MTECH PROJECT REPORT

INTELLIGENT EMPLOYEE SCHEDULER

REASONING SYSTEMS project report submitted in part fulfilment of the degree of **M-Tech Intelligent Systems**

TEAM MEMBERS

YONG QUAN ZHI, TOMMY A0195353Y
SANTHIYAPILLAI RAJEEVAN PAUL A0195399E
RITESH MUNJAL A0195304H
GARY NG JIAN ZHE A0195367N
ANI SIVARAM PORKALANGAD AYYAPPAN A0195404E
SOOREJ MOHANADAS GANGA A0195397J

April 21, 2019

Table of Contents

Abstract

Most organizations face problems in creating a well-planned roster to manage shift works of their employees, given there are limited set of constrained resources.

With a multitude of scheduler software available in the market today, Intelligent Employee Scheduler (IES) created by our group of NUS M-Tech students stands out of the crowd by leveraging on OptaPlanner to allow a manager assign shift work to his employees.

The optimal goal prioritizes the needs of employees for maximized satisfaction. IES can assign them to various shifts based on several hard and soft constraints such as availability of a worker subject to their leaves, proximity of a worker to work-site and assigning workers of relevant expertize appropriately.

Thanks to NUS ISS Lecturer Sam Gu, who has guided us on the working of OptaPlanner and how it could be used to tackle employee rostering problem, using Hybrid Reasoning System approach.

Project Specification

Objective: To create Intelligent Employee Scheduler for LTA to meet their business requirements as specified below.

Stakeholders: LTA manager in charge of maintenance operations, staff under the manager for whom shift works are assigned.

Task Force: NUS ISS MTech Part Time students – Tommy, Paul, Ritesh, Gary, Ani & Soorej

Requirement: Operations & maintenance activities for Train & Bus facilities in Singapore is mostly conducted during wee hours to minimise inconvenience to commuters. With 200+ of these facilities where maintenance is required, it becomes a challenging task for managers to assign their workers to the various locations under certain given constraints.

Scope: Current scope includes a team in LTA with 15 employees. These employees have to be assigned shift work across nation-wide train/bus stations. The works are carried out only during night, from Monday to Friday.

Constraints: Maintenance works are carried out only at most 10 stations per night out of the 206 stations based on issues/problems reported through various means. There are 3 different category of workers – Assistant Engineer (Grade 1), Engineer (Grade 2) & Senior Engineer (Grade 3). As seniority increases, their shift work assigned decreases. There can be up to 3 shifts assigned to the junior most personnel. CLOPEN (word coined from Close&Open) need to be managed well when assigning shifts – eg. A person working day shift should not be assigned the same night and vice-versa. However consecutive nights are acceptable. It is also good to assign workers to their nearby train stations for shift work for easy commuting to work.

Timeline: The works start on 7th Apr 2019 and gone through various stages such as: Requirement gathering – 3 days, developing system design & architecture – 3 days, code development - 5 days, testing -2 days and finally deploy on 21st April 2019.

Chapter 1: **Introduction**

With over 200 train & bus facilities in Singapore, the maintenance activity scheduling is a difficult task to LTA managers, which eats up to 20% of their time every week. If there is any error in the roster, it could lead to unattended maintenance call, which will not only affect revenue, but also have serious impact on Government reputation since these services are public facing.

Following are some common problems LTA face when it comes to employee scheduling.

1. Last minute scheduling: Having that predictability should help keep everyone happy. Employees know what to expect and once the rhythm has been established, they will be able to reach out to those who are free if they are unable to do a shift and things need to be juggled around.

2. Confusion among employees: Unpredictability of shifts and last-minute changes can lead to unsatisfied employees. If there is no consistency or people are able to swap and change the shifts with no system to track, it can cause havoc at the workplace. For consistency, it is important to hold employees accountable for their own shifts and to find a replacement if they are unable to work on a particular day.

3. Clopen: This nightmare scenario would keep any employee disgruntled - being chosen for the dreaded ‘clopen’ shift. It means that an employee has the last shift and then has the first shift next day. IES tackles this problem as easy as a breeze.

4. Unfair distribution: Overtime can become an issue and affect productivity. It’s an extra cost to the business and affects the employee morale, hence should be ideally avoided.

5. Unplanned cancellations: At times, an employee might take leave on urgent notice due to medical reasons or whatsoever. In such cases, overtime for another worker is the easy band-aid solution. IES, instead recalculates the entire shift assignment and helps avoid such issues.

