4. Risk Management Plan

This section describes how project risks for the Personal Food Log App were identified, analyzed, and managed throughout the project lifecycle. The primary artifact is the Risk Register (IDs R1–R12 in the Charter), which is maintained and updated by the Project Manager and reviewed by the whole team.

4.1 How Risks Were Identified

Risk identification was an iterative process carried out from project initiation to closure. The team used the following activities and sources:

1. Charter and Scope Review

- The initial set of risks (R1–R12) was derived from the assumptions, constraints, and scope boundaries defined in the Project Charter (for example, mandatory use of AWS, support for both Android and iOS, privacy regulations, and non-negotiable delivery deadlines).

- For each assumption or constraint, the team asked, "What could go wrong if this is not true?" and recorded the result as a risk.

2. Work Breakdown Structure (WBS) and Schedule Analysis

- When decomposing the work into milestones M1–M8, the team identified schedule and resource risks such as R2 – Key developer unavailable and R8 – Schedule crunch pre-11/20.

- Any task on the critical path (such as mobile capture, AWS integration, and machine learning components) was examined for potential delays or dependencies.

3. Technical Architecture and Technology Review

- During design of the mobile app and AWS backend, the team identified technical risks like R3 – Mobile OS fragmentation breaks build, R4 – AWS cost/limits, R5 – Auto-calibration not reliable, and R6 – Segmentation accuracy too weak to convince.

- Risks specific to machine learning, such as dataset quality and bias (R12), were identified using established ML project checklists and prior project experience.

4. Stakeholder Discussions and Lessons from Similar Projects

- Discussions with the Sponsor, Customer, and customer during feedback meetings surfaced additional risks related to expectations and success criteria (for example, R11 – Misalignment on success criteria).

- The team also reviewed similar past projects to anticipate common issues such as demo failure, unclear requirements, and last-minute scope changes.

5. Ongoing Brainstorming and Pre-Mortem Sessions

- At the start of major milestones (M2, M3, M5), the team conducted short pre-mortem sessions: assuming the project had failed, they brainstormed reasons why (such as privacy incident, demo crash, or unusable UI) and translated these into new or updated risk entries.

- Any new risk identified during development, testing, or customer interactions was added to the Risk Register with a unique ID.

4.2 How Risks Were Analyzed

The project used a qualitative risk analysis approach appropriate for a full, production-grade application.

1. Likelihood and Impact Scales

- Each risk was rated for Likelihood (L) and Impact (I) using a three-level scale:

Likelihood: Low (1), Medium (2), High (3)

Impact: Low (1), Medium (2), High (3)

2. Risk Exposure / Priority Score

- A Risk Score was calculated as: Risk Score = L × I

- Scores ranged from 1 to 9 and were interpreted as:

7–9: High priority

4–6: Medium priority

1–3: Low priority

3. Ranking and Heat Map

- Risks were sorted by Risk Score on a 3×3 risk matrix.

- High-priority risks included R1, R5, R6, R8, and R9.

4. Review and Re-Assessment Cadence

- The Risk Register was reviewed weekly and before major milestones.

- Ratings were updated as needed and all changes were documented.

4.3 Risk Mitigation and Monitoring Plan

Risk responses included avoid, mitigate, transfer/share, or accept. Each risk was assigned actions, owners, and triggers.

1. Mitigation Planning

Examples:

- R1: Scope creep. Strategy: Mitigate/Avoid. Actions: Enforce CCB process; freeze scope at M5.

- R5: Auto-calibration risk. Strategy: Mitigate. Actions: Implement baseline early; provide manual override.

- R7: Privacy issues. Strategy: Avoid/Mitigate. Actions: Use synthetic images; strip metadata; enforce consent.

- R9: Cloud outage. Strategy: Mitigate. Actions: Offline mode; backup devices; fallback rehearsal.

2. Integration into Project Planning

- High-priority mitigation tasks were added to the backlog with owners and due dates.

3. Triggers and Contingency Execution

Examples:

- Burn-down chart misalignment triggers feature de-scoping.

- Calibration errors trigger simplified flow fallback.

- Connectivity issues trigger offline mode.

4. Monitoring and Reporting

- Weekly updates maintained by the Project Manager.

- Summaries sent to Sponsor and customer ahead of major reviews.

5. Closure and Lessons Learned

- At project close, the team will document which risks occurred, which mitigations worked, and insights for future large-scale application projects involving mobile, AWS, and machine learning.