



EBO SYSTEM

Full System Documentation Pack

System specification, design documentation, database overview, user flows, module guide, operations guide, deployment notes, security guidance, and support procedures.

Document	Details
System	EBO SYSTEM
Platform Type	Expandable multi-tenant business operating system
Framework	Django 5, django-tenants, PostgreSQL, Tailwind templates
Deployment	Heroku, public domain ebosystem.co.za, tenant subdomains
Generated	2026-08-03
Audience	Developers, administrators, support staff, tenant admins, client stakeholders

Table of Contents

- 1. Executive Summary
- 2. SSD - Software/System Specification Document
- 3. Architecture and Tenancy Design
- 4. User and Role Diagram
- 5. Database Overview Diagram
- 6. Module Documentation
- 7. Admin Portal Guide
- 8. Tenant User Guide
- 9. Security, Privacy, and Compliance
- 10. Deployment and Operations
- 11. Testing, Support, and Maintenance
- 12. Future Expansion Roadmap

1. Executive Summary

EBO SYSTEM is an expandable multi-tenant business operating system for organizations that need one secure workspace for business modules, collaboration, administration, billing, and industry workflows. The platform supports a public marketing site, a public-schema platform admin portal, and isolated tenant workspaces. Current implemented module families include tenant operations, HCM, CRM, Finance, Legal Practice, Smart Chat, Meetings, Notifications, Billing, Tenant Branding, and support documentation. The platform is intentionally not limited to these modules; new modules can be added for new industries and client-specific operations.

Capability	Current Scope
Multi-tenancy	Public schema for platform/admin data, tenant schemas for tenant business operations.
Branding	Tenant-specific colors, logos, portal name, tagline, and default fallback branding.
Access Control	Super admin access for platform admin, tenant admins for internal users, and module-level permissions.
Communication	Smart Chat, EBO AI conversations, support conversations, notifications, announcements, email alerts, and meetings.
Documents	PDF invoices/quotes/legal invoices, uploads, legal matter documents, static documentation PDF.
Integrations	SMTP email, Twilio click-to-call, Jitsi-compatible meeting provider, AI integration points.

2. SSD - Software/System Specification Document

2.1 Purpose

The purpose of EBO SYSTEM is to give SMEs and professional practices a modular platform for running daily operations in one tenant workspace. It should reduce scattered tools, centralize business records, support role-based access, and allow new modules to be added over time.