Maintaining a functional and fair schedule that keeps every employee happy itself is a full time job on its own. Often, a staff manager will find himself buried under piles of spreadsheets, timetables, calendars and employee leave forms for this purpose. Managing shift schedule also require great communication and organisational skills.

IES is a one-stop solution to all the problems above mentioned. This software has been developed based on inputs from LTA, and it has been considered high business value in terms of the problems it solves.

Chapter 2: **Project Lifecycle**

The first phase was **PROJECT INITIATION** where all pre-project work was done. This involved our team defining the opportunity, discussions on the merits of project options, and getting the consensus to move forward.

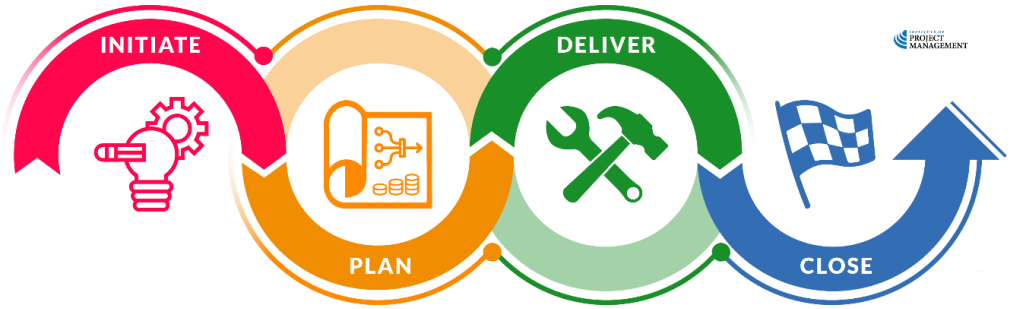


Figure 1: Project Life Cycle

Our team had identified different projects as listed in Table 1, from which we decided to go ahead with hybrid reasoning system project – **IES,** after careful consideration on the factors such as:

a. Implementation of Business rule-based reasoning techniques
b. Efficient usage of business resource optimization using OptaPlanner
c. Business value it brings to an organization by implementation
d. Availability of resources for developing front-end, back-end and integration
e. Complexity & Feasibility of implementation within the stipulated timeline

No.	Title	Description	Data Resources
1	Intelligent Employee Scheduler	Prepare roster for LTA employees on maintenance regime	Station names, Staff names & location, work days per week
2	Intelligent Lucky Draw Planner	Compose lucky draw prizes based on an inventory, budget and no. of prizes	Online retail sales datasets
3	Intelligent Course Scheduler	Course scheduler for NUS ISS	NUS ISS

Table 1: Project Initiation

In the **PROJECT PLANNING** phase, our team sat down for discussions on requirement analysis.

For the requirement analysis, the key tasks include:

1. Break down all the work that needs to be done into tasks
2. Estimate the time, resources and budget required to complete each task, and
3. Define acceptance criteria

Timeline	Deliverables	Owner
7th Apr	Requirement Specifications	Ani
10th Apr – 12th Apr	System Design & System Architecture	Paul, Tommy, Gary, Ritesh
13th Apr – 17th Apr	Frontend Code development	Tommy, Soorej
13th Apr – 17th Apr	Rules definition	Gary
13th Apr – 17th Apr	Optaplanner solver building	Paul, Ritesh, Gary
13th Apr – 17th Apr	Bug fixing and back-end code development	Tommy, Paul, Gary, Ritesh, Soorej
18th Apr	Integration of front-end & back-end	Paul, Tommy, Gary, Ritesh, Soorej
19th Apr – 20th Apr	Testing	Tommy, Gary, Ritesh, Paul, Ani
18th Apr – 20th Apr	Project report	Ani, Gary, Ritesh, Tommy
19th Apr	Project Video	Ani
21st Apr	Deployment (Github & Luminus)	Gary

Table 2: Requirement Analysis

With the completion of planning phase, our team entered the **PROJECT IMPLEMENTATION** phase, which will be covered in detail in the next sections.

Our group first set out to build a sizeable knowledge base by conducting interviews with LTA managers. While building the system, we utilized tools such as Java & OptaPlanner. The frontend was developed in Angular and NodeJS.

Chapter 3: **Knowledge Model**

Search methods are useful for business resource optimization where the aim is to find an optimal solution among feasible solutions, according to an objective / evaluation / score function.

