**“Local Activity Finder”**

**Handoff Guide**

**Version 1.0**

**10/18/2025**

### **Version 1.0** - *10/18/2025* – M. Rado – Initial document creation.

## 1) What you’re inheriting (quick brief)

* **Backend**: Laravel API exposing /api/geocode/\*, /api/weather, /api/activities, /api/recommendations. Validation, caching, retry/backoff, and **Mock Mode** with fixtures per request via ?mock=<set>.
* **Fixtures**: storage/app/private/fixtures/<set>/{weather,activities}.json (e.g., default, raining). (test/Fixtures for commits, copied on install)
* **Contracts**: Normalized JSON shapes; recommendation sorts indoor/outdoor by weather.
* **CI**: install → lint → test on push; .env.testing + fixtures seeded for backend tests; UI lints with check-only formatting.

## 2) Backlog (next steps, prioritized)

**P0 – must-do before mobile E2E**

1. **Stabilize API contracts**
   * Freeze response shapes; add /v1 prefix (or doc a “v0 freeze” window).
   * Add OpenAPI snippet or typed models for mobile.
2. **Environment parity**
   * Confirm staging base URL; enable MOCK\_MODE=false by default with per-request ?mock= override for QA scripts.
   * Verify rate limits and cache TTLs are configured in env.
3. **Monitoring & error surfaces**
   * Add structured logs for upstream failures; expose minimal error codes to clients.
   * Healthcheck endpoint (readiness/liveness) for mobile CI.
4. **Fixture packs for mobile QA**
   * Ship default, raining, and poor-coverage sets; document scenarios and expected UI states.
5. **Mobile SDK shims (thin)**
   * Define one small client per endpoint (Kotlin/Swift), with timeouts, retries (client-side), and parsing.

**P1 – should-do soon**  
6. **Auth envelope (future-proofing)**

* Define bearer header (no-op for now); document 401/403 semantics.

1. **Pagination & limits**
   * Cap activities results; agree default limit and radius bounds.
2. **Recommendation tuning hooks**
   * Config flags for wind/precip thresholds; expose in config, not API contract.
3. **Accessibility & i18n notes**
   * Confirm units (metric/imperial) and localized category names.

**P2 – nice-to-have**  
10. **Offline UX**  
- Optional offline cache of last-good results; integrate with fixtures for demo flights.  
11. **Analytics**  
- Minimal event taxonomy (search → open detail → get directions).

## 3) Milestones (suggested)

Dates are placeholders—adjust with PM.

* **M1 — API freeze & staging ready (1–2 weeks)**  
  Contracts frozen, /v1 routing (or “v0 freeze”), healthcheck up, fixtures published, smoke tests green.
* **M2 — Mobile integration alpha (2–3 weeks)**  
  Native clients for all endpoints; mock-mode tests; baseline E2E (geocode → weather → recs).
* **M3 — Field test & tuning (1–2 weeks)**  
  Test in 2–3 cities; adjust category mapping; finalize thresholds; bug bash.
* **M4 — Release candidate (1 week)**  
  Perf pass, error budgets, on-call runbook; freeze for app-store submission.

## 4) Implementation plan (how to execute)

**Phase A — Contracts + scaffolding**

* Add /v1 routes or freeze v0 for 30 days; publish OpenAPI or typed client models.
* Create lightweight mobile clients (Kotlin/Swift) with consistent timeouts & error mapping.

**Phase B — Feature wiring**

* Integrate **Geocode** (search & reverse) → pick place → fetch **Weather** → fetch **Activities** → call **Recommendations**.
* Implement chip filters (indoor|outdoor|all), radius selector (1/3/5 km), and retry UX.

**Phase C — Reliability**

* Verify server caching and retry/backoff behavior under throttling.
* Add client-side backoff guardrails; show user-friendly toasts on 429/5xx.

