

Functional Requirements

Must Have

- **HR:**
 - Create an account to hire new employees
 - Create a draft of the new employee's contract
 - Finalise the contract once the employee has agreed
 - Terminate a contract
 - Accept employee's modifications to the contract
 - E.g. initial salary steps
- **Employee:**
 - Log in with credentials provided by HR
 - Ask to modify contract
 - Contracts should be set in stone and if the employee wants to modify their contract they should make a new one
 - Agree on the contract
 - View the contract
- Authenticate Users using NetID and password
- Make a distinction between users' role
- Salary steps & scale follow the guidelines set by government
- **Contract includes:**
 - Name of employer
 - Name & Address of candidate
 - Duration of contract (temporary, part-time, permanent)
 - Work Hours per week (min. 8 hours, max. 40 hours)
 - Number of vacation/holiday days
 - Pension scheme
 - Salary scale & steps
 - Additional benefits (e.g. travel allowance, insurance)

Should Have

- **HR:**
 - Approve employee requests
 - E.g. vacation days, sick leave, maternity/paternity leave
- **Employee**
 - Ask to terminate their contract
 - Make request for vacation days, sick and maternity leave (yearly vacation days: 15-30)
 - This action can also be done in person rather than through the application

- **Contract**
 - Temporary contracts can't be renewed more than once
 - Temporary contracts can't be renewed for more than 5 years
 - Contracts can start only on the 1st or the 15th of each month

Could Have

- **HR:**
 - Notify the employees with a close-to-terminate contract before the end date of the contract
 - This action can be done in person rather than through the application
 - Promote employees (possibly after recommendation from Dean)
 - E.g. from associate to full professor
- **Employee:**
 - Ask for documents and certificates (contract, salary slips, proof of enrollment)

Won't Have

- GUI
- Private messaging
 - The purpose of this application is to manage people rather than communicate between them

Non-functional Requirements

- Built with Spring Boot and Gradle
- Use microservice architecture
- Microservices must be scalable
- Application works solely with the terminal (does not include a GUI)
- Should be scalable
- Written in Java 11
- Design with modularity in mind (system should be extendable with extra functionality at a later stage)
- Support easy integration with 3rd party systems using an API