NEOX Infinity App - User Management Module

Complete User Administration System

Version: 3.0
Last Updated: 2025-10-28
Module: User Management

Overview

The User Management module provides comprehensive user administration including CRUD operations, role-based access control, permissions management, bulk operations, and user directory features.

Key Features

- User list with advanced filtering & search
- Create/Edit users with full profile management
- Role & permission management
- Bulk user import/export
- User directory with organizational hierarchy
- Self-service profile management
- Session management & monitoring
- Password policies & security settings

Screen Specifications

1. User List Screen

Route: `/users`

Layout: Data table with filters, search, and bulk actions

Components:

```
// Main table with columns
- Avatar + Name
- Email
- Role
- Department
- Status (Active/Inactive badge)
- Last Login
- Actions (Edit, Delete, More)

// Filters (top)
- Search (name, email)
- Role filter (dropdown)
- Status filter (Active, Inactive, All)
- Department filter (tree select)
- Date range (registration date)

// Bulk Actions
- Activate selected
- Deactivate selected
- Export selected
- Delete selected (with confirmation)

// Pagination
- 10, 25, 50, 100 per page
- Total count display
```

API: `GET /api/users?page=1&limit=25&search=&role=&status=`

2. Create User Screen

Route: `/users/create`

Form Fields:

```
 // Personal Information
- First Name (required, 2-50 chars)
- Last Name (required, 2-50 chars)
- Email (required, unique, email format)
- Phone (optional, international format)
- Profile Photo (upload, max 5MB, JPG/PNG)

// Account Information
- Role (dropdown: Admin, Manager, User)
- Department (tree select)
- Job Title (optional)
- Employee ID (optional, alphanumeric)

// Access Settings
- Password (auto-generate or manual, 8+ chars)
- Send welcome email (checkbox)
- Force password change on first login (checkbox)
- Account status (Active/Inactive)

// Permissions (if custom role)
- Permission tree with checkboxes
- Module-level permissions
```

Validation:

- Email uniqueness check (async)
- Password strength indicator
- Required field validation
- Format validation (phone, email)

API: `POST /api/users`

---

3. Edit User Screen

Route: `/users/:id/edit`

Features:

- Pre-populated form with current data
- Change tracking (highlight modified fields)
- Unsaved changes warning
- Delete user button (with confirmation modal)
- Deactivate/Activate toggle
- Reset password option
- Activity log (sidebar)

API: `PUT /api/users/:id`

---

4. User Detail Screen

Route: `/users/:id`

Tabs:

1. **Overview**
   - Profile information display
   - Contact details
   - Department & role
   - Account status
   - Quick actions (Edit, Deactivate, Reset Password)
2. **Activity**
   - Login history (date, IP, device, location)
   - Recent actions timeline
   - Filter by date range
   - Export activity log

3. **Permissions**
    - Current permissions matrix
    - Inherited vs custom permissions
    - Edit permissions button
4. **Settings**
    - Notification preferences
    - Language & timezone
    - Two-factor authentication status
    - API access settings

API: `GET /api/users/:id`

---

5. Roles Management Screen

Route: `/users/roles`

Components:

```
// Role List
- System roles (cannot delete)
- Custom roles (editable)
- User count per role
- Actions (Edit, Duplicate, Delete)

// Create/Edit Role Modal
- Role Name (required)
- Description
- Permission Matrix (module-based checkboxes)
  - User Management (Create, Read, Update, Delete)
  - Visitor Management (...)
  - Parking Management (...)
  - [All modules...]
- Inherit from role (optional)

// Permission Groups
- Full Access
- Read Only
- Custom
```

API:

- `GET /api/roles`
- `POST /api/roles`
- `PUT /api/roles/:id`
- `DELETE /api/roles/:id`

---

6. Permissions Screen

Route: `/users/permissions`

Features:

- Permission tree view (hierarchical)
- Bulk permission assignment
- Permission templates
- Search permissions
- Filter by module
- Role-permission mapping table

---

7. Bulk User Import Screen

Route: `/users/import`

Workflow:

