# **1. Tóm tắt dự án**

**Tên:** LotusMiles — POC Loyalty System (Member Portal + Admin Portal)  
**Mục tiêu:** Triển khai MVP trong 2 tuần để phục vụ cuộc thi hoán đổi vai trò: cho phép user gửi request tích dặm thủ công, admin duyệt/từ chối, và minh họa tích điểm qua SessionM (mock hoặc real).

# **2. Mục tiêu kinh doanh & tiêu chí thành công**

* Cho phép demo luồng end-to-end: gửi yêu cầu tích dặm → admin duyệt → cộng dặm → log transaction → (gọi SessionM).
* Đạt các tiêu chí chấm thi (không crash, FE/BE rõ ràng, có mock/real SessionM, logging, spec & test) — mapping chi tiết ở phần 11.
* Kết quả bàn giao: repo + README + Postman collection + 1 demo script để trình bày.

# **3. Phạm vi (Scope) — POC (cần làm trong 14 ngày)**

**Bao gồm (MUST):**

* Member Portal (Web responsive): login, profile, dashboard, gửi yêu cầu tích dặm thủ công, xem lịch sử & trạng thái request.
* Admin Portal (Web): list requests, filter/search, review detail, approve/reject, add manual transaction, audit log.
* SessionM integration: gọi API earn (mock hoặc real) khi approve/ghi nhận transaction; retry + queue khi lỗi.
* Cách tính dặm cơ bản + business rules cho manual claim. (mô tả chi tiết bên dưới)

**Không bao gồm (OUT of scope):**

* Phân hạng tự động (hiển thị giả lập nếu cần).
* Mobile app native (chỉ responsive web).

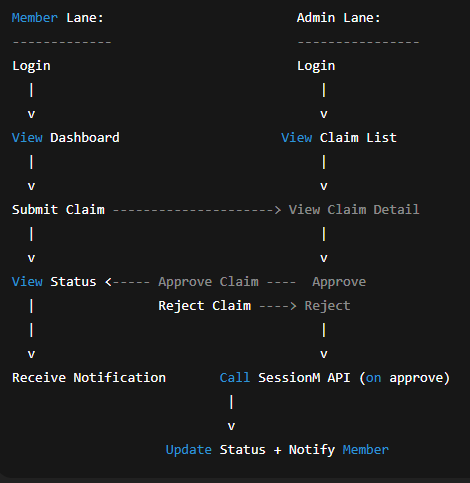
# **4. Personas (người dùng)**

* **Member**: Đăng nhập, gửi claim, xem trạng thái, xem lịch sử dặm.
* **Admin / Agent**: Duyệt claim, chỉnh sửa, ghi chú, thêm giao dịch manual.
* **FinOps/BI user**: Tải báo cáo hàng ngày/tháng.

# **5. Luồng nghiệp vụ chính (User flows)**

1. Member đăng nhập → Dashboard (balance, dặm năm nay, tiến độ thăng hạng).
2. Member chọn “Gửi yêu cầu tích dặm” → form (loại: chuyến bay / mua sắm / khác) → upload biên lai / vé → submit → request status = “Đang xử lý”.
3. Admin mở list → xem chi tiết (file đính kèm, metadata) → Approve/Reject:  
   * Approve → hệ thống ghi transaction (earn dặm) → cập nhật balance → gọi SessionM event (earn) → user nhận thông báo + lịch sử cập nhật.
   * Reject → status “Từ chối” + lý do → user nhận thông báo.
4. Nếu gọi SessionM thất bại → transaction vẫn lưu local với trạng thái SYNC\_PENDING → queue retry; Admin có thể re-sync.

**Swimlane Diagram (Member / Admin)**

****

# **6. Chi tiết chức năng (Function list) — mỗi function có Acceptance Criteria**

## **6.1 Member Portal — “Gửi yêu cầu tích dặm” (Manual Miles Claim)**

**Form fields:**

* Loại claim: flight | purchase | other (required)
* Ngày giao dịch / ngày chuyến bay (required)
* Mã giao dịch / Booking reference (optional)
* Hành trình (SGN → HAN) hoặc tên shop (tùy loại)
* Số tiền (nếu purchase) — dạng number, currency
* Fare class (economy/premium/business) — nếu flight (optional)
* Mô tả (max 1000 chars)
* Tệp đính kèm (pdf/jpg/png) — tối đa 5 file, mỗi file ≤ 5MB.  
   **Validation & rules:**
* Required fields kiểm tra front + back.
* Attachment type/size validation.
* Duplicate detection heuristic: if (user, transaction\_ref) trùng hoặc hash file trùng trong 30 ngày ⇒ mark possible\_duplicate (auto flag).  
   **Acceptance criteria:**
* User submit thành công → tạo record manual\_claim với status PENDING và gửi email/notification.
* Nếu thiếu field required → trả lỗi rõ ràng.

