# Epic 4: System Administration

**Epic ID:** EPIC-004 **Priority:** P0 (High - MVP Foundation) **Status:** Not Started **Owner:** Product Owner (Sarah) **Target Release:** MVP (Week 1-4, ongoing)

## Epic Goal

Enable system administrators to configure the platform, manage users, set up departments and fiscal years, and monitor system health - providing the foundation for all other epics.

## Business Value

**Problem Solved:** - No way to onboard new users and assign roles - Manual user management via database (not sustainable) - Need to configure departments and fiscal years before departments can create plans - No visibility into system usage or errors

**Expected Outcomes:** - Self-service user management - Proper role-based access control - Clean administrative interface for configuration - Audit trail for administrative actions

**Success Metrics:** - 100% of users onboarded via admin interface (not manual database) - Zero unauthorized access incidents - All departments and fiscal years configured before plan creation begins - Audit logs capture 100% of system changes

## Epic Description

This epic provides administrative tools for system setup, user management, configuration, and monitoring. These are foundational capabilities that must exist before departments can use the system.

**Key Capabilities:**

1. **User Management**
   * Create/edit/delete users
   * Assign roles (admin, department\_director, staff, city\_manager, finance, council, public)
   * Assign users to departments
   * Deactivate users (soft delete)
   * View user list with filters
2. **Department Management**
   * Create/edit/delete departments
   * Configure department metadata (director, mission, etc.)
   * Activate/deactivate departments
   * View department list
3. **Fiscal Year Configuration**
   * Create fiscal years (year, start date, end date)
   * Mark current fiscal year
   * View fiscal year list
   * Cannot delete fiscal years with associated plans
4. **System Settings**
   * Configure city information (municipality record)
   * City strategic priorities (used in goal alignment)
   * System-wide settings (e.g., fiscal year start month)
5. **Audit Log Viewer**
   * View all system changes
   * Filter by: Table, User, Date range, Action (insert/update/delete)
   * Export audit log to CSV
6. **System Health Monitoring** (Future: Phase 2)
   * Dashboard showing: Active users, Plan count, System errors
   * Database performance metrics

## User Personas

**Primary User:** - **System Administrator** (IT staff, initial setup) - **City Manager** (May have admin privileges)

**Secondary Users:** - **Department Directors** (View-only access to their dept info)

## User Journey

### Setup Journey (One-time, before launch)

1. Admin logs in to system (Supabase Auth configured)
2. Create municipality record (City of Carrollton)
3. Create fiscal years (FY2025, FY2026, FY2027)
4. Create departments (Water & Field Services, Parks & Rec, IT, Finance, etc.)
5. Create users and assign to departments with roles
6. Configure city strategic priorities
7. System ready for department use

### Ongoing User Management

1. New employee joins city
2. Admin creates user account, assigns role and department
3. User receives email invitation to set password
4. User logs in, sees their department’s data only (RLS enforced)

## Stories

### Story 4.1: User Management - List Users

**As an** Admin **I want to** view a list of all users **So that** I can manage user accounts

**Acceptance Criteria:** - Admin can navigate to “Users” page - Page displays table of all users with: Full name, Email, Role, Department, Status (active/inactive), Last login - Admin can filter by: Role, Department, Status - Admin can search by name or email - Admin can sort by any column - Pagination supports 50 users per page

**Story Points:** 8 **Priority:** P0 **Dependencies:** None

### Story 4.2: User Management - Create User

**As an** Admin **I want to** create a new user account **So that** employees can access the system

**Acceptance Criteria:** - Admin clicks “Create User” button - Form prompts for: Full name, Email, Role (dropdown), Department (dropdown, optional for city-wide roles), Title - Admin submits form - System creates user record in users table - System sends invitation email to user (Supabase Auth) - User appears in user list - User can log in and set password via invitation link

**Story Points:** 13 **Priority:** P0 **Dependencies:** Story 4.1, Supabase Auth configured

### Story 4.3: User Management - Edit User

**As an** Admin **I want to** edit a user’s role or department **So that** I can update access as responsibilities change

**Acceptance Criteria:** - Admin can click “Edit” on user row - Form displays current user data (pre-filled) - Admin can change: Full name, Role, Department, Title, Status (active/inactive) - Admin cannot change Email (linked to auth) - Admin submits changes - System updates user record - Changes reflected immediately in user list - Audit log captures change

**Story Points:** 8 **Priority:** P0 **Dependencies:** Story 4.1, Story 4.2

### Story 4.4: User Management - Deactivate User

**As an** Admin **I want to** deactivate a user account **So that** former employees cannot access the system

**Acceptance Criteria:** - Admin can click “Deactivate” on user row - System prompts for confirmation - System sets is\_active = false on user record - User cannot log in (Supabase Auth disabled) - User still appears in audit logs and historical records - Admin can reactivate user later if needed

**Story Points:** 5 **Priority:** P0 **Dependencies:** Story 4.1

### Story 4.5: Department Management - List Departments

**As an** Admin **I want to** view a list of all departments **So that** I can manage department configuration

**Acceptance Criteria:** - Admin can navigate to “Departments” page - Page displays table of departments with: Name, Director name, Director email, # Staff, # Plans, Status (active/inactive) - Admin can filter by: Status - Admin can search by name - Admin can sort by any column

**Story Points:** 5 **Priority:** P0 **Dependencies:** None

### Story 4.6: Department Management - Create Department

**As an** Admin **I want to** create a new department **So that** departments can create strategic plans

**Acceptance Criteria:** - Admin clicks “Create Department” button - Form prompts for: Name, Slug (auto-generated from name, editable), Director name, Director email, Mission statement, Core services (list) - Admin submits form - System creates department record - Department appears in department list - Department available in user assignment dropdowns