For this project, Knowledge model is combination of 2 main aspects:

- Knowledge Acquisition
- Knowledge Representation

Knowledge Acquisition was pretty straight forward in that one of our team member Ani is working in LTA.

3.1 Knowledge Acquisition

For Intelligent Employee Scheduler, main method of knowledge acquisition was by elicitation of tacit knowledge through face-to-face interview with LTA manager. Through these interviews, our team was able to identify the problems and challenges LTA is facing today, the objectives they need to meet, as well as available resources and their constraints.

In Singapore, there are 141 MRT stations, 43 LRT stations & 30 bus interchanges where systems maintenance are carried out on a daily basis. Ad-hoc problems are reported by

1. Monitoring systems
2. Public transport operator staff (SMRT/SBST) who use these systems
3. Members of public

There are also planned maintenance cases such as :

1. Preventive maintenance carried out based on the system commissioned date. For eg, if the system was deployed in 2014 and the lifespan is 5 years, then a planned maintenance is carried out for Tech refresh of the system.
2. For IT systems, there are regular monthly / quarterly patching exercise covering IT security, feature enhancements etc.

The list of these public transport nodes, segregated by various lines are shown in table 3 & 4 below:

EWL	NSL	CCL	NEL	DTL
Pasir Ris	Bukit Gombak	Bras Basah	Boon Keng	Bugis
Tampines	Yew Tee	Esplanade	Buangkok	Promenade
Simei	Kranji	Promenade	Chinatown	Bayfront
Tanah Merah	Marsiling	Nicoll Highway	Clarke Quay	Downtown
Kembangan	Woodlands	Mountbatten	Farrer Park	Telok Ayer
Aljunied	Admiralty	Dakota	Harbour Front	Chinatown
Kallang	Future station at Canberra	MacPherson	Dhoby Ghaut	Bukit Panjang
Lavendar	Yio Chu Kang	Tai Seng	Hougang	Cashew
Bugis	Braddell	Bartley	Kovan	Hillview
Tanjong Pagar	Toa Payoh	Lorong Chuan	Little India	Hume Avenue
Tiong Bahru	Novena	Marymount	Potong Pasir	Beauty World
Redhill	Newton	Caldecott	Punggol	King Albert Park
Queenstown	Somerset	Bukit Brown (not in service)	Sengkang	Sixth Avenue
Commonwealth	Marina Bay	Botanic Gardens	Serangoon	Tan Kah Kee
Dover	Marina South Pier	Farrer Road	Woodleigh	Botanic Gardens
Chinese Garden	Ang Mo Kio	Holland Village		Stevens
Lakeside	Bishan	One-North		Newton
Boon Lay	Bukit Batok	Kent Ridge		Little India
Pioneer	Choa Chu Kang	Haw Par Villa		Rochor
Joo Koon	City Hall	Pasir Panjang		Fort Canning
Future Station at Gul Circle	Dhoby Ghaut	Labrador Park		Bencoolen
Future Station at Tuas	Jurong East	Telok Blangah		Jalan Besar
Future Station at Tuas West	Khatib	Bayfront (sharing DTL)		Bendemer
Future Station at Tuas Link	Orchard	Marina Bay		Geylang Bahru
Expo	Raffles Place	Buona Vista		Mattar
Changi Airport	Sembawang	Bishan		Macpherson
Bedok	Yishun	Harbour Front		Ubi
Buona Vista		Dhoby Ghaut		Kaki Bukit
City Hall		Paya Lebar		Bedok North
Eunos		Serangoon		Bedok Reservoir
Clementi		Stadium		Tampines West
Jurong East				Tampines
Outram Park				Tampines East
Paya Lebar				Upper Changi
Raffles Place				Expo

