**Assessment 2 Project Report (Group)   
Fortune Living Victoria – Real Estate Web Application**

**Course/Unit:** *COIT20273* **Team:** [Abhaykumar Dobariya, Rahul Patel, Harsh Adhiyol, Parthesh Patel] **Date:** [4 sep 2025]

# 1. User Stories (Formal) with Acceptance Criteria

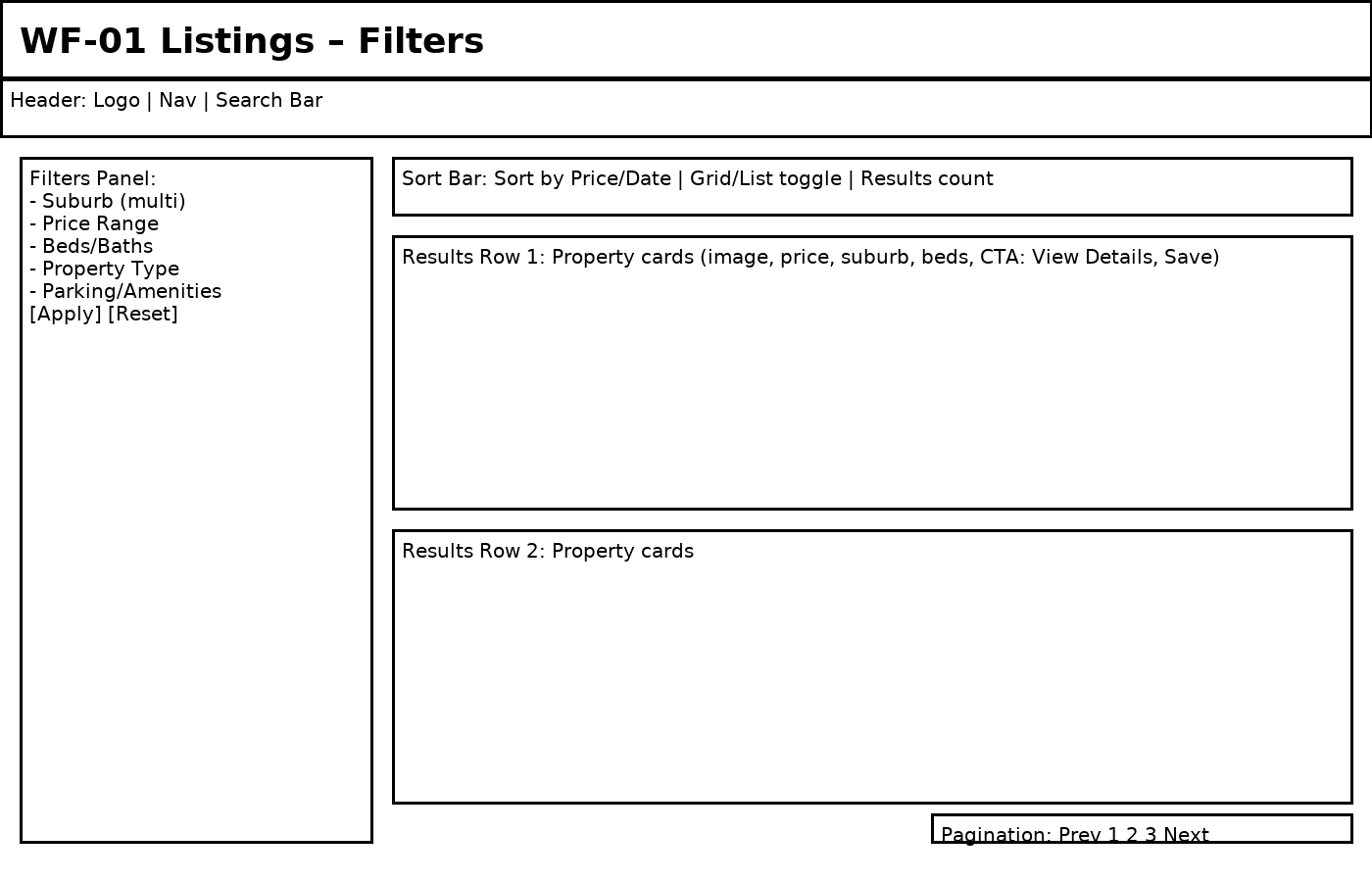
Actors: Visitor/Buyer (anonymous), Agent, Admin. Priorities use MoSCoW (Must/Should/Could). Each story traces to planned wireframes (WF-xx) that will be provided in Section 2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Actor | User Story (As a…, I want…, so that…) | Priority | Acceptance Criteria (Given/When/Then) | Traces (Wireframes) | Notes |
| US-01 | Visitor | As a visitor, I want to filter property listings by suburb, price range, bedrooms, bathrooms, property type, and parking, so that I can quickly find suitable properties. | Must | • Given properties exist with varied attributes, when I select one or more filters and click Apply, then the results update to show only matching properties and display the total count.  • Given I clear filters, when I click Reset, then all filters return to default and full results display. | WF-01 Listings (filters panel), WF-02 Listings Results |  |
| US-02 | Visitor | As a visitor, I want to view a property details page with image gallery, price, address/suburb, description, amenities, agent contact, and a map, so that I can evaluate the property. | Must | • Given a property exists, when I click a result card, then I am taken to the property details page showing images, key facts, amenities, and agent contact.  • Given the details page, when I click on a thumbnail, then the main image changes accordingly; when I click Map, then a map view shows the location. | WF-03 Property Details |  |
| US-03 | Visitor | As a visitor, I want to submit an enquiry form for a property, so that the agent can contact me about that property. | Must | • Given I am on a property details page, when I fill in name, email/phone, and message and click Send, then the system validates inputs and creates an enquiry linked to that property.  • Given my enquiry is submitted successfully, when processing completes, then I see a success message/confirmation and the agent receives a notification. | WF-03 Property Details (Enquiry), WF-04 Enquiry Confirmation |  |
| US-04 | Visitor | As a visitor, I want to request an inspection (choose preferred date/time), so that I can schedule a viewing with the agent. | Should | • Given available date/time inputs, when I submit a viewing request, then the system records the request and notifies the agent.  • Given a conflicting/invalid time, when I attempt to submit, then I see an error explaining the issue and how to correct it. | WF-05 Request Inspection |  |
| US-05 | Visitor | As a visitor, I want to see the property location on an interactive map, so that I can judge the neighbourhood and commute. | Should | • Given a property has coordinates, when I open the map tab, then a map renders with a marker at the property location.  • Given I zoom or pan, when I interact with the map, then the map responds and maintains the property marker. | WF-03 Property Details (Map) |  |
| US-06 | Visitor (registered) | As a registered buyer, I want to save properties to a favourites list, so that I can revisit them later. | Could | • Given I am signed in, when I click Save on a listing, then the property appears in my Saved list.  • Given a saved property, when I click Remove in my Saved list, then it is removed and the list updates. | WF-06 Saved Listings |  |
| US-07 | Agent | As an agent, I want to log in securely, so that I can manage my property listings and view enquiries. | Must | • Given I have valid credentials, when I enter them on the login page and submit, then I am authenticated and redirected to the Agent Dashboard.  • Given invalid credentials, when I attempt login, then I see an error without revealing which field is incorrect. | WF-07 Agent Login, WF-08 Agent Dashboard |  |
| US-08 | Agent | As an agent, I want to create, edit, publish/unpublish, and archive listings with multiple images, amenities, and key attributes, so that I can keep my portfolio up to date. | Must | • Given I am on the listing form, when I provide required fields (title, price, address/suburb, bedrooms, bathrooms, type) and save, then a property record is created/updated with images and amenities.  • Given a listing status control, when I set Publish and save, then the listing becomes visible on the public site; when Unpublish/Archive, then it is hidden from the public site. | WF-09 Listing Form (Create/Edit), WF-10 Manage Listings |  |
| US-09 | Agent | As an agent, I want to view and respond to enquiries for my listings, so that I can follow up with potential buyers promptly. | Must | • Given enquiries exist for my listings, when I open Enquiries, then I see a list with property, sender name, contact, and message preview sorted by newest.  • Given an enquiry, when I click it, then I can view full details and copy the contact information to reply; the system marks it as Read. | WF-11 Enquiries Inbox (Agent) |  |
| US-10 | Admin | As an admin, I want to manage users and roles (Admin, Agent, Buyer), so that access is controlled and extendable as the system grows. | Must | • Given I am an admin, when I create a user and assign one or more roles, then the user can log in and access only the permitted sections.  • Given role updates, when I change a user’s roles, then the user’s permissions update immediately upon next request. | WF-12 Admin Users & Roles |  |
| US-11 | Admin | As an admin, I want to manage master data (amenities, property types, suburbs), so that agents use consistent taxonomy across listings. | Should | • Given master data exists, when I add/edit/deactivate an amenity or type, then agents see the updated options on listing forms.  • Given a deactivated amenity, when agents open existing listings, then the amenity remains on old records but is not selectable for new assignments. | WF-13 Admin Master Data |  |
| US-12 | Visitor | As a visitor, I want to search using natural language (e.g., “3-bed under $700k in Tarneit with parking”), so that I can find properties without clicking many filters. | Could (AI) | • Given I enter a natural-language query, when I submit search, then the system parses the query into filters (beds, price, suburb, amenities) and applies them to the listing results.  • Given an unrecognized term, when I search, then the system returns results for recognized parts and shows a hint indicating which terms were not understood. | WF-01 Listings (search bar) |  |
| US-13 | Agent | As an agent, I want AI-suggested first-reply templates for enquiries (editable), so that I can respond faster while maintaining quality. | Could (AI) | • Given an enquiry message, when I open it, then the system generates a reply draft referencing the property and buyer’s questions for me to edit and send via email client.  • Given the draft suggestion, when I click Regenerate, then a new draft is produced without overwriting the previous saved notes unless I confirm. | WF-11 Enquiries Inbox (suggest reply) |  |
| US-14 | Visitor | As a visitor, I want clear validation and feedback on forms (enquiry, inspection), so that I can correct errors and know when submissions succeed. | Must | • Given required fields, when I submit with missing or invalid inputs, then field-level error messages explain what to fix and the submit button is disabled until valid.  • Given a successful submission, when processing completes, then I see a success toast/confirmation and receive guidance on next steps. | WF-03 Property Details (Enquiry), WF-05 Request Inspection |  |

Note: Wireframe IDs (WF-xx) will match the figures added in Section 2. AI-related stories (US-12, US-13) are explicitly tied to system functionality to satisfy the 'AI component' requirement.

# 2. Wireframes & Screen Descriptions

This section provides wireframes (PNG) for all key user interfaces with navigation and action descriptions. Each figure aligns with user stories in Section 1 (see 'Traces').



**Figure WF-01: Listings – Filters**

Covers user stories: US-01, US-12, US-14

Navigation:

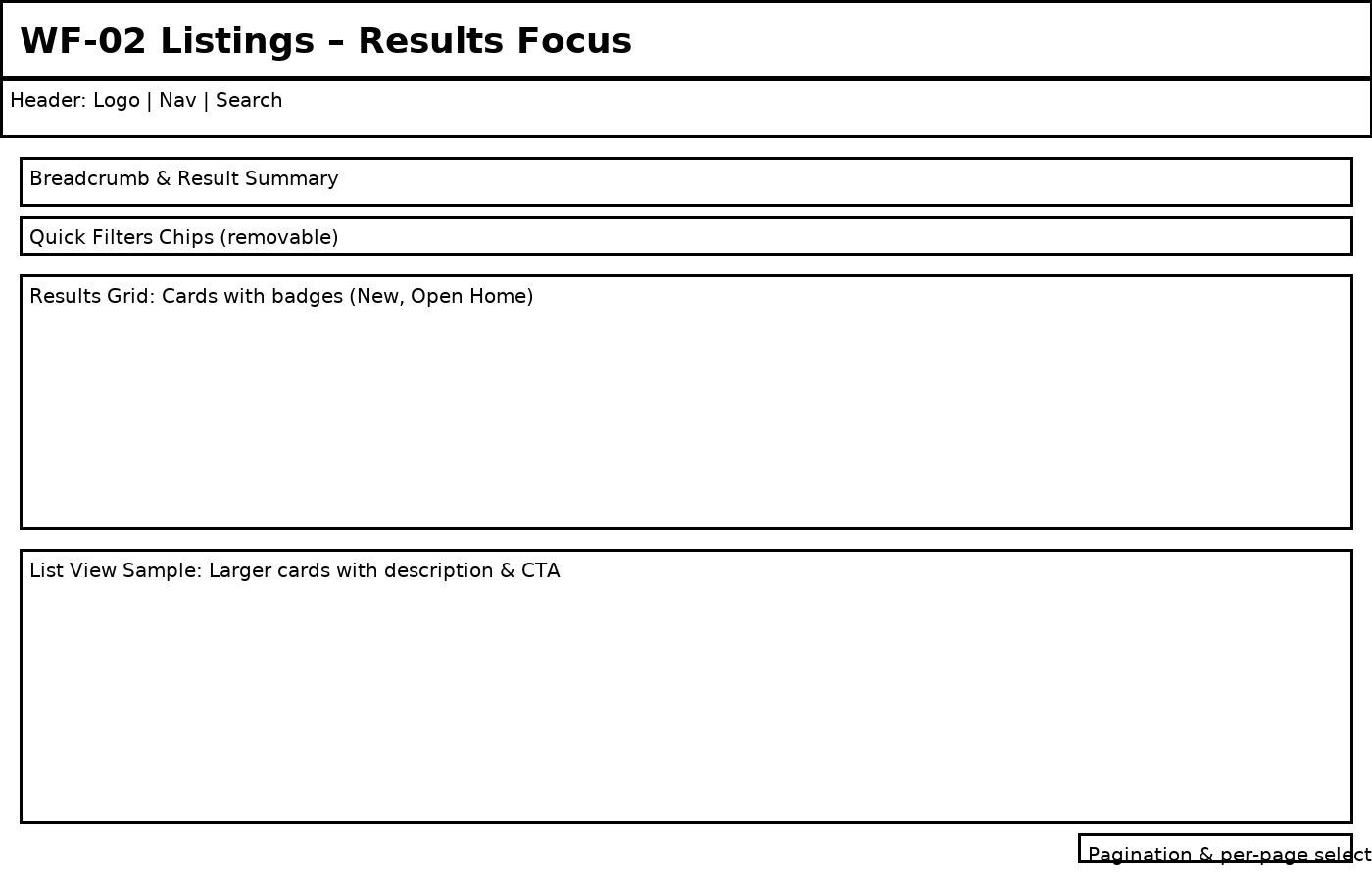
• From Home → Listings.