1. **Upload CSV**
    - Download template button

    - Drag & drop or browse
    - File validation (max 10MB, .csv only)

2. **Field Mapping**

    - Auto-detect headers
    - Map CSV columns to user fields
    - Preview first 5 rows
    - Required field indicators

3. **Validation**

    - Show validation errors
    - Line-by-line error display
    - Skip invalid rows option
    - Download error report

4. **Import**

    - Progress bar
    - Success/error count
    - View imported users button

API: `POST /api/users/import`

---

8. User Directory Screen

Route: `/users/directory`

Views:

- **Grid View**: User cards with avatars
- **List View**: Compact table
- **Org Chart**: Hierarchical tree

Features:

- Department tree navigation (left sidebar)
- Search users
- Quick contact (email, call buttons)
- Export to contacts (vCard)
- Presence indicators (online/offline)

---

9. User Profile Screen (Self-Service)

Route: `/profile`

Sections:

1. **Personal Information**

    - Edit own details
    - Change profile photo
    - Contact information

2. **Account Security**

    - Change password
    - Enable/disable 2FA
    - Backup codes
    - Active sessions
    - Login history

3. **Preferences**

    - Language
    - Timezone
    - Date/time format
    - Notification settings
    - Email digest preferences

4. **Activity Log**

    - My recent actions
    - Download my data (GDPR)

API:

- `GET /api/profile`
- `PUT /api/profile`

---

10. User Deactivation Screen

Route: `/users/:id/deactivate`

Deactivation Checklist:

- □ Transfer ownership of resources
- □ Reassign pending tasks
- □ Backup user data
- □ Revoke access tokens
- □ Archive email correspondence

Options:

- Deactivate immediately
- Schedule deactivation (date picker)
- Permanent deletion (30-day retention)

API: `POST /api/users/:id/deactivate`

---

11. Session Management Screen

Route: `/users/:id/sessions`

Display:

```
 // Active Sessions Table
- Device (icon, name)
- Browser
- IP Address
- Location (city, country)
- Last Active
- Actions (Revoke)

// Actions
- Revoke session (individual)
- Revoke all sessions
- Force logout
```

API:

- `GET /api/users/:id/sessions`
- `DELETE /api/users/:id/sessions/:sessionId`

---

12. User Settings Screen

Route: `/users/settings`

Configuration:

```
// Password Policies
- Minimum length (8-32)
- Require uppercase
- Require lowercase
- Require numbers
- Require special characters
- Password expiry (days)
- Password history (prevent reuse)

// Account Lockout
- Failed attempts threshold (3-10)
- Lockout duration (minutes)
- Auto-unlock after duration

// Session Settings
- Session timeout (minutes)
- Concurrent sessions limit
- Remember me duration (days)

// Registration Settings
- Allow self-registration
- Email verification required
- Default role for new users
- Auto-approve registrations

// MFA Settings
- Require MFA for roles
- MFA methods (SMS, Email, Authenticator)
- Grace period (days)
```

API:

- `GET /api/users/settings`
- `PUT /api/users/settings`

API Endpoints

Users

GET /api/users

```
// Response
{
  "users": [
    {
      "id": "usr_123",
      "firstName": "John",
      "lastName": "Doe",
      "email": "john@example.com",
      "role": { "id": "role_1", "name": "Admin" },
      "department": { "id": "dept_1", "name": "IT" },
      "status": "active",
      "avatar": "https://...",
      "lastLogin": "2025-10-28T10:00:00Z",
      "createdAt": "2025-01-01T00:00:00Z"
    }
  ],
  "total": 150,
  "page": 1,
  "limit": 25
}
```

POST /api/users

```
// Request
{
  "firstName": "Jane",
  "lastName": "Smith",
  "email": "jane@example.com",
  "phone": "+1234567890",
  "roleId": "role_2",
  "departmentId": "dept_1",
  "password": "SecurePass123!",
  "status": "active",
  "sendWelcomeEmail": true
}

// Response (201)
{
  "user": { "id": "usr_124", ... },
  "message": "User created successfully"
}
```