Table 3: MRT Stations

LRT	Bus Interchange
Choa Chu Kang	Boon Lay
South View	Woodlands Regional
Keat Hong	Tampines
Teck Whye	Toa Payoh
Phoenix	Ang Mo Kio
Bukit Panjang	Yishun
Petir	Choa Chu Kang
Pending	Pasir Ris
Bangkit	Bukit Batok
Fajar	SengKang
Segar	Bishan
Jelapang	Hougang Central
Senja	Serangoon Central
Ten Mile Junction	Yio Chu Kang
Sengkang	Bukit Merah
Compassvale	Eunos
Rumbia	HarbourFront
Bakau	Sembawang
Kangkar	Bedok
Ranggung	Clementi
Cheng Lim	Jurong East Temp
Farmway	Punggol Temp
Kupang	Bukit Panjang
Thanggam	Changi Airport PTB1
Fernvale	Changi Airport PTB2
Layar	Changi Airport PTB3
Tongkang	Changi Village
Renjong	Ghim Moh
Punggol	New Bridge Road
Cove	Kent Ridge
Meridian	
Coral Edge	
Riviera	
Kadaloor	
Oasis	
Damai	
Sam Kee	
Teck Lee	
Punggol Point	
Samudera	
Nibong	
Sumang	
Soo Teck	

Table 4: LRT Stations & Bus Interchagnes

In the department that we interviewed in LTA, there are the following team members, grouped according to their nature of work represented in grades.

Sl. No.	Name	Grade	Max shifts per week	Home Location
1	John	1	3	Lavender
2	Eric	1	3	Bugis
3	Jasmine	1	3	Bedok
4	Amelia	1	3	Toa Payoh
5	Tom	1	3	Tampines
6	David	1	3	Pasir Ris
7	Janet	1	3	Woodlands
8	Roji	2	2	Admiralty
9	Regalado	2	2	Bukit Panjang
10	Glenn	2	2	Boon Lay
11	Joseph	2	2	Sengkang
12	Mathew	2	2	Punggol
13	Arun	3	1	Sembawang
14	Tracy	3	1	Choa Chu kang
15	Andrew	3	1	Ang Mo Kio

Table 5: Staff Details

The constraints identified as follows:

A. Staff grade based
1. Grade 1 officer can be assigned max 3 shifts a week
2. Grade 2 officer can be assigned max 2 shifts a week
3. Grade 3 officer can be assigned max 1 shift a week

B. Time of day based
1. Always assign consecutive night shifts for a particular staff.
2. If there is a day duty, the staff should not be assigned same date night shift. Instead assign one night after or 2 nights before the day shift.
3. If there is no planned night maintenance works in a week, then can assign consecutive day shifts for a staff.
4. Priority of day shift decreases according to grade. Means grade 3 officer has higher preference for day shift.

C. Staff absence based
1. System should be capable of adjusting itself at minimal disruption when there are unexpected absence of staff due to sick leave

D. Staff location based
   1. Assign site to a staff based on their home location

E. Station based
   1. There must be provision to add/remove stations on 24 hour notice, with minimal disruption to planned schedule. This is to cater for ad-hoc urgent maintenance requests.

Ideally, IES is supposed to create weekly schedule as shown below, when we input the stations where maintenance are to be carried out in that week and the list of available employees.

Week yy	Monday		Tuesday		Wednesday		Thursday		Friday	
	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night
Station 1	<staff>	<staff>	<staff>	<staff>	<staff>	<staff>	<staff>	<staff>	<staff>	
Station 2		<staff>	<staff>	<staff>			<staff>			<staff>
Station 3	<staff>					<staff>	<staff>	<staff>	<staff>	
Station 4	<staff>	<staff>	<staff>	<staff>	<staff>					<staff>
Station 5		<staff>			<staff>	<staff>	<staff>	<staff>	<staff>	
Station 6	<staff>		<staff>	<staff>		<staff>			<staff>	
Station 7		<staff>	<staff>	<staff>	<staff>	<staff>	<staff>	<staff>	<staff>	<staff>
Station 8	<staff>	<staff>	<staff>				<staff>	<staff>	<staff>	<staff>
Station 9				<staff>		<staff>				
Station 10	<staff>	<staff>					<staff>	<staff>	<staff>	<staff>