• Apply/Reset filters updates results without page reload.

Active elements & actions:

• Filter controls: multi-select suburb; numeric ranges for price; toggles for beds/baths/type/parking/amenities.

• Sort by price/date; switch grid/list view; pagination; Save card adds to Saved (if signed in).



**Figure WF-02: Listings – Results Focus**

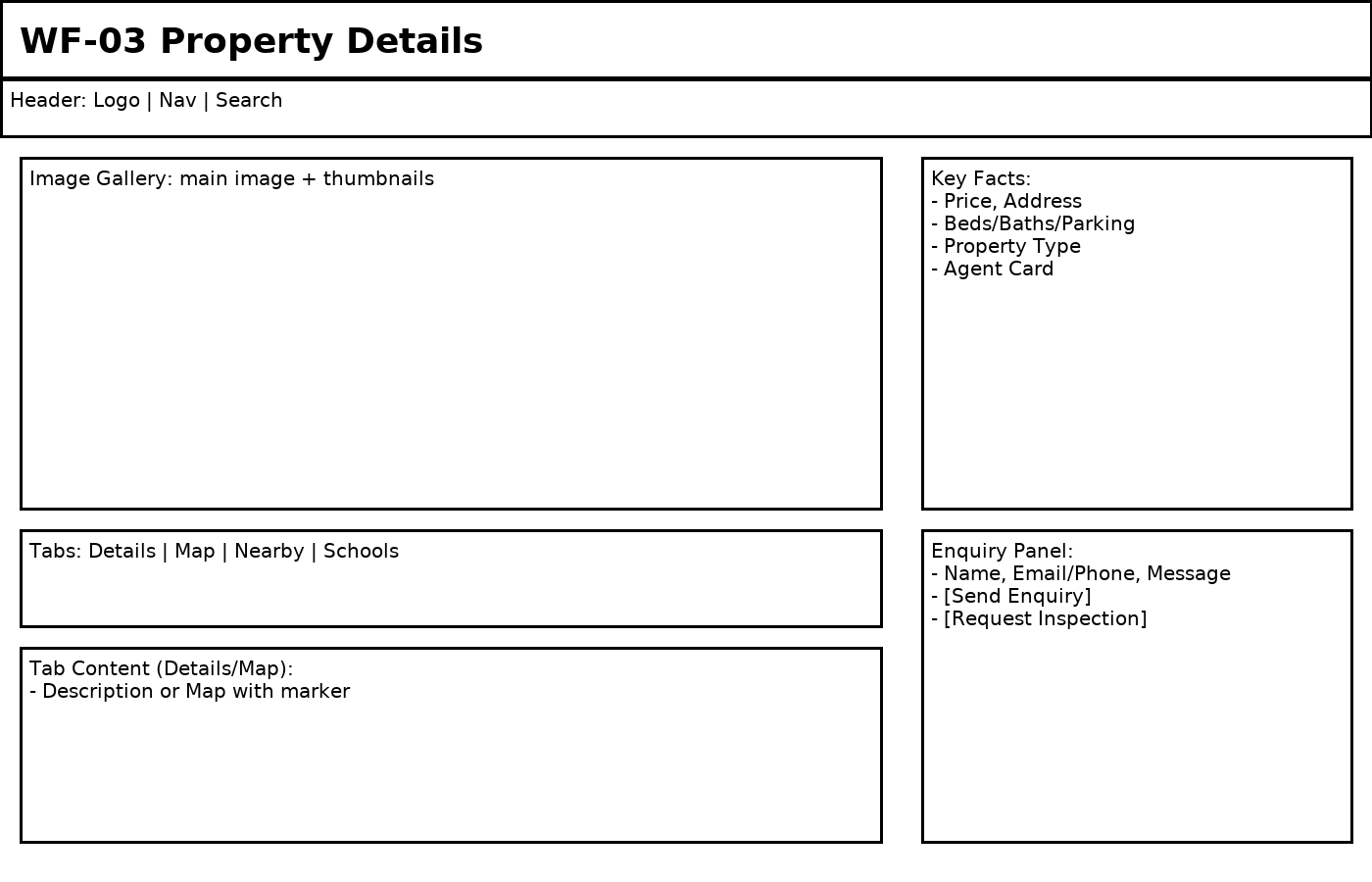
Covers user stories: US-01, US-06

Navigation:

• Listings → Results view (grid/list).

Active elements & actions:

• Remove filter chips to widen results; change per-page; click a card → Property Details; Save/Unsave from card.



**Figure WF-03: Property Details**

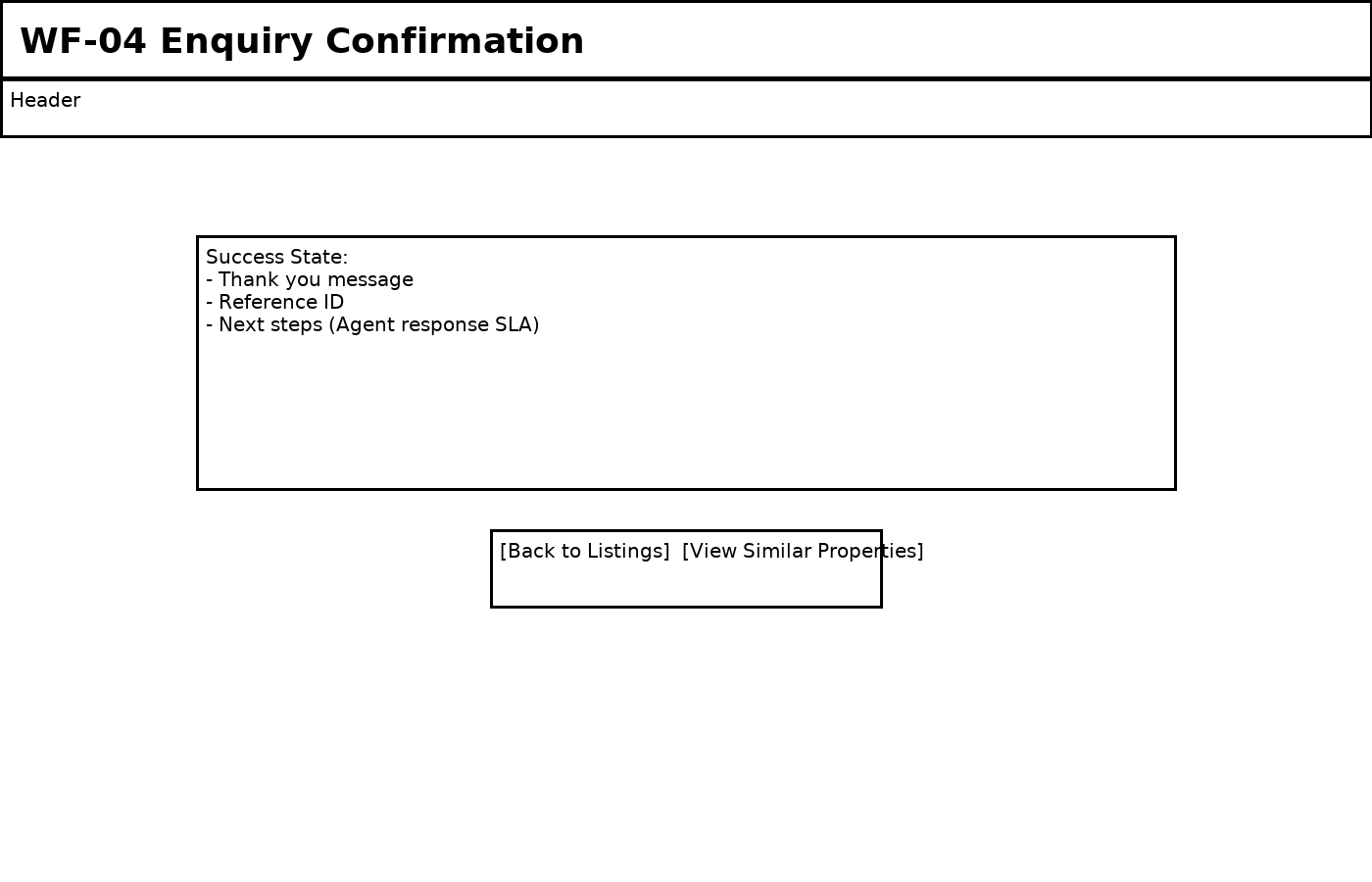
Covers user stories: US-02, US-03, US-05, US-14

Navigation:

• Listings → Property Details.

Active elements & actions:

• Thumbnail click swaps main image; tabs switch content; Enquiry validates fields; Request Inspection opens dedicated screen.



**Figure WF-04: Enquiry Confirmation**

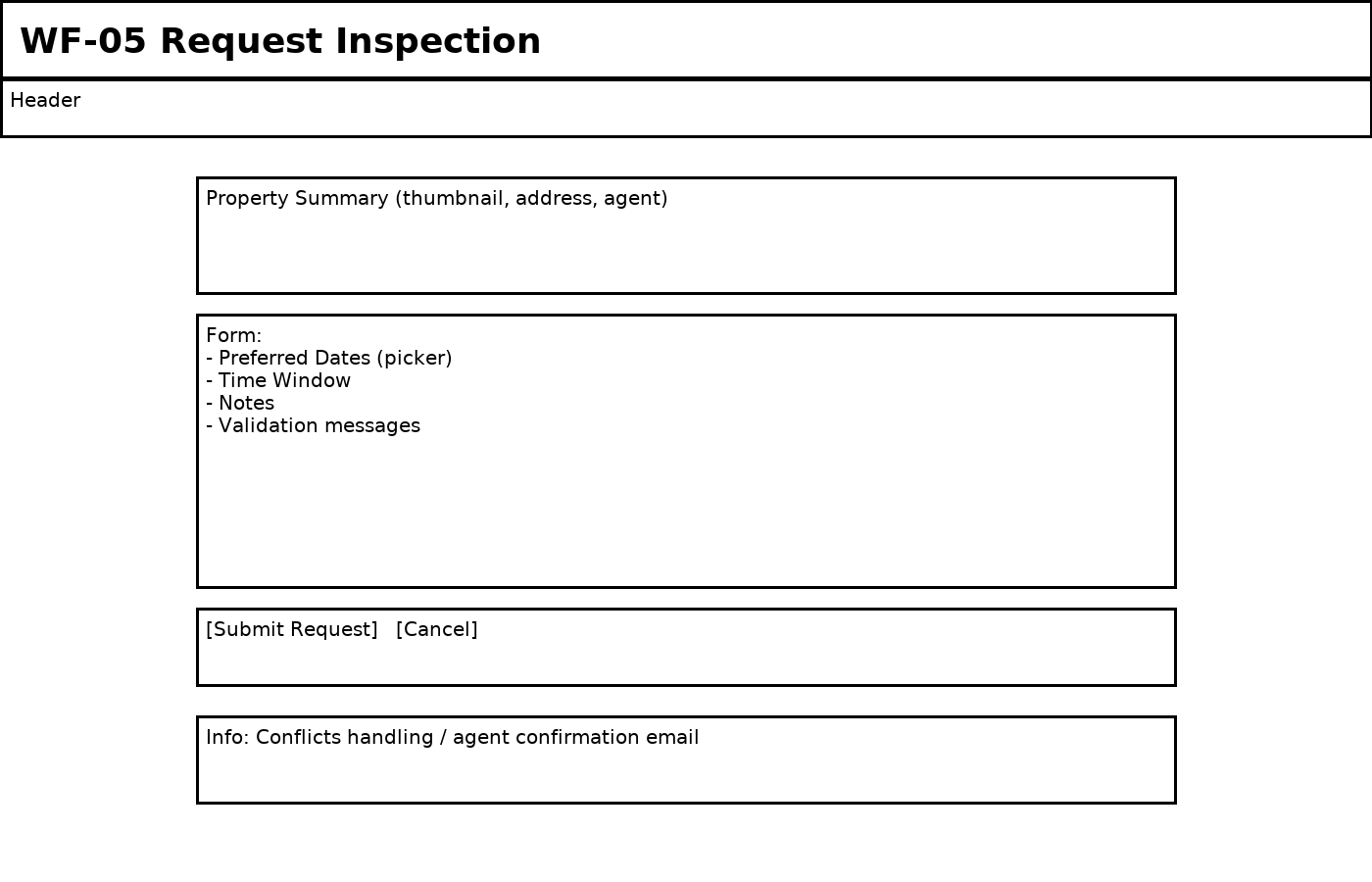
Covers user stories: US-03

Navigation:

• After successful Enquiry submission.

