Configuration Management Plan (CMP)

Project: Food Logging & Nutrition Intelligence System
Version: 1.0
Date: 19-11-2025

1. Introduction

1.1 Purpose

This Configuration Management Plan (CMP) defines the processes, tools, responsibilities, and policies used to manage configuration items (CIs) throughout the lifecycle of the Food Logging & Nutrition Intelligence System. The plan ensures consistency, traceability, version control, and controlled change management across all system components.

1.2 Scope

The CMP applies to all artifacts related to the system including:

- Source code (mobile app, backend APIs, ML services)
- Machine learning model versions & training datasets
- Infrastructure configuration (AWS resources)
- Documentation (design docs, requirements, API specs)
- Test artifacts and deployment packages

It is applicable across development, testing, staging, and production environments.

1.3 Objectives

- Protect the integrity of all configuration items
- Ensure controlled and auditable changes
- Maintain reliable version history and rollback capability
- Provide transparency across teams (mobile, backend, ML, DevOps)

- Support CI/CD pipelines and automated deployments

2. Configuration Management Organization

2.1 Roles & Responsibilities

Role	Responsibilities
CM Manager	Oversees implementation of CMP, approves baselines, maintains CI registry
DevOps/QA Engineer	Maintains CI/CD pipelines, manages AWS infrastructure configs. Validates releases, ensures test artifacts reflect the latest baseline.
Mobile Team Lead	Oversees mobile source code changes and release versions
Backend Team Lead	Reviews and approves API/backend CI modifications

2.2 Communication

- Twice a week sync between all team leads
- Weekly CM review meeting
- Automated alerts for key changes in Git (protected branches, PR reviews)

3. Configuration Identification

3.1 Configuration Items (CIs)

All major components are designated as controlled CIs.

3.1.1 Software CIs

- **Mobile Application**
  - Flutter codebase
  - Build artifacts (APK/AAB)

    - UI assets
- **API Layer**
    - Lambda functions
    - OpenAPI/Swagger specifications
    - Deployment packages
- **ML Services**
    - Model artifacts (trained weights, inference scripts)
    - Preprocessing & postprocessing modules
    - Feature extraction code
    - Training pipeline scripts

3.1.2 Data CIs

- USDA/Canadian nutrition datasets
- Model training sets (raw & curated)
- User correction/feedback datasets
- DynamoDB schemas

3.1.3 Infrastructure CIs

- AWS configuration (S3 buckets, API Gateway configs, IAM roles)
- CICD pipeline definitions

3.1.4 Documentation CIs

- Requirements specification
- Architecture and design documents
- CMP, test plans, user guides

4. Configuration Versioning & Baselines

4.1 Version Control System

- **Git** is used for all software/code & documentation.
- Hosted on **GitHub**.

- ML models + datasets stored in **S3**.

4.2 Branching Strategy

- **main** → Production-ready
- **feature_name** → Individual features
- **develop** → Integration branch
- **hotfix** → Fast production patches

Protected branches require:

- Pull Request
- Code review (minimum 1 reviewer)
- Passing CI tests

4.3 Baseline Types

- **Development Baseline**: End of sprint
- **Test Baseline**: Candidate for QA testing
- **Release Baseline**: Fully validated, versioned release
- **Model Baseline**: Tagged model version (e.g., model-v1.3.2) including:
    - Model weights
    - Training dataset hash
    - Model config file
    - Metrics summary

5. Configuration Control

5.1 Change Control Procedures

All changes follow a standardized approval workflow:

1. **Change Request (CR) Submission**
    a. Developer or team lead submits a CR (GitHub Issues).
2. **Impact Analysis**
    a. Affected components identified

    b. Complexity, risk, and dependencies evaluated
3. **CR Review**
    a. CM Manager + relevant team leads review request
4. **Approval or Rejection**
5. **Implementation & Code Review**
    a. Code committed to feature branch
    b. PR approval required
6. **Verification**
    a. QA reviews functionality and regression impacts
7. **Baseline Update**

5.2 Emergency Changes

- Allowed only for critical issues
- Must be documented retroactively within the following 24 hours
- Requires approval from CM Manager and team lead

6. Configuration Status Accounting

6.1 CI Tracking

A Configuration Item Registry is maintained containing:

- CI name
- Owner
- Version history
- Location (Git repository, S3 bucket)
- Baseline associations

6.2 Reporting

Automated reports generated weekly:

- Git commits & merged PRs
- Model version updates
- CI/CD pipeline deployment log

7. Configuration Verification & Audit

7.1 Verification

Ensures CIs meet their requirements and are correctly versioned:

- CI integrity checks (hash validation)
- PR checks for code changes
- Model reproducibility validation (dataset + config hashing)

7.2 Audits

Two audit types:

1. **Functional Audit**
    a. Ensures implemented features match approved specifications
2. **Physical Audit**
    a. Ensures the correct versions of all CIs exist and are stored properly

Audits occur:

- Before major release
- After architectural changes
- Quarterly for ML model + dataset integrity

8. Tools & Automation

8.1 Tooling

- **GitHub –** Version control
- **GitHub Actions –** CI/CD
- **AWS SageMaker –** Model hosting & monitoring
- **S3 versioning –** Dataset management

8.2 Automated Policies

- Auto-linting and testing on every commit
- GitHub security scans for dependencies
- S3 object versioning enabled for model/data artifacts
- Automatic rollback if deployment fails CI validations

9. Release Management

9.1 Release Packaging

A release must include:

- Compiled mobile app build
- Deployed backend Lambdas
- Versioned ML model endpoints
- Updated documentation

9.2 Release Numbering

Follows semantic versioning:

MAJOR.MINOR.PATCH

Examples:

- 2.1.0 — Feature update
- 2.1.3 — Bug fix
- 3.0.0 — Breaking changes

9.3 Deployment Stages

- **Development → Staging → Production**
- Manual approval required between stages

10. Glossary

- **CI:** Configuration Item
- **CMP:** Configuration Management Plan
- **PR:** Pull Request
- **ML:** Machine Learning
- **S3:** Amazon Simple Storage Service