

MealMap - Requirements document

Functional requirements

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
FR-01	Authentication & Profiles (Java Web Tokens)	Secure sign-up/login with JWT; user profile stores optional kcal/macro targets.	<ul style="list-style-type: none">- Email/password sign-up, login, logout.- JWT access (≤ 15 min) + refresh (≤ 7 days) with rotation & revoke on logout.- Password reset via email link.- View/update profile (kcal/macros optional).- Delete account (hard delete of user data).	Core security & personalization; prerequisite for all user data.
FR-02	Recipe Library	Maintain a library of reusable recipes that contain only ingredients + measurements , plus an optional external link (no instructions/notes stored).	<ul style="list-style-type: none">- Recipe fields: name (required) and link (optional).- Ingredients per recipe: ingredient ref, quantity (metric g/ml), unit, package note (e.g., "1×400 g can").- Create/read/update/delete recipes.- Prevent duplicates via name + ingredient set suggestion (non-blocking).	Minimal data model aligns with your preference; enables reuse without bloating storage.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
FR-03	Weekly Planner	Plan meals on a Mon–Sun grid with breakfast, morning snack, lunch, afternoon snack, dinner slots; insert items from Recipe Library or ad-hoc ingredients.	<ul style="list-style-type: none"> - Week view (Mon–Sun), breakfast, morning snack, lunch, afternoon snack, dinner per day. - Drag & drop saved recipes into slots. - For each placement: allow a portion multiplier (scales ingredient quantities at plan-time). - Move/remove items; persist per user; duplicate last week. 	Primary flow; connecting library to planning while keeping servings logic in the planner.
FR-04	Smart Grocery List	Generate a category-grouped , de-duplicated list from the week’s plan; subtract pantry quantities (optional) .	<ul style="list-style-type: none"> - Aggregate all planned items (including scaled recipes). - Group by category . - De-duplicate + sum quantities in metric units. - Subtract pantry stock automatically (optional). - Check/uncheck items; persist list state. 	Efficient shopping and waste reduction.
FR-05	N-Trip Split	One-click split of the list into desired ranges.	<ul style="list-style-type: none"> - Generate N lists by date range. - Maintain separate check states per list. - Show per-list totals. 	Matches real shopping cadence; improves usability.
FR-06	Pantry Basics	Track staple inventory in metric units; impacts list computation.	<ul style="list-style-type: none"> - Add/edit/remove pantry items with quantities in g/ml or package count. - Bulk adjust after shopping. - Pantry affects list subtraction. 	Accurate lists rely on accurate pantry.
FR-07	Exports	Export lists to CSV and PDF with clear date ranges.	<ul style="list-style-type: none"> - CSV/PDF export for each list. - Filenames include date range (e.g., MealMap_Sun-Wed_2025-09-14.pdf). 	Offline/phone-friendly; easy sharing/printing.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
FR-08	Privacy & Data Control	Users can export or delete their data; clear privacy notice.	<ul style="list-style-type: none"> - Data export (JSON/ZIP). - Account deletion self-service. - Privacy notice & consent for optional features (e.g., OCR). 	Compliance & user trust.
FR-09	Plan Templates	Save a complete weekly plan as a template and reuse it later.	<ul style="list-style-type: none"> - Save current plan → named template. - Apply template to any target week (with date shift). - Template library: list, rename, delete. - When applying, allow per-meal portion multipliers. 	Speeds up recurring weeks; complements recipe reuse.
FR-10 (Optional)	Nutrition Hints	Show approximate kcal/macros per recipe/plan using local/public data.	<ul style="list-style-type: none"> - Lookup from local/public dataset (no cloud AI). - Display per-recipe & daily totals. - Show approximation disclaimer. 	Lightweight insight without feature bloat.
FR-11 (Optional)	Recipe Import (URL)	Extract recipe title/ingredients from a URL with fallback to manual paste.	<ul style="list-style-type: none"> - Server-side best-effort extractor (title + ingredient lines). - If fail, prefill paste form for user review. - On save, store only ingredients + measurements and optionally link. 	Faster content entry while honoring minimal recipe storage.
FR-12 (Optional, AI)	Meal-Plan Optimization	Solver-based plan that fits macros ($\pm 10\%$), budget, and variety; fallback to greedy.	<ul style="list-style-type: none"> - OR-Tools CP-SAT with soft macro constraints ($\pm 10\%$), budget 2nd, variety 3rd. - Timeout 2.5 s → greedy fallback. - Feature flag: AI_OPTIMIZER. 	High utility when enabled; deterministic and fast.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
