STATUS ACCOUNTING PLAN – GYM ACCESS MANAGEMENT SYSTEM

1. INTRODUCTION

This document defines the status accounting process for tracking changes produced by change requests (CRs) in the Gym Access Management System. The goal is to monitor implementation progress, manage issues, and document deviations from the baseline.

2. STATUS ACCOUNTING DATA STRUCTURE

Field	**Description**
CR ID / Class	Identifier and classification (Electronic Payment, Rewards, Facial Recognition)
CI Reference	Configuration Item affected (Requirements, Design, Code, Baseline versions)
Changes on Baseline	Functional/non-functional requirements added or modified, design changes, code updates
Persons Assigned	Backend, Frontend, Database, skill levels (Junior, Mid, Senior)
Maintenance Status	Integrated / Partially Integrated / Not Integrated
Estimated Time	Based on development plan (4 weeks for payments, 3 for rewards, 6+ for facial recognition)
Time Elapsed	Time consumed so far since CR approval
Coding Time	Hours logged on feature development
Testing Time	Time spent on test case development and execution
Optimization Time	Effort in performance tuning or UI polishing
Budget Assigned	Budget estimated per CR
Budget Spent	Real expenditure to date
Infrastructure Used	Fingerprint reader, cloud DB, servers, camera (for face recognition)
Issues Encountered	Technical or logistical challenges (API integration failures, device incompatibility)
Handling of Issues	Corrective actions, workaround, pending resolution status
Feature Completion (%)	Implementation percentage
Implementation Status	Planned / On Track / Delayed / Completed

Additional Metric	User Engagement Impact (None / Low / Medium / High) – based on relevance to gym clients

3. STATUS ACCOUNTING RULES (CUSTOMIZED)

If the estimated time is exceeded without completion, a report will be generated describing:

- Main issues encountered
- Context and affected components
- Partial outcomes

If the budget exceeds 10% of the initial estimate, the cost overrun must be justified with:

- A breakdown of unexpected expenses
- Areas where optimization is required

If features remain unimplemented after time and budget limits:

- A team competency analysis must be performed
- Skill gaps will be matched with additional training or resource reallocation

Weekly Meetings will be held to:

- Review CR implementation status
- Address issues
- Record decisions, responsibilities, and tasks

Final Report Requirements:

- Summary of all changes to baseline
- Chart comparing expected vs actual time
- Chart comparing scheduled vs actual budget
- List and analysis of issues encountered}
- Percentage of goal completion

4. IMPLEMENTATION SPEED CATEGORIES

- Faster Implementation: Completed before estimated time with <90% of budget spent
- Planned Implementation: Completed within time and budget limits

- Slower Implementation: Exceeds time or budget (over 10%)

5. STATUS ACCOUNTING REPORT TEMPLATE

- CR ID and Date of Approval
- CR Description
- CI Versions Affected (Requirements, Design, Code)
- Team Members Involved (Name, Role, Skill Level)
- Estimated Budget
- Estimated Time
- Roles Assigned
- Budget Used
- Time Used
- HR Performance (Efficiency, Task Completion)
- Issues
  - Description
  - Author (who reported it)
  - Personnel involved
  - Associated risk
  - Potential solution
  - Outcome (solved, addressing, unsolved) with rationale
- Outcome Summary (status of CR implementation, user engagement, final percentage complete)

6. MONITORING AND FOLLOW-UP PLAN

- Weekly progress logs are maintained by the project manager.
- Mid-phase evaluations during week 4, 10, and 16.
- Final audit during week 20, before project closure.
- Reports to be stored in the project repository and presented during review meetings.

7. APPROVAL AND MAINTENANCE

This status accounting plan is approved by the project manager and is subject to revision based on project progress or team changes.

This section demonstrates the application of the status accounting plan in three different project phases with two approved CRs: CR-01 (Electronic Payments) and CR-02 (Rewards System).

Scenario 1: Beginning of the project – minimal code written

CR-01 Issue: API integration documentation from the payment gateway is inconsistent, causing initial delays.

- Handling: Assigned team to contact provider and request updated specs. Temporary sandbox created for simulation.
- Outcome: Addressing. Still awaiting updated documentation.

CR-02 Issue: No ruleset defined for point allocation frequency (e.g., daily vs. per check-in).

- Handling: Requirements analyst drafted a logic tree proposal, pending validation in next meeting.
- Outcome: Addressing. Requires functional confirmation.

Scenario 2: Mid-development – ~50% of code written

CR-01 Issue: Unexpected behavior in backend after integrating sandboxed payment test – it conflicts with membership renewal logic.

- Handling: Bug isolated by backend lead. A new database schema proposed to separate transaction logs.
- Outcome: Solved. Retested and merged.

CR-02 Issue: UI designer delayed layout for rewards tracking dashboard due to prioritization of core features.

- Handling: PM reallocated frontend intern to assist with design.
- Outcome: Addressing. Design to be completed within the week.

Scenario 3: Final stages – ~90% code completed

CR-01 Issue: Security vulnerability found in payment callback verification.

- Handling: Escalated to senior backend developer verification module introduced.
- Outcome: Solved. Passed QA penetration test.

CR-02 Issue: Gamification logic causes race conditions when simultaneous point calculations occur.

- Handling: Critical bug. Database locking strategy revised. Testing extended 2 more days.
- Outcome: Addressing. QA retesting pending.

RESULTS BASED ON STATUS ACCOUNTING

- Real-time tracking of issue resolution: Each issue has traceable metadata – responsible person, risk level, and resolution.
- Improved team coordination: Scenarios show reassignments and collaboration across roles.
- Risk mitigation in early stages: Critical architectural decisions (e.g., schema separation, logic trees) made early avoided larger refactors.
- Real-time status awareness: “Addressing” vs. “Solved” outcomes show where extra effort is needed.
- Documentation of budget/timeline impacts: e.g., extra 2 days for testing in Scenario 3 noted for timeline adjustment.
- Scalability of reporting: Each CR report serves as a template for future change assessments and audit-ready review.