**Story Points:** 8 **Priority:** P0 **Dependencies:** Story 4.5

### Story 4.7: Department Management - Edit Department

**As an** Admin **I want to** edit department information **So that** I can keep department metadata current

**Acceptance Criteria:** - Admin can click “Edit” on department row - Form displays current department data (pre-filled) - Admin can change: Name, Director, Mission, Core services, Status - Admin submits changes - System updates department record - Changes reflected in department list and all references

**Story Points:** 5 **Priority:** P0 **Dependencies:** Story 4.5

### Story 4.8: Fiscal Year Configuration - List Fiscal Years

**As an** Admin **I want to** view all configured fiscal years **So that** I can manage the fiscal year calendar

**Acceptance Criteria:** - Admin can navigate to “Fiscal Years” page - Page displays table of fiscal years with: Year, Start date, End date, Is current (checkbox/badge), # Plans - Fiscal years sorted by year descending (most recent first) - Current fiscal year highlighted

**Story Points:** 3 **Priority:** P0 **Dependencies:** None

### Story 4.9: Fiscal Year Configuration - Create Fiscal Year

**As an** Admin **I want to** create a fiscal year **So that** departments can create plans for that year

**Acceptance Criteria:** - Admin clicks “Create Fiscal Year” button - Form prompts for: Year (integer, e.g., 2026), Start date, End date, Is current (checkbox) - System validates: End date > Start date, Year not duplicate - Admin submits form - System creates fiscal year record - Fiscal year appears in fiscal year list - Fiscal year available in strategic plan creation

**Story Points:** 5 **Priority:** P0 **Dependencies:** Story 4.8

### Story 4.10: Municipality Configuration

**As an** Admin **I want to** configure municipality information **So that** the system displays correct city branding and settings

**Acceptance Criteria:** - Admin can navigate to “System Settings” page - Form displays municipality info: Name, State, Settings (JSONB) - Admin can edit: City name, State, Timezone, Fiscal year start month, Currency - Admin can add city strategic priorities (list) - used in goal alignment - Admin submits changes - System updates municipality record - Changes reflected throughout application (e.g., city name in headers)

**Story Points:** 8 **Priority:** P1 **Dependencies:** None

### Story 4.11: Audit Log Viewer

**As an** Admin **I want to** view audit logs of all system changes **So that** I can investigate issues or ensure accountability

**Acceptance Criteria:** - Admin can navigate to “Audit Log” page - Page displays table of audit log entries with: Timestamp, User, Table, Record ID, Action (insert/update/delete), Changes summary - Admin can filter by: Table name, User, Date range, Action - Admin can click entry to see full before/after values (JSONB) - Admin can export filtered audit log to CSV - Pagination supports 100 entries per page - Logs sorted by timestamp descending (most recent first)

**Story Points:** 13 **Priority:** P0 **Dependencies:** Audit log triggers configured in database

### Story 4.12: Admin Dashboard

**As an** Admin **I want to** see a dashboard of system status **So that** I can quickly understand system health

**Acceptance Criteria:** - Admin sees dashboard on login (or dedicated “Admin Home”) - Dashboard displays: - Total users (active vs. inactive) - Total departments - Total strategic plans by status - Total initiatives - Recent audit log entries (last 10) - System alerts (e.g., no current fiscal year marked) - Links to: User management, Department management, Fiscal years, Audit log

**Story Points:** 8 **Priority:** P1 **Dependencies:** Stories 4.1, 4.5, 4.8, 4.11

## Feature Requirements References

* FR-007: User Roles & Permissions
* FR-010: Audit Trail

## Technical Considerations

**Authorization:** - Only users with role “admin” can access admin pages (RLS + UI routing) - Admin role has unrestricted access to all data (RLS policies)

**User Invitation:** - Use Supabase Auth invite flow - Send email with password setup link - Set user is\_active = false until they accept invitation (optional)

**Data Integrity:** - Cannot delete departments with associated plans or users - Cannot delete fiscal years with associated plans - Soft delete preferred (set is\_active = false)

**Audit Logging:** - All admin actions logged (user creation, role changes, department edits) - Audit triggers on key tables

## Definition of Done

**Epic is complete when:**

* All 12 stories completed with acceptance criteria met
* Admin can create and manage users with proper roles
* Admin can configure departments and fiscal years
* Admin can view audit logs
* RLS policies enforce admin permissions correctly
* User invitation flow works (email sent, password setup)
* System configuration persists and displays correctly
* All admin actions logged in audit trail
* Unit and integration tests complete
* Accepted by Product Owner in UAT

## Dependencies & Risks

**Dependencies:** - Supabase Auth configured and working - Database migrations complete (users, departments, fiscal\_years, audit\_logs) - Email service configured (Supabase or SMTP)

**Risks:**

| Risk | Likelihood | Impact | Mitigation |
| --- | --- | --- | --- |
| User invitation emails not delivered | Medium | High | Test email flow early, configure SPF/DKIM, have manual backup process |
| Admin accidentally deletes critical data | Low | High | Implement soft deletes, confirmation dialogs, audit trail |
| RLS policies too restrictive or too permissive | Medium | Critical | Extensive testing of all roles, security review |

## Notes

* This epic is foundational - must be completed early in MVP
* Prioritize Stories 4.1-4.4 (user management), 4.5-4.7 (departments), 4.8-4.9 (fiscal years) first
* Audit log viewer (4.11) is critical for government accountability - don’t skip
* Municipality configuration (4.10) can be simplified (hardcoded for Carrollton in MVP)
* Admin dashboard (4.12) is nice to have but not critical for launch

**Epic Created:** January 9, 2025 **Last Updated:** January 9, 2025