Active elements & actions:

• Back to Listings; View Similar triggers filtered search by suburb/price/type.



**Figure WF-05: Request Inspection**

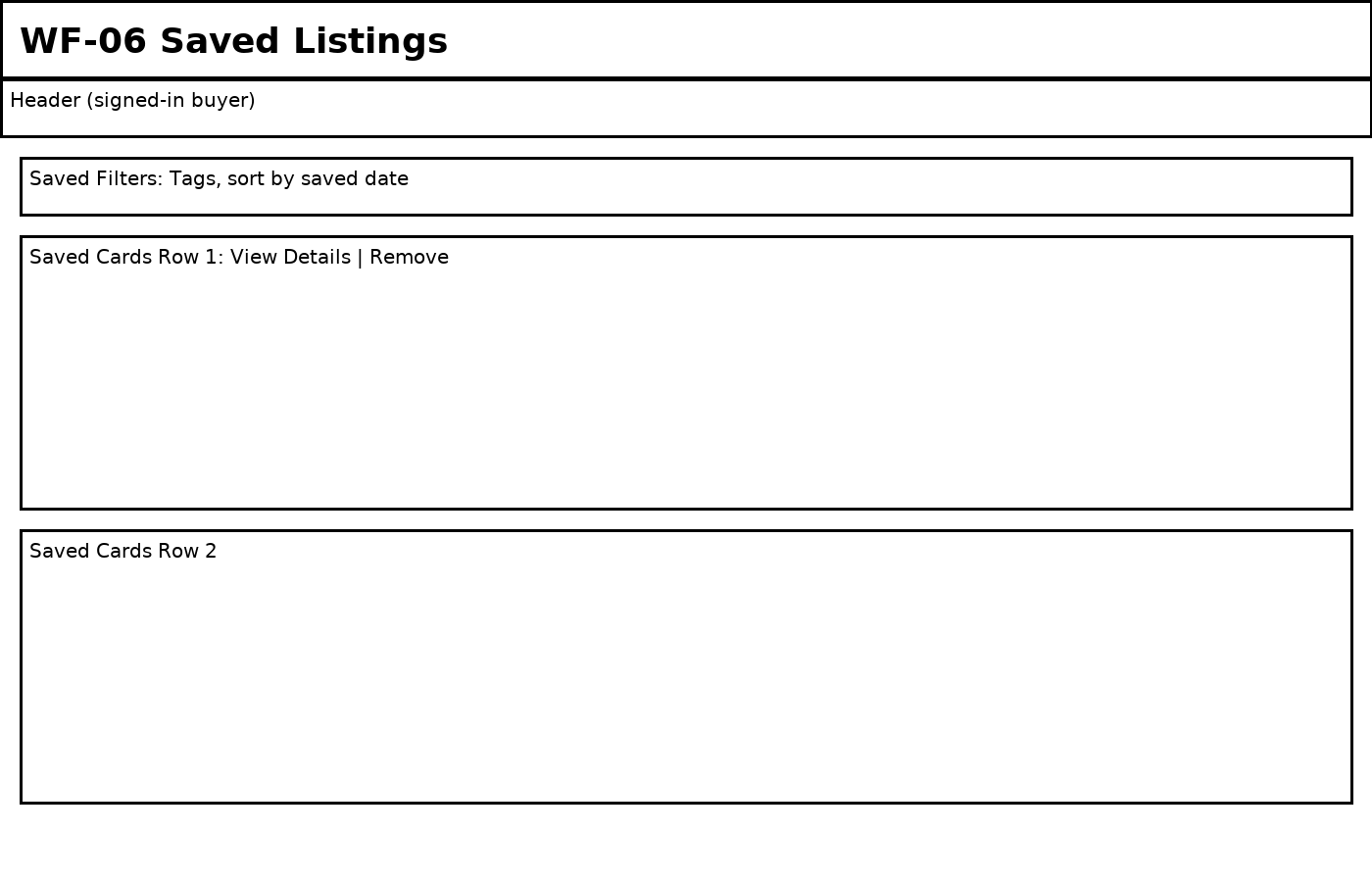
Covers user stories: US-04, US-14

Navigation:

• Property Details → Request Inspection.

Active elements & actions:

• Date/time validation; conflicting times show inline errors; Submit records request and notifies agent.



**Figure WF-06: Saved Listings**

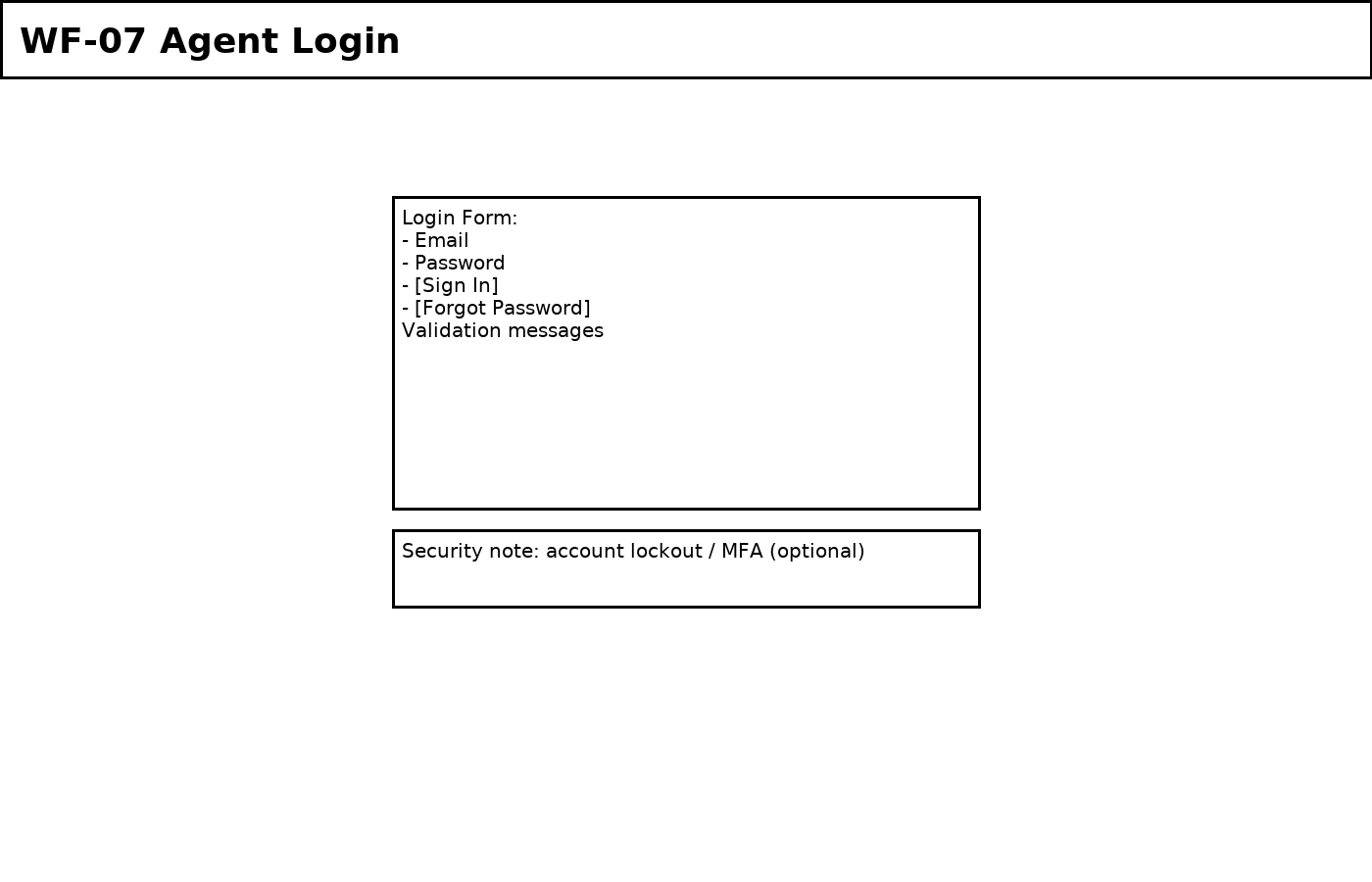
Covers user stories: US-06

Navigation:

• User menu → Saved Listings (requires sign-in).

Active elements & actions:

• Remove saved; open details; sort by saved date or price.



**Figure WF-07: Agent Login**

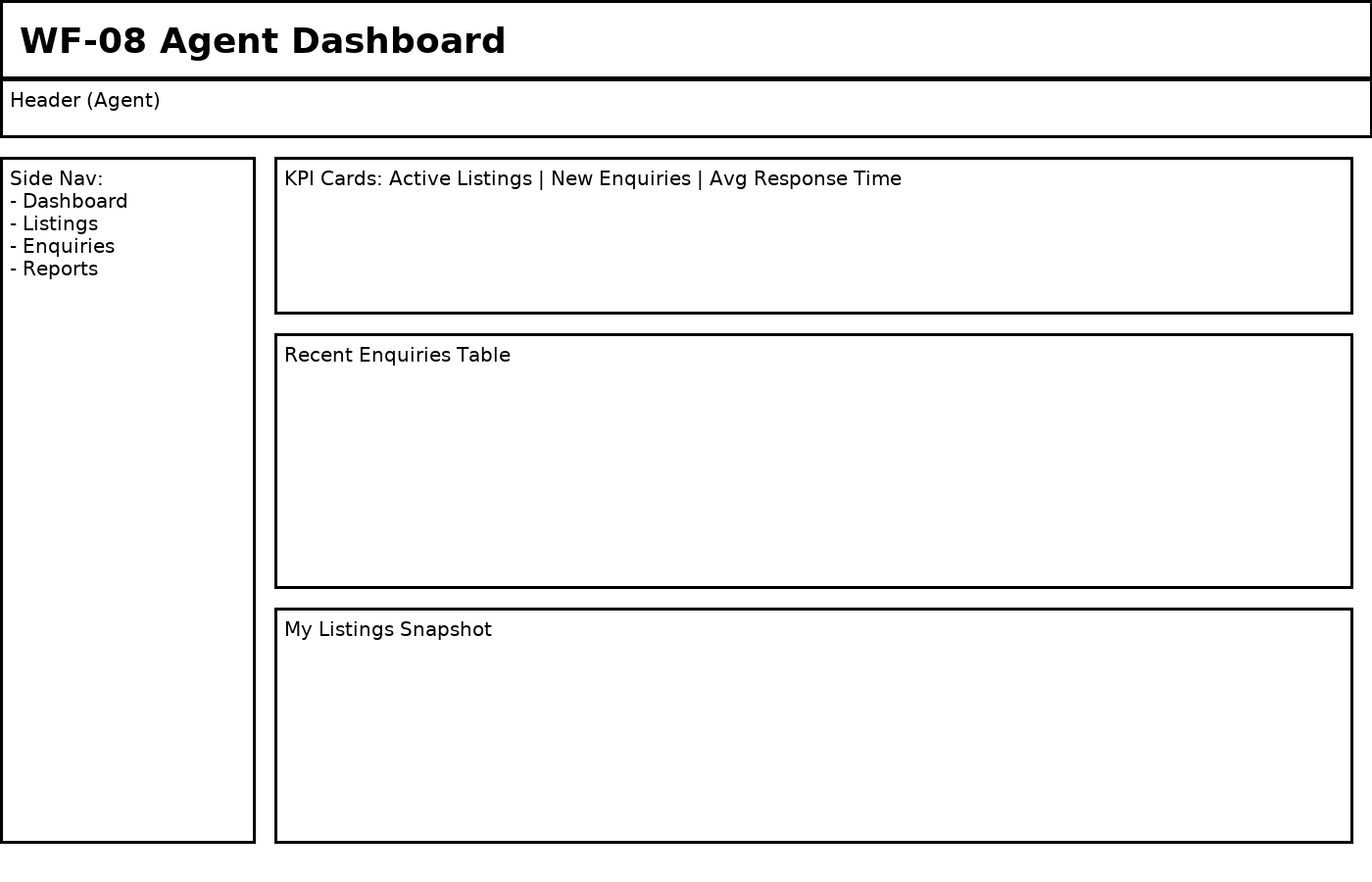
Covers user stories: US-07

Navigation:

• Agent/Staff → Login.

