

Software Requirements Specification (SRS)

Full-Stack Employee & Family Registry

Date: 15 Feb, 2026

Project: Employee Management System for Bangladesh Context

Version: 1.0

1. System Scope

1.1 What the Software Does

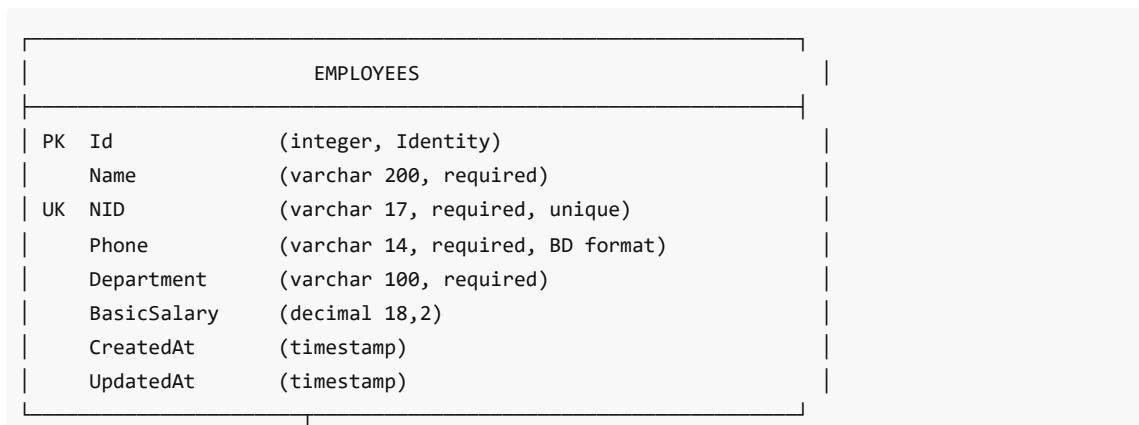
The Employee Management System is a full-stack web application designed for the Bangladesh context that:

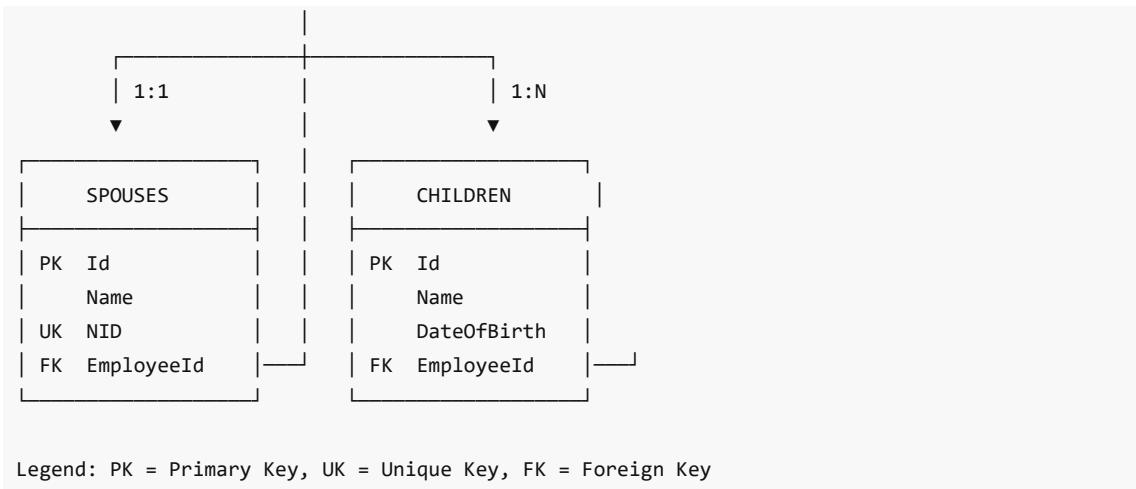
- **Manages employee profiles** with fields: Name, NID (10 or 17 digits), Phone (Bangladesh format), Department, and Basic Salary
- **Manages family data** with One-to-One relationship for Spouse (Name, NID) and One-to-Many relationship for Children (Name, Date of Birth)
- **Provides global search** with a single input field filtering by Name, NID, or Department (case-insensitive)
- **Generates PDF reports:**
 - Table View: PDF of the current filtered employee list
 - Individual CV: PDF summary of one employee including their family details
- **Implements role-based access:**
 - **Viewer:** Read-only access (view employees, search, export PDFs)
 - **Admin:** Full Create, Update, and Delete capabilities

