Document Control

- **Title:** Student Housing Swipe App — Technical Specification
- **Version:** 1.0 (Initial Draft)
- **Date:** 2025-10-12
- **Owner:** Ty Mabee (Co-Founder)
- **Audience:** Ishit Bhatia (Co-Founder)
- **Status:** Draft for review

1. Executive Summary

An iOS-first mobile application that helps post-secondary students discover, evaluate, and match with off-campus housing using a swipe-based (Tinder-style) interface. The platform supports rapid discovery, rich comparison of leases, verified listings (with third-party photo verification performed by us), shortlisting, filters on favorites, ratings/reviews for landlords (roommate reviews later), a roommate-finder (“Find a Homie”) with group chats, and privacy-preserving profile disclosure during the leasing journey. Initial launch targets iOS with a cloud-hosted backend and an admin web console.

2. Goals

- Rapid, low-friction discovery of student-friendly rentals via swipe UI.
- Each **good swipe** = favorite; each **bad swipe** = ignore (soft-hide), powering personalization.
- Deep comparison of listings across **complete lease feature sets** (term, monthly cost, utilities, cleaners, rules, amenities, etc.).
- Verified listings with **independent media capture** by our verification workflow.
- Favorites workspace with **filters & sort**.
- Secure messaging and **viewing requests**.
- Ratings & reviews for landlords (0–5 stars); roommate reviews post-MVP.
- **“Find a Homie”** roommate & group chat finder, with profile-based matching and explicit consent for information sharing.
- Scalable backend (multi-tenant for multiple schools/regions), robust moderation & safety.
- Analytics instrumentation to measure funnel, matching quality, and trust signals.
- Implement Chexy (?) or some third-party payment app to pay rent.
- Find third-party database to use for storage.

3. Personas & Use Cases

Student Renter: 18–28, budget-constrained; expects quick swipes, comparison, maps, photos/videos, and DMs. **Landlord/Manager:** Lists properties, manages availability, responds to inquiries, passes verification, receives reviews. **Campus Partner (optional):** Housing office that curates/flags listings. **Internal Admin (Tech Lead):** Verifies listings, moderates users/content, handles reports and audits.

Primary Use Cases

1. Student signs up → sets preferences → swipes. Right (good) = favorite; left (bad) = ignore. Student filters favorites and starts comparisons.
2. Student compares 2–5 favorited listings side-by-side across lease attributes; requests a viewing; messages landlord.
3. We verify a listing with independent photos; Admin approves/rejects.
4. Student uses **Find a Homie** to discover roommate matches, form a group chat, and share profile details progressively with consent.
5. Student leaves a landlord review after stay; landlords respond once.

4. Scope (V1 Feature List)

- **Onboarding & Auth:** Apple Sign-In, Email+OTP; optional campus email verification.
- **Profile & Preferences:** program/major, school, bio, interests; housing preferences (budget, radius, term, pets, furnished, accessibility).
- **Discovery:** swipe deck; list & map views; listing details with rich media; similar listings.
- **Swipe Semantics:** Good swipe → add to Favorites; Bad swipe → Ignore (soft-hide); undo (last action) limit.
- **Favorites Workspace:** filter/sort favorites (price, distance, term fit, rating, newest).
- **Comparison:** side-by-side comparison view of selected listings with normalized lease attributes.
- **Verification:** third-party (us) verification with independent photos; verification badge & audit trail.
- **Messaging & Viewings:** 1:1 chat; propose/accept viewing slots; ICS export.
- **Reviews:** 0–5 star + text for landlords; photo evidence optional; report/appeal flow. (Roommate reviews V1.1)
- **Find a Homie:** roommate discovery, profile matching, interest tags, privacy-guarded info reveal, group creation & group chat.
- **Trust & Safety:** reporting, moderation queue, fraud heuristics, suspensions/bans, audit logs.
- **Notifications:** push, email (critical), in-app inbox.

- **Admin Console (Web):** verification tools, moderation, metrics, feature flags.
- **Analytics:** tracking across swipe→favorite→compare→message→viewing; trust metrics; roommate match funnel.