Active elements & actions:

• Invalid credentials show error; optional MFA/lockout; successful login redirects to Agent Dashboard.



**Figure WF-08: Agent Dashboard**

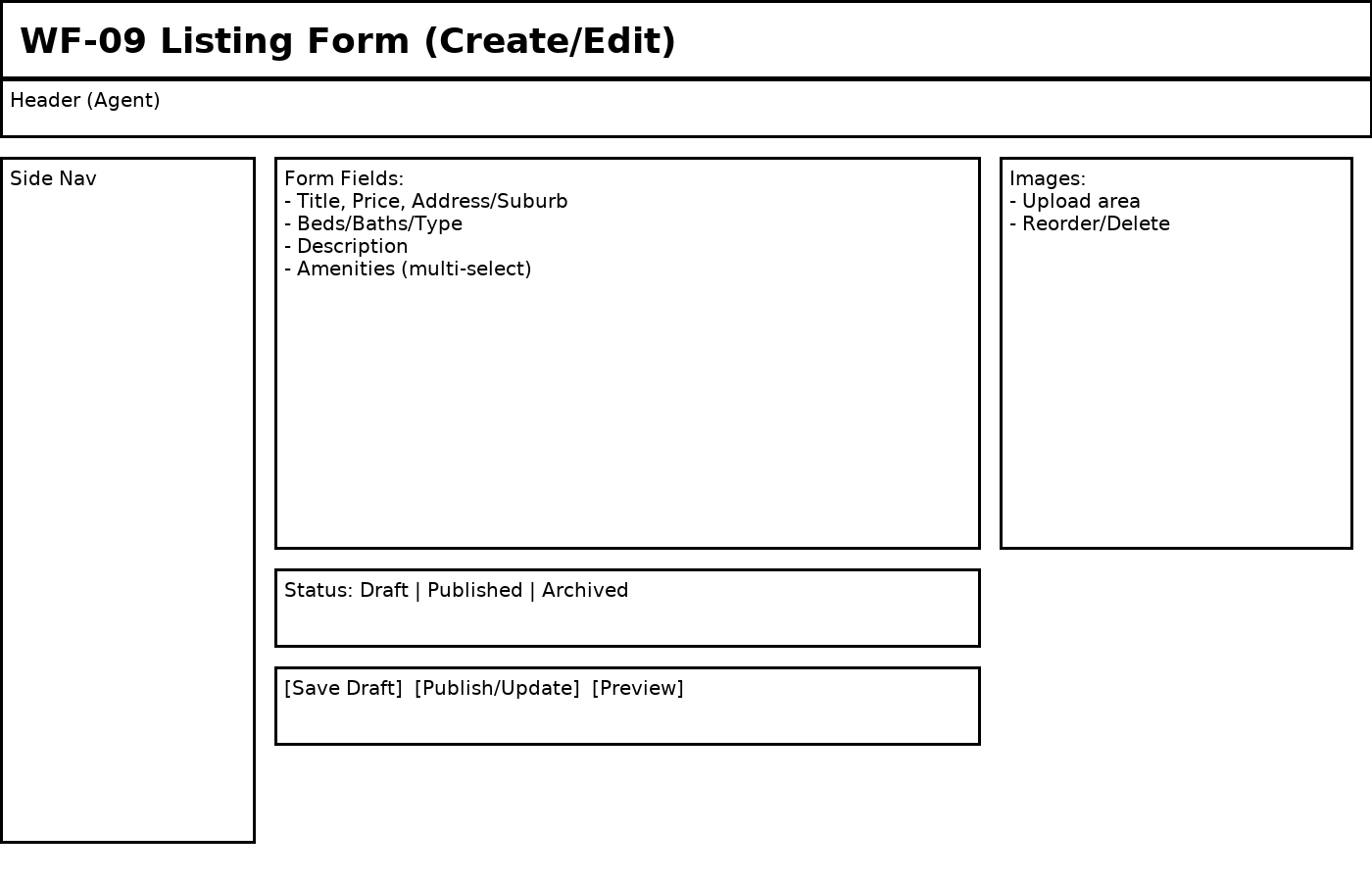
Covers user stories: US-07, US-08, US-09

Navigation:

• Login → Agent Dashboard.

Active elements & actions:

• Open Recent Enquiries; navigate to Manage Listings; KPI cards link to filtered views.



**Figure WF-09: Listing Form (Create/Edit)**

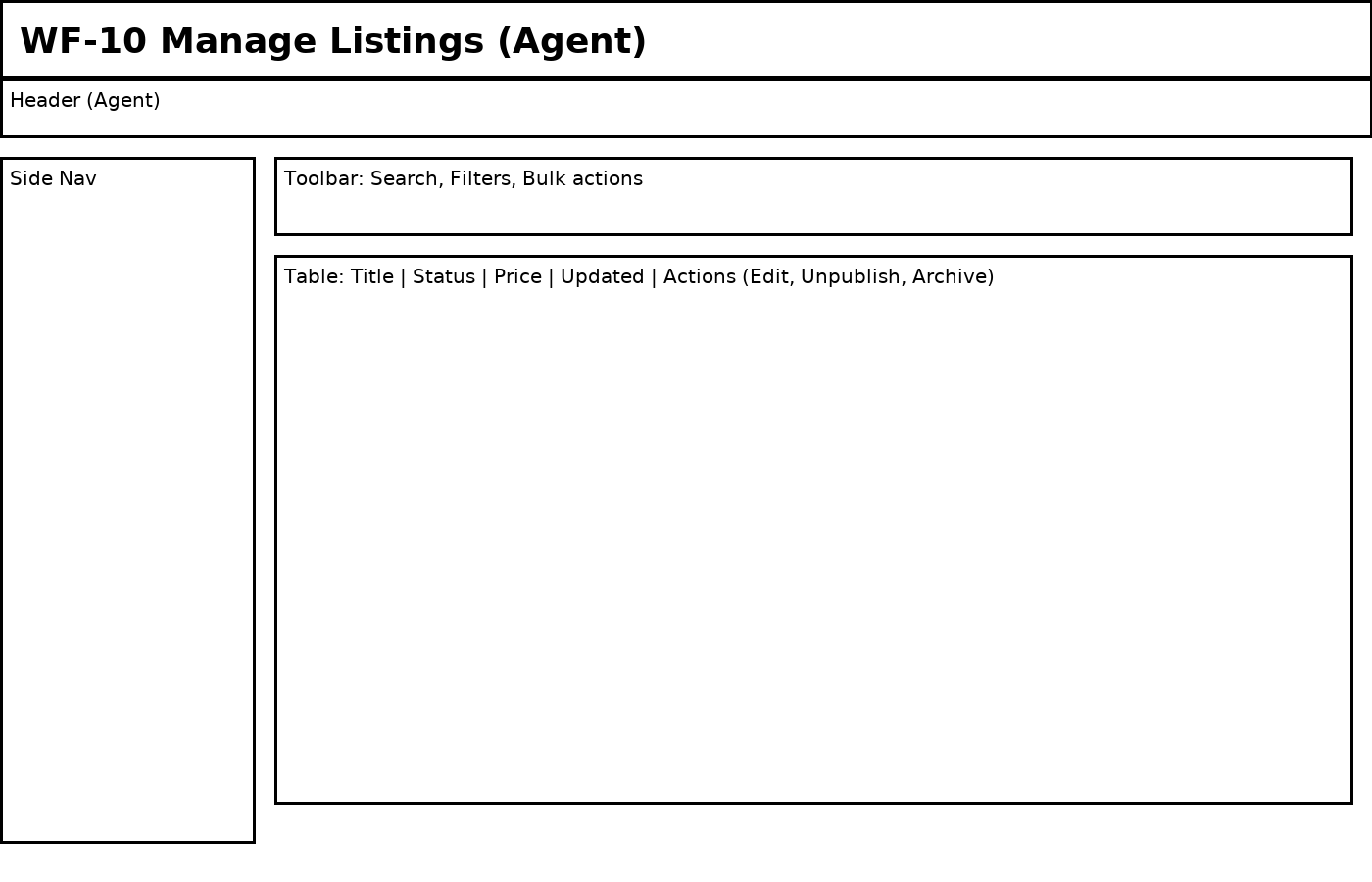
Covers user stories: US-08, US-11

Navigation:

• Agent Dashboard → Create/Edit Listing.

Active elements & actions:

• Required field validation; upload/reorder images; set status Draft/Published/Archived; Save/Preview.



**Figure WF-10: Manage Listings (Agent)**

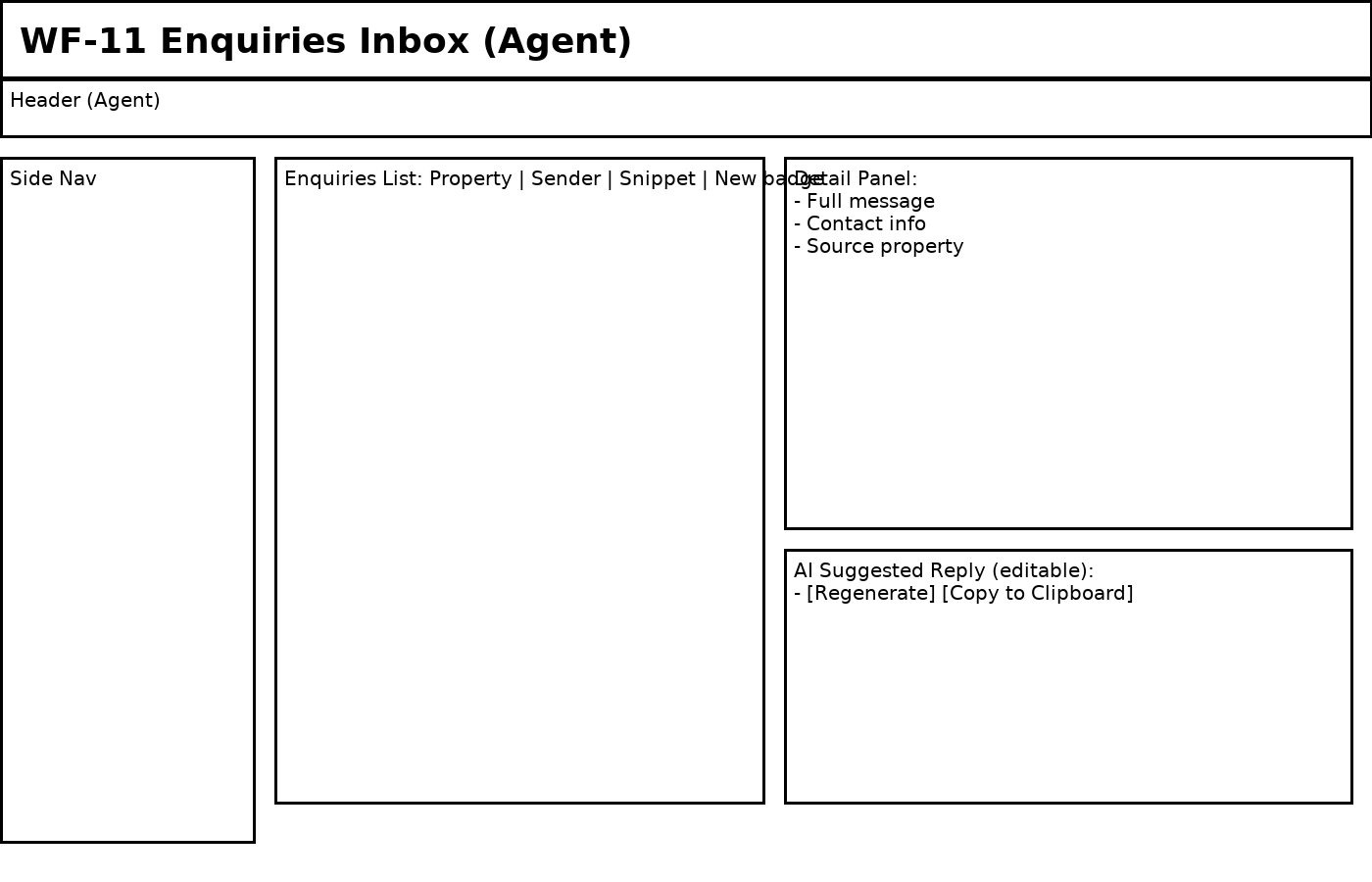
Covers user stories: US-08

Navigation:

• Agent Dashboard → Manage Listings.