**Phase D — QA & release**

* Drive all happy-path and error-path scenarios with fixtures (?mock=default, ?mock=raining).
* Record contract tests (snapshot JSON) in mobile CI.
* Sign off on SLOs: p95 < 800ms (staging), error rate < 1% (retriable excluded).

## 5) Risk register

| Risk | Impact | Likelihood | Owner | Mitigation |
| --- | --- | --- | --- | --- |
| Provider 429/timeouts (Geoapify/Nominatim) | High | Med | Backend | Exponential backoff + short TTL cache; client retry UX; prefetch on map idle. |
| Data sparsity/unnamed POIs | Med | Med | Mobile | Fallback to category label; guard empty names; UX copy for “nearby but unnamed”. |
| Fixture drift vs live | Med | Med | QA | Keep fixtures minimal but realistic; re-gen quarterly; ensure same normalization in both paths. |
| Contract changes post-integration | High | Low | PM | /v1 freeze; changelog; deprecation window. |
| Localization/units mismatch | Low | Med | Mobile | Unit toggle (metric/imperial); formatters; basic i18n pass. |
| Key handling / attribution | Med | Low | DevOps | Server-side keys; add attribution strings in app settings/about. |
| Cache staleness | Low | Med | Backend | TTLs documented; client “pull to refresh” forces bypass. |

## 6) Final testing checklist (you own)

**Functional**

* Geocode: valid/invalid inputs; reverse pin on map; multiple results sorted by relevance.
* Weather: normalized keys present; unit formatting correct.
* Activities: type filters work; pagination/limit respected; distance formatting accurate.
* Recommendations: rainy/windy → indoor-first; fair → outdoor-first.
* Error paths: 429/5xx show toast; retry works; timeouts handled.

**Fixture-driven (mock mode)**

* ?mock=default renders mixed items; ?mock=raining yields indoor-first list.
* Snapshot tests on JSON for all endpoints (mobile CI).

**Performance**

* p95 latency under target on staging (warm cache).
* Network failure resilience (airplane mode, captive portals).

**Accessibility**

* Dynamic type, color contrast, VoiceOver/TalkBack on key screens.

## 7) Maintenance & operations

**Runbook (short)**

* **Health**: /health (readiness) should report 200; watch error rate & p95.
* **Logs**: filter by request id; include upstream status and retry counts.
* **Config**: change thresholds (wind/precip) via config, not code; restart OK.
* **Incidents**: rotate to mock-mode in QA to isolate provider issues; roll back via env.

**SLOs (initial)**

* **Availability**: 99.5% monthly.
* **Latency**: p95 ≤ 800ms (cached paths faster).
* **Error budget**: 0.5% (excl. client cancels).

**Change management**

* Changelog entries for any response contract change.
* Version bump (v0 → v1) for breaking changes; keep 2-week overlap.

## 8) Deliverables to expect in this repo

* docs/
  + handoff/downstream-mobile.md (this)
  + architecture.md, ADRs (docs/adr/\*.md), research summary
* **Contracts**: OpenAPI JSON/YAML or typed DTOs for mobile
* **Fixtures**: tests/Fixtures/<set> (source of truth) and storage/app/fixtures/<set> for runtime
* **CI**: workflows for backend and UI; .env.testing + seeded fixtures

## 9) Acceptance criteria (for sign-off)

* All endpoints validated and stable; OpenAPI/typed models published.
* Mobile E2E passes with **mock mode** and with **live mode** on staging.
* Risk mitigations in place (backoff, caching, attribution).
* Monitoring + runbook adopted; SLOs tracked.
* Changelog updated; RC tagged.

## 10) One-page implementation calendar (template)

* **Week 1**: Contracts freeze; OpenAPI; staging healthcheck; fixtures verified.
* **Week 2**: Mobile client wiring; mock-mode tests; baseline UI flows.
* **Week 3**: Field tests (3 cities); tuning thresholds/category map.
* **Week 4**: Perf + resilience pass; RC sign-off and handover to release.