Tamiti Studio Backend Roadmap & Architecture

Author: Enos Kyeza\ **Date:** July 27, 2025\ **Project:** Tamiti Studio

---

🧭Development Roadmap (Modular Sprints)

A progressive roadmap aligning with Django 5.x best practices, DRF modular architecture, and async-ready infrastructure.

Foundational Sprint: Core & Auth

Apps: `core`, `users`, `auth`, `common`

Module	Key Features
core	Abstract models (`TimeStampedModel`, `UUIDModel`), signals, utilities
users	Custom `User` model with roles and profile logic
auth	JWT (SimpleJWT), login, registration, email verification, password reset
common	Enums, shared constants, generic pagination, throttling logic

---

🚀Sprint 1: Task & Project Management

Apps: `projects`, `tasks`

Module	Key Features
projects	Project creation, milestones, team roles, client metadata
tasks	Global task CRUD, tagging, due dates, status tracking

Phase 2: Gantt charts, project filters, and assignment matrix.

🏠Sprint 2: Finance Manager

Apps: `finance`, `media`

Module	Key Features
finance	Income/Expense tracking, requisition approval flows, goals, creditors & debtors
media	Upload and associate receipts/invoices via `MediaAsset`

Tech: Celery async reminders, payment schedule triggers

Sprint 3: Digital & Content Manager

Apps: `digital`, `content`, `media`

Module	Key Features
digital	Scheduled posts, platform-specific formatting, campaign calendar
content	Asset tagging, gallery curation, media usage logs
media	File roles, approval flow (draft > review > approved)

🧭**Sprint 4: Field & Lead Tracker**

Apps: `field`, `tasks`, `users`

Module	Key Features
field	Lead logging, conversion status, syncable offline support
tasks	Auto-linked follow-up tasks
users	Agent tagging, zone-based reporting

⚓**Sprint 5: Virtual Staff & Chat**

Apps: `chat`, `notifications`, `core.ai`

Module	Key Features
chat	AI personas, summaries, chat-based interface
notifications	Daily reminders, approvals, summary digests
tasks	Task summaries processed by Virtual Staff

Infra: Celery for async summaries, OpenAI/LLM integration

🚢**Sprint 6: Productivity & Notification Engine**

Apps: `focus`, `notifications`, `users`

Module	Key Features
focus	Login streak tracking, timer, motivational prompts
notifications	Alert levels, digest control
users	Behavior stats dashboard (logins, streaks)

App Structure Proposal

```
tamiti/
├── config/                 # Settings, entrypoints (ASGI, WSGI)
│   ├── settings/
│   │   ├── base.py
│   │   ├── dev.py
│   │   ├── prod.py
├── core/                   # Shared models, utils, signals
├── common/                 # Enums, pagination, throttles
├── users/                  # Custom user model, profile logic
├── auth/                   # Auth endpoints, password reset, email
├── tasks/                  # Global task manager
├── projects/               # Project team assignments, milestones
├── finance/                # Requisitions, income/expense, goals
├── media/                  # File uploads, approval flows
├── content/                # Gallery, asset usage, filters
├── digital/                # Social campaigns & scheduling
├── field/                  # Lead tracking, feedback forms
├── chat/                   # Virtual staff, chat prompts
├── notifications/          # Alerts, digest logic
├── focus/                  # Productivity tracking
├── templates/              # Emails, HTML templates
├── tests/                  # Global factories and test utils
```

Infrastructure & Tooling

Tool	Purpose
PostgreSQL	Primary DB with UUIDs and full-text search
Redis	Caching + Celery broker
Celery	Async task queue for summaries, reminders
DRF + drf-spectacular	API schema + docs

Tool	Purpose
SimpleJWT	Token-based authentication
pytest + factory_boy	Testing suite
Docker	Environment and deployment
GitHub Actions	CI/CD pipelines
Whitenoise / S3	Static & media file serving
Sentry	Monitoring and logging

🚚Security, Permissions & Observability

Concern	Strategy
Authentication	SimpleJWT (refresh/rotate)
Email Verification	Token + Celery email task
Role Management	Enum-based roles + field perms
Object-Level Security	Custom `permissions.py` per app
Throttling	DRF throttle classes, login throttles
Secrets Management	`django-environ`, `.env`, Docker secrets
Logging & Auditing	Django signals + admin logs

This roadmap is built to scale with Tamiti Studio's modular needs while supporting future AI integrations, role-specific dashboards, and white-label deployments.

Next step: Sprint 0 scaffolding (`core`, `users`, `auth`, `common`) + settings infrastructure