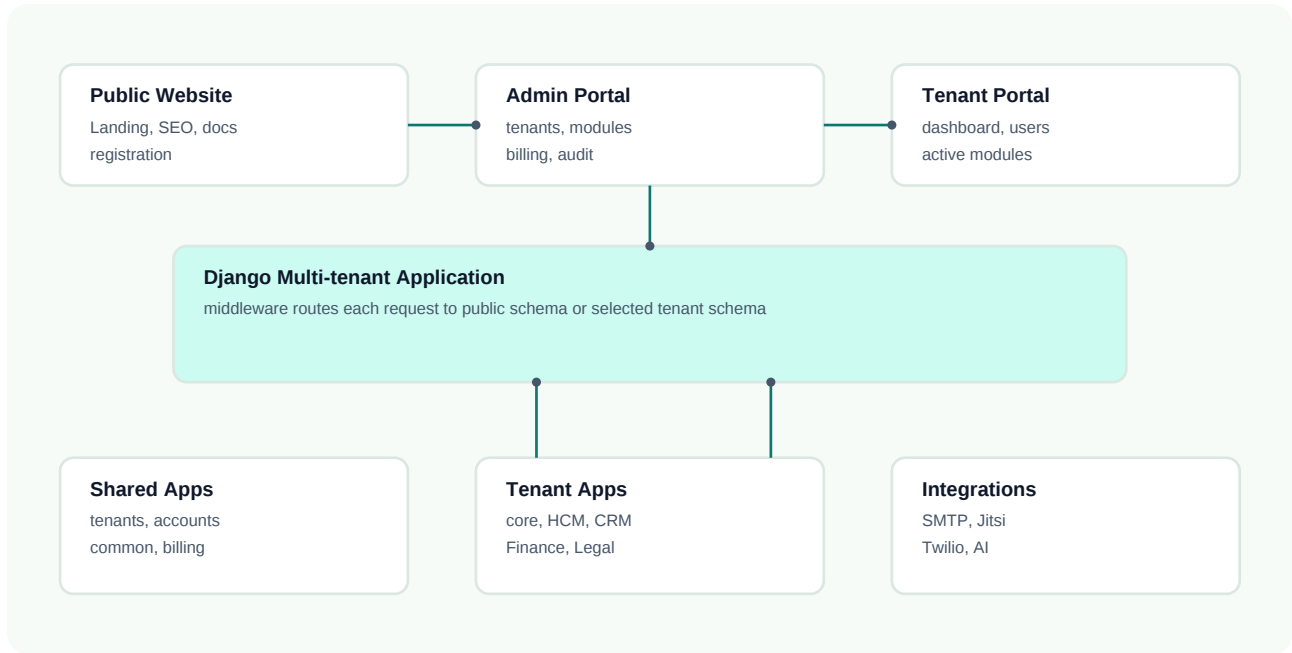
2.2 Functional Requirements

Area	Requirement
Public Website	Landing page, SEO metadata, pricing, features, contact, documentation, support, user guide, sitemap, robots.txt.
Tenant Registration	Create new tenant workspace and domain/subdomain records.
Admin Portal	Dashboard, tenants, modules, subscriptions, billing, users, roles, chat, announcements, audit logs, settings, tenant branding.
Tenant Portal	Dashboard, smart calendar, sidebar module dropdowns, notifications, users, module access, chat, meetings.
HCM	Employee records, departments, job positions, shifts, attendance, leave, payroll records, contracts, performance reviews.
CRM	Contacts, companies, leads, pipeline, opportunities, activities, quotes, sales orders, contracts, campaigns, CRM settings.
Finance	Accounts, journals, invoices, bills, payments, banking, reconciliation, assets, budgets, tax, SARS/IFRS-ready reports.
Legal	Matters, matter types, courts, checklists, documents, work sessions, calls, billable events, invoices, emails, reports.
Notifications	In-app notifications, announcements, email notifications, support chat updates, module event notifications.
Security	Tenant isolation, authentication, CSRF, secure cookies in production, HSTS, role and module permission checks.

2.3 Non-functional Requirements

- Responsive design for desktop, tablet, and mobile screens.
- Clean professional UI using Django Templates and Tailwind CSS classes.
- Tenant data must remain isolated through django-tenants schema routing.
- Public admin works on the public schema and should be limited to super admins/platform roles.
- System must support future modules without redesigning the tenant shell.
- Reports, invoices, and legal billing documents should be exportable to PDF where required.
- Email and telephony integrations must be controlled by environment variables.

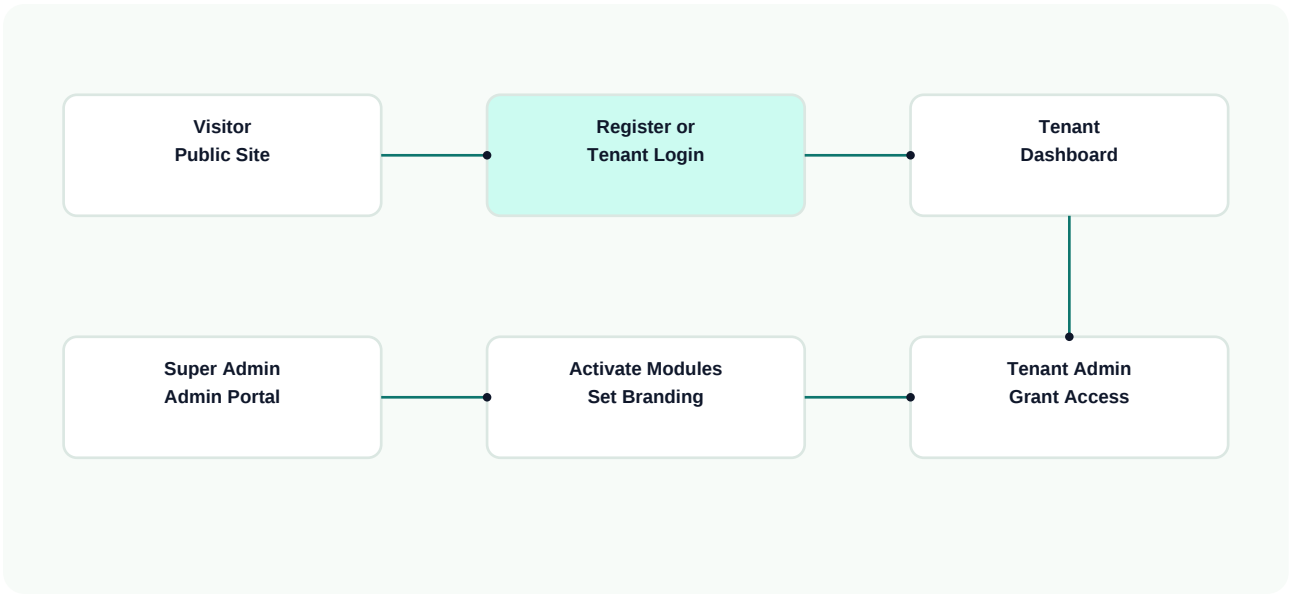
3. Architecture and Tenancy Design



The system uses django-tenants. Requests to public domains such as ebosystem.co.za use the public URL configuration and public schema. Requests to tenant subdomains such as spar.ebosystem.co.za use the tenant URL configuration and the tenant schema selected by the middleware.

