Nurse Rostering with Spring Boot User Guide

20-Apr-2019 | Version 1.0

Team Name: ai.Orz

Dai Yirui, Dong Meirong, Gu Lijian, Guo Feng, Wong Yoke Keong, Zhang Le

Prepared for NUS-ISS Master of Technology in Intelligent Systems ISY5001 Reasoning Systems Group Project

Table of Contents

1.0 System Overview

Resource constraints issues in Singapore hospitals have long been an challenge due to the increasing number of patients and shortage of nurses and having effective rosters for nurses are vital for timely care for patients and efficient operations of hospitals.

OptaPlanner by KIE provides an enterprise grade solution to this type of resource constraints problems. Considering future integration requirements with existing systems or services, the industry standard micro-services oriented architecture design has been adopted in this implementation of Nurse Rostering as a core planning engine utilizing the Spring Boot framework, where it can be run as an independent service, or as part of a larger system.

1.1 User Interface

The front-end system is built with simple HTML page and javascript, which calls the REST API from the backend.

2.0 Requirements

2.1 Recommended Browsers

Nurse Rostering demonstration supports the following web browsers:

- Firefox 63 and above
- Google Chrome Version 73 and above

2.2 Dependencies

Nurse Rostering for Spring Boot demo requires the following application dependencies:

- Oracle Java 8 JDK or JRE or compatible distributions
- IntelliJ 2019.1 and above (for launch via IDE)
- Apache Maven 3.3.9 and above (for launch via Maven)

3.0 Deployment

Our system is capable of running on both Windows or Linux Systems as .jar executable, through Maven or IntelliJ IDEA.

3.1 Prepare and run Nurse Rostering demo for Spring Boot through command line

1. Download source code from the below URL or via **git clone** command in your desired directory

https://github.com/guofeng201507/IRS-RS-2019-03-09-IS1PT-GRP-ai.Orz-NurseRostering

2. Change directory: **cd IRS-RS-2019-03-09-IS1PT-GRP-ai.Orz-NurseRostering**
3. Execute command: **mvn clean install**
4. Execute command: **mvn spring-boot:run** or **java -jar target/NurseRostering.jar**

3.2 Prepare and run Nurse Rostering demo for Spring Boot through IntelliJ IDEA

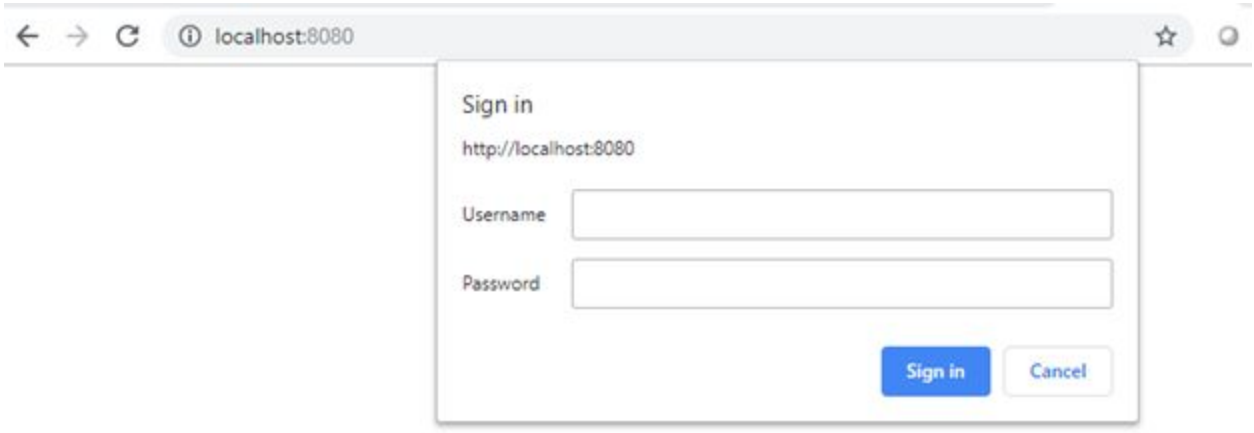
1. Download source code from the below URL or via **git clone** command in your desired directory
   https://github.com/guofeng201507/IRS-RS-2019-03-09-IS1PT-GRP-ai.Orz-NurseRostering
2. Import Maven project into IntelliJ IDEA
3. Execute the following Maven plugin commands through the IDE
   a. Clean
   b. Install
   c. spring-boot:run