Active elements & actions:

• Search, filter, bulk actions; row actions: Edit, Publish/Unpublish, Archive.



**Figure WF-11: Enquiries Inbox (Agent)**

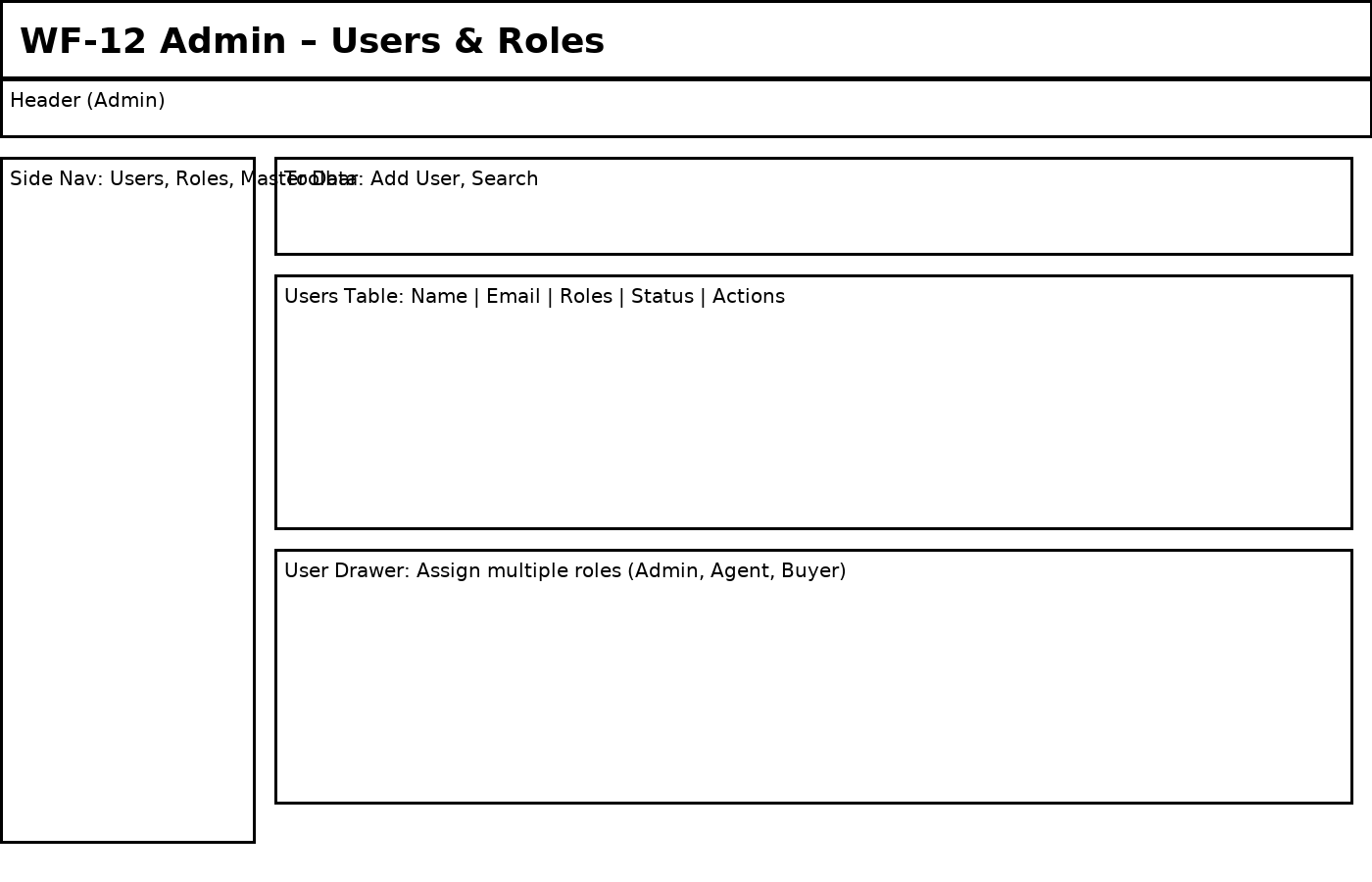
Covers user stories: US-09, US-13

Navigation:

• Agent Dashboard → Enquiries.

Active elements & actions:

• Open enquiry to view details; AI generates reply; Regenerate and Copy to clipboard; mark as Read.



**Figure WF-12: Admin – Users & Roles**

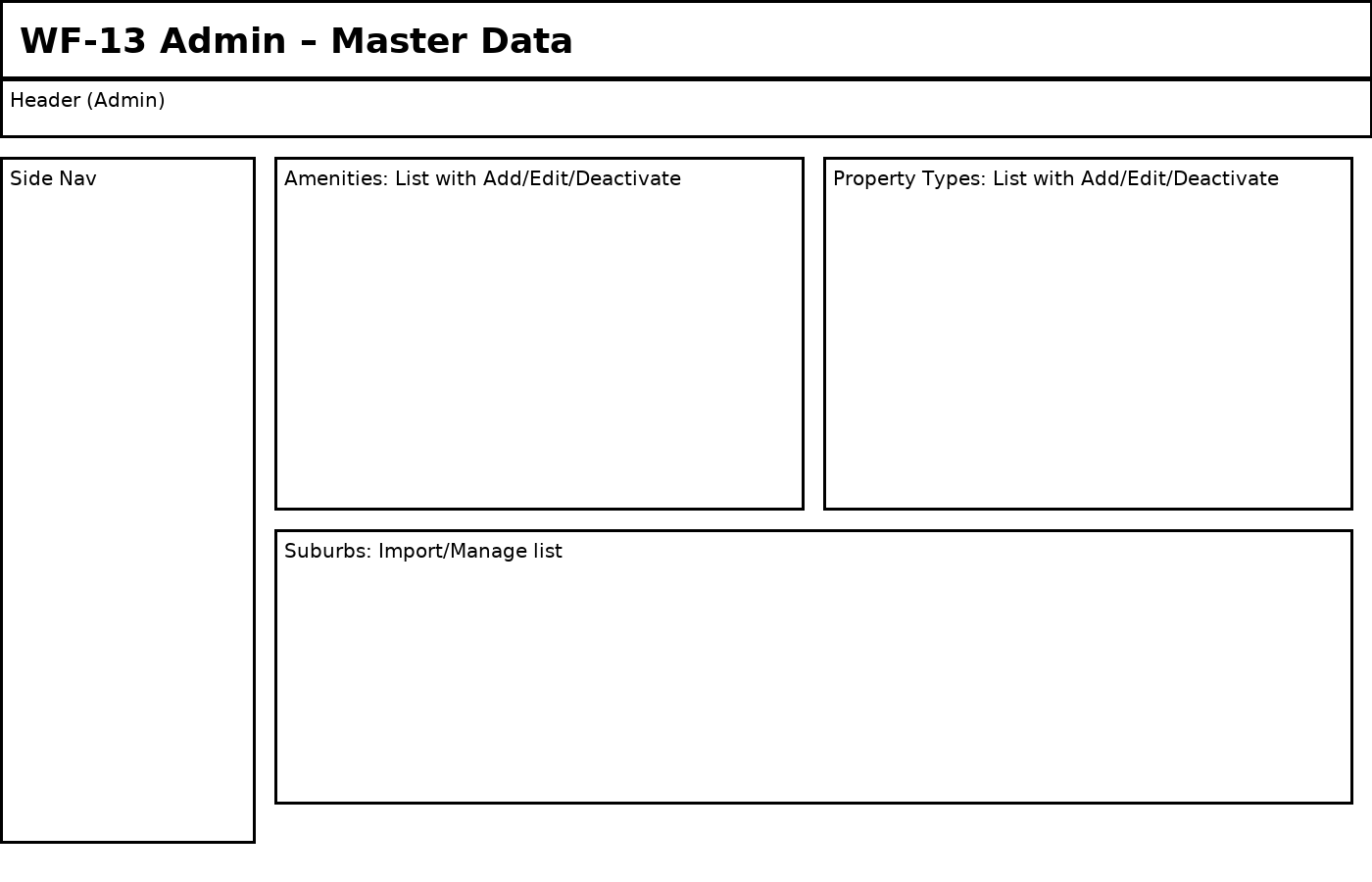
Covers user stories: US-10

Navigation:

• Admin → Users & Roles.

Active elements & actions:

• Create user; assign multiple roles; activate/deactivate; changes apply on next request.



**Figure WF-13: Admin – Master Data**

Covers user stories: US-11

Navigation:

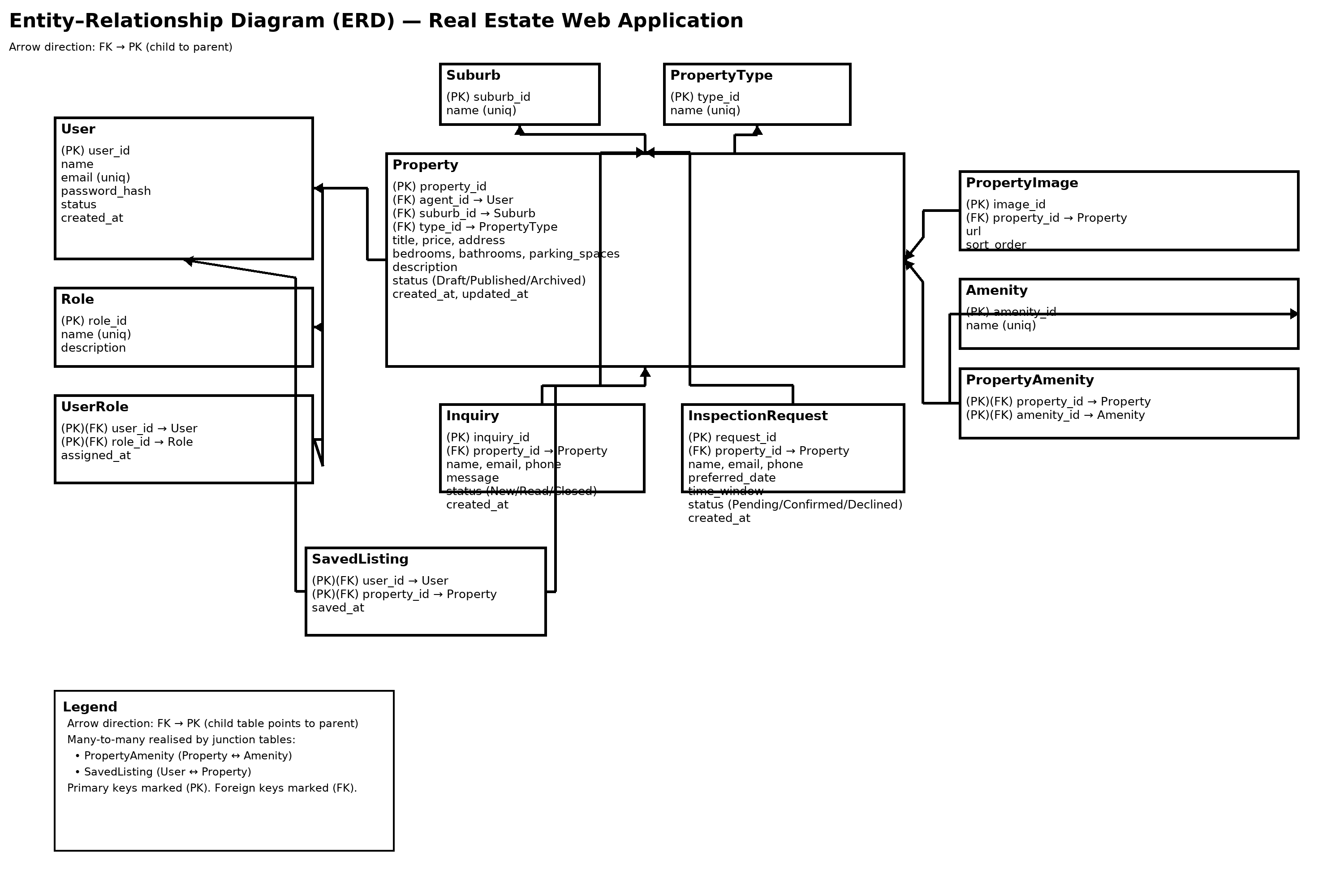
• Admin → Master Data.

Active elements & actions:

• Add/Edit/Deactivate amenities & types; manage suburbs; deactivated items remain on old records but hidden for new selections.

