Software Project Management Plan (SPMP)

Project: Personal Food Log App
Course: SEG 4105 – Software Project Management
Team #18
Date: November 05, 2025

Team Members

Name	Role
Pradyu Vasudev	Project Manager / AWS Infrastructure
Luke DuSautoy	Analyst & UX Lead
Wilt Moise	Analyst & ML Lead
Samuel Braun	Analyst/Programmer & Mobile Lead

1. Introduction

1.1 Purpose

This SPMP defines how Team #18 will plan, execute, monitor, and close the *Personal Food Log App* project. It establishes responsibilities, schedules, risks, and control mechanisms consistent with IEEE 1058 and SEG 4105 requirements.

1.2 Scope

The project delivers a proof-of-concept mobile application that enables users to log daily food intake through smartphone images. Photos are uploaded to AWS, processed by a machine-learning stub that identifies visible ingredients and returns estimated calories. Advanced nutrition analytics, social features, and external-platform integration are excluded.

1.3 Definitions, Acronyms and Abbreviations

AWS – Amazon Web Services, the mandated cloud platform.

CCB – Change Control Board, composed of the Project Manager and TA.

CI – Continuous Integration, the automated build process for each commit.

PoC – Proof of Concept.

SPMP – Software Project Management Plan.

VBM – Vision-Based Meal system, the AI technique for food detection.
Definitions of specialized ML or AWS services appear in the project glossary within /docs/glossary.md.

1.4 References

- SEG 4105 Course Outline and Deliverable Schedule
- *Project Charter v1.0* (26 Sep 2025)
- IEEE Std 1058-1998 — Software Project Management Plans
- AWS Free-Tier Documentation

2. Project Organization

2.1 Stakeholders and Roles

Role	Name	Responsibilities
Sponsor	Shervin Shirmohammadi	Oversight, approval of milestones
Customer	Tushar Bhatia	Feedback
Project Manager	Pradyu Vasudev	Planning, risk management, AWS infrastructure
Analyst & UX Lead	Luke DuSautoy	Requirements, wireframes, competitor benchmarking
Analyst & ML Lead	Wilt Moise	Data pipeline stub and calibration mock
Mobile Lead	Samuel Braun	Mobile capture, API integration, CI builds

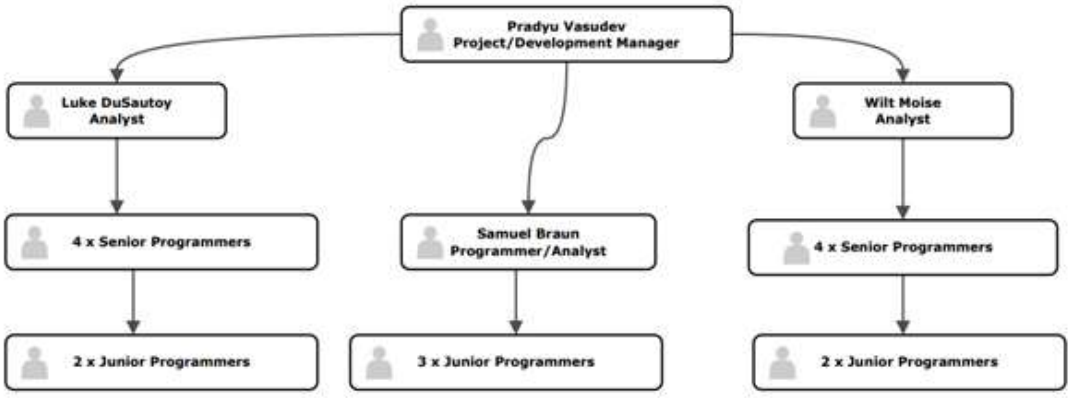
2.2 Organizational Structure

Structure: Functional structure, People are divided based on their specialities.

Change Control Board (CCB): PM + Customer. Sponsor consulted as needed.

Decision Rights: Component owners → PM tie-break → CCB for scope/time issues.

2.3 Organizational Diagram



3. Managerial Process Plans

3.1 Start-Up Plan

Charter approval by Sep 30.

Git repository and AWS set up.

Low-fidelity wireframes and API contract ready by Nov 10.

POC demonstration, and stakeholder approval to move forward.

3.2 Work Breakdown Structure (Level 2 Summary)

3.2.1 Project Definition and Initial Planning (Inception and Elaboration Phases)

This phase focuses on establishing the project's scope, boundaries, initial architecture, and detailed requirements.