4.0 Walkthrough

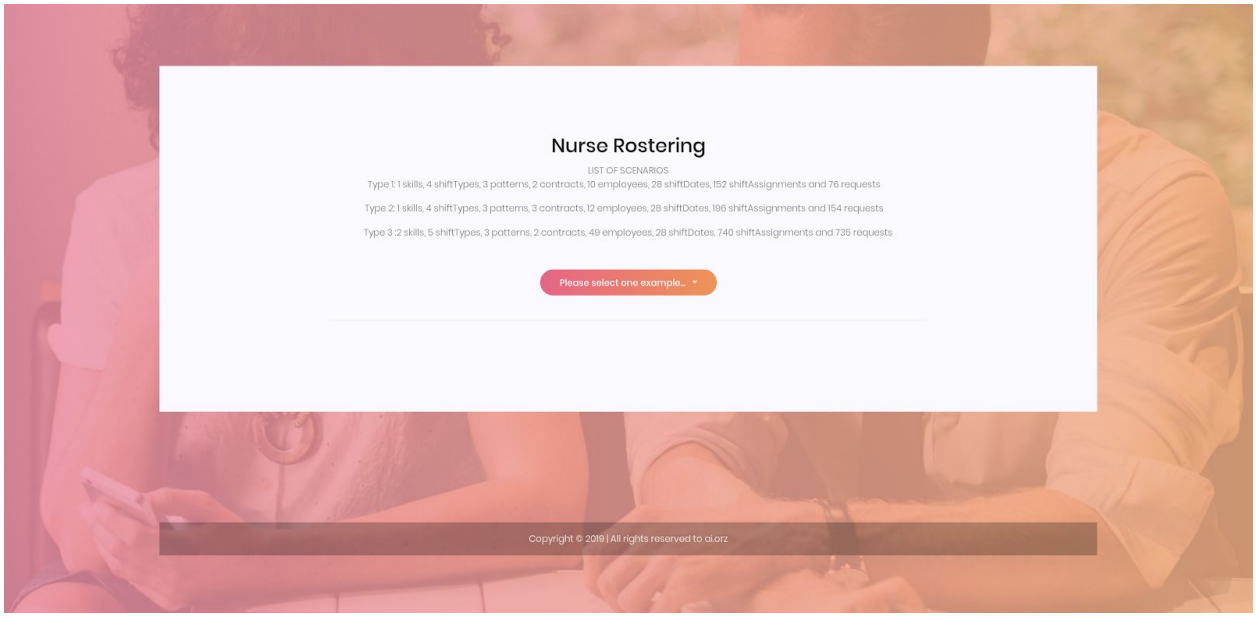
After the application is started. Open the URL in your browser: http://localhost:8080

Note: If you are using the iss-vm provided, please use **Mozilla Firefox** as your browser.

Enter the username/password as: **john/john@pwd1**



It will bring you the home screen, where use can choose the test scenarios, they are pre-configured according to the business scenarios.



Select any one of the scenarios and the application will start optimizing the roster.