PUT /api/users/:id **DELETE /api/users/:id**

Roles

GET /api/roles **POST /api/roles** **PUT /api/roles/:id** **DELETE /api/roles/:id**

Permissions

GET /api/permissions - Get all available permissions **GET /api/users/:id/permissions** - Get user's effective permissions **PUT /api/users/:id/permissions** - Update user's custom permissions

---

Database Schema

```
// User Model
interface User {
  id: string;
  firstName: string;
  lastName: string;
  email: string;
  phone?: string;
  avatar?: string;
  password: string; // hashed
  roleId: string;
  departmentId?: string;
  jobTitle?: string;
  employeeId?: string;
  status: 'active' | 'inactive';
  emailVerified: boolean;
  twoFactorEnabled: boolean;
  lastLogin?: Date;
  passwordChangedAt?: Date;
  preferences: UserPreferences;
  createdAt: Date;
  updatedAt: Date;
  deletedAt?: Date;
}

// Role Model
interface Role {
  id: string;
  name: string;
  description?: string;
  permissions: string[]; // array of permission IDs
  isSystem: boolean; // cannot be deleted
  createdAt: Date;
  updatedAt: Date;
}

// Permission Model
interface Permission {
  id: string;
  name: string;
  resource: string; // e.g., "users"
  action: string; // e.g., "create", "read", "update", "delete"
  module: string; // e.g., "user_management"
}

// Session Model
interface Session {
  id: string;
  userId: string;
  token: string;
  refreshToken: string;
  device: string;
  browser: string;
  ip: string;
  location?: string;
  lastActive: Date;
  expiresAt: Date;
  createdAt: Date;
}
```

Validation Rules

User Creation/Update:

- First name: 2-50 characters, letters only
- Last name: 2-50 characters, letters only
- Email: Valid format, unique across system
- Phone: International format, optional
- Password: 8+ characters, uppercase, lowercase, number, special char
- Employee ID: Alphanumeric, unique if provided

Role Management:

- Role name: 3-50 characters, unique
- Cannot delete role if users assigned
- Cannot edit system roles (Admin, User)

Bulk Import:

- Max 1000 users per import
- CSV max 10MB
- Required columns: firstName, lastName, email
- Email must be unique across import

Security Considerations

1. **Password Security**
    - Use bcrypt with salt rounds 10+
    - Never log passwords
    - Enforce password policies
    - Require password change on first login
2. **Permission Checks**
    - Verify permissions on every API call
    - Use middleware for consistent checks
    - Log permission denials
3. **Session Security**
    - Use httpOnly cookies for tokens
    - Implement refresh token rotation
    - Track session activity
    - Allow remote session revocation
4. **Data Privacy**
    - Mask sensitive data in logs
    - Implement data retention policies
    - Provide data export (GDPR)
    - Soft delete with 30-day retention

Implementation Checklist

Frontend

- □ Create all 12 screen components
- □ Implement data table with sorting, filtering, pagination
- □ Add form validation with real-time feedback
- □ Create role & permission matrix UI
- □ Build bulk import wizard
- □ Add user directory with org chart
- □ Implement session management UI
- □ Create settings configuration screens
- □ Add loading states for all async operations
- □ Implement error handling and user feedback
- □ Add accessibility (ARIA labels, keyboard nav)
- □ Write unit tests for components
- □ Write E2E tests for user flows

Backend

- □ Implement all API endpoints
- □ Set up database models and migrations
- □ Add authentication middleware
- □ Implement permission checking middleware
- □ Create password hashing utilities
- □ Build email notification service

- □ Implement session management
- □ Add bulk import processing (queue-based)
- □ Create audit logging
- □ Implement rate limiting
- □ Add input validation
- □ Write API tests
- □ Set up monitoring and alerting

DevOps

- □ Configure database indexes for performance
- □ Set up Redis for session storage
- □ Configure email service (SendGrid/SES)
- □ Set up file storage for avatars (S3/CloudStorage)
- □ Implement backup strategy
- □ Configure monitoring (Sentry, DataDog)

End of User Management Module Specification

This module provides complete user administration capabilities with role-based access control, suitable for enterprise deployments.