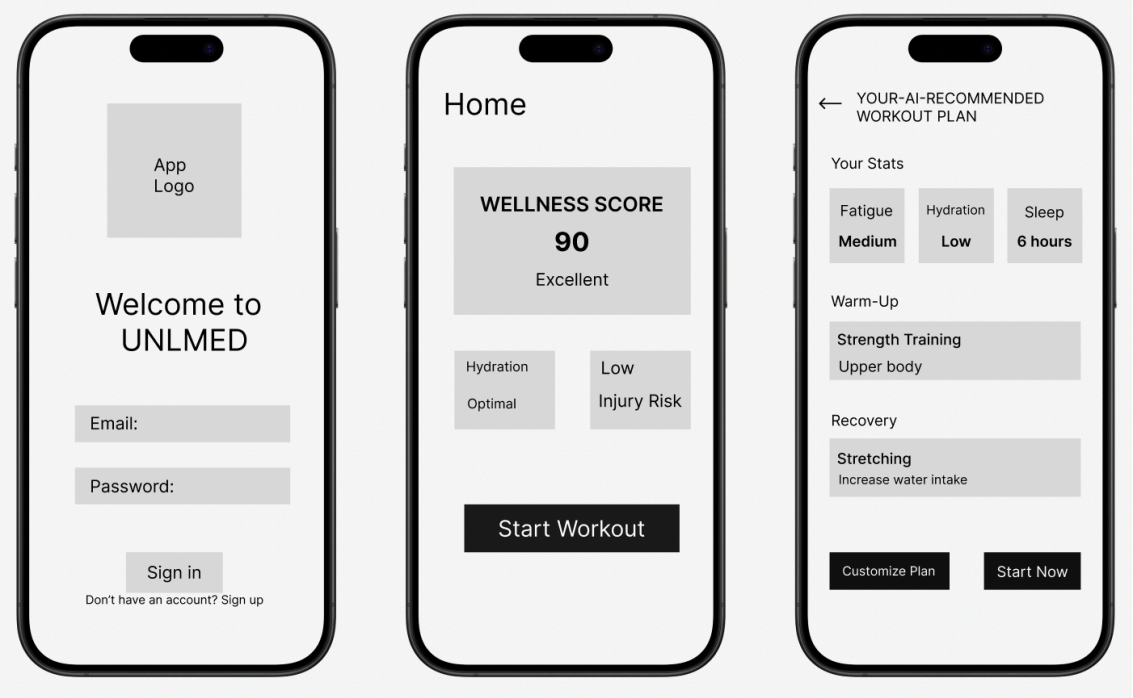
- The system must support secure login, logout, password resets, and session management.
- The system must support two-factor authentication.
- The system must store encrypted credentials using secure hashing algorithms (e.g., bcrypt).
- The system must manage and store basic profile data for each user (linked to wearable data).

2.5 User Dashboard (Priority: 2)

- The dashboard must display a visual summary of:
    - Recovery status
    - Fatigue level
    - HR/HRV trends
    - Hydration status
    - Users must be able to view historical data with filters (date ranges, metrics).
- The dashboard must be updated in near real-time as data is received from the wearable.

3. UI Mockups and UX Flows

- Mockup Tool: Figma



- User Story Flow
    a. **Login Screen (Welcome to UNLMTD)**
        - **Function**: Allows returning users to sign into their account.
        - **Elements**: App logo, email & password fields, sign-in button, and a sign-up link for new users.
        - **UX Goal**: Ensure quick, accessible entry into the app with clear form fields.

    b. **Home Dashboard**
        - **Function**: Displays a wellness summary with hydration level and injury risk.
        - **Elements**: Wellness Score (e.g., 90 - Excellent), Hydration status, Injury Risk indicator, and a "Start Workout" and more.
        - **UX Goal**: Clean overview of user wellness and encourage action via the Start Workout button.

    c. **Workout Plan Page**
        - **Function**: Delivers a tailored workout plan based on AI-analyzed stats (fatigue, hydration, sleep).

        - **Elements**: Your Stats section, Warm-Up and Recovery suggestions, and navigation buttons for plan customization or starting the workout.
        - **UX Goal**: Personalize the experience and enhance usability by guiding users to relevant actions based on their real-time condition.

