

# 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"><li>- Email/password sign-up, login, logout.</li><li>- JWT access (<math>\leq 15</math> min) + refresh (<math>\leq 7</math> days) with rotation &amp; revoke on logout.</li><li>- Password reset via email link.</li><li>- View/update profile (kcal/macros optional).</li><li>- Delete account (hard delete of user data).</li></ul>	Core security & personalization; prerequisite for all user data.
FR-02	Recipe Library	Maintain a library of <b>reusable recipes</b> that contain only <b>ingredients + measurements</b> , plus an <b>optional external link</b> (no instructions/notes stored).	<ul style="list-style-type: none"><li>- Recipe fields: <b>name (required)</b> and <b>link (optional)</b>.</li><li>- Ingredients per recipe: ingredient ref, quantity (metric g/ml), unit, package note (e.g., "1×400 g can").</li><li>- Create/read/update/delete recipes.</li><li>- Prevent duplicates via name + ingredient set suggestion (non-blocking).</li></ul>	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 <b>Recipe Library</b> or ad-hoc ingredients.	<ul style="list-style-type: none"> <li>- Week view (Mon–Sun), breakfast, morning snack, lunch, afternoon snack, dinner per day.</li> <li>- Drag &amp; drop <b>saved recipes</b> into slots.</li> <li>- For each placement: allow a <b>portion multiplier</b> (scales ingredient quantities at plan-time).</li> <li>- Move/remove items; persist per user; duplicate last week.</li> </ul>	Primary flow; connecting library to planning while keeping servings logic in the planner.
FR-04	Smart Grocery List	Generate a <b>category-grouped</b> , de-duplicated list from the week’s plan; subtract pantry quantities (optional) .	<ul style="list-style-type: none"> <li>- Aggregate all planned items (including scaled recipes).</li> <li>- Group by <b>category</b> .</li> <li>- De-duplicate + sum quantities in <b>metric</b> units.</li> <li>- Subtract pantry stock automatically (optional).</li> <li>- Check/uncheck items; persist list state.</li> </ul>	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"> <li>- Generate N lists by date range.</li> <li>- Maintain separate check states per list.</li> <li>- Show per-list totals.</li> </ul>	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"> <li>- Add/edit/remove pantry items with quantities in g/ml or package count.</li> <li>- Bulk adjust after shopping.</li> <li>- Pantry affects list subtraction.</li> </ul>	Accurate lists rely on accurate pantry.
FR-07	Exports	Export lists to <b>CSV</b> and <b>PDF</b> with clear date ranges.	<ul style="list-style-type: none"> <li>- CSV/PDF export for each list.</li> <li>- Filenames include date range (e.g., <b>MealMap_Sun-Wed_2025-09-14.pdf</b>).</li> </ul>	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"> <li>- Data export (JSON/ZIP).</li> <li>- Account deletion self-service.</li> <li>- Privacy notice &amp; consent for optional features (e.g., OCR).</li> </ul>	Compliance & user trust.
FR-09	Plan Templates	Save a complete weekly plan as a <b>template</b> and <b>reuse</b> it later.	<ul style="list-style-type: none"> <li>- Save current plan → named template.</li> <li>- Apply template to any target week (with date shift).</li> <li>- Template library: list, rename, delete.</li> <li>- When applying, allow per-meal portion multipliers.</li> </ul>	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"> <li>- Lookup from local/public dataset (no cloud AI).</li> <li>- Display per-recipe &amp; daily totals.</li> <li>- Show approximation disclaimer.</li> </ul>	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"> <li>- Server-side best-effort extractor (title + ingredient lines).</li> <li>- If fail, prefill paste form for user review.</li> <li>- On save, store only <b>ingredients + measurements</b> and <b>optionally link</b>.</li> </ul>	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"> <li>- OR-Tools CP-SAT with soft macro constraints (<math>\pm 10\%</math>), budget 2nd, variety 3rd.</li> <li>- Timeout 2.5 s → greedy fallback.</li> <li>- Feature flag: <b>AI_OPTIMIZER</b>.</li> </ul>	High utility when enabled; deterministic and fast.