5. System Architecture

5.1 High-Level Diagram (conceptual)

- **iOS App (SwiftUI)** → **API Gateway** → **Core Backend Services**: Auth, Users/Profiles, Listings, Search/Ranking, Swipe/Interaction, Favorites, Comparison, Verification, Reviews, Messaging, Roommates/Groups, Media, Moderation, Notifications, Analytics.
- **Data Stores:** PostgreSQL + PostGIS; Redis (cache/session/decks); OpenSearch (search); S3 + CloudFront (media); SQS/Kafka (async); Feature Flags (Config service).
- **Admin Web:** Next.js→Admin API with elevated scopes.
- **Third-party:** APNs, SES/SendGrid, MapKit/Mapbox geocoding, MediaConvert/FFmpeg, Sentry, Datadog, Amplitude/Segment.

5.2 Tech Stack (Recommended)

- **Mobile:** Swift 5+, SwiftUI, async/await, MapKit, CoreLocation, GRDB/SQLite or Realm.
- **Backend:** TypeScript + NestJS (Node 20), PostgreSQL 15 + PostGIS, Redis 7, OpenSearch 2.x, AWS (ECS Fargate/EKS, RDS, ElastiCache, OpenSearch, S3, CloudFront, SQS, WAF, ACM). IaC Terraform. CI/CD GitHub Actions.
- **Admin Web:** Next.js 14, TypeScript, Tailwind, shadcn/ui.

6. Detailed Component Specs

6.1 Authentication & Identity

- Methods: Sign in with Apple, Email+OTP; optional Google later. Campus verification via edu/college domain token.
- Tokens: OAuth2.1/OIDC; access (15m) + rotating refresh; device binding; PKCE; TLS 1.2+; rate limits.
- Roles & Scopes: STUDENT, LANDLORD, ADMIN; fine-grained permissions for verification and moderation.

6.2 Listings Service

- CRUD with versioning; availability windows; amenities/utilities schema; policy/rules fields (quiet hours, pets, cleaners, furnishings, parking, accessibility).

- Geocoding + campus distance (PostGIS).
- Workflow: DRAFT → PENDING_VERIFICATION → VERIFIED → ACTIVE → PAUSED → ARCHIVED → REMOVED.

6.3 Search & Ranking (Swipe Deck)

- Index: denormalized docs in OpenSearch with geo filters.
- Scoring inputs: distance, price fit, availability vs. move-in, bed/bath/type, amenities, recency, quality (complete profile, verified), landlord rating.
- Personalization: online re-rank based on recent favorites/ignores; exploration epsilon.
- Deck lifecycle: candidate query → score → business rules → Redis deck per user → consume on swipe; replenish below threshold; hourly freshness.

6.4 Swipe/Interaction Service

- Actions: LIKE (good), PASS (bad), optional SUPERLIKE.
- Behavior: LIKE ⇒ create Favorite; PASS ⇒ create Ignore (TTL or permanent unless filters change materially). Undo last action once.
- Idempotency keys, anti-spam throttling, anomaly detection.

6.5 Favorites Service

- Store/set of favorited listings; filter/sort endpoints; computed labels (“verified”, “price-within-budget”, “near campus”, “new”).
- Bulk actions (select, compare, remove, share).

6.6 Comparison Service

- Normalization layer to map heterogeneous lease attributes into a canonical schema (numbers, booleans, enums, money, dates).
- Side-by-side up to 5 listings; diff highlighting; export/share summary (PDF later).
- Data completeness scoring; prompts landlords to fill gaps.

6.7 Verification Service (Third-Party by Us)

- Flow: landlord submits → pre-checks → we schedule independent photo capture or remote guided capture → tamper checks (EXIF, GPS, hash) → reviewer approval → badge issuance.
- Evidence store with immutable audit log; re-verification on significant changes or age (e.g., 12 months).

6.8 Reviews Service

- Entities: Review(id, target_type=LANDLORD|ROOMMATE, target_id, reviewer_id, rating 0–5, text, media?).

- Landlord reviews enabled V1; roommate reviews V1.2 with consent & safeguards.
- Moderation: profanity, doxxing filters, dispute/resolution workflow; one right of reply.
- Aggregation: Bayesian average to prevent early skew; expose count and distribution.

6.9 Messaging & Viewing Requests

- 1:1 threads (student↔landlord); per-listing context; attachments (images only V1).
- Viewing: propose slots, accept/decline, reminders; ICS export.

