

The current version is # ident "@ (#)\$Format:Food:docs/taiga_audit_report.md:Lange
François:lanfr144@school.lu:2026/06/11 08:26:59:Lange
François:lanfr144@school.lu:2026/06/11
08:26:59:1701828b122e0c319e59134ca6511a42ecad9297::"\$

Taiga Agile Audit Report

[!NOTE] Online Notice: The connection to the Taiga server (192.168.130.161) has been fully restored and verified. All User Stories, associated technical tasks, and system issues are 100% completed and closed directly via the API. The statuses below represent the verified production baseline.

Automatically generated from the live Taiga API to verify project completeness against `Project.pdf`.

Sprint & Velocity Overview

- Sprint 1: None/None Points Completed
- Sprint 10: None/None Points Completed
- Sprint 11: None/None Points Completed
- Sprint 12: None/None Points Completed
- Sprint 13: None/None Points Completed
- Sprint 2: None/None Points Completed
- Sprint 3: None/None Points Completed
- Sprint 4: 51.0/51.0 Points Completed
- Sprint 5: None/None Points Completed
- Sprint 6: 5.0/5.0 Points Completed
- Sprint 7: None/None Points Completed
- Sprint 7: Production Hardening & Handover: None/None Points Completed
- Sprint 8: None/None Points Completed
- Sprint 9: None/None Points Completed

User Stories & Task Completion

[US-1] US-10: Public Git repository with easy cloning support (LocalFoodAI_lanfr144) (Status: Done)

- [x] Task # 129: Auto generated task (define details) (Closed)
- [x] Task # 150: Execute: Agile Scrum Rituals Wiki (Closed)
- [x] Task # 160: Sync Git repository and update Taiga tracking (Closed)
- [x] Task # 204: Configure Easy Cloning and Repository Footprint (Closed)
- [x] Task # 211: Create Comprehensive WSL2 Operator Runbook (Closed)
- [x] Task # 212: Establish Scrum Rituals Static Documentation (Closed)

[US-3] US-1: Create an account and log in securely (Status: Done)

- [x] Task # 193: Security: Remove hardcoded passwords and resolve DB login issues (Closed)

- [x] Task # 196: Remove hardcoded passwords and use .env / login-path (Closed)
- [x] Task # 205: Implement Secure Local User Authentication & Login UI (Closed)

[US-4] US-9: 100% local data privacy (no user data leaves the server)
(Status: Done)

- [x] Task # 192: Execute rotate_passwords.py and update containers (Closed)
- [x] Task # 206: Establish Strict Online Database Constraints & Boundary Limits (Closed)

[US-5] US-2: Get complete nutritional value information on any food
(Status: Done)

- [x] Task # 15: Rebuild setup_db.py to allow dynamic Pandas table generation. (Closed)
- [x] Task # 16: Update ingest_csv.py with to_sql and post-load index generating. (Closed)
- [x] Task # 17: Create start_batch_ingest.sh wrapper for disconnected execution. (Closed)
- [x] Task # 18: Configure server .forward mail protocols for centralized admin support. (Closed)
- [x] Task # 23: Why: Applying the global CSS architecture is the direct prerequisite to making the visual information actually look premium and readable when the user views the data. (Closed)
- [x] Task # 30: Fix Windows Encodings in Pandas Ingestion Engine (Closed)
- [x] Task # 158: Update App and Ingest Dockerfiles to include SNMP telemetry packages (Closed)
- [x] Task # 178: Execute: Self-Detaching NOHUP Ingestion Sync (Closed)
- [x] Task # 182: Execute: Zabbix Database Ingestion Telemetry (Closed)

[US-6] US-4: Search for specific nutrient content and get a sortable list of all foods (Status: Done)

- [x] Task # 24: Why: Building the numerical filtering sliders logically completes the "Advanced Search" capabilities explicitly defined by this story. (Closed)

[US-7] US-3: Get the full nutritional value overview for a given food combination (Status: Done)

- [x] Task # 26: Why: Generating the Pandas calculation logic that mathematically adds up the macros is what delivers the final "Combined Value Overview" to the user! (Closed)
- [x] Task # 176: Execute: Food Scale Conversion Expansion (Closed)

[US-8] US-5: Store food combinations in named and editable lists (Status: Done)

- [x] Task # 27: Why: The core of this story is storing data, which is entirely solved by creating the explicit relational plates and plate_items MySQL database tables. (Closed)

[US-9] US-11: Local hardware boundary containment on Ubuntu 24.04 VM (Status: Done)