FR-13 (Optional, AI)	Ingredient Normalization & Unit Conversion	Convert free-text lines to canonical ingredient + metric units.	<ul style="list-style-type: none"> - RapidFuzz + rule-based parsing; Pint for units. - Package awareness (e.g., 2×400 g). - Flag low-confidence lines for review. - Feature flag: AI_NORMALIZE. 	Cleaner data → better lists; optional to keep MVP lean.
FR-14 (Optional, AI)	Pantry-Aware Substitutions	Suggest swaps that use pantry items while staying within macro tolerance.	<ul style="list-style-type: none"> - Local embeddings + category + macro tolerance. - Show reason & confidence. - Feature flag: AI_SUBSTITUTIONS. 	Reduces cost/waste when enabled.
FR-15 (Optional, AI)	Barcode → Open Food Facts	Add pantry items via barcode lookup against Open Food Facts.	<ul style="list-style-type: none"> - ZXing barcode scanning (web/mobile upload). - Lookup Open Food Facts only; store nutrition & package size. - No barcode images stored. - Feature flag: BARCODES. 	Fast, accurate pantry entry; privacy-respecting.
FR-16 (Optional, AI)	Receipt OCR → Auto-Pantry	Parse receipts to update pantry; images deleted immediately after parsing.	<ul style="list-style-type: none"> - Tesseract/OpenCV preprocessing. - Parse lines → match to ingredients; user confirm. - Immediate image deletion after parse. - Feature flag: OCR. 	Convenience with strict privacy controls.
FR-17 (Optional, AI)	Natural-Language → Constraints	Parse text prompts into structured constraints for planning.	<ul style="list-style-type: none"> - Deterministic parser (regex/Lark) → JSON. - Optional local LLM fallback in dev only (AI_NLP=ollama), ** off in prod**. - Prefilled form for edits. 	Friendly UX; remains optional and transparent.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
FR-18	Profile Types & Permissions	Support three user profile types with simple, clear rights.	<ul style="list-style-type: none"> - Personal: full control on own resources; can create/revoke share links. - Household: view/edit shared plans/lists/pantry; cannot change sharing settings; cannot delete owner data. - Coach/Nutritionist: read-only to plans/lists; (optional) comment-only; no pantry edits. - Enforce server-side authorization on every API and WebSocket destination. 	Enables collaboration while keeping ownership and boundaries clear.
FR-19	Ad-Hoc Sharing via Invite Links	Share a specific plan or list via a time-limited invite link with scoped permissions.	<ul style="list-style-type: none"> - Create link with role (Viewer/Editor) and expiry (1 day/7 days/custom). - Links are signed tokens scoped to resource ID and role; revocable at any time. - Accepting a link binds the resource to the invitee's account; email confirmation supported. - Show active shares and revoke individually. 	Lightweight collaboration without global "household" accounts or extra friction.
FR-20	Real-Time Planner Co-Editing	Multiple users co-edit a weekly planner with item-level optimistic merge .	<ul style="list-style-type: none"> - WebSocket/STOMP channels per plan; connect with JWT. - Operations are item-scoped (add/move/remove/scale). - Last-write-wins per item with server-side versioning; client receives authoritative state diffs. - Soft "editing" indicator on slots being modified. 	Smooth collaboration with minimal conflict; keeps latency low and UX predictable.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
FR-21	Real-Time Grocery List Sync	Keep list items in sync across users instantly.	<ul style="list-style-type: none"> - Item-level ops (check/uncheck, qty change, add/remove). - Debounced local updates; server confirms with versioned patch. - Merge conflicts resolved last-write-wins; toast shows if user was overwritten. - Split lists (Sun–Wed / Wed–Sun) maintain separate channels/states. 	Shopping is often shared; instant updates prevent duplicates and confusion.
FR-22	Presence & Editing Indicators	Show who is online and who is editing at a glance.	<ul style="list-style-type: none"> - Presence per resource via WS heartbeats; avatar stack of online participants. - Show “X editing this item” chip on planner slots/list items. - Privacy: show display name and avatar only; no typing content shared. 	Builds trust and reduces accidental conflicts.
FR-23	Notifications (In-App & Email)	Notify participants of important collaboration events.	<ul style="list-style-type: none"> - In-app toasts: user joined/left, item edited, list finalized, share revoked. - Email: invite issued/accepted, plan assigned, list finalized; respect email opt-out. - Notification center stores last 20 events per user (14-day retention). 	Keeps everyone aligned without requiring constant presence.
FR-24	Activity Log (Per Resource)	Maintain a simple, exportable change log for each plan/list.	<ul style="list-style-type: none"> - Entries: timestamp, actor, action, target (e.g., “Anna checked ‘400 g tomatoes’”). - Retain 30 days by default; include in plan export (JSON). - No content analytics; purely operational history. 	Transparency, debugging, and accountability for shared resources.