# 3. Database Schema (ER Diagram)

The ERD below supports multi-user roles and authorization (Admin, Agent, Buyer) and covers properties, images, amenities, enquiries, inspection requests, and saved listings. Roles are extensible via a User–Role many-to-many mapping.



**Figure ERD-01: Entity–Relationship Diagram for the Real Estate Web Application**

Key entities and relationships:

• User–Role (UserRole) enables multiple roles per user and easy addition of new roles (normalised RBAC).

• Property belongs to an Agent (User with Agent role) and references Suburb and PropertyType as master data.

• PropertyImage is 1–N from Property; sort\_order supports gallery arrangement.

• Amenity is many-to-many with Property via PropertyAmenity.

• Inquiry links a buyer’s enquiry to a Property with status tracking (New/Read/Closed).

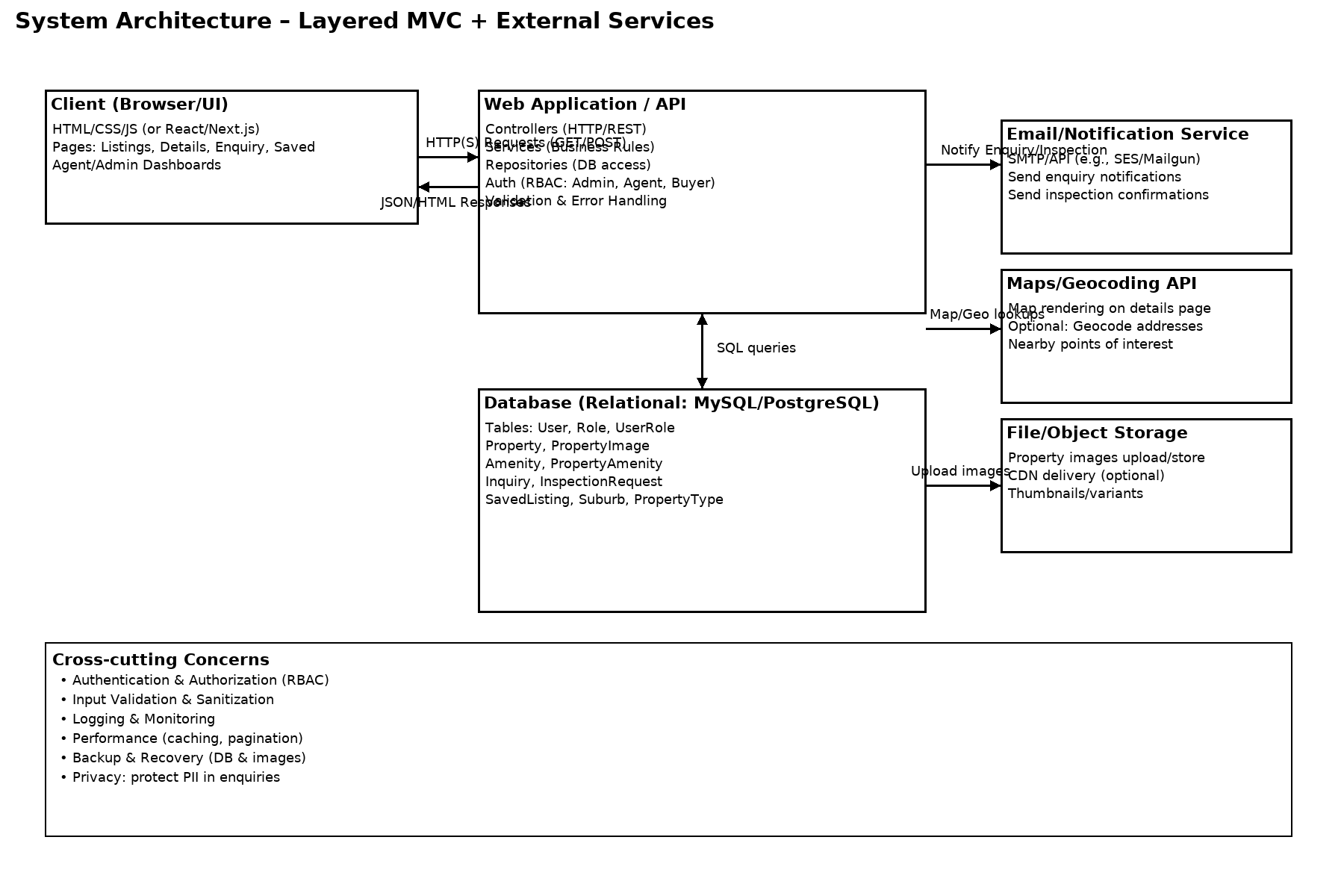
• InspectionRequest captures preferred date/time windows and workflow status (Pending/Confirmed/Declined).

• SavedListing links a signed-in buyer to saved properties (composite key prevents duplicates).

Cardinalities: User 1..N UserRole; Role 1..N UserRole; User (Agent) 1..N Property; Property 1..N PropertyImage; Property M..N Amenity; Property 1..N Inquiry; Property 1..N InspectionRequest; User (Buyer) M..N Property via SavedListing; Property 1..N SavedListing; Property N..1 Suburb; Property N..1 PropertyType.

# 4. System Architecture & UML

## 4.1 Architecture Diagram (Layered MVC + External Services)



**Figure ARCH-01: Layered architecture with Client → Web/API → Database and integrations (Email, Maps, Storage).**

Design pattern & justification:

• Layered MVC cleanly separates concerns: controllers handle HTTP, services encapsulate business rules, repositories isolate data access.

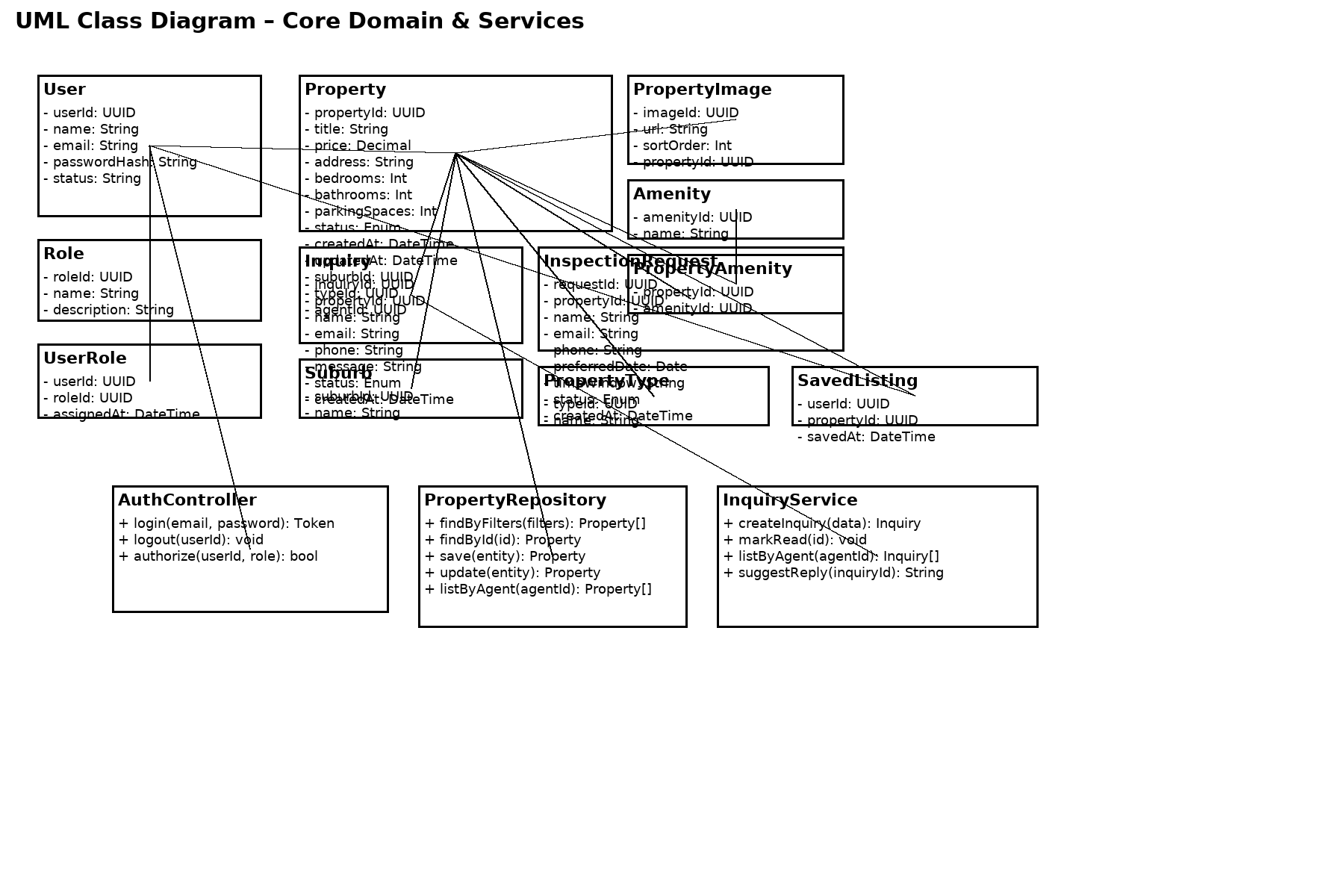
• RBAC via User/Role enables secure, extensible authorization (Admin, Agent, Buyer).

• Externalized services (Email, Maps, Storage) align with real estate needs (enquiry notifications, map views, image hosting).

• Performance: pagination on listings, caching of metadata (suburbs, amenities), and CDN for images.

• Reliability: database backups + stored image backups; logs for enquiries and errors.

## 4.2 UML Class Diagram (Core Domain & Services)



**Figure UML-01: Core domain classes and service/controller layer with key associations.**

Notes: The domain mirrors the ERD (entities and associations). Service/Controller classes orchestrate use cases: AuthController for authentication/authorization; PropertyRepository for search and CRUD; InquiryService for enquiry lifecycle and AI reply suggestions.

# 5. AI/GenAI Components (Project-Specific)

## 5.1 Features mapped to user stories

**US-12 – NLP Search**

NLP query parsing converts natural language into filters (beds, price, suburb, amenities). Example: “3-bed under $700k in Tarneit with parking”.

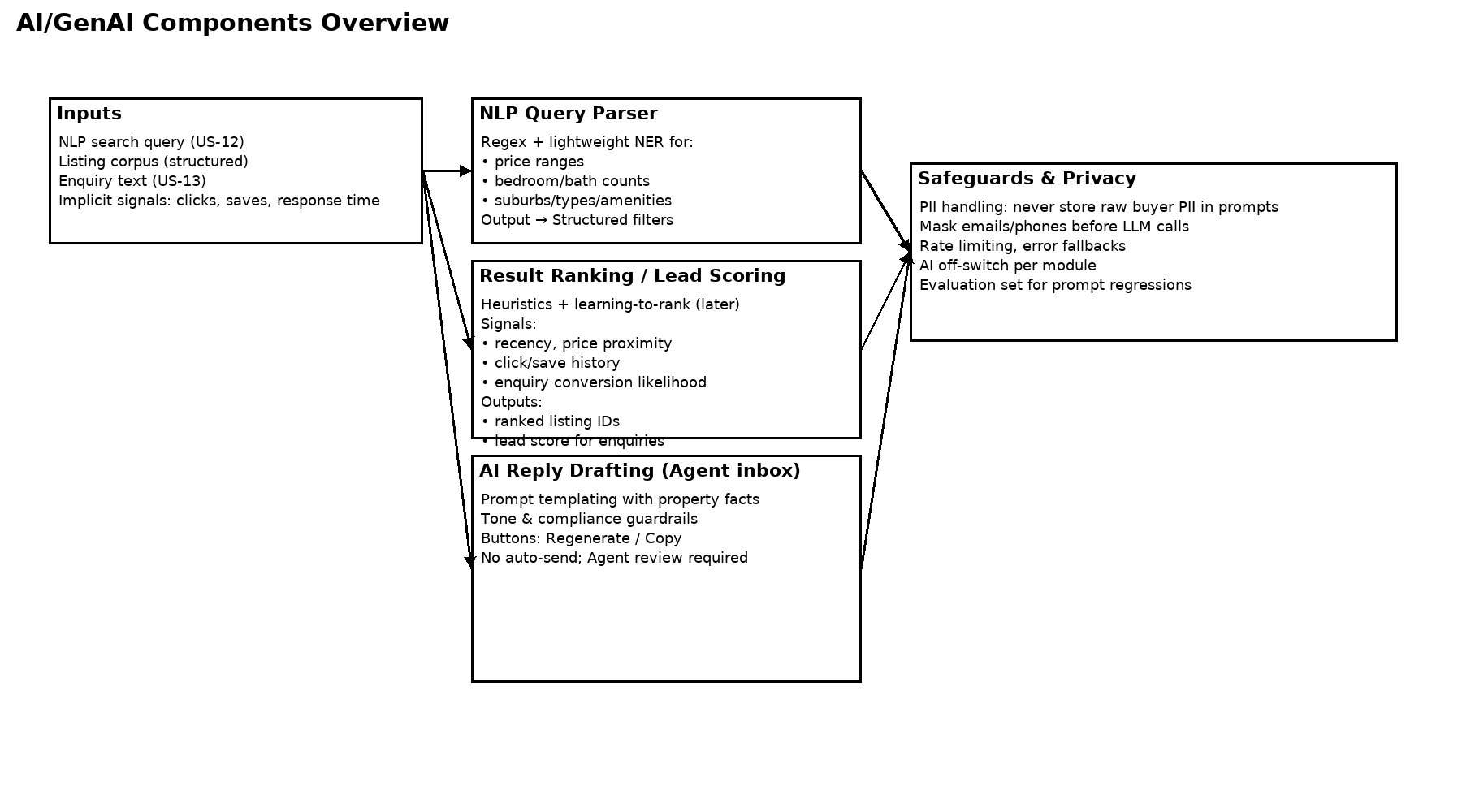
**US-13 – Agent Reply Drafting**

For each enquiry, the system drafts a reply that references the property, answers the buyer’s questions, and remains editable. No auto-send; the agent reviews and sends from their email client.