6.10 Roommates ("Find a Homie") & Groups

- **Profiles:** academic program, year, campus, budget share, lifestyle (quiet hours, cleanliness, guests, smoking, pets), interests, personality tags.
- **Matching:** similarity scoring + hard constraints (gender preferences if enabled, pet allergies, budget ranges); opt-in discovery.
- **Groups:** create group, invite, join codes; **group chat** channel; shared favorites & comparisons.
- **Privacy & Consent:** progressive disclosure—show high-level compatibility first; reveal contact/social only with explicit consent or on group formation step.

6.11 Media Handling

- Client compression; pre-signed S3 upload; server normalization (WebP/AVIF+JPEG), EXIF strip; content moderation; duplicate detection.
- Video optional V1.1 via HLS pipeline.

6.12 Moderation & Trust

- Reporting pipeline with SLA; actions: warn, hide, suspend, ban; ban-evasion checks.
- Fraud heuristics: price anomalies, external contact leakage, geo mismatches, duplicate photos.
- Full audit logging of privileged actions.

6.13 Notifications

- APNs topics: messages, viewings, verification, group activity.
- Email: transactional only.
- In-app inbox for low-priority events.

6.14 Admin Web Console

- Verification workbench (photo sets, checklists), moderation queues, review disputes, metrics dashboards, feature flags.

7. Data Model (Core Entities)

User(id, role, email, phone?, campus_id?, first_name, last_name, avatar_url, program, school, interests_json, bio, created_at, updated_at, status, preferences_json)

RoommateProfile(user_id FK, lifestyle_json, budget_min_cents, budget_max_cents, roommate_count_pref, hard_constraints_json, discovery_opt_in)

LandlordProfile(user_id FK, company_name?, verification_status, contact_methods)

Listing(id, landlord_user_id FK, title, description, price_cents, currency, address, lat, lng, campus_id, distance_km, bedrooms, bathrooms, room_type, sqft?, lease_term_months, available_from, available_to?, furnished, pets_allowed, utilities_included_json, amenities_json, rules_json, cleaners_included bool, cleaners_details?, media_cover_url, status, created_at, updated_at, verification_badge, verification_version)

ListingMedia(id, listing_id FK, type [image|video], url, sort_order, captured_by [landlord|verification], captured_at)

Swipe(id, user_id FK, listing_id FK, action [LIKE|PASS|SUPERLIKE], context_json, created_at)

Favorite(id, user_id FK, listing_id FK, created_at)

Ignore(id, user_id FK, listing_id FK, created_at, ttl_at?, reason)

ComparisonSet(id, user_id FK, listing_ids[], created_at, title?)

Thread(id, listing_id FK, student_id FK, landlord_id FK, status)

Message(id, thread_id FK, sender_id FK, body, attachments_json, created_at, read_at)

Review(id, target_type, target_id, reviewer_id, rating_smallint, text, media_json?, created_at, status)

Group(id, name, owner_id FK, campus_id, created_at)

GroupMembership(group_id FK, user_id FK, role [owner|member], joined_at)

GroupChatMessage(id, group_id FK, sender_id FK, body, attachments_json, created_at)

Campus(id, name, domain_list, lat, lng, radius_km)

VerificationCase(id, listing_id FK, stage, evidence_json, reviewer_id?, opened_at, closed_at, decision)

Report(id, reporter_id, target_type, target_id, reason_code, details, state, created_at, resolved_at)

AuditLog(id, actor_id, action, target_type, target_id, metadata_json, created_at)

Indexes

- GIST geo on (lat,lng); composite: (user_id, listing_id) on Favorite/Ignore/Swipe; (landlord_user_id, status) on Listing; (target_type,target_id) on Review; (group_id,created_at) on GroupChatMessage.

