

# Software Requirements Document (SRD)

## 1. Overview

This document outlines the requirements for the frontend development of three microservices/applications within the DNA ecosystem: Dashboard, LMS (Leave Management System), and PODs. The focus is on UI/UX design, API contracts for integration, and overall frontend development guidelines.

---

## 2. UI/UX Design Guidelines

### 2.1 Color Scheme

- **Primary Color:** #007bff (Blue - for primary actions)
- **Secondary Color:** #6c757d (Gray - for secondary actions)
- **Background Color:** #f8f9fa (Light Gray - for application background)
- **Success Color:** #28a745 (Green - for success messages)
- **Error Color:** #dc3545 (Red - for error messages)

### 2.2 Typography

- **Font Family:** "Inter", sans-serif
- **Heading Font Size:** 24px (Bold)
- **Subheading Font Size:** 18px (Medium)
- **Body Text Size:** 16px (Regular)
- **Button Text Size:** 14px (Bold)

### 2.3 Components

- **Buttons:** Rounded corners (8px), filled for primary actions, outlined for secondary actions.
  - **Cards:** Shadow (box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1)) with padding (16px).
  - **Modals:** Centered popups with a semi-transparent background.
  - **Forms:** Input fields with a border radius of 5px and padding of 10px.
- 

## 3. Application Features

### 3.1 Dashboard

- Displays multiple tiles reflecting highlights from various applications.
- Each tile fetches real-time data from APIs.
- Users can customize which tiles appear on their dashboard.

## API Contract

### Fetch Dashboard Data

#### Request:

```
GET /api/dashboard
Headers: { Authorization: Bearer <token> }
```

#### Response:

```
{
  "tiles": [
    { "id": "1", "title": "Leave Summary", "content": "10 leaves remaining" },
    { "id": "2", "title": "Pod Members", "content": "3 active members" }
  ]
}
```

---

## 3.2 LMS (Leave Management System)

### General User Features

- Apply for leave.
- View granted leaves.
- Check available leave balance.

### Manager Features

- Approve/reject leave requests.
- View team leave history.

## API Contract

### Apply for Leave

#### Request:

```
POST /api/lms/leave/apply
Headers: { Authorization: Bearer <token> }
Body:
{
  "startDate": "2025-03-15",
  "endDate": "2025-03-18",
}
```

```
    "reason": "Family event"
  }
```

### **Response:**

```
{
  "message": "Leave request submitted successfully",
  "status": "pending"
}
```

### **Approve Leave (Manager Only)**

#### **Request:**

```
POST /api/lms/leave/approve
Headers: { Authorization: Bearer <token> }
Body:
{
  "leaveId": "12345",
  "status": "approved"
}
```

#### **Response:**

```
{
  "message": "Leave request approved",
  "status": "approved"
}
```

---

## **3.3 PODs**

- Managers can add employees to a pod.
- Employees can view their pod details.
- Employees can recommend other employees to join their pod.

### **API Contract**

#### **View Pod Details**

#### **Request:**

```
GET /api/pods/details
Headers: { Authorization: Bearer <token> }
```

#### **Response:**

```
{
  "podId": "56789",
  "podName": "Innovation Team",
  "members": [
```

```
{ "id": "1", "name": "John Doe", "role": "Lead Developer" },
{ "id": "2", "name": "Jane Smith", "role": "UI/UX Designer" }
]
```

### **Recommend an Employee for a Pod**

#### **Request:**

```
POST /api/pods/recommend
Headers: { Authorization: Bearer <token> }
Body:
{
  "podId": "56789",
  "recommendedUserId": "3"
}
```

#### **Response:**

```
{
  "message": "Recommendation sent successfully"
}
```

---

## **4. Authentication & Authorization**

- Users authenticate via JWT-based login.
- Role-based access control (RBAC) to differentiate between managers and employees.

### **API Contract**

#### **Login**

#### **Request:**

```
POST /api/auth/login
Body:
{
  "email": "user@example.com",
  "password": "securepassword"
}
```

#### **Response:**

```
{
  "token": "jwt-token-here",
  "user": { "id": "1", "role": "manager" }
}
```

#### **Fetch Current User**

**Request:**

```
GET /api/auth/me
Headers: { Authorization: Bearer <token> }
```

**Response:**

```
{
  "id": "1",
  "name": "John Doe",
  "role": "manager"
}
```

---

## 6. Deployment & Hosting Considerations

- **Frontend Framework:** Angular with Angular with Material UI.
  - **Hosting:** Vercel, Netlify, or S3 with CloudFront.
  - **CI/CD:** GitHub Actions or GitLab CI/CD for automated deployment.
- 

## 7. Conclusion

This document provides frontend implementation guidelines, UI/UX specifications, API contracts, and integration details for Dashboard, LMS, and PODs applications.

---