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
<b>FR-13</b> <b>(Optional, AI)</b>	Ingredient Normalization & Unit Conversion	Convert free-text lines to canonical ingredient + metric units.	<ul style="list-style-type: none"> <li>- RapidFuzz + rule-based parsing; <b>Pint</b> for units.</li> <li>- Package awareness (e.g., 2×400 g).</li> <li>- Flag low-confidence lines for review.</li> <li>- Feature flag: <b>AI_NORMALIZE</b>.</li> </ul>	Cleaner data → better lists; optional to keep MVP lean.
<b>FR-14</b> <b>(Optional, AI)</b>	Pantry-Aware Substitutions	Suggest swaps that use pantry items while staying within macro tolerance.	<ul style="list-style-type: none"> <li>- Local embeddings + category + macro tolerance.</li> <li>- Show reason &amp; confidence.</li> <li>- Feature flag: <b>AI_SUBSTITUTIONS</b>.</li> </ul>	Reduces cost/waste when enabled.
<b>FR-15</b> <b>(Optional, AI)</b>	Barcode → Open Food Facts	Add pantry items via barcode lookup against Open Food Facts.	<ul style="list-style-type: none"> <li>- ZXing barcode scanning (web/mobile upload).</li> <li>- Lookup <b>Open Food Facts only</b>; store nutrition &amp; package size.</li> <li>- No barcode images stored.</li> <li>- Feature flag: <b>BARCODES</b>.</li> </ul>	Fast, accurate pantry entry; privacy-respecting.
<b>FR-16</b> <b>(Optional, AI)</b>	Receipt OCR → Auto-Pantry	Parse receipts to update pantry; images deleted immediately after parsing.	<ul style="list-style-type: none"> <li>- Tesseract/OpenCV preprocessing.</li> <li>- Parse lines → match to ingredients; user confirm.</li> <li>- <b>Immediate image deletion</b> after parse.</li> <li>- Feature flag: <b>OCR</b>.</li> </ul>	Convenience with strict privacy controls.
<b>FR-17</b> <b>(Optional, AI)</b>	Natural-Language → Constraints	Parse text prompts into structured constraints for planning.	<ul style="list-style-type: none"> <li>- Deterministic parser (regex/Lark) → JSON.</li> <li>- Optional local LLM fallback in dev only (<b>AI_NLP=ollama</b>), ** off in prod**.</li> <li>- Prefilled form for edits.</li> </ul>	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"> <li>- <b>Personal:</b> full control on own resources; can create/revoke share links.</li> <li>- <b>Household:</b> view/edit shared plans/lists/pantry; cannot change sharing settings; cannot delete owner data.</li> <li>- <b>Coach/Nutritionist:</b> read-only to plans/lists; (optional) comment-only; no pantry edits.</li> <li>- Enforce server-side authorization on every API and WebSocket destination.</li> </ul>	Enables collaboration while keeping ownership and boundaries clear.
FR-19	Ad-Hoc Sharing via Invite Links	Share a <b>specific plan or list</b> via a time-limited invite link with scoped permissions.	<ul style="list-style-type: none"> <li>- Create link with <b>role (Viewer/Editor)</b> and <b>expiry</b> (1 day/7 days/custom).</li> <li>- Links are <b>signed tokens</b> scoped to resource ID and role; revocable at any time.</li> <li>- Accepting a link binds the resource to the invitee's account; email confirmation supported.</li> <li>- Show active shares and revoke individually.</li> </ul>	Lightweight collaboration without global "household" accounts or extra friction.
FR-20	Real-Time Planner Co-Editing	Multiple users co-edit a weekly planner with <b>item-level optimistic merge</b> .	<ul style="list-style-type: none"> <li>- WebSocket/STOMP channels per plan; connect with JWT.</li> <li>- Operations are item-scoped (add/move/remove/scale).</li> <li>- <b>Last-write-wins per item</b> with server-side versioning; client receives authoritative state diffs.</li> <li>- Soft "editing" indicator on slots being modified.</li> </ul>	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"> <li>- Item-level ops (check/uncheck, qty change, add/remove).</li> <li>- <b>Debounced</b> local updates; server confirms with versioned patch.</li> <li>- Merge conflicts resolved last-write-wins; toast shows if user was overwritten.</li> <li>- Split lists (Sun–Wed / Wed–Sun) maintain separate channels/states.</li> </ul>	Shopping is often shared; instant updates prevent duplicates and confusion.
FR-22	Presence & Editing Indicators	Show <b>who is online</b> and <b>who is editing</b> at a glance.	<ul style="list-style-type: none"> <li>- Presence per resource via WS heartbeats; avatar stack of online participants.</li> <li>- Show “X editing this item” chip on planner slots/list items.</li> <li>- Privacy: show display name and avatar only; no typing content shared.</li> </ul>	Builds trust and reduces accidental conflicts.
FR-23	Notifications (In-App & Email)	Notify participants of important collaboration events.	<ul style="list-style-type: none"> <li>- In-app toasts: user joined/left, item edited, list finalized, share revoked.</li> <li>- Email: invite issued/accepted, plan assigned, list finalized; respect email opt-out.</li> <li>- Notification center stores last 20 events per user (14-day retention).</li> </ul>	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"> <li>- Entries: timestamp, actor, action, target (e.g., “Anna checked ‘400 g tomatoes’”).</li> <li>- Retain 30 days by default; include in plan export (JSON).</li> <li>- No content analytics; purely operational history.</li> </ul>	Transparency, debugging, and accountability for shared resources.