8. API Specification (REST JSON)

8.1 Auth

- `POST /v1/auth/apple`
- `POST /v1/auth/email/request`
- `POST /v1/auth/email/verify`
- `POST /v1/auth/refresh`

8.2 Users & Profiles

- `GET /v1/me`
- `PATCH /v1/me`
- `PUT /v1/me/preferences`
- `GET /v1/users/{id}` (public profile fields by role/consent)
- `PUT /v1/roommates/profile`
- `GET /v1/roommates/matches` (query: constraints)

8.3 Listings

- `GET /v1/listings` (filters: campusId, priceMin/Max, distanceKm, beds, baths, amenities[], rules[], term, availableFrom, verifiedOnly, ratingMin, sort)
- `GET /v1/listings/{id}`
- `POST /v1/listings` (landlord)
- `PATCH /v1/listings/{id}`
- `POST /v1/listings/{id}/submit`
- `POST /v1/listings/{id}/state`

8.4 Swipe, Favorites & Ignores

- `GET /v1/deck`
- `POST /v1/swipes` {listingId, action, context}
- `GET /v1/favorites` (filters/sort: price, distance, term, verified, rating, newest)
- `DELETE /v1/favorites/{listingId}`
- `GET /v1/ignores`
- `DELETE /v1/ignores/{listingId}`

8.5 Comparison

- `POST /v1/comparisons` {listingIds[]}
- `GET /v1/comparisons/{id}`
- `DELETE /v1/comparisons/{id}`

8.6 Verification

- `GET /v1/admin/verification/cases`
- `POST /v1/admin/verification/cases/{id}/decision` {approve|reject, notes}
- `POST /v1/uploads/presign` (scope=verification)

8.7 Reviews

- `POST /v1/reviews` {targetType, targetId, rating, text}
- `GET /v1/reviews` (query by target)
- `GET /v1/reviews/summary` (target)
- `POST /v1/reviews/{id}/report`
- `POST /v1/reviews/{id}/reply` (landlord only)

8.8 Messaging & Viewings

- `POST /v1/threads` {listingId, toUserId}
- `GET /v1/threads`
- `GET /v1/threads/{id}`
- `POST /v1/threads/{id}/messages`
- `POST /v1/listings/{id}/viewings` {proposedSlots[]}
- `PATCH /v1/viewings/{id}` {accept|decline|cancel}

8.9 Groups & Group Chat

- `POST /v1/groups` {name, campusId}
- `POST /v1/groups/{id}/invite` {userId|email}
- `POST /v1/groups/{id}/messages` {body}
- `GET /v1/groups/{id}/messages`
- `POST /v1/groups/{id}/favorites/share` {listingIds[]}

8.10 Media & Uploads

- `POST /v1/uploads/presign` {contentType, sizeBytes, scope}
- `POST /v1/uploads/complete` {key, hash}

Standards: JSON:API or simple JSON; error envelope `{error:{code,message,details}}`; idempotency via `Idempotency-Key`.

9. iOS App Architecture

- **Pattern:** MVVM + SwiftUI; async/await networking; local cache for deck, favorites, messages, groups.
- **Modules:** Auth, Onboarding, Home/Deck, Map, Search/Filters, Listing Detail, **Favorites**, **Comparison**, Messages, **Find a Homie**, Groups/Group Chat, Profile, Settings.
- **Key Screens:** Swipe Deck; Favorites (filters/sort); Comparison (matrix diff); Verification Badge details; Reviews; Roommate Matches; Group Chat.
- **Accessibility:** Dynamic Type, VoiceOver, WCAG 2.1 AA; large tap targets.

10. Matching & Scoring

10.1 Listing Relevance (initial heuristic)

$$\text{score}(L, U) = w_d f_d + w_p f_p + w_a f_a + w_b f_b + w_m f_m + w_r f_r + w_l f_l$$

- `f_l`: landlord rating prior (Bayesian); others as defined previously (distance, price, availability, beds/baths, amenities, recency, quality).
- Online learning loop adjusts weights using swipe outcomes (favorites) with guardrails.

10.2 Roommate Matching (Find a Homie)

- Hard filters (budget overlap, pet constraints, location/campus, roommate count) → similarity (cosine/Jaccard over interests & lifestyle) → exploration.
- Progressive disclosure: only compatibility score and non-sensitive fields until consented.

11. Privacy, Security & Compliance

- **Progressive Disclosure:** Stage-gated reveal of personal info (contacts/socials only upon group formation or explicit consent). Consent artifacts stored with timestamps.
- **Data Residency:** Primary in ca-central-1 (PIPEDA). Encryption at rest (KMS) & in transit.
- **Access Controls:** RBAC; least privilege IAM; row-level checks enforced in service layer.
- **Trust & Safety:** content moderation; block/report; rate limits; audit logs.
- **Legal:** ToS, Privacy Policy, Acceptable Use; age 16+ default (configurable); roommate review policy.