## **6.2 Admin Portal — Review & Approve/Reject**

**List view:**

* Columns: claim\_id, user (AmazeID), type, submitted\_at, amount\_info, attachments\_count, status, assigned\_admin, actions (view/approve/reject).
* Filters: status, date range, type, possible\_duplicate flag, user\_id.  
   **Detail view:**
* Hiện toàn bộ form, thumbnails cho attachments (click để phóng to), metadata, previous claims history của user.  
   **Actions:**
* Approve (option: accept as-is or override points) — reason optional. Khi Approve: create miles\_transaction (status SUCCESS), update user.balance. Gọi SessionM earn API (see Integration).
* Reject — set status REJECTED + reason (required).
* Bulk approve/reject (nếu nhiều request nhỏ).  
   **Acceptance criteria:**
* Approve → user.balance tăng; transaction traceable; event queued for SessionM; user notified.
* Reject → user notified + reason visible.

## **6.3 Manual “Add transaction” (Admin)**

* Admin có thể thêm giao dịch earn trực tiếp (dùng để sửa balance). Field tương tự claim nhưng không require attachment. Must create audit log.

## **6.4 Transaction History & Dashboard**

* Member sees lịch sử: mỗi record hiển thị: date, type (earn/burn/refund), source (manual\_claim/flight/purchase/admin), points, balance\_after, status.
* Dashboard hiển thị: total miles, miles this year, progress bar toward next tier (configurable target). (Tier calc can be static for POC.)

## **6.5 Reporting (FinOps)**

* Daily & Monthly reports theo spec trong PRD mẫu: daily transactions, total burned/earned, unique redeemers, mapping campaign\_id if any. Xuất CSV/Excel và gửi email theo schedule.

## **6.6 SessionM integration (POC)**

* On approve/create transaction: POST event to SessionM (earn). Nếu real API unavailable, call mock endpoint that simulates success/failure. Log response. Retry policy: exponential backoff, 5 attempts, then put to manual queue for admin re-sync.

# **7. Business rules — Cách tính dặm (algorithms)**

**Nguyên tắc POC:** có 3 nguồn earn: (A) chuyến bay, (B) mua sắm, (C) manual/admin. Mỗi source có công thức riêng, có thể cấu hình bởi admin.

## **7.1 Flight (khoảng cách-based)**

Inputs: distance\_km (nếu có) hoặc from/to → dùng lookup city-pair → khoảng cách km.  
 Formula:

ini

distance\_miles = distance\_km \* 0.621371

base\_miles = round(distance\_miles)

miles\_awarded = base\_miles \* fare\_class\_multiplier \* promo\_multiplier

* fare\_class\_multiplier: economy=1.0, premium=1.25, business=1.5 (configurable).
* promo\_multiplier: default 1.0, có thể set cho campaign.
* Round: round to nearest integer.  
   **Ví dụ:** SGN→HAN ~968 km → distance\_miles ≈ 602 → base\_miles = 602. Nếu economy => awarded = 602.

## **7.2 Purchase (amount-based)**

Inputs: amount\_local\_currency, earn\_rate (points per 1 unit currency, admin-config).  
 Formula:

ini

miles\_awarded = floor(amount \* earn\_rate)

* Ví dụ: nếu earn\_rate = 0.01 (1 point per 100 currency), amount=50,000 → miles\_awarded = floor(50000 \* 0.01) = 500 points.

## **7.3 Manual claim override**

* Member có thể gửi các thông tin trên; Admin có quyền **approve as computed** (system computes using above formulas if applicable) hoặc **override** (set custom points). Nếu override, audit log phải ghi rõ admin\_id + reason.

