Employer-Worker Registration System

A comprehensive Java desktop application for organisational management, featuring hierarchical employee structures and department coordination

Technology Stack: Java 8+, Swing UI Framework, MySQL 8.0+, JDBC Connectivity

Project Overview

Purpose & Core Functionality

A desktop application designed for managing employer and worker registrations within a hierarchical organisational structure. The system provides comprehensive CRUD operations across departments, employers, and workers.

- Complete department management hierarchy
- Employer registration with company details
- Worker assignment and tracking
- Consolidated record viewing with search capabilities

Architecture & Database

Built on a three-tier architecture ensuring clear separation of concerns. The MySQL database (employee_worker_db) maintains referential integrity throughout all operations.

- Presentation layer with Java Swing components
- Business logic with DAO pattern implementation
- Data persistence via MySQL with JDBC

System Architecture

Presentation Layer

Java Swing UI components with Nimbus Look and Feel provide a modern desktop interface

- MainFrame with tabbed interface
- Department, Employer, Worker tabs
- View Records with search functionality

Business Logic Layer

DAO pattern implementation ensures clean separation and maintainable code architecture

- DepartmentDAO for department operations
- EmployerDAO for employer management
- WorkerDAO for worker administration

Data Persistence Layer

MySQL database with JDBC connectivity manages all data storage and retrieval operations

- DatabaseConnection class for connectivity
- Prepared statements for security
- Transaction management for integrity

Technology Stack Components

Component	Technology	Version	Purpose
Programming Language	Java	8+	Core application logic and implementation
UI Framework	Java Swing	Built-in	Desktop GUI with Nimbus Look and Feel
Database	MySQL	8.0+	Relational data persistence and storage
Database Driver	MySQL Connector/J	8.0.33	JDBC connectivity and database operations
Build System	Batch Scripts	N/A	compile.bat and run.bat for deployment

This carefully selected technology stack provides a robust foundation for enterprise desktop application development, balancing reliability with modern Java capabilities.

Database Schema & Entity Relationships

Three-Table Architecture

Departments Table: Foundation entity containing id (PK), name (UNIQUE), description, and created_at timestamp

Employers Table: id (PK), name, email (UNIQUE), phone, company_name, department_id (FK), created_at

Workers Table: id (PK), name, email (UNIQUE), phone, employer_id (FK), department_id (FK), position, hire_date, created_at

Relationship Rules

- Departments contain multiple employers through foreign key relationships
- Employers supervise multiple workers with hierarchical assignment
- Referential integrity enforced throughout database operations
- Cascade rules protect data consistency

Core Features & Functionality

1

Department Management

Create, update, delete, and view departments forming the foundation of organisational structure. Departments serve as the primary organisational unit.

2

Employer Registration

Register employers with complete company details and assign to appropriate departments. Business rules prevent deletion if workers exist under supervision.

3

Worker Registration

Register workers under designated employers with position details and hire dates. Automatic department assignments maintain hierarchical consistency throughout the system.

4

Comprehensive View

Search and filter all worker records with complete hierarchical information displayed. Consolidated view enables efficient record management and reporting.

User Interface Components

Departments Tab

Comprehensive department CRUD operations with validation and real-time updates

Employers Tab

Employer management with input validation and department assignment capabilities

Workers Tab

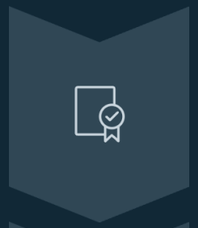
Worker registration interface with position tracking and hire date management

View Records Tab

Consolidated search and display with hierarchical information presentation

The interface features modern Nimbus Look and Feel, comprehensive input validation, real-time data refresh on tab switching, and a status bar displaying database connection status.

Data Flow & Operations



User Input Validation

UI layer performs comprehensive validation ensuring data quality before processing

DAO Method Invocation

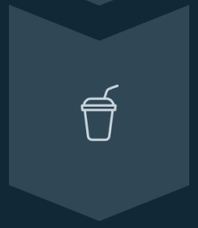
Business logic layer processes validated data through appropriate DAO methods

Database Connection

DatabaseConnection.getConnection() establishes secure JDBC connection for operations

SQL Execution

Prepared statements execute queries preventing SQL injection vulnerabilities



Result Processing

Results processed and UI updated with transaction management ensuring data integrity

Project Structure

```
employeeworkerreg/
├── src/com/employeemanagement/
│   ├── Main.java              # Application entry point
│   ├── model/                 # Entity classes
│   │   ├── Department.java
│   │   ├── Employer.java
│   │   └── Worker.java
│   ├── dao/                   # Data access layer
│   │   ├── DepartmentDAO.java
│   │   ├── EmployerDAO.java
│   │   └── WorkerDAO.java
│   └── ui/                    # UI components
│       ├── MainFrame.java
│       ├── DepartmentsTab.java
│       ├── EmployersTab.java
│       └── WorkersTab.java
├── lib/
│   └── mysql-connector-java-8.0.33.jar
├── sql/
│   └── schema.sql             # Database schema
├── compile.bat                # Build script
└── run.bat                    # Execution script
```

Architecture Benefits

This organised structure ensures maintainability and scalability whilst adhering to Java best practises.

- Clear separation of concerns across layers
- Modular components for easy maintenance
- Logical grouping of related functionality
- Simple build and deployment process

Key Highlights & Future Roadmap

Technical Achievements

Design Patterns Implementation

DAO pattern for data abstraction, MVC-like UI separation, factory pattern for database connections

Data Integrity Features

Unique constraints on emails, foreign key relationships, cascade operations, application-level validation

Performance Optimisation

Efficient per-operation connections, optimised JOIN queries, lazy loading of tab data

Future Enhancements

- Export records to CSV and PDF formats
- Advanced search with multiple filter criteria
- User authentication and role-based access control
- Comprehensive reporting and analytics dashboard
- Automated email notifications for key events
- Database backup and restore functionality