Workload Tracker Access by Role

Generated: January 9, 2026

Access Model Summary

- Backend default permissions: IsAuthenticated + RoleBasedAccessPermission when AUTH_ENFORCED is on.
- Admin is is_staff or is_superuser; Manager is in group "Manager"; User is in group "User" or no group.
- Frontend routes are gated only by RequireAuth (token check), not by role.
- Sidebar shows all pages to any authenticated user; role enforcement is server-side.
- Settings sections are hidden client-side using is_staff checks for admin-only sections.
- If AUTH_ENFORCED is disabled, most endpoints fall back to AllowAny except those explicitly using IsAdminUser/is_staff checks.

Role Overview (High Level)

- Admin: full read/write across the app, including admin-only endpoints and settings.
- Manager: full read/write on most non-admin endpoints; blocked from explicit admin-only endpoints.
- User: read-only across most endpoints; can edit only their own Person and their own Assignments.

Admin Capabilities

Pages: can access all pages (same as any authenticated user), with full write privileges.

- User management: create users, list users, change roles, delete users, invite users, link users to people, view admin audit logs.
- Backups: list/create/download/delete/restore/upload+restore.
- Integrations: manage connections, credentials, syncs, and provider config.
- Pre-deliverables: global settings and backfill operations.
- Calendar feed tokens and feed settings.
- Department project roles add/remove; project role CRUD/reorder.
- Project pre-deliverable settings updates.
- Role list reorder (global roles).
- Admin-only reports: pre-deliverable team performance.

Settings UI: sees all sections (Admin Users, Backup/Restore, Integrations, Calendar Feeds, Pre-Deliverables Backfill, Audit Log, Dept Project Roles, Role Management, Utilization Scheme).

Manager Capabilities

Pages: can access all pages, but admin-only settings sections are hidden in the UI.

- Backend permissions allow full CRUD on most core entities: People,

Departments, Projects, Assignments, Deliverables, Skills, Reports, etc.
- Managers are blocked from endpoints explicitly gated by IsAdminUser or is_staff checks.

Admin-only restrictions for managers include: user management, backups, integrations, calendar feed tokens, pre-deliverable global settings/backfill, department project role mutations, project role CRUD/reorder, project pre-deliverable settings updates, and role reordering.
Settings UI: only Role Management and Utilization Scheme are visible; Utilization Scheme is read-only (non-staff).

User Capabilities

Pages: can access all pages (same as any authenticated user).
- Read-only across most APIs.
- Write access only to their own Person record and their own Assignments, if linked via UserProfile.person.
- Can update own profile settings, link/unlink their own Person (with email checks), change password, set notification preferences.

Restrictions: cannot create/edit/delete other people, projects, departments, assignments (outside own), deliverables, roles, or admin-only endpoints.
Settings UI: Role Management and Utilization Scheme are visible, but write actions will 403 on the backend for users.

Page-Level Access Summary

All authenticated routes are reachable for Admin, Manager, and User (routing is token-only).
- Dashboard, People, Departments, Assignments, Project Assignments, Projects, Reports, Skills, Calendar, Settings, Profile, Help, My Work: all protected by RequireAuth only.
- Actual permissions are enforced by backend endpoints, not by route guards.

Notable Gaps / Mismatches

- Managers are not treated as admins in the UI because most checks use is_staff; managers can have broader backend access than the UI suggests.
- Users can see Role Management UI but will be blocked by backend on write actions.

Key Source Files

- backend/accounts/permissions.py (RoleBasedAccessPermission)
- backend/config/settings.py (REST_FRAMEWORK defaults, AUTH_ENFORCED)
- frontend/src/main.tsx (RequireAuth routing)
- frontend/src/components/layout/Sidebar.tsx (navigation visibility)
- frontend/src/pages/Settings/Settings.tsx and sections/index.tsx (section gating)
- backend/accounts/views.py (admin-only user management)
- backend/core/backup_views.py (backup/restore admin endpoints)

- backend/integrations/views.py (integrations admin endpoints)
- backend/projects/views.py (project pre-deliverable settings admin check)
- backend/projects/views_roles.py (department project roles admin endpoints)
- backend/roles/views.py (role reorder staff-only)
- backend/reports/views.py (admin-only report endpoint)