ID	Stories	Owner	Deliverable	Due
3.2.1.1	**Project Charter & Setup**	**Pradyu**	**Approved Charter**	**Sep 30**
	Description: Formal documentation recognizing project existence, defining goals, scope, objectives, quality level, and validating business justification.			
3.2.1.2	**Requirements & UX Flows**	**Luke**	**Wireframes + Backlog**	**Nov 10**
	Description: Defining user needs, system constraints, and interface visualization. Deliverables include wireframes (design element) and a backlog (collection of user stories/requirements).			
3.2.1.3	**Final Plan & PoC**	**Samuel**	**IEEE-1058 SPMP + PoC**	**Nov 20**
	3.2.1.1.1	SPMP Documentation		IEEE-1058 SPMP
	Description: Finalized Software Project Management Plan (SPMP) documentation.			
	3.2.1.1.2	Proof of Concept Delivery		PoC
	Description: Delivering the final Proof of Concept software.			
3.2.1.4	**Demo Presentation**	**Wilt**	**Slides + Live Demo**	**Nov 28**
	3.2.1.4.1	Demo Preparation		Slides
	3.2.1.4.2	Live Demonstration		Live Demo

3.2.2 Core Development and System Assembly (Construction Phase)

This phase involves achieving useful, working versions of the product, often referred to as alpha or beta versions.

ID	Stories	Owner	Deliverable	Due
3.2.2.1	**Mobile Capture & Upload** *Description:* Implementation of the mobile application functionality to capture and upload data.	**Samuel**	**Android/iOS Upload Demo**	**Nov 15**
3.2.2.2	**ML Pipeline Stub** *Description:* Implementation of the machine learning backend component, resulting in a server stub that returns data.	**Wilt**	**Server Stub Return Data**	**Nov** 18
3.2.2.3	**Integration & Testing** *Description:* Combining individual modules (Mobile, ML Stub) and ensuring they interact correctly (integration testing). Goal is to achieve adequate quality.	**Pradyu**	**Stable Demo Build**	**Nov 25**
3.2.2.4	**AI training and fine tuning** *Description:* Fine tune AI model to achieve at least 94% accuracy against real world test data.	**Samuel**	**AI model**	**Dec 15**
3.2.2.5	**Completed UI interface** *Description:* Fine tune AI model to achieve at least 94% accuracy against real world test data.	**Luke**	**Flutter App**	**Jan 20**

3.2.3 Project Finalization and Deployment (Transition Phase)

This phase ensures stakeholder concurrence of completeness and achieves the final product baseline and begins the process of deploying the application.

ID	Stories	Owner	Deliverable	Due
3.2.3.1	**Final Stakeholder Approval**	**Luke**	**Working Application**	**Jan 30**
	3.2.3.1.1 *Description:* Finalized Software Project Management Plan (SPMP) documentation.	Non-deployed app		All documentation, Final app prototype
	3.2.3.1.2	Deployment plan		Detailed plan for app release

Description: Documentation detailing the release of the application.

3.2.3.2	**Deployment**	**Samuel**	**User Facing Application**	**Feb 15**
	3.2.3.2.1	List application in stores		Approval

3.2.4 Project Closure (Post-Performance Analysis)

This activity occurs after the project is functionally complete, supporting Continuous Process Improvement.

ID	Stories	Owner	Deliverable	Due
3.2.4.1	**Post-Performance Analysis**	**Pradyu + Luke**	**Performance Report**	**Feb 26**

Description: Measuring and analyzing process metrics, collecting data on lessons learned, and providing strategies for improving the process for future projects.

3.3 Estimation Methods

Two complementary effort-estimation techniques were used to ensure that schedule and workload projections were realistic, unbiased, and cross-validated:

3.3.1 Blitz Estimation

The team performed a rapid, collaborative estimation session where the entire WBS was reviewed as a group. Each member independently proposed rough effort ranges for each task (in hours), followed by a group discussion to quickly converge on a consensus estimate.
Blitz Estimation was chosen because:

- It accelerates initial forecasting in early-phase planning.
- It exposes hidden tasks and missing dependencies through fast group feedback.
- It suits projects with small teams and compressed timelines, such as SEG4105. Blitz produced the first pass estimate, which served as the baseline for review.

3.3.2 Wideband Delphi Estimation

After Blitz results were established, a three-round Wideband Delphi cycle refined the numbers. Each member estimated task effort anonymously, after which:

1. Individual estimates were collected and averaged.
2. The range and variance were analyzed.
3. Outliers triggered a discussion moderated by the Project Manager.