**US-01/US-02/US-09 – Ranking & Lead Scoring**

Rank listings by relevance and estimate enquiry likelihood to prioritize follow-ups.

## 5.2 Design & Data Flow



**Figure AI-01: AI components and safeguards integrated into the application flow.**

• NLP Query Parser: Regex + small NER dictionary for suburbs/amenities; outputs canonical filters.

• Ranking/Lead Scoring: start with heuristics (recency, price proximity, engagement signals), later upgrade to learning-to-rank when data is available.

• Reply Drafting: prompt templates that insert property facts (title, price, suburb, amenities) and enquiry snippets; explicit controls (Regenerate/Copy).

## 5.3 Model and Technology Options (with justification)

• Phase 1 (MVP): Deterministic NLP (regex + gazetteers) to ensure predictable filters; avoids heavy infra and privacy concerns.

• Phase 1 (MVP): Heuristic ranking (no personalisation) to meet performance and explainability requirements.

• Phase 1 (MVP): Prompt-based reply drafts with strict redaction of PII; no automatic sending; agent is always in control.

• Phase 2+: Optional hosted LLM for better paraphrasing and query understanding after privacy review and guardrails; optional lightweight vector search for semantic matching.

## 5.4 Privacy, Security, and Risk Controls

• PII Redaction: strip or mask Emails/Phones before prompt assembly; pass only the minimum context needed.

• Data Minimisation: keep prompts ephemeral; store only structured results (filters, draft text) not raw prompts.

• Rate-limiting & Monitoring: prevent abuse and detect failures; clear fallback behavior (show non-AI results if AI fails).

• Opt-out Switches: Admin can disable AI modules individually.

• Bias & Quality Checks: maintain a small evaluation set; review drafts for tone/compliance; log regenerations (without PII).

## 5.5 Acceptance Criteria & KPIs

|  |  |  |
| --- | --- | --- |
| Area | Acceptance Criteria (UAT) | Initial KPI (Post‑MVP) |
| NLP Search (US‑12) | Given sample queries, When submitted, Then filters match expected beds/price/suburb/amenities in ≥ 9/10 cases. | ≥ 90% correct parsing on UAT set; median search < 400ms. |
| Reply Draft (US‑13) | Given an enquiry, When opening in Agent Inbox, Then a contextually accurate, editable draft is produced without PII leakage. | Agent edits < 30% of tokens on average; zero PII exposure incidents. |
| Ranking/Scoring | Given a fixed dataset, When ranking executes, Then top‑N contains relevant properties per rubric. | Click‑through rate +10% vs. baseline; time‑to‑first‑reply < 24h. |

## 5.6 Fallbacks and Graceful Degradation

If any AI sub‑system fails or is disabled, the application defaults to deterministic behavior: standard keyword search and filter application, chronological sorting, and manual agent replies.

# 6. Platforms, Languages, and Tools (with Justification)

This section explains each selected technology and why it fits the project’s scope, skills, and constraints. It also clarifies the pivot from a static brochure site (A1) to a dynamic, database‑backed MVP (A2) based on client requirements (search, admin, enquiries).

|  |  |  |  |
| --- | --- | --- | --- |
| Layer / Area | Choice | Why this choice (justification) | Alternatives (if constraints change) |
| Frontend UI | Next.js (React) + Tailwind CSS | Component reusability, fast developer velocity, SSR/SSG options for SEO on listings pages, broad ecosystem; Tailwind for consistent design system. | Plain HTML/CSS/JS (simpler), Vue/Nuxt, Angular |
| Backend API | Laravel 10 (PHP) | Team familiarity; expressive routing/controllers; Eloquent ORM for fast CRUD; built‑in auth; mail/queue integration; ecosystem (spatie/laravel-permission) for RBAC. | Node.js (Express/NestJS), Django (Python), Spring Boot (Java) |
| Auth & RBAC | Laravel Breeze/Fortify + spatie/laravel-permission | Rapid, standards‑aligned authentication; role/permission model aligns with ERD; reduces custom security code. | Custom JWT in Node; Auth0 (hosted) if SSO needed |
| Database | MySQL 8 | Relational integrity for listings/enquiries; team familiarity; easy hosting; query tooling. | PostgreSQL (advanced features), MariaDB |
| Email/Notifications | Mailgun (SMTP/API) | Reliable deliverability; easy Laravel integration; sandbox domains for UAT. | AWS SES, SendGrid, Postmark |
| Maps/Geocoding | Google Maps Platform | Rich mapping and POI data; familiar to users; straight integration with frontend. | Mapbox, Leaflet + OSM |
| Object Storage (Images) | S3‑compatible (AWS S3 / Wasabi) + CDN | Cost‑effective, scalable image hosting; CDN for performance; presigned uploads supported. | Local storage (dev only), Cloudflare R2 |
| Project Tracking | ClickUp (Kanban + Docs) | Simple board + statuses; assignees/dates; easy evidence via screenshots. | Jira (more enterprise), Trello |
| Version Control | GitHub | Branching/PRs/reviews; actions for CI later; easy collaboration and evidence screenshots. | GitLab, Bitbucket |
| Testing/Tools | Postman (API), Browser DevTools, phpunit (unit), manual UAT | Covers API checks, UI validation, regression; aligns with A2’s emphasis on UAT evidence. | Cypress/Playwright (later automation) |

Pivot note: The client requested searchable listings and an admin area; therefore, a dynamic backend with database was chosen for A2 to meet real requirements.

# 7. Test Plan & User Acceptance Testing (UAT)

## 7.1 Objectives & Scope

Validate that the MVP meets functional requirements (search, details, enquiry), data validation rules, business logic (publish/unpublish, role‑based access), and basic non‑functionals (responsiveness, accessibility checks, performance smoke).

## 7.2 Test Types & Approach

• Functional/UI: filters, navigation, forms, CRUD flows.

• Data validation: required fields, formats, ranges; clear error messages.

• Business logic: listing status workflow; role permissions (Admin/Agent/Buyer).

• Compatibility/responsiveness: modern desktop + mobile widths (≥ 360px).

• Accessibility (basic): keyboard focus, alt text for images, labels for form fields.

• Performance (smoke): listings pages paginate; images optimised; LCP target < 3s on staging.

• Security (basic): auth protection on dashboards; server‑side validation; no sensitive data in client logs.

## 7.3 Roles & Responsibilities

• QA Lead (Student A): plans UAT, writes cases, tracks defects, coordinates re‑tests.

• Product Rep / Client (Agent role): validates agent workflows and content accuracy.

• Admin (Student B): validates admin features (users/roles, master data).

• Developers (Team): bug triage/fixes; provide testable builds on staging.

• PM (Student C): schedule, scope, sign‑off coordination.

## 7.4 Environments & Test Data

Environments: DEV (feature branches), STAGING (UAT build). Test accounts: Admin, Agent; sample listings with varied amenities; fake buyer data (non‑PII). Mail in sandbox mode. Maps with test keys.

## 7.5 Entry / Exit Criteria

Entry: features complete on staging; smoke tests pass; seed data loaded; access provided to UAT users.

Exit: all Must‑Have stories pass UAT; no open High/Critical defects; agreed Known Issues list documented.

## 7.6 Schedule (Who / When / How)

|  |  |  |  |
| --- | --- | --- | --- |
| When | Who | What | How/Tools |
| Day 1 | QA Lead + Dev | Prepare UAT build & data; sanity checks | GitHub Actions (optional), seed scripts |
| Day 2 | QA Lead | Run functional + validation tests | Manual, Browser DevTools |
| Day 3 | Agent (client) | Agent dashboard flows, enquiries | Manual, guided script |
| Day 4 | Admin | Users & roles, master data | Manual |
| Day 5 | QA Lead | Regression on fixes; performance smoke | Manual + Lighthouse |

## 7.7 Defect Reporting & Triage

Log defects in GitHub Issues (title, steps, expected vs actual, screenshots). Link ClickUp task to the GitHub issue. Triage daily: Critical/High fixed before sign‑off; Medium/Low scheduled.

## 7.8 UAT Test Cases (Representative Set)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Title | Actor | Preconditions | Steps | Test Data | Expected Result | Owner | When | How/Tools |
| UAT‑01 | Filter listings by suburb+price | Visitor | Listings exist across suburbs and prices | 1) Open Listings 2) Select 'Tarneit' 3) Set price < $700k 4) Apply | Suburb: Tarneit; Price < 700k | Results show only Tarneit under $700k; count updates; pagination intact | QA Lead | Day 2 | Manual UI |
| UAT‑02 | View property details | Visitor | At least one property matching filters | 1) Click a result card 2) Switch gallery image 3) Open Map tab | N/A | Details page renders key facts; gallery and map work | QA Lead | Day 2 | Manual UI |
| UAT‑03 | Submit enquiry with validation | Visitor | On details page | 1) Leave required fields empty 2) Attempt submit 3) Fill valid data 4) Submit | Name, Email, Message | Inline errors appear for invalid; success confirmation on valid; enquiry saved | QA Lead | Day 2 | Manual UI |
| UAT‑04 | Request inspection | Visitor | Details page; inspection form enabled | 1) Open Request Inspection 2) Choose date/time 3) Submit | Preferred date/time | Request recorded; agent notified (sandbox email) | QA Lead | Day 2 | Manual UI |
| UAT‑05 | Agent login (auth) | Agent | Agent user exists | 1) Open login 2) Enter valid creds 3) Submit 4) Logout | agent@example.com / \*\*\*\* | Successful login redirects to dashboard; invalid creds show error | Agent | Day 3 | Manual UI |
| UAT‑06 | Create & publish listing | Agent | Logged in as Agent | 1) Open Create Listing 2) Fill required fields 3) Upload images 4) Publish | Title, price, address, beds/baths, amenities, images | Listing visible on public site when status=Published | Agent | Day 3 | Manual UI + images |
| UAT‑07 | Manage listings status | Agent | Agent has multiple listings | 1) Open Manage Listings 2) Unpublish one 3) Archive one | N/A | Unpublished hidden publicly; archived hidden from agent defaults | Agent | Day 3 | Manual UI |
| UAT‑08 | View enquiries & mark read | Agent | Enquiries exist | 1) Open Enquiries 2) Open latest enquiry 3) Mark as Read | N/A | Detail panel shows message/contact; status updates to Read | Agent | Day 3 | Manual UI |
| UAT‑09 | Admin users & roles | Admin | Admin account exists | 1) Add user 2) Assign Agent+Buyer roles 3) Save 4) Login as the new user | Name/email; roles | User can access only permitted sections by role | Admin | Day 4 | Manual UI |
| UAT‑10 | Admin master data | Admin | Amenities/types list available | 1) Add a new amenity 2) Deactivate an old one | Amenity name | New amenity appears on listing form; deactivated hidden from new selections | Admin | Day 4 | Manual UI |
| UAT‑11 | NLP search parsing | Visitor | Dictionary seeded with common suburbs/amenities | 1) Enter '3‑bed under $700k in Tarneit with parking' 2) Submit | NL query text | Filters auto‑populate: beds=3, price<700k, suburb=Tarneit, amenity=parking | QA Lead | Day 2 | Manual UI |
| UAT‑12 | AI reply drafting | Agent | Enquiry exists; AI module enabled (sandbox) | 1) Open enquiry 2) Generate draft 3) Regenerate | Enquiry body sample | Draft references property facts; no PII leaked; regenerate produces variation | Agent | Day 3 | Manual UI |