## **7.4 Capping & Limits**

* Max points per claim: default 100,000,000 (configurable).
* Duplicate detection: if transaction\_ref hoặc attachment hash matches existing APPROVED claim → reject/flag.
* Backdated claims: if claim date > today or older than 365 days → auto flag for manual review.

## **7.5 Refund / Compensation**

* Nếu phát hiện issuance failure (ví dụ calling coupon issuance fails) → system phải automatically create refund transaction: order\_type\_no = 24 (Return burn point) theo PRD mẫu. Hiển thị trong point history.

## **7.6 Quy Tắc Tiers**

#### **1. Tier Entrance Requirements**

* **Điều kiện tham gia:**
  + **Bronze Tier:** Tối thiểu 100 điểm tích lũy.
  + **Silver Tier:** Tối thiểu 500 điểm tích lũy.
  + **Gold Tier:** Tối thiểu 1000 điểm tích lũy.

#### **2. Tier Maintenance Requirements**

* **Điều kiện duy trì cấp bậc:**
  + **Bronze Tier:** Cần tích lũy ít nhất 50 điểm trong 12 tháng.
  + **Silver Tier:** Cần tích lũy ít nhất 300 điểm trong 12 tháng.
  + **Gold Tier:** Cần tích lũy ít nhất 600 điểm trong 12 tháng.

#### **3. Thưởng cho thành viên**

* **Thưởng theo cấp bậc:**
  + **Bronze Tier:** 1 điểm cho mỗi 10 USD chi tiêu.
  + **Silver Tier:** 1 điểm cho mỗi 8 USD chi tiêu.
  + **Gold Tier:** 1 điểm cho mỗi 5 USD chi tiêu.

#### **4. Quy tắc khuyến khích**

* **Khuyến khích nâng cấp cấp bậc:**
  + Thành viên nhận thêm 100 điểm khi nâng cấp từ Bronze lên Silver.
  + Thành viên nhận thêm 200 điểm khi nâng cấp từ Silver lên Gold.

#### **5. Quy tắc đặc biệt**

* **Thưởng cho khách hàng quay lại:**
  + Nếu khách hàng không đạt yêu cầu duy trì cấp bậc trong 12 tháng, họ sẽ tự động hạ cấp, nhưng có thể nhận lại cấp bậc cũ nếu tích lũy đủ điểm trong 3 tháng tiếp theo.

# **8. Data model (tóm tắt bảng chính)**

* users (id, email, name, tier, balance, created\_at, updated\_at)
* manual\_claims (id, user\_id, type, submitted\_data JSON, attachments refs, status[PENDING/APPROVED/REJECTED], flagged\_duplicate BOOL, assigned\_admin, created\_at, updated\_at)
* miles\_transactions (id, user\_id, type[earn/burn/refund], points, balance\_after, source, source\_id, metadata JSON, status[SUCCESS/PENDING/FAILED], created\_at)
* sessionsm\_events (id, transaction\_id, payload, status, attempt\_count, last\_response, created\_at)
* admin\_audit\_logs (id, admin\_id, action, target\_id, note, created\_at)
* reports (pre-generated CSV refs)

(Chi tiết DDL có thể bổ sung nếu cần.)

# **9. API contracts (mẫu)**

Base: /api/v1

**Auth**

* POST /auth/login → {email,password} → token

**Member**

* GET /me → profile + balance
* GET /me/transactions?limit=&page= → list transaction history
* POST /claims → create manual claim (multipart/form-data) → returns claim\_id, status=PENDING
* GET /claims/{id} → get claim detail

**Admin**

* GET /admin/claims?status=&type=&date\_from=&date\_to=&user\_id= → list
* GET /admin/claims/{id} → detail
* POST /admin/claims/{id}/approve → {points: optional, note: optional} → returns transaction\_id, sessionm\_event\_status
* POST /admin/claims/{id}/reject → {reason}
* POST /admin/transactions → create direct transaction (for corrections)
* GET /admin/report/daily?date=yyyy-mm-dd → csv link

**Integration / Webhooks**

* POST /webhooks/sessionm/callback → SessionM callbacks (if any)
* POST /internal/sessionm/emit → internal endpoint to trigger SessionM push (used by job/process)

(Specs JSON schema to be added in dev tasks.)