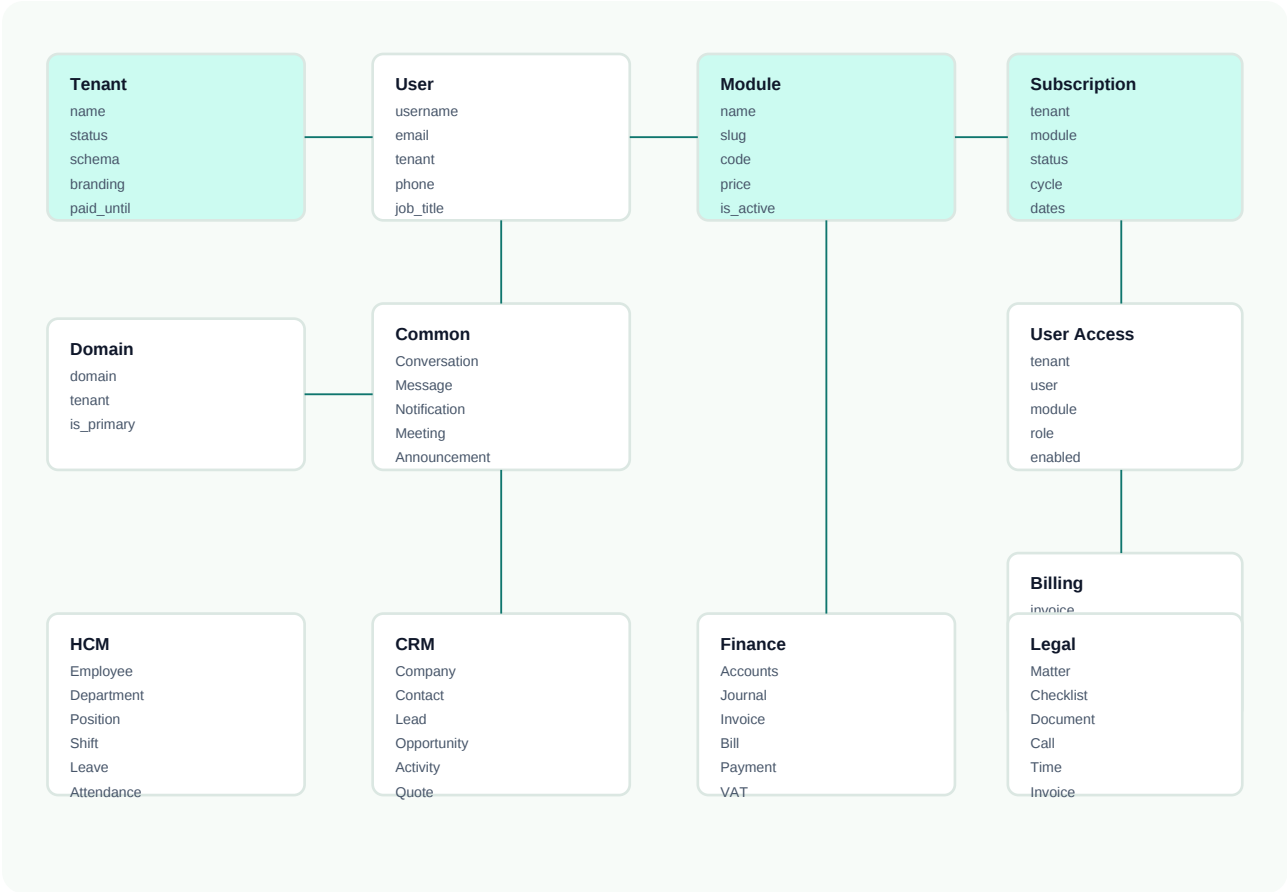
Layer	Responsibility
Public Site	Marketing, documentation, support, registration, SEO, sitemap, robots.txt.
Admin Portal	Platform operations across tenants, modules, subscriptions, branding, invoices, payments, audit logs.
Tenant Shell	Dashboard, sidebar, active modules, tenant branding, user management, notifications, chat, meetings.
Module Apps	HCM, CRM, Finance, Legal Practice, and future modules.
Persistence	PostgreSQL with public schema plus tenant schemas.
Static/Media	WhiteNoise for static files; media is local filesystem in current configuration.