- UX Principles Review

Principle	Implementation
Useful	Tracks real-time fatigue and recovery to prevent injuries and improve performance.
Usable	Simple app interface with clear feedback and minimal steps to view insights.
Desirable	Clean design, motivating insights, and visual progress tracking to engage users.
Accessible	Supports large fonts, color contrast, and voice-friendly summaries.
Findable	Key data (fatigue, hydration, readiness) is shown on the main screen/dashboard.
Credible	AI feedback is backed by real data and coach-approved recovery science.

- Tool Used
    - **Figma** for wireframe design.
    - **iPhone Frame Templates** for mobile responsiveness

4. High Level Architecture & Database Organization

4.1 Database Organization: MongoDB (noSQL)

users	{
	"user_id": "String",
	"full_name": "String",
	"email": "String",
	"phone_number": "String",
	"role": "String",
	"profile_settings": {
	"notifications": "Boolean",
	"language": "String",
	"privacy": "String"
	},
	"age": "Number",
	"height_cm": "Number",
	"weight_kg": "Number",
	"fitness_level": "String"
	}

biometric_data	{ "timestamp": "Date", "heart_rate_bpm": "Number", "hrv_ms": "Number", "hydration_level": "Number", "muscle_fatigue_index": "Number",

	"step_count": "Number", "body_temp_c": "Number", "movement_pattern": "String", "device_id": "String", "session_id": "String", "user_id": "String" }

workout_sessions	{
	"session_id": "String",
	"user_id": "String",
	"start_time": "Date",
	"end_time": "Date",
	"workout_type": "String",
	"intensity_level": "String",
	"calories_burned": "Number",
	"notes": "String"
	}

recovery_plans	{ "plan_id": "String", "user_id": "String", "recommended_actions": ["String"], "plan_type": "String", "priority_level": "String", "duration_days": "Number",

	"customized_by": "String", "hydration_target_ml": "Number", "sleep_target_hours": "Number" }

alerts	{
	"alert_id": "String",
	"user_id": "String",
	"alert_type": "String",
	"severity_level": "String",
	"trigger_data_id": "String",
	"timestamp": "Date",
	"resolved": "Boolean"
	}

progress_logs	{
	"log_id": "String",
	"user_id": "String",
	"date_range": {
	"start": "Date",
	"end": "Date"
	},
	"avg_fatigue": "Number",
	"hydration_score": "Number",
	"adherence_score": "Number",
	"coach_notes": "String",

	"ai_summary": "String" }

4.2 Add/Delete/Search Architecture

users

- Add: When a new user registers.
- Search: By user_id, email, or role.
- Update: User profile info, settings, or password.
- Delete: If a user deletes their account.

biometric_data

- Add: Real-time sensor data is logged automatically.
- Search: By user_id, timestamp, session_id, or movement_pattern.
- Delete: Outdated or erroneous data (admin only).
- Display: On the user dashboard for real-time insights.

workout_sessions

- Add: Each time a workout session starts and ends.
- Search: By user_id, session_id, date, or workout_type.
- Update: Notes or calories if recalculated.
- Delete: Invalid sessions (optional).

recovery_plans

- Add: After each AI recommendation or coach-generated plan.
- Search: By user_id, plan_type, priority_level.
- Update: If user/coach modifies the plan.
- Delete: Expired or replaced plans.

alerts

- Add: Triggered by threshold crossings (e.g. high fatigue).
- Search: By user_id, alert_type, resolved.
- Update: To mark as resolved.
- Delete: Optional cleanup or retention policy.

progress_logs

- Add: Automatically or by user/coaches over a date range.
- Search: By user_id, date_range, or adherence_score.
- Update: Notes or AI summaries may be edited.

- Delete: Old logs (based on data retention rules).

4.3 Backend endpoint API's

/api/users

- POST /api/users/register — Create new user account.
- POST /api/users/login — Log in with email/password.
- GET /api/users/:id — Get user profile by ID.
- PUT /api/users/:id — Update user settings or personal info.
- DELETE /api/users/:id — Delete user account.

/api/biometric-data

- POST /api/biometric-data — Log new biometric reading.
- GET /api/biometric-data/:userId — Fetch data for a user.
- GET /api/biometric-data/session/:sessionId — Fetch data by workout session.