4. A second and third round were conducted until convergence (variance < 20%).

Wideband Delphi was chosen because it:

- Reduces bias from dominant personalities.
- Forces consideration of uncertainty and hidden risk.
- Produces statistically stable estimates despite a small team size.

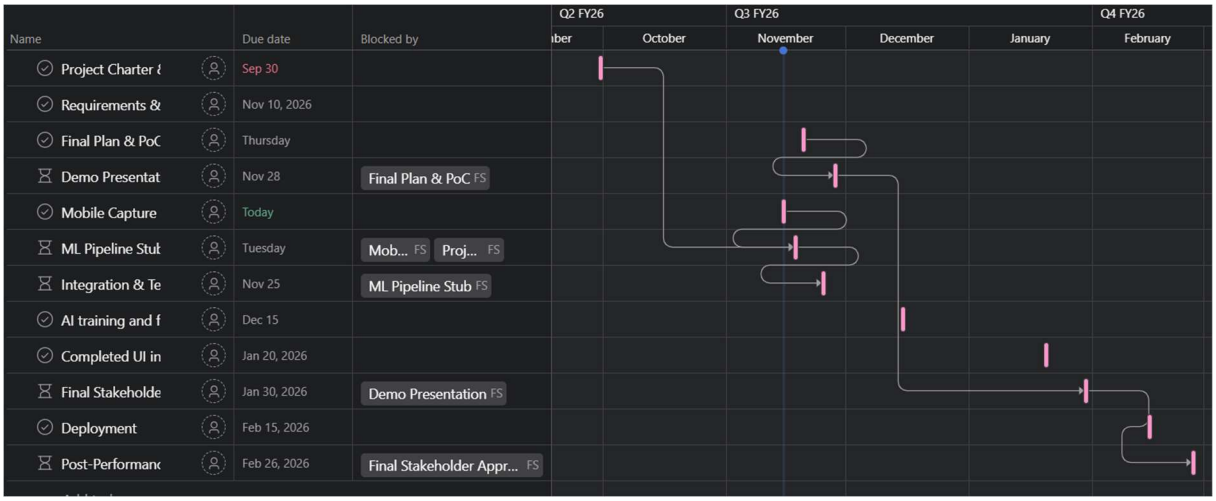
3.3.3 Resulting Estimates

The final effort values used in the WBS and schedule represent the converged Wideband Delphi output**,** cross-validated against the Blitz session for consistency.
These estimates were then converted to calendar time using an availability constraint of **around 40 hours/week per employee**, and applied against the schedule in and milestone deadlines defined in Section 3.3.

3.4 Milestone

ID	Milestones	Owner	Deliverable	Due
M1	Project Charter & Setup	Pradyu	Approved Charter	Sep 30
M2	Requirements & UX Flows	Luke	Wireframes + Backlog	Oct 6
M3	Mobile Capture & Upload	Samuel	Android/iOS Upload	Nov 10
M4	ML Pipeline Stub	Wilt	Server Stub Return Data	Nov 15
M5	Integration & Testing	All	Stable Demo Build	Nov 20
M6	Final Plan & PoC	All	IEEE-1058 SPMP + PoC	Nov 20
M7	Demo Presentation	All	Slides + Live Demo	Nov 28
M8	Final Stakeholder Approval	Luke	Working Application, Deployment plan	Jan 30
M9	Post-Performance Analysis	Pradyu + Luke	Final Report	Feb 26

3.4.1 Gantt Chart



3.4.2 Schedule Dependencies

Story 3 (Mobile Capture) → Story 4 (ML Pipeline Stub) → Story 5 (Integration & Testing).

Charter approval (M1) → Pipeline Stub (M4).

PoC stability (M6) → Presentation (M7) → Final Approval (M8) → PPA (M9)

3.4 Budget and Resources

Human: 18 employees, 40 h/week each.

Technical: AWS, GitHub, Development Devices.

The estimated total project budget is **$246500 CAD equivalent**, based on nominal rates and not billed costs.

