****

**ALAMAIT BOARDING SCHOOL PROJECT DOCUMENTATION**

**PROJECT SCOPE**

To build a comprehensive, user-friendly platform that allows Alamait to independently manage operations, housing, student data, and finances. The platform is designed to support multiple boarding houses (tenants), each operating independently under its own subdomain (e.g., belvedere.alamait.co.zw). A super administrator has visibility and control across all tenants.

**DETAILED SCOPE FOR END USERS**

**Super Admin Capabilities:**

* Create and configure new boarding houses (tenants), each with its own subdomain
* Access consolidated reports across all tenants for oversight and auditing
* Post and monitor financial transactions across tenants for centralized reporting
* Manage global Chart of Accounts, currency settings, and exchange rates
* Review system-wide audit logs showing user activities and data changes
* Manage platform-level settings, roles, and security access

**Boarding House Admin Functions:**

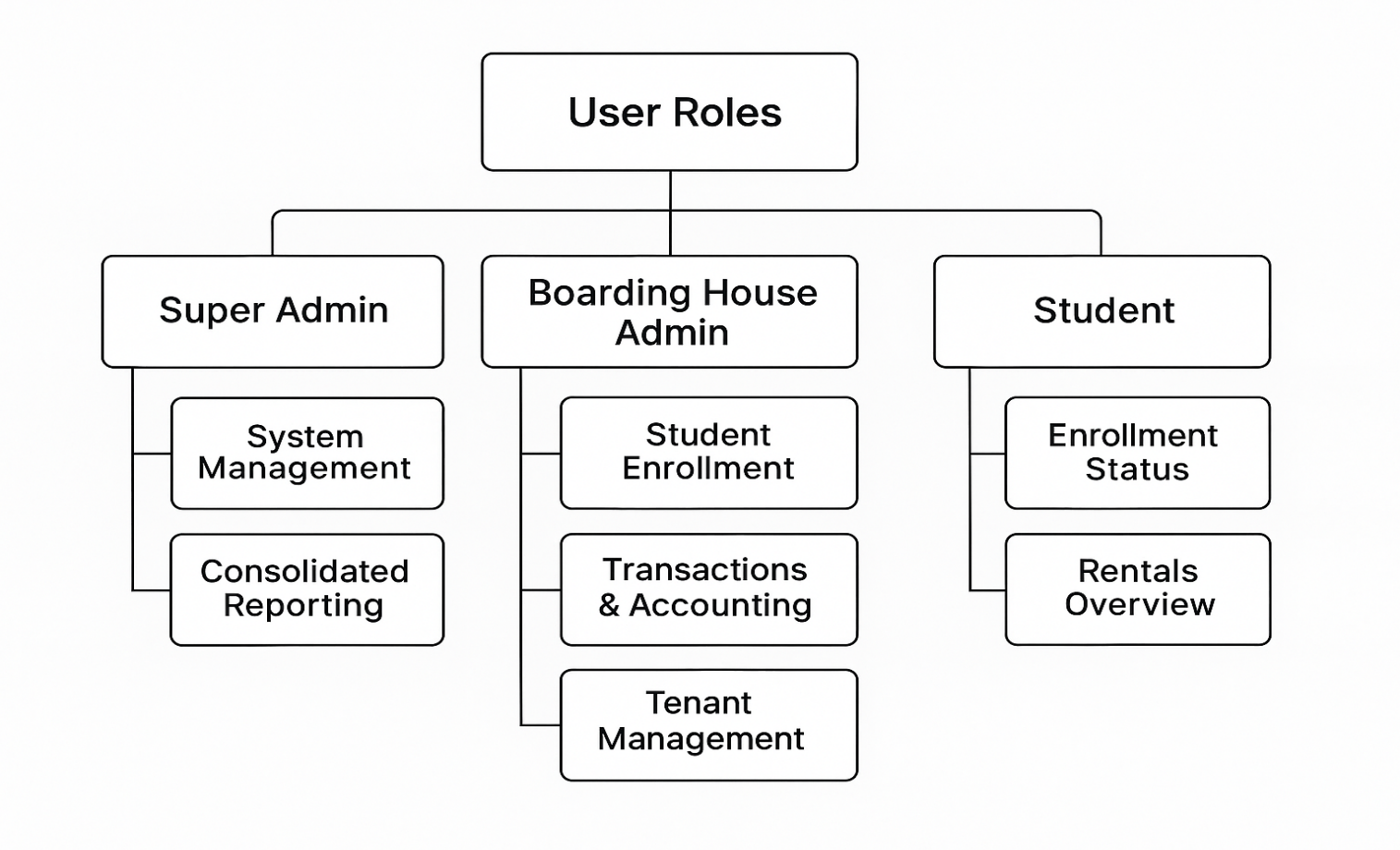
* Define and manage room inventory, bed capacity, and room pricing
* Register new students and assign them to specific rooms
* Set expected enrollment durations, including start and end dates
* Link students to rental agreements and manage expected payment plans
* Post and manage financial transactions specific to the boarding house
* Maintain a localized Chart of Accounts for house-level financial tracking
* Optionally post transactions to the main admin ledger if required by the house policy
* Require a supporting document or image (invoice, receipt, etc.) for every posted transaction to ensure traceability
* Generate house-specific reports, including financials, occupancy statistics, and student performance summaries
* Log incidents, comments, and administrative updates related to individual students
* Upload and manage student-related documents, such as national IDs, academic forms, or contracts
* Track changes in student assignments, including room moves, deactivations, and soft deletions for audit purposes