## Non functional requirements

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

Requirement ID	Requirement Name	Requirement Description	Sub-Requirements	Reasoning
<b>NFR-02A</b>	Real-Time Performance	Set strict latency targets for collaboration features.	<ul style="list-style-type: none"> <li>- <b>P95 WS round-trip <math>\leq 300</math> ms</b> within EU; <b>P99 <math>\leq 800</math> ms</b>.</li> <li>- Presence updates every 10 s; missed 2 heartbeats → mark offline.</li> <li>- Co-edit ops batched to <math>\leq 10</math> ops/sec/user; server fanout completes in <b><math>\leq 200</math> ms</b> typical.</li> </ul>	Feels instantaneous; aligns with your <3s overall budget for heavy ops.
<b>NFR-03</b>	Availability & Reliability	Reasonable uptime for a student project; fail safely.	<ul style="list-style-type: none"> <li>- <b>Uptime <math>\geq 99.0\%</math> monthly</b> (Azure Web App + Static Web Apps).</li> <li>- Error budget: <b><math>\leq 1\%</math> 5xx</b> monthly.</li> <li>- Graceful degradation: AI features can be toggled off without breaking core flows.</li> <li>- Safe retries/backoff on transient errors.</li> </ul>	Practical Service Level Objectives for portfolio-grade reliability.
<b>NFR-03A</b>	Reliability for WS	Ensure stable WS behavior and graceful fallbacks.	<ul style="list-style-type: none"> <li>- Automatic <b>reconnect with jittered backoff</b>; show "Reconnecting..." banner.</li> <li>- If WS unavailable, <b>read-only</b> view; edits queue or are disabled with clear messaging.</li> <li>- Server maintains <b>version counter</b> per resource; clients reconcile on reconnect.</li> </ul>	Real-world connectivity is flaky; make failures visible and safe.
<b>NFR-04</b>	Data Integrity & Persistence	Durable storage and recoverability.	<ul style="list-style-type: none"> <li>- Microsoft SQL.</li> <li>- Migrations via Flyway; no destructive changes without backups.</li> </ul>	Prevents data loss; enables controlled schema evolution.
<b>NFR-05</b>	Usability & Accessibility	Minimal, desktop-first; list is mobile-friendly; accessible core flows.	<ul style="list-style-type: none"> <li>- WCAG 2.1 AA basics (labels, focus order, contrast, keyboard nav).</li> <li>- New user can generate a list in <b><math>\leq 5</math> minutes</b> without docs.</li> <li>- Clear empty states and inline validation.</li> </ul>	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"> <li>- Latest Chrome/Edge, Firefox, Safari; mobile Safari/Chrome for list view.</li> <li>- Europe/Amsterdam default; ISO week handling; 24-hour time.</li> </ul>	Realistic browser support for target users.
NFR-07	Maintainability & Quality	Keep code clean, tested, and documented.	<ul style="list-style-type: none"> <li>- OpenAPI docs; README run/deploy steps; C4 diagrams (Context/Container/Component).</li> <li>- Static analysis: SpotBugs/Checkstyle (Java), ESLint/TS; fail build on high severity.</li> <li>- Tests: backend unit+integration (Testcontainers); frontend unit+E2E (Playwright).</li> <li>- <b>Coverage floors:</b> backend <math>\geq 70\%</math>, frontend <math>\geq 60\%</math> (lines).</li> <li>- No function &gt;200 LOC; lint rules enforce.</li> </ul>	Easier to extend/debug; portfolio credibility.
NFR-07A	Testability (Real-Time)	Add focused tests for WS flows.	<ul style="list-style-type: none"> <li>- Contract tests for WS destinations (connect/auth/subscribe/publish).</li> <li>- Simulated multi-client integration tests for merge logic and ordering.</li> <li>- Load test WS fanout to <math>\geq 200</math> <b>concurrent clients</b> per resource in CI-lite.</li> </ul>	Prevent regressions; prove behavior under basic load.
NFR-08	Observability	See system health without tracking users.	<ul style="list-style-type: none"> <li>- Basic system metrics: CPU/mem, request rate/latency, error counts.</li> <li>- JSON logs with correlation IDs; retention <math>\leq 14</math> <b>days</b>.</li> <li>- Optional alert: error rate &gt;1% or health check failures → email.</li> </ul>	Operability without privacy trade-offs.
NFR-09	Internationalization & Units	Keep scope focused; use metric.	<ul style="list-style-type: none"> <li>- English-only UI/content.</li> <li>- <b>Metric units</b> (g/ml) with package awareness by default.</li> </ul>	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"><li>- Metrics: active connections, subscribes/sec, send/receive rates, WS error counts.</li><li>- Logs include correlation IDs and resource IDs; <b>no PII</b>.</li><li>- Alert when WS error rate &gt; 1% or connections drop &gt; 30% in 5 min.</li></ul>	Operability without analytics or privacy trade-offs.