12. Non-Functional Requirements (NFRs)

- **Availability:** 99.9% core APIs.
- **Latency p95:** Deck <300ms (cached), Search <600ms, Favorites fetch <300ms, Comparison build <500ms, Message send <250ms.
- **Scale:** tbd.
- **Security:** tbd.
- **Observability:** tbd.
- **Localization:** tbd.
- **Accessibility:** tbd.

13. DevOps & Environments

- Trunk-based dev; Dev→Staging→Prod; separate AWS accounts.
- CI/CD: GitHub Actions; build/test/lint; IaC via Terraform; blue/green; Flyway/Prisma migrations; PITR backups.
- Secrets via OIDC→AWS; no long-lived keys.

14. Quality Strategy

- Unit/integration/e2e; contract tests (Pact) for mobile↔API; performance tests (K6) on deck, favorites, comparison, messaging.
- Security: CodeQL, ZAP; dependency scanning; review moderation test cases.
- Beta via TestFlight; analytics validation plan.

15. Analytics & Metrics

- **Core Funnel:** impressions → swipes → favorites rate → compare sessions → messages → viewing requests → attended viewings.
- **Trust:** verification pass rate, time to verify, report rate, takedown rate.
- **Reviews:** avg rating, distribution, review volume, dispute rate.
- **Roommates:** match views, group creations, group activity, group→lease conversion proxy.

16. Accessibility & Inclusive Design

- Contrast/alt text; content warnings; avoid discriminatory filters; plain-language copy.

17. Risks & Mitigations

- **Fraud/Scams:** verification & moderation, anomaly detection, user education.
- **Legal/Policy:** fair housing analogues, reviews liability—add moderation & appeals.
- **Cold Start:** seed inventory, campus partnerships, landlord onboarding, referral incentives.
- **Privacy:** progressive disclosure, consent logs, minimal data sharing by default.

18. Roadmap & Milestones (Indicative)

- **M0 Discovery (2–3 wks):** UX research, legal review, verification playbook.
- **M1 Foundation (6–8 wks):** Auth, profiles, listings CRUD, media uploads, search, deck, swipe→favourite/ignore, favourites filters, admin MVP.
- **M2 Comparison & Verification (4–6 wks):** comparison matrix, verification pipeline & badges.
- **M3 Messaging & Viewings (4–6 wks):** threads, push notifications, viewing requests.
- **M4 Reviews & Find a Homie (4–6 wks):** landlord reviews, roommate matching, groups & group chat (beta).
- **M5 Beta (3–4 wks):** campus pilot, perf, analytics, moderation hardening.
- **M6 GA:** launch; post-launch V1.0: roommate reviews, video, PDF export for comparisons.

19. Open Questions

- Scope of remote vs. in-person verification and SLAs per campus.
- Roommate review safeguards (cool-down, mutual review windows, identity confirmation).
- Undo limits & daily swipe caps.
- Exposure of landlord contact info pre-verification.

20. Appendices

20.1 Example JSONs

Favorite List (filtered)

```
{
  "filters": {"verifiedOnly": true, "sort": "priceAsc", "priceMax": 150000},
  "items": [
    {"listingId": "lst_123", "title": "2BR near campus", "price_cents": 120000, "verified": true, "rating": 4.6},
    {"listingId": "lst_558", "title": "Studio downtown", "price_cents": 14500
0, "verified": true, "rating": 4.2}
  ]
}
```

Comparison Matrix (normalized)

```
{
  "listingIds": ["lst_123", "lst_558"],
  "attributes": [
    {"key": "monthly_cost_cents", "values": [120000, 145000]},
    {"key": "lease_term_months", "values": [12, 8]},
    {"key": "cleaners_included", "values": [true, false]},
    {"key": "utilities_included", "values": [["WATER","HEAT"],["NONE"]]}
  ]
}
```

Review Payload

```
{ "targetType": "LANDLORD", "targetId": "ll_42", "rating": 5, "text": "Responsive and fair." }
```

End of Spec v1.0