1.2 What the Software Does NOT Do

- **Does not integrate with external HR systems** (e.g., payroll, attendance)
- **Does not support bulk import/export** of employee data (except PDF export)
- **Does not manage organizational hierarchy** (managers, departments as entities)
- **Does not provide audit trails** or change history
- **Does not support multi-tenancy** or multiple organizations
- **Does not validate NID authenticity** against government databases—only format and uniqueness
- **Does not support file uploads** (e.g., photos, documents)

2. Entity Relationship Diagram (ERD)





Relationships:

- **Employee ↔ Spouse:** One-to-One (optional). Each employee can have at most one spouse.
- **Employee ↔ Children:** One-to-Many. Each employee can have zero or more children.
- **Cascade Delete:** Deleting an employee automatically deletes associated spouse and children.

3. Edge Cases & Handling

3.1 NID Duplicates

Scenario	Handling
Employee NID duplicate on Create	FluentValidation + service-level check. Returns 400 Bad Request with message: "An employee with NID '{nid}' already exists."
Employee NID duplicate on Update	Service checks NID uniqueness excluding current employee. Same error response.
Spouse NID duplicate	Spouse NID has unique index in database. Checked in EmployeeService.CreateEmployeeAsync and UpdateEmployeeAsync. Returns same DuplicateNidException.
Employee and Spouse same NID	Allowed (different entities). Only uniqueness within same entity type is enforced.

3.2 Invalid Phone Numbers

Scenario	Handling
Invalid BD format	FluentValidation: BevalidBDPhone() validates format. Accepts: +8801[3-9]xxxxxxxx (14 chars) or 01[3-9]xxxxxxxx (11 chars). Returns 400 with validation message.
Empty phone	NotEmpty() rule. Returns validation error.

3.3 Invalid NID Format

Scenario	Handling

Wrong length	Must be exactly 10 or 17 digits. FluentValidation <code>BeValidNID()</code> enforces.
Non-numeric	Must contain only digits. Validation rejects.

3.4 Other Edge Cases

Scenario	Handling
Empty search query	API returns 400 Bad Request when <code>q</code> is null or whitespace.
Child DoB in future	FluentValidation: <code>DateOfBirth < DateTime.Now</code> . Rejected.
Employee not found	Returns 404 Not Found for GetById, Update, Delete, and CV PDF.
Unauthorized access	JWT required for Admin-only endpoints. Returns 401 Unauthorized OR 403 Forbidden.

4. Assumptions

1. **One spouse per employee:** An employee can have at most one spouse. The system does not support multiple spouses or complex family structures.
2. **NID format:** Bangladesh NID is either 10 digits (old) or 17 digits (new). No verification against government database.
3. **Phone format:** BD mobile numbers use operators 013–019. Landline or other formats are not supported.
4. **Authentication:** Simple username/password stored in configuration. No user registration; predefined Admin and Viewer accounts.
5. **Seeding:** Seed data is applied via EF Core migrations. On first run, `Database.Migrate()` runs migrations which include seed data. No separate seed command.
6. **PDF export for filtered list:** Uses the same search parameter `q` as the search API. Empty `q` exports all employees.
7. **Case-insensitive search:** Implemented in backend using `ToLower()` for Name and Department; NID compared as-is (numeric).

5. Technical Architecture Summary

- **Backend:** .NET 10, ASP.NET Core Web API, EF Core, PostgreSQL, FluentValidation, QuestPDF, JWT
- **Frontend:** React 18, Vite, Tailwind CSS, Axios
- **Database:** PostgreSQL with EF Core Migrations
- **Auth:** JWT Bearer tokens; roles: Admin, Viewer