Nurse Rostering

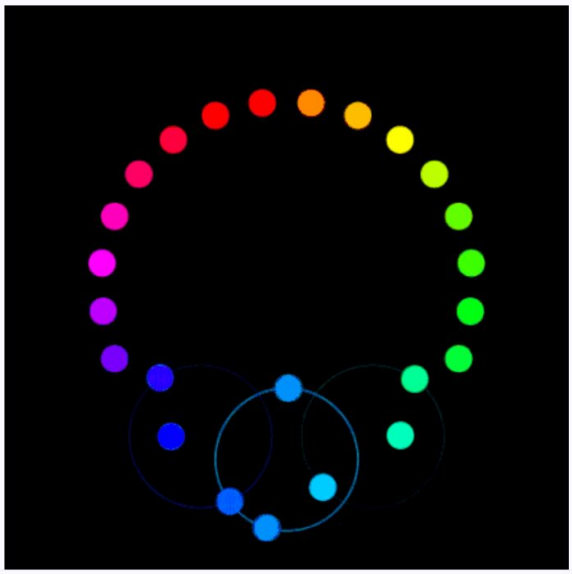
LIST OF SCENARIOS

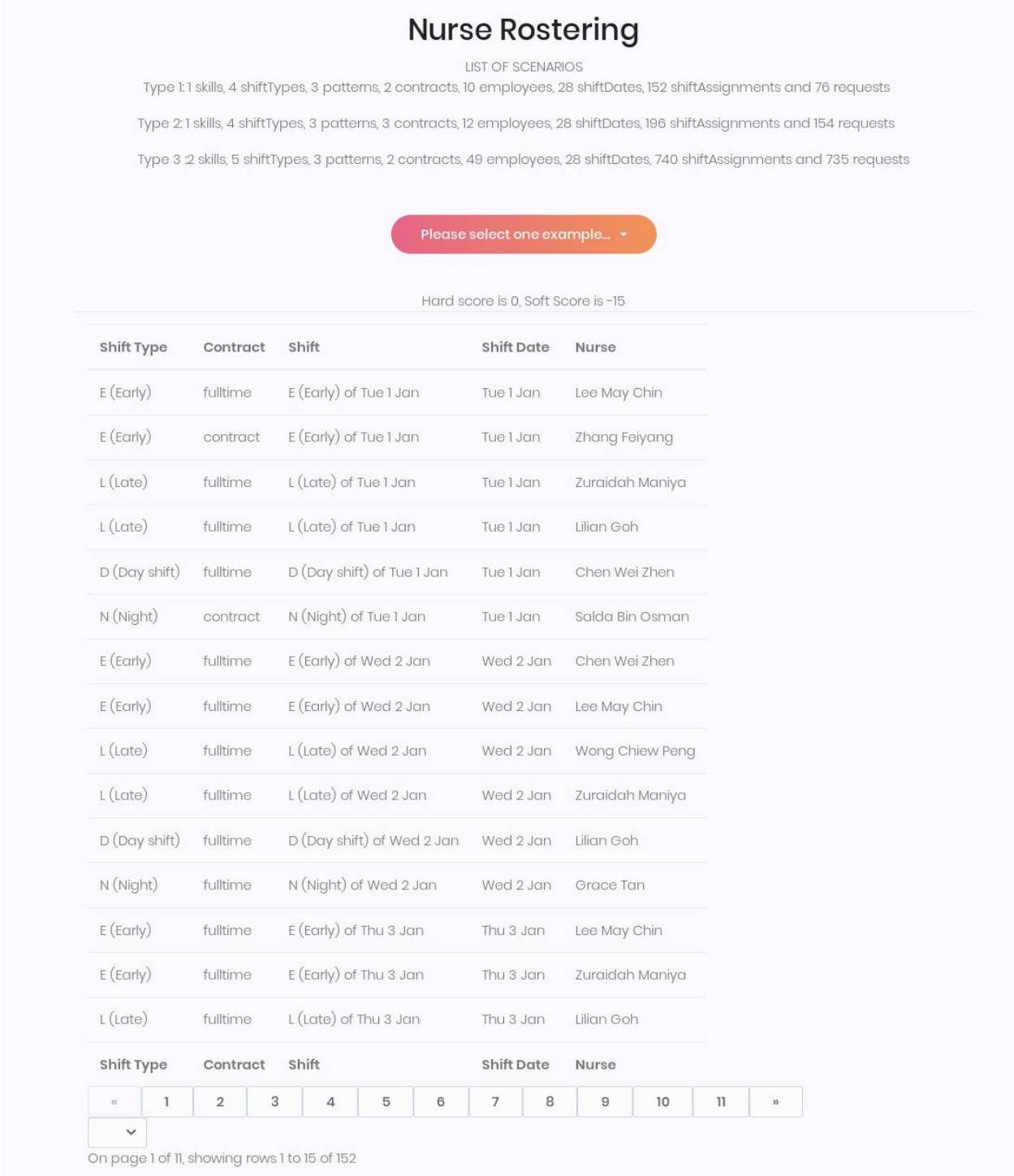
Type 1: 1 skills, 4 shiftTypes, 3 patterns, 2 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 76 requests

Type 2: 1 skills, 4 shiftTypes, 3 patterns, 3 contracts, 12 employees, 28 shiftDates, 196 shiftAssignments and 154 requests

Type 3 :2 skills, 5 shiftTypes, 3 patterns, 2 contracts, 49 employees, 28 shiftDates, 740 shiftAssignments and 735 requests

Please select one example...





After approximately 20 seconds, the optimized roster will be presented.

How to interpret the results

1. At the top of the table, you can find the the hard score and soft score, providing an indication of the degree of optimization possible given the set of constraints configured. The most optimized scenario will be when both hard scores and soft scores are 0

2. The table contains the roster, it shows on which day, which shift, who is assigned, and whether the employee is full time or contract.

5.0 Day 0 System Configuration

In our Day 0 system configuration, the following defaults have been set.

1. To reduce wait time, the team has opted to speed up the planning process by configuring optaplanner to automatically use the highest thread count possible.
    a. In nurseRosteringSolverConfig.xml
    `<moveThreadCount>AUTO</moveThreadCount>`
2. Termination condition has been set to 20 seconds, as we have empirically tested various scenarios and found out that 20s is **sufficient** for optaplanner to compute the optimal result.
    a. In nurseRosteringSolverConfig.xml
    `<secondsSpentLimit>20</secondsSpentLimit>`

6.0 Business Scenarios - Sample Input and System Output

6.1 Scenario 1

SCENARIO 1 Baseline planning load	1x Skill 4x Shift Types 3x Shift Patterns 4x Contracts 10x employees (nurses) 28x Shift Dates (4 weeks) 152x Shift Assignments 150x Leave Requests

6.2 Scenario 2

SCENARIO 2 Baseline with Light leave request load	1x Skill 4x Shift Types 3x Shift Patterns 2x Contracts 10x employees (nurses) 28x Shift Dates (4 weeks) 152x Shift Assignments 76x Leave Requests

6.3 Scenario 3

SCENARIO 3 Baseline with high shift assignment load	1x Skill 4x Shift Types 3x Shift Patterns 3x Contracts 12x employees (nurses) 28x Shift Dates (4 weeks) 196x Shift Assignments 154x Leave Requests

6.4 Scenario 4

SCENARIO 4 Large and complex ward load	2x Skills 5x Shift Types 3x Shift Patterns 2x Contracts 49x employees (nurses) 28x Shift Dates (4 weeks) 740x Shift Assignments 735x Leave Requests