## 7.9 UAT Sign‑off

### 7.9.1 UAT Sign‑off Form

|  |  |  |  |
| --- | --- | --- | --- |
| Project / Application | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| Build / Version | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| Environment | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| UAT Period (From) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| UAT Period (To) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| Client Representative (Name / Role / Org) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| Project Manager (Name) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |

#### Signatures

|  |  |
| --- | --- |
| Client Representative | Project Manager |
| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 7.9.2 UAT Sign‑off Checklist

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item | Description | Status (Pass/Fail/N/A) | Evidence (screenshot/link) | Owner | Date |
| Must‑Have stories passed | US‑01, US‑02, US‑03, US‑07, US‑08, US‑09, US‑10, US‑14 all pass UAT |  |  |  |  |
| No Critical/High defects | Zero open Critical/High at sign‑off time |  |  |  |  |
| UAT executed per schedule | All planned cases run & results recorded (Section 7.8) |  |  |  |  |
| Regression on fixes | Fixed defects re‑tested; no breakages |  |  |  |  |
| Performance smoke | Listings paginate; LCP < 3s on staging |  |  |  |  |
| Accessibility basics | Keyboard focus, alt text, labels validated |  |  |  |  |
| Data validation | Required fields, formats, ranges enforced |  |  |  |  |
| Privacy/PII checks | No PII in logs/prompts; sandbox emails only |  |  |  |  |
| Known Issues list | Documented and accepted by client rep |  |  |  |  |

### 7.9.3 Known Issues (for Sign‑off)

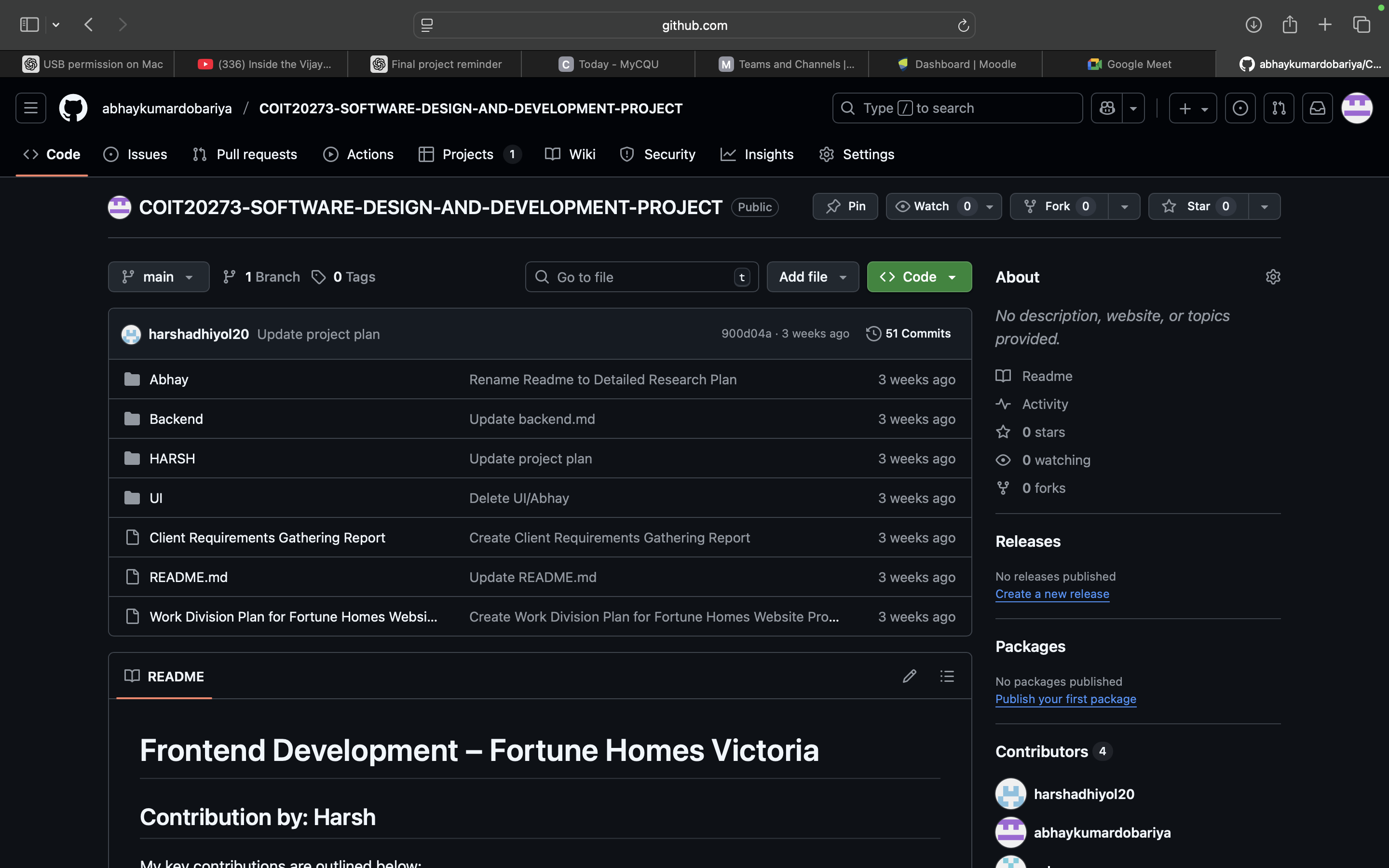
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Title | Severity | Description | Workaround | Target Fix Version | Owner / Status |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Sign‑off requires: all Must‑Have stories pass; no open Critical/High defects; documented Known Issues list accepted by client representative (Agent role).

# 8. Evidence of Version Control & Project Tracking

Insert screenshots (or append in Appendix) to provide concrete evidence, as required.

## 8.1 GitHub Evidence (screenshots)

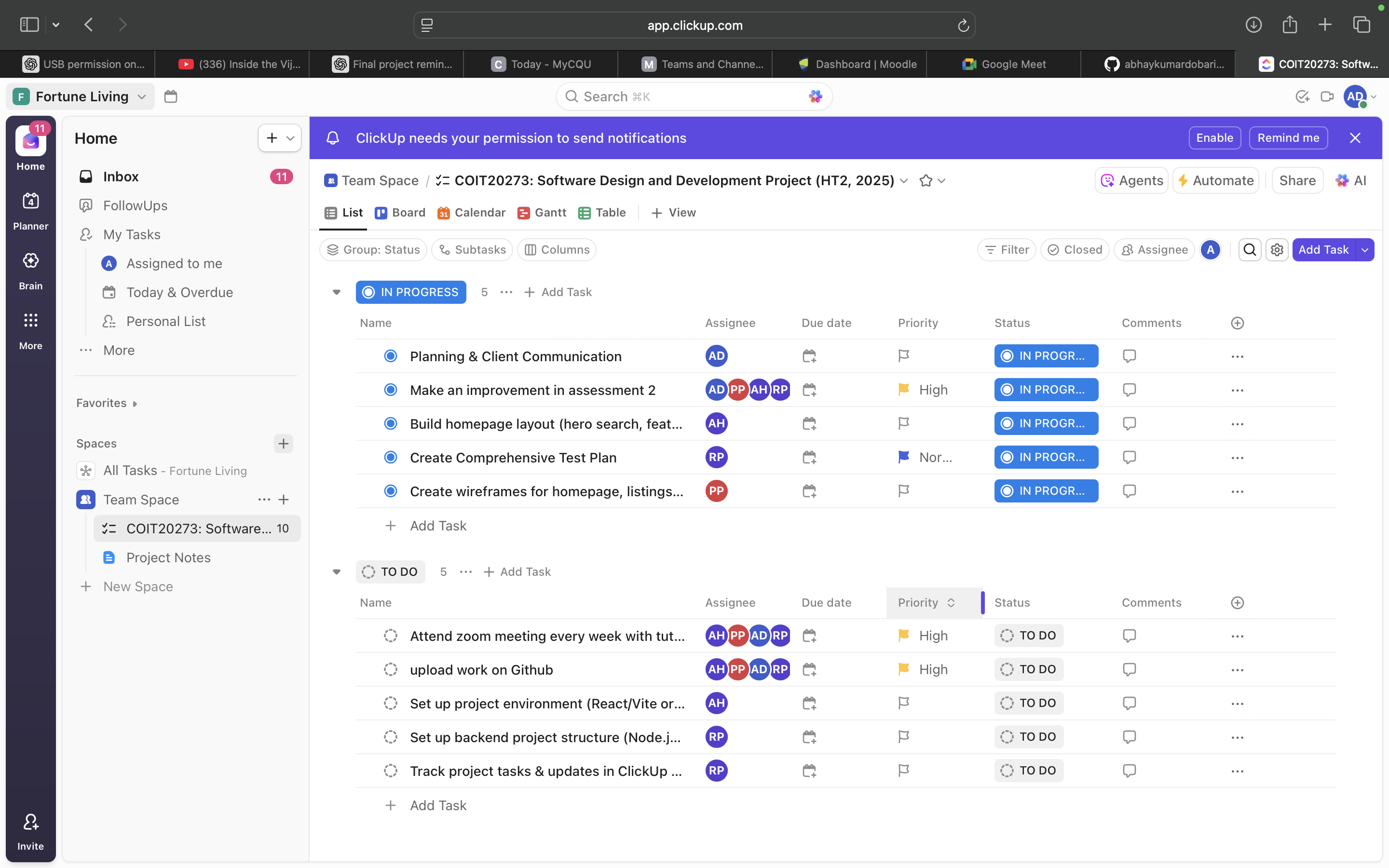
• Repository home page (URL visible).

• Branch list (main + feature branches).

• Commits graph (show active development).

• One pull request (title, description, code diff).

## 8.2 Tracking Board (ClickUp/Jira) Evidence

• Kanban board with columns (Backlog → In Progress → Testing → Done).

• Cards for key user stories (ID matches this report), assignees, due dates.

• A card linked to a GitHub issue/PR (bi‑directional reference if possible).

## 8.3 Traceability Example (Story → Task → PR → UAT)

Example chain: US‑08 Create/Publish Listing → ClickUp task CU‑123 → GitHub PR #10 → UAT‑06 passed (screenshot attached).

# 9. Requirements Traceability Matrix (RTM)

This matrix links each user story to its UI wireframe(s), database entities, architecture/UML components, and UAT test case(s). It demonstrates end‑to‑end coverage from requirements to verification and addresses the assessor’s comments about alignment and completeness.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Story (ID & Title) | Wireframes (WF‑xx) | ERD Entities | Architecture / UML | UAT Case(s) |
| US‑01 Filter listings | WF‑01, WF‑02 | Property, Amenity, PropertyAmenity, Suburb, PropertyType | PropertyRepository | UAT‑01 |
| US‑02 View property details | WF‑03 | Property, PropertyImage, Amenity, PropertyAmenity, Suburb, PropertyType | PropertyRepository | UAT‑02 |
| US‑03 Submit enquiry | WF‑03, WF‑04 | Inquiry, Property | InquiryService | UAT‑03 |
| US‑04 Request inspection | WF‑05 | InspectionRequest, Property | (Controller/Repo for InspectionRequest) | UAT‑04 |
| US‑05 Map on details | WF‑03 | Property, Suburb | PropertyRepository | UAT‑02 |
| US‑06 Saved listings (optional) | WF‑06 | SavedListing, User, Property | PropertyRepository | TBD (Phase 2) |
| US‑07 Agent login | WF‑07 | User, Role, UserRole | AuthController | UAT‑05 |
| US‑08 Manage listings (CRUD + status) | WF‑09, WF‑10 | Property, PropertyImage, Amenity, PropertyAmenity | PropertyRepository | UAT‑06, UAT‑07 |
| US‑09 View/respond to enquiries | WF‑11 | Inquiry | InquiryService | UAT‑08 |
| US‑10 Admin users & roles | WF‑12 | User, Role, UserRole | AuthController | UAT‑09 |
| US‑11 Admin master data | WF‑13 | Amenity, PropertyType, Suburb | PropertyRepository | UAT‑10 |
| US‑12 NLP search (AI) | WF‑01 | Property, Amenity, PropertyAmenity, Suburb, PropertyType | PropertyRepository | UAT‑11 |
| US‑13 AI reply drafting | WF‑11 | Inquiry, Property | InquiryService | UAT‑12 |
| US‑14 Form validation & feedback | WF‑03, WF‑05 | Inquiry, InspectionRequest | AuthController (guarded routes) | UAT‑03, UAT‑04 |

Coverage summary:

• User stories covered by UAT in MVP: 12 of 14 (US‑06 is optional; US‑05 validated within UAT‑02).

• All Must‑Have stories (US‑01, US‑02, US‑03, US‑07, US‑08, US‑09, US‑10, US‑14) have at least one UAT case.

• ERD entities referenced by stories are present in Section 3; architecture/UML mapping in Section 4.