# **10. SessionM mapping (POC)**

* Event type: earn\_points. Payload example:

json

POST /sessionm/events

{

"event\_type": "earn\_points",

"event\_id": "earn-20250801-0001",

"user\_id": "user\_123",

"points": 1250,

"currency": "VND",

"source": "manual\_claim",

"metadata": {

"claim\_id": 9876,

"transaction\_ref": "INV-12345"

}

}

* On success → mark sessionsm\_events.status = SUCCESS.
* On failure → set PENDING, retry. Admin UI shows queue.

# **11. Acceptance criteria & mapping tới tiêu chí chấm (contest grading)**

Mình map từng tiêu chí chấm (file tiêu chí) sang deliverables để bạn dễ đạt điểm:

* **Hoàn thiện sản phẩm (30đ):** Member + Admin hoàn chỉnh, claim flow ok → checklists & demo script. (tương ứng 15đ BRD + 10đ độ ổn định + 5đ UX).
* **Chất lượng kỹ thuật (15đ):** code structure, separation BE/FE, basic unit tests, Postman collection.
* **Tích hợp SessionM (25đ):** implement call (real or mock), log events, retry queue. (10đ call, 5đ log).
* **Sáng tạo & sản phẩm (15đ):** thêm 1–2 feature extra (ví dụ auto-flag duplicate, admin override reason templates, dashboard nhỏ) — mô tả trong backlog.
* **Teamwork (15đ):** commit history, demo bằng tiếng Anh, phân công task (mô tả phân chia trong timeline).

# **12. Non-functional requirements**

* Auth: JWT token, RBAC (member/admin).
* Performance: list endpoints return < 500ms for <1000 rows.
* File storage: attachments in S3 (or local for POC) with virus-scan step (if possible).
* Logs: audit trail for all admin actions.
* Monitoring: basic error alert (Slack/email).

# **13. Testing / QA**

* Unit tests for business rules (calculation formulas).
* Integration tests for approve → balance change → SessionM event.
* E2E demo script covering: submit claim (member) → review/approve (admin) → SessionM success + balance update.
* Test cases include duplicate claims, oversized attachments, past-date claims, SessionM failures (simulate 500).

# **14. Edge cases & failure handling**

* Duplicate detection & manual override.
* SessionM unavailable → queue + retry. Admin manual re-sync.
* If coupon issuance fails after burning points → automatic refund transaction (order\_type\_no=24).
* Partial approve (admin approves smaller points than requested) — record original requested and approved values.

# **15. Security & privacy**

* Only accept image/pdf attachments; limit file size.
* RBAC: endpoints protected. Logging of admin identity for any balance changes.
* Personal data kept minimal; comply with local privacy rules.

# **16. Reporting & FinOps (the spec from PRD Reward Redemption)**

* Implement daily and monthly report jobs sending CSV to FinOps as in PRD mẫu (fields: transaction\_id, user\_id/Amaze\_ID, points, point\_value\_amount, campaign\_id, created\_at). Schedule & recipients per PRD mẫu.

# **17. Sprint plan (14 ngày POC) — high level**

* Day 0: Kickoff, repo + infra skeleton.
* Day 1–3: BE models (users, claims, transactions), auth, DB.
* Day 4–6: Member UI (submit claim, history), file upload.
* Day 7–9: Admin UI (list, detail, approve/reject), manual transaction.
* Day 10: SessionM mock + integration + retry worker.
* Day 11: Reporting job + FinOps CSV.
* Day 12: Tests, bugfix.
* Day 13: Polish UI, write demo script & docs.
* Day 14: Buffer + rehearsals + final demo.  
   (Phân công: FE/BE/QC/PO hoán đổi role như quy định cuộc thi.)

# **18. Deliverables (khi kết thúc POC)**

* Source code (FE + BE) trên GitHub (private).
* README + runbook (how to run locally + env variables).
* Postman collection / API docs.
* Test cases list + basic unit tests.
* Demo script (EN) + 10–15 phút demo flow.
* Generated sample daily report (Excel) for FinOps.

# **19. Checklist demo (tối thiểu để đạt điểm)**

* Member submit claim → request visible for admin.
* Admin approve → points appear trong balance.
* Event sent to SessionM (or mock) and logged.
* FinOps sample report generated.
* All team members có commit & có phần trình bày bằng tiếng Anh.