Category	Description	Estimated Cost (CAD)
Development Salaries	18 developers × 40 h/week × 10 weeks × $30/h	$216,000 (in-kind)
Software Licenses	IDEs (VS Code free), design tools (Figma license), Git Hub	$500
Cloud Rental	AWS (S3, Lambda, Cognito)	$10,000
Hardware	Personal laptops and smartphones for testing	$20,000
Total		**$246,500 in effort value**

3.5 Risk Management Plan

ID	Risk	Likelihood	Impact	Mitigation
R1	Scope Creep	Med	High	Formal CCB; freeze scope by M5
R2	Key Developer Unavailable	Low	High	Cross-train, shared docs
R3	Mobile Build Issues	Med	Med	Latest SDKs, CI tests
R4	AWS Quota Limits	Low	Med	Local mocks
R5	Auto-Calibration Fails	Med	High	Baseline method + manual override

3.6 Monitoring and Control Plan

Progress Reviews: Weekly stand-ups + GitHub burndown.

Status Reports: Weekly summary to Customer via Brightspace forum.

Metrics:

- Schedule variance $\leq \pm 10$ %
- Defect density < 0.3/LOC (qualitative check)
- Demo readiness index ≥ 90 % of planned features operational.

Change Control: Request → Impact (24 h) → CCB Decision (48 h).

3.6.1 Communication Plan

Project communication follows a structured, transparent pattern:

- **Weekly Team Syncs** – 30 min meetings to review progress, blockers, and upcoming tasks.
- **Customer Status Reports** – Concise summaries submitted via Brightspace every Friday.
- **Issue Tracking** – All tasks, bugs, and change requests logged and discussed in GitHub Issues.
- **Emergency Channels** – Slack DMs or email (PM → Customer only) for time-critical issues.

- **Documentation Updates** – Major decisions recorded in Decision_Log.md within the repo.
  The Project Manager is the sole liaison to the TA; all other communication occurs internally.

3.7 Deliverable Traceability Matrix

Milestone	Deliverable	Primary Owner	Evaluation Criterion
M1	Charter	Pradyu	Customer approval
M2	UX Flows & Backlog	Luke	Feedback Meeting 1
M3	Mobile Capture Upload	Samuel	Functional Demo
M4	ML Stub	Wilt	Feedback Meeting 2
M5	End-to-End Demo	Samuel	Stable PoC
M6	Project Plan + PoC	Pradyu	Customer review
M7	Presentation	Luke	Sponsor evaluation
M8	Final Stakeholder approval	Luke	Sponsor evaluation
M9	Post-Performance Analysis	Pradyu	Final submission

4. Technical and Supporting Process Plans

4.1 Process Model

Hybrid Agile incremental model: each milestone = 1 iteration (analysis → prototype → feedback).

4.2 Methods/Tools/Techniques

Languages: Dart (Flutter), Python (backend).

Cloud: AWS S3, Lambda, Cognito.

VCS: GitHub with branch protection.

Testing: Manual functional + CI build verification.

4.3 Configuration Management

Repo structure: frontend/, backend/, docs/.

Change tracking via GitHub issues and tags (v1-M3, v1-M5…).

Artifacts stored in /docs and submitted to Brightspace.

4.4 Quality Assurance and V&V Plan

Peer review of each deliverable by non-author.

Acceptance criteria defined per milestone.

Verification: Each function tested against expected input/output.

Validation: End-to-end demo and Customer approval confirm objectives O1–O4 met.

4.5 Documentation Plan

Artifacts include Charter, SPMP, Risk Log, Meeting Minutes, PPA. Each document versioned and dated.

4.6 Training Plan

Internal 15-minute micro-sessions on AWS setup, Flutter build, and Git workflow; how-to notes in repo.

Training is two-tiered:

1. **Onboarding of Junior Developers** – Pair programming sessions and 15-minute "micro-labs" on Flutter and GitHub workflows led by senior members.
2. **Cross-Team Machine Learning Upskilling** – All members complete the Coursera "Intro to TensorFlow" module and study dataset annotation using Kaggle Food-101.
   Each training activity is logged, and completion is reviewed during weekly stand-ups.
   The intent is to ensure shared competency in both mobile and ML domains before PoC integration.

4.7 Problem Resolution Plan

1 → 1:1 discussion; 2 → PM mediation; 3 → time-boxed prototype decision; 4 → retrospective and reassignment if recurring.

4.8 Process Improvement

Lessons learned documented in Post-Performance Analysis for continuous improvement.

5. Project Close-Out

Deploy approved application.

Submit PPA.

Conduct team retrospective and archive GitHub repo (tag v1.0).

Deliver demo video and final documentation bundle for continuous support.

Revision History

Version	Date	Author(s)	Description
1.0	Nov 5, 2025	Team #18	Initial SPMP per IEEE 1058
1.1	Nov 5, 2025	Pradyu Vasudev + Luke DuSautoy	Added traceability, metrics, and Gantt overview