/api/workout-sessions

- POST /api/workout-sessions — Log a new session.
- GET /api/workout-sessions/:userId — Get all sessions for user.
- PUT /api/workout-sessions/:sessionId — Update notes or metrics.
- DELETE /api/workout-sessions/:sessionId — Delete session.

/api/recovery-plans

- POST /api/recovery-plans — Create a new recovery plan.
- GET /api/recovery-plans/:userId — Get all plans for a user.
- PUT /api/recovery-plans/:planId — Update a plan.
- DELETE /api/recovery-plans/:planId — Delete a plan.

/api/alerts

- POST /api/alerts — Create an alert.
- GET /api/alerts/:userId — View user alerts.
- PUT /api/alerts/:alertId — Mark alert as resolved.

/api/progress-logs

- POST /api/progress-logs — Add new log.
- GET /api/progress-logs/:userId — Fetch logs for user.
- PUT /api/progress-logs/:logId — Update notes or summary.
- DELETE /api/progress-logs/:logId — Delete old log.

5. Major AI Functions – UNLMTD

UNLMTD integrates multiple AI-driven features to deliver real-time insights, personalized recovery guidance, and predictive alerts. These functions are central to the product's competitive value and user experience.

5.1 Recovery Score Generation

- **Function:** Calculate a **daily recovery readiness score** (0.0 to 1.0) based on biometric inputs like HRV, fatigue level, sleep, hydration, and activity volume.
- **Input Data:**
    - Heart rate variability (HRV)
    - Sleep duration
    - Muscle fatigue level
    - Hydration status
    - Workout intensity (from session data)
- **Output:**
    - Numeric score (recovery readiness)
    - Status label (e.g., "Ready", "Moderate", "Recovering")
- **AI Technique:** Weighted regression model using multi-day moving averages + threshold logic

5.2 Fatigue Risk Classification

- **Function:** Detect signs of **overtraining or muscle overuse** in real time using biometric and movement pattern data.
- **Input Data:**
    - EMG signals (via muscle fatigue index)
    - Recent workout volume
    - Movement patterns (irregular/repetitive)
    - HR spikes and variability
- **Output:**
    - Binary or risk-scaled flag (low, moderate, high)

    - Alert: “High Fatigue Risk – Recommended Rest”
- **AI Technique:** Supervised classification model trained on labeled fatigue cases.

5.3 Personalized Recovery Recommendations

- **Function:** Recommend **next steps** (rest, light activity, hydration, etc.) based on the user’s recovery and performance profile.
- **Input Data:**
    - Daily recovery score
    - Fatigue risk level
    - Adherence to past recommendations
- **Output:**
    - Text-based recommendation (e.g., “Light cardio & mobility today”)
    - Optional hydration and sleep targets
- **AI Technique:** Rules-based engine enhanced with collaborative filtering (learns from similar user responses)

5.4 Adaptive Recovery Plan Generator

- **Function:** Create and adjust multi-day recovery or performance plans using user data and ongoing response.
- **Input Data:**
    - Historical performance trends
    - User role (athlete, casual, recovering)
    - Past injuries or soreness reports
- **Output:**
    - Multi-day plan (stored in recommended_activities JSON)
    - Triggers auto-adjustments based on adherence or fatigue events
- **AI Technique:** Reinforcement logic with rule-based adaptation; integrates user feedback loops.

5.5 Alert Prioritization & Notification AI

- **Function:** Automatically determine **when and how** to notify users or their coach about significant recovery issues.
- **Input Data:**

    - All biometric + workout trends
    - AI-generated risk tags
    - Notification settings/preferences
- **Output:**
    - Push or in-app alerts (e.g., “High fatigue risk – take a rest day”)
    - Coach alerts for shared profiles
- **AI Technique:** Decision tree w/ ranking logic; sensitivity tuned per user.

Summary Table

AI Function	Purpose	Primary Input	Output
Recovery Score Generation	Estimate readiness to train	HRV, fatigue, sleep, hydration	Numeric score and label
Fatigue Risk Classification	Detect overtraining risk	EMG, HR, movement patterns	Risk level and alert
Personalized Recovery Recs	Suggest daily recovery actions	Recovery score, risk level	Rest, activity, hydration suggestions
Adaptive Recovery Plan Generator	Build/adjust recovery plans	Historical data, fatigue trends	3-7 day AI-generated plan
Alert Prioritization Engine	Smart alerting	All biometric and settings	Notification delivery to user/coach