4. User and Role Diagram



Role	What the role can do
Visitor	View public website, pricing, features, documentation, support, and register interest/workspace.
Super Admin	Access platform admin portal; manage tenants, modules, billing, platform roles, settings, logs, branding.
Platform Finance	Manage invoices, payments, revenue reporting, overdue accounts, and subscription billing.
Platform Support	Support tenants through chat, announcements, settings guidance, and issue triage.
Tenant Admin	Manage internal tenant users, grant module permissions, monitor dashboard, coordinate operations.
Tenant User	Use modules assigned to them, chat, join meetings, respond to notifications, perform daily work.
Module Role	Finance Admin, Accountant, HCM manager, Sales user, Attorney, Clerk, Viewer, or similar module-level access.

5. Database Overview Diagram



The diagram is a high-level ERD. In django-tenants, shared apps live in the public/shared schema, while tenant apps are synchronized to tenant schemas. Module access links a tenant, a module, and a user so the sidebar and permissions reflect active subscriptions and assigned access.

6. Module Documentation

Module	Main Models	Core Workflows
Platform/Admin	Tenant, Domain, Module, TenantSubscription, Invoice, Payment, AuditLog, PlatformSettings	Create tenants, activate modules, manage billing, set branding, invite platform admins, publish announcements, audit actions.
Tenant Core	Conversation, ChatMessage, Notification, Announcement, Meeting, MeetingParticipant, TenantUserModuleAccess	Dashboard, smart calendar, tenant users, module permissions, chat, AI support, tenant-to-admin support, video meetings.
HCM	Employee, Department, JobPosition, WorkShift, LeaveRequest, AttendanceRecord, SalaryStructure, EmploymentContract, PerformanceReview	Manage employees, shifts, attendance lateness, leave approvals, contracts, payroll preparation, and reviews.
CRM	Company, Contact, Lead, Opportunity, Activity, Campaign, Quote, SalesOrder, Contract, Territory, Competitor	360 customer records, lead scoring, pipeline, activities, quotes, sales orders, contracts, campaign tracking, import/export.
Finance	ChartOfAccount, JournalEntry, Invoice, Bill, Payment, BankAccount, BankTransaction, Budget, Asset, TaxRule, EMP201Record, IRP5Record	General ledger, sales invoicing, purchases, banking, reconciliation, reports, VAT, assets, budgets, SARS integration points.
Legal Practice	Matter, MatterType, ChecklistTemplateItem, MatterDocument, MatterWorkSession, MatterCall, BillableEvent, LegalInvoice	Matter files, checklist templates, missing documents, calls, work timers, billable time, invoices, PDFs, emails, legal audit trail.

Module Expansion Pattern

- Create a Django app for the new module and add it to tenant apps if data belongs inside tenant schemas.
- Create models, forms, URLs, class-based views, templates, permissions, and navigation entries.
- Create a Module record with slug, code, category, icon, URL namespace, pricing, and active status.
- Use TenantSubscription to activate the module per tenant.
- Use TenantUserModuleAccess to grant users access inside the tenant portal.
- Add sidebar dropdown entries for important sub-pages.
- Add notifications, audit records, emails, reports, and PDFs where the workflow requires them.

7. Admin Portal Guide

Page	Purpose
/admin/dashboard/	Overview cards, recent activity, MRR/manual revenue, subscriptions, signups, and pending actions.
/admin/tenants/	Tenant CRUD, filters, detail pages, activate/suspend/delete, reset password, reset branding, export CSV.
/admin/modules/	Manage modules, code, category, icons, prices, ordering, and global active/inactive status.
/admin/subscriptions/	Tenant vs Module matrix, activation, expiry, billing cycle, status, notes, activated_by, history, invoice creation.
/admin/billing/	Create invoices, record payments, monitor pending/overdue revenue, report monthly/yearly amounts.
/admin/users/	Invite platform admins and assign platform roles.
/admin/chat/	Respond to tenant support conversations.
/admin/announcements/	Publish global or tenant-specific announcements.
/admin/audit-logs/	Search and filter platform action history.
/admin/settings/	Manage company settings, contact details, default currency, tax and email settings.