**Student Portal Features:**

* Login using secure credentials to a personalized student portal
* View assigned room, room price, and enrollment dates
* Track rental dues, paid and outstanding balances
* View personal documents uploaded by the administration
* Update or review next-of-kin details

**Platform-Wide Functional Modules:**

* Room Management: Manage room inventory, bed capacity, and room activation status
* Student Enrollment: Capture admission data, track rental agreements, log start and expected end dates
* Rentals and Transactions: Track all student payments, assign payment statuses, allow transaction posting
* Chart of Accounts (CoA): Define structured accounts per tenant (assets, liabilities, expenses, income)
* Exchange Rates: Store and use currency exchange data to report in multiple currencies
* Audit Logging: Every system action is recorded (e.g., student moved, transaction created, user deleted)
* Soft Delete: Users and records can be soft-deleted and retained for history without showing in active views
* Multi-Tenant Subdomain Routing: Each tenant has a unique subdomain with isolated data and user access

****

**PROJECT TIME LINES**

|  |  |  |
| --- | --- | --- |
| **Day** | **Phase** | **Tasks** |
| Day 1–2 | Planning & Setup | • Finalize feature scope and user roles• Design multi-tenant database schema• Prepare core project structure (Node.js + MySQL + Vite) |
| Day 3 | Authentication & Tenant Bootstrapping | • Implement user authentication (Super Admin, Boarding House Admin, Student)• Set up tenant creation and subdomain resolution |
| Day 4 | Student & Room Module | • Develop student enrolment, document upload, next of kin• Implement room creation, pricing, and bed capacity |
| Day 5 | Rental Transactions & Local CoA | • Build rental transaction logic (student-to-room link)• Implement boarding house-level Chart of Accounts |
| Day 6 | Journal & Exchange Rate Engine | • Link transactions to journal entries with rules• Integrate exchange rate table for multi-currency support |
| Day 7 | Reporting & Audit Logging | • Generate house-level and consolidated financial reports• Set up system-wide audit logging |
| Day 8 | Portals & Soft Delete | • Build admin dashboard and student portal• Implement soft delete logic for all modules |
| Day 9 | Testing & Finalization | • Perform full module integration and QA• Fix critical bugs, optimize queries |
| Day 10 | Go-Live | • Deploy first tenant instance• Create super admin and house admin onboarding flows• Launch pilot boarding house for real-time use |

**PROJECT SUCCESS INDICATORS**

* Each boarding house operates independently with full control of data and users
* Super admin maintains full visibility across all tenants
* Accurate and compliant accounting per tenant, with optional consolidated view
* Students can securely access their own portals and records
* The platform is secure, audit-ready, scalable, and user-friendly
* Lease agreements are digitized and automatically generate student onboarding records
* Petty cash processes are tracked digitally and reconcile against reports
* Maintenance issues are digitally recorded and assignable to responsible staff
* Receivables and monthly financials are generated live from actual system usage
* Marketing inquiries and tenant availability are tracked through a dedicated module