Non functional requirements

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
NFR-01	Security & Privacy	Protect data in transit/at rest; minimize/erase sensitive data; comply with GDPR.	<ul style="list-style-type: none"> - Password hashing BCrypt. - TLS 1.2+ end-to-end. - JWT access ≤ 15 min, refresh ≤ 7 days; rotation & revoke. - Server-side validation (Bean Validation); client schemas (Zod). - OWASP Top-10 mitigations; rate-limit auth endpoints. - No PII in logs; redact tokens; structured logs only. - OCR images deleted immediately; external calls only to Open Food Facts; data export/delete self-service. 	Meets course standards & user trust; reduces breach/abuse risk.
NFR-01A	Collaboration Security	Harden sharing and sockets.	<ul style="list-style-type: none"> - WS connects only with valid JWT; re-auth on refresh/expiry. - Signed invite tokens (short TTL; single-resource scope; one-time redeem flag optional). - Per-destination authorization (Spring Security + interceptors). - Rate-limit WS subscribe/send per user to prevent abuse. - Email invites include resource name and permissions; warn on external sharing. 	Prevents token abuse and unauthorized access.
NFR-02	Performance	Keep the app responsive under normal use.	<ul style="list-style-type: none"> - P95 API/page latency ≤ 800 ms (authenticated, warm). - Grocery list generation from saved plan: P95 ≤ 3.0 s, typical ≤ 1.5 s. - N-trip split ≤ 800 ms. - Open Food Facts barcode lookup typical 100–400 ms, P95 ≤ 800 ms. - Optimizer typical ≤ 1.0 s, timeout 2.5 s \rightarrow fallback. - Normalization per line P95 ≤ 300 ms. 	Ensures snappy UX; aligns with < 3 s overall suggestion target.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
NFR-02A	Real-Time Performance	Set strict latency targets for collaboration features.	<ul style="list-style-type: none"> - P95 WS round-trip ≤ 300 ms within EU; P99 ≤ 800 ms. - Presence updates every 10 s; missed 2 heartbeats → mark offline. - Co-edit ops batched to ≤ 10 ops/sec/user; server fanout completes in ≤ 200 ms typical. 	Feels instantaneous; aligns with your <3s overall budget for heavy ops.
NFR-03	Availability & Reliability	Reasonable uptime for a student project; fail safely.	<ul style="list-style-type: none"> - Uptime $\geq 99.0\%$ monthly (Azure Web App + Static Web Apps). - Error budget: $\leq 1\%$ 5xx monthly. - Graceful degradation: AI features can be toggled off without breaking core flows. - Safe retries/backoff on transient errors. 	Practical Service Level Objectives for portfolio-grade reliability.
NFR-03A	Reliability for WS	Ensure stable WS behavior and graceful fallbacks.	<ul style="list-style-type: none"> - Automatic reconnect with jittered backoff; show "Reconnecting..." banner. - If WS unavailable, read-only view; edits queue or are disabled with clear messaging. - Server maintains version counter per resource; clients reconcile on reconnect. 	Real-world connectivity is flaky; make failures visible and safe.
NFR-04	Data Integrity & Persistence	Durable storage and recoverability.	<ul style="list-style-type: none"> - Microsoft SQL. - Migrations via Flyway; no destructive changes without backups. 	Prevents data loss; enables controlled schema evolution.
NFR-05	Usability & Accessibility	Minimal, desktop-first; list is mobile-friendly; accessible core flows.	<ul style="list-style-type: none"> - WCAG 2.1 AA basics (labels, focus order, contrast, keyboard nav). - New user can generate a list in ≤ 5 minutes without docs. - Clear empty states and inline validation. 	Low friction; inclusive experience; demonstrable UX quality.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
NFR-06	Compatibility	Work on modern browsers; timezone-aware.	<ul style="list-style-type: none"> - Latest Chrome/Edge, Firefox, Safari; mobile Safari/Chrome for list view. - Europe/Amsterdam default; ISO week handling; 24-hour time. 	Realistic browser support for target users.
NFR-07	Maintainability & Quality	Keep code clean, tested, and documented.	<ul style="list-style-type: none"> - OpenAPI docs; README run/deploy steps; C4 diagrams (Context/Container/Component). - Static analysis: SpotBugs/Checkstyle (Java), ESLint/TS; fail build on high severity. - Tests: backend unit+integration (Testcontainers); frontend unit+E2E (Playwright). - Coverage floors: backend $\geq 70\%$, frontend $\geq 60\%$ (lines). - No function >200 LOC; lint rules enforce. 	Easier to extend/debug; portfolio credibility.
NFR-07A	Testability (Real-Time)	Add focused tests for WS flows.	<ul style="list-style-type: none"> - Contract tests for WS destinations (connect/auth/subscribe/publish). - Simulated multi-client integration tests for merge logic and ordering. - Load test WS fanout to ≥ 200 concurrent clients per resource in CI-lite. 	Prevent regressions; prove behavior under basic load.
NFR-08	Observability	See system health without tracking users.	<ul style="list-style-type: none"> - Basic system metrics: CPU/mem, request rate/latency, error counts. - JSON logs with correlation IDs; retention ≤ 14 days. - Optional alert: error rate >1% or health check failures → email. 	Operability without privacy trade-offs.
NFR-09	Internationalization & Units	Keep scope focused; use metric.	<ul style="list-style-type: none"> - English-only UI/content. - Metric units (g/ml) with package awareness by default. 	Scope control; consistency for NL/EU context.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
NFR-09A	Observability (WS)	Minimal signals to operate WS without user tracking.	<ul style="list-style-type: none">- Metrics: active connections, subscribes/sec, send/receive rates, WS error counts.- Logs include correlation IDs and resource IDs; no PII.- Alert when WS error rate > 1% or connections drop > 30% in 5 min.	Operability without analytics or privacy trade-offs.