8. Tenant User Guide

Daily tenant workflow

- Open the tenant subdomain and sign in with username/email and password.
- Use Forgot Password/Username if login details are lost.
- Review dashboard cards, active modules, workspace status, notifications, and calendar events.
- Use the sidebar menu button to open or close the sidebar on smaller screens.
- Open a module dropdown and select the relevant sub-page.
- Use Smart Chat for internal tenant chat, support chat with EBO admins, or AI assistant questions.
- Create meetings from the tenant portal and join video meetings when scheduled.
- Ask the tenant admin if a module is active but not visible for the user.

Tenant admin workflow

- Create and update internal tenant users.
- Grant module permissions only for modules that are active for the tenant.
- Disable access for inactive employees.
- Monitor notifications, announcements, module activity, and user support needs.
- Keep company records, user emails, and phone numbers accurate for notifications and calls.

9. Security, Privacy, and Compliance

Control	Implementation/Expectation
Tenant Isolation	django-tenants schema isolation separates tenant business data.
Authentication	Django authentication with custom User model and account recovery flows.
Authorization	Super admin/platform role restrictions for admin portal; module access restrictions for tenant users.
CSRF	Django CSRF middleware enabled.
Transport Security	Production uses SSL redirect, secure cookies, HSTS, and proxy SSL header.
Auditability	Platform AuditLog and module timelines capture important operational actions.
Email Secrets	SMTP credentials stored in environment variables, not code.
Telephony Secrets	Twilio SID/token/from-number stored in environment variables.
South Africa Compliance	Finance supports VAT concepts, SARS-ready report areas, EMP201/IRP5 integration points, CIPC/VAT fields.
Legal Compliance	Legal matters support FICA, POPIA consent, conflict checks, confidentiality, matter audit timeline, documents, and billing history.

10. Deployment and Operations

Item	Current Configuration
Runtime	Python 3.11, Django 5, Gunicorn, WhiteNoise.
Database	PostgreSQL through django-tenants backend. Heroku DATABASE_URL supported.
Static Files	collectstatic with WhiteNoise compressed manifest storage.
Media Files	FileSystemStorage currently configured. For production scale, move user uploads to object storage.
Domain	Public domain ebosystem.co.za; tenants use subdomains such as spar.ebosystem.co.za.
Release	Heroku Procfile includes release process that runs tenant migrations and seeds default modules.
Environment	SECRET_KEY, DEBUG, ALLOWED_HOSTS, PUBLIC_DOMAINS, TENANT_BASE_DOMAIN, DATABASE_URL, SMTP, Twilio, meeting domain.

Operational Checklist

- Before deployment, run python manage.py check.
- Commit only scoped changes and avoid unrelated working-tree files.
- Push to Heroku and confirm release command finishes.
- Confirm public page, tenant dashboard, admin dashboard, login, and critical module pages after deployment.
- Keep environment variables updated for production email, Twilio, allowed hosts, trusted origins, and domain settings.

11. Testing, Support, and Maintenance

Recommended Test Coverage

- Tenant registration creates Tenant and Domain correctly.
- Admin portal rejects non-superusers and unauthorized users.
- Module activation changes tenant sidebar and user module access.
- Tenant user management cannot expose users from other tenants.
- HCM attendance evaluates late/early status from shifts.
- CRM lead conversion creates customer/opportunity records.
- Finance journals must balance before posting.
- Legal call and work-session timers create billable entries correctly.
- PDF generation works for finance invoices, CRM quotes, and legal invoices.
- Email notifications send to correct tenant admins, clients, attorneys, clerks, and assigned users.

Support Process

Priority	Definition	Action
Urgent	Tenant down, login broken for many users, billing blocked, critical data issue.	Investigate immediately, check logs, confirm release, roll forward with hotfix.
High	Key module workflow blocked for one tenant.	Reproduce, inspect permissions/data, patch and deploy.
Normal	User question, form issue, report improvement, minor layout issue.	Guide user or schedule maintenance fix.
Enhancement	New module, new report, integration, workflow automation.	Capture requirement, estimate, design, build, test, deploy.

12. Future Expansion Roadmap

- Public downloadable documentation and versioned release notes.
- Object storage for tenant logos and uploaded documents.
- In-app help center with searchable module articles.
- Full AI assistant with permission-aware data retrieval per active module.
- Rocket.Chat or equivalent collaboration layer integration.
- Mobile applications for Android and iOS after web workflows stabilize.
- Desktop wrapper for clients that want a desktop-style app.
- Advanced telephony with production Twilio account or SIP/FreePBX integration.
- Additional industry modules: Legal, Hospice, Salon, Internship, Projects, Inventory, Help Desk, Field Service, Recruitment.
- More automated tests and CI/CD quality gates.

Document End