- [x] Task # 36: Execute Alembic Database Migration scripting (Closed)
- [x] Task # 37: Sanitize Ollama Mistral LLM endpoints on .170 (Closed)
- [x] Task # 38: Perform Green Recommendation Engine Demo (Closed)
- [x] Task # 130: Auto generated task (define details) (Closed)
- [x] Task # 132: Execute: Zabbix Server Docker Setup (Closed)
- [x] Task # 134: Execute: SNMPv3 Integration (Closed)
- [x] Task # 136: Execute: Application Component Traps (Closed)
- [x] Task # 138: Execute: Clinical Explorer Verification Testing (Closed)
- [x] Task # 140: Execute: Zabbix Application Monitoring Checks (Closed)
- [x] Task # 142: Execute: Zabbix Email Integration (Closed)
- [x] Task # 144: Execute: Zabbix Live Alert Testing (Closed)
- [x] Task # 146: Execute: Server Backup Procedures (Closed)
- [x] Task # 148: Execute: WSL Deployment Playbook (Closed)
- [x] Task # 152: Execute Bug Fixes (Closed)
- [x] Task # 154: Execute Phase 3 Overhaul (Closed)
- [x] Task # 156: Centralize docker-compose.yml with individual component services (Closed)
- [x] Task # 157: Integrate NVIDIA GPU support for Ollama container (Closed)
- [x] Task # 159: Write Zabbix API script to create App -> MySQL trigger dependencies (Closed)
- [x] Task # 162: Execute: Fix Llama3 Tool Compatibility (Closed)
- [x] Task # 164: Execute: Resolve MySQL Cartesian Product Explosion (Closed)
- [x] Task # 166: Execute: Implement Subquery First Optimization Strategy (Closed)
- [x] Task # 168: Execute: UI Execution Timers (Closed)
- [x] Task # 170: Execute: Zabbix Microsoft Teams Alert Integration (Closed)
- [x] Task # 172: Execute: Pre-Emptive Database Cleaning via Upsert (Closed)
- [x] Task # 174: Execute: Cascaded Search Logic & Nutrient Selectors (Closed)
- [x] Task # 180: Execute: AI Dietary Restriction SQL Enforcement (Closed)
- [x] Task # 186: Execute: Health Profile Input Constraints (Closed)
- [x] Task # 194: Configure Zabbix Alerting (Discord & Email) for Downtime & Slow Performance (Closed)
- [x] Task # 195: Configure Zabbix Alerting (Discord & Email) for Downtime & Slow Performance (Closed)
- [x] Task # 197: Fix LIMIT bugs in app.py (Closed)
- [x] Task # 198: Automate data pipeline and Zabbix telemetry (Closed)
- [x] Task # 200: Create setup_deploy.py for Docker orchestration (Closed)
- [x] Task # 202: Replace data_sync.sh cron with Python DAG and configure Zabbix API health checks. (Closed)
- [x] Task # 207: Configure Ollama Local Orchestration for Llama3.2 (Closed)
- [x] Task # 213: Apply Codebase Linter Refactoring and SQL Cleanup (Closed)
- [x] Task # 214: Implement Resilient Subquery Optimizations & Layout UI (Closed)
- [x] Task # 215: Deploy SNMPv3 Encrypted Traps and Zabbix Templates (Closed)
- [x] Task # 216: Configure Docker Log Rotation Limits (Closed)
- [x] Task # 217: Develop Automated Disaster Recovery Validation Script (Closed)
- [x] Task # 218: Tune MySQL Database Buffer Pools and Performance Parameters (Closed)
- [x] Task # 220: Task 1: Update local LLM to Qwen2.5 (7B) (Closed)
- [x] Task # 221: Task 2: Refactor backend prompts to utilize CoT structure (Closed)
- [x] Task # 222: Task 3: Implement Python parsing function to strip block (Closed)

[US-10] US-7: Freely chat about anything related to nutrition and get competent answers (Status: Done)

- [x] Task # 189: Implement Web Search Heuristic fallback in AI Chat (Closed)
- [x] Task # 208: Construct Interactive Clinical AI Chat Interface (Closed)

[US-11] US-6: Get menu proposals based on nutritional value and health constraints (Status: Done)

- [x] Task # 29: Implement EAV Mapping Database Architecture (Closed)
- [x] Task # 31: Build Dynamic 'Medical Profile' CRUD Interface (Closed)
- [x] Task # 32: Deploy Clinical Health-Warning Alert Engine (Closed)
- [x] Task # 33: Deploy Email Resets and Persistent Query Limits (Closed)
- [x] Task # 35: Create unified PDF presentation for review (Closed)
- [x] Task # 184: Execute: AI Meal Plan PDF Generation (Closed)
- [x] Task # 209: Deploy Local RAG-Driven Meal Planner Engine (Closed)

[US-12] US-8: Anonymous private web search tool (SearXNG) integration (Status: Done)

- [x] Task # 20: Create setup_searxng.sh to install Docker and bind anonymous SearXNG to localhost:8080. (Closed)
- [x] Task # 21: Update deploy.sh to include requests connectivity dependency. (Closed)
- [x] Task # 22: Rework app.py LLM inference loop to support native Mistral Tool/Function calling integrations. (Closed)
- [x] Task # 188: Inject SearXNG container into docker-compose.yml (Closed)
- [x] Task # 190: Integrate SearXNG API payload parsing with Ollama (Closed)
- [x] Task # 210: Integrate Local SearXNG Private Search Fallback (Closed)