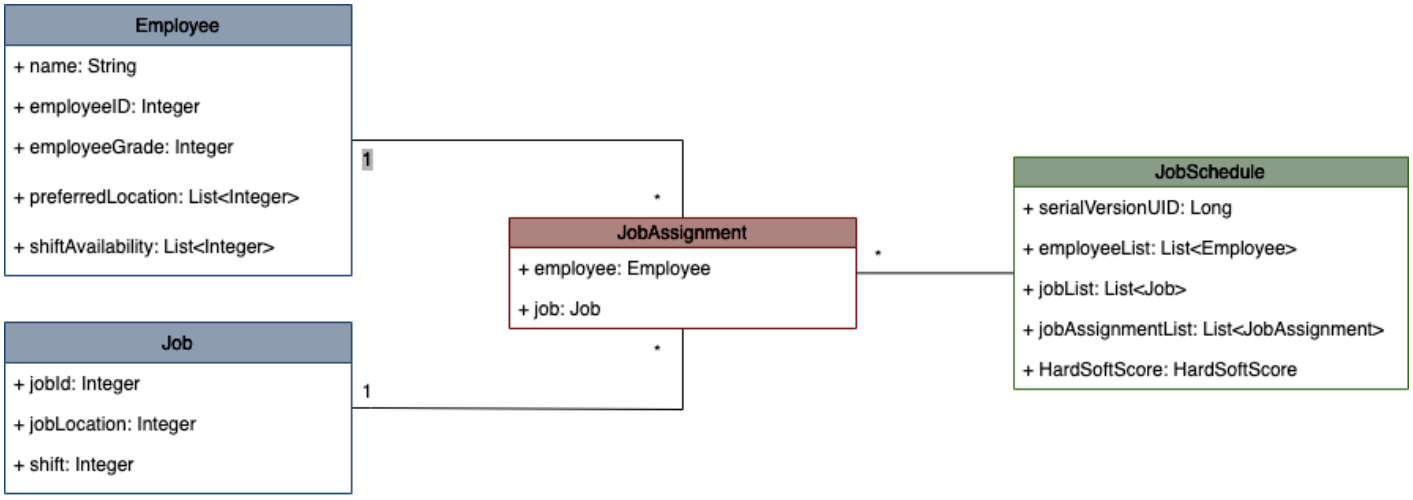
Table 6 : Employee Shift Sample

3.2 Knowledge Representation

Acquired knowledge was further refined to identify the crux of requirements and represented in the knowledge model. This process is very important as it clearly highlights the goals of the system and can be used directly to start implementation of knowledge-based system.

There are four main classes that define the domain:

- Employee
- Job
- JobAssignment &
- JobSchedule



Employee and **Job** define a maintenance worker and a maintenance work to be assigned respectively. They are the basic classes that encapsulate this real-world problem's core entities: the worker, and the work that needs to be done. Hence, they contain attributes that reflect real-world properties that are pertinent to the problem space:

Employee

- Employee ID: to uniquely identify each worker
- Name: for user-friendly display purposes
- Shift Availability: to indicate the availability of each worker to be assigned for each day for that week (useful when there are planned leaves for an employee)
- Preferred Location: to indicate which job locations are most convenient for each worker (usually employees prefer location near their home to avoid excessive cab charges after night shifts)
- Employee Grade: to indicate the number of times the worker can work per week (eg. Grade3 officer works only one night per week, whereas Grade1 officer works upto 3 nights per week)

Job

- Job ID: to uniquely identify each job
- Job Location: *where* this job will be (could be any one out of the 214 from table 3&4 above)
- Shift: *when* this job will be (to denote whether it's a day or night shift)

JobAssignment and **JobSchedule** are classes that define the relationship between the Employee and Job classes.

JobAssignment

- is one Job assigned to one Employee; the work has been designated to a worker.

JobSchedule

- is the solution class: it represents all the assigned Jobs and so if one were to peek into the JobSchedule, one would find a list of JobAssignments (among other ancillary things), each with its own unique Job and designated Employee.

3.3 Business Rules & Score Function

Hard and soft scores are used to quantify hard and soft constraints respectively. The constraints are determined by the business rules as follows:

Business Rule	Description	Hard Score	Soft Score
No unassigned job	Every job must have a designated worker. A schedule is not complete unless all jobs are assigned to a worker.	-1	-
No unavailable employee assigned	Jobs cannot be assigned to workers that are not available at the particular day and time of day	-1	-
Number of JobAssignment follow grade	Each worker has a grade (1 to 3). It determines how many jobs they can take per week (3 to 1, respectively). Workers cannot work more than what their grade allows.	-1	-
No consecutive shifts/same shift	Workers cannot work two jobs of the same shift (same day, same time of day). Workers also cannot work two consecutive shifts (e.g. Monday PM, then Tuesday AM).	-1	-
Employee given preferred location	Workers can be assigned job locations that are more preferred (due to convenience).	-	+1

The rules are determined by the actual business context.

Each job must be assigned one worker. A job cannot be unfilled as it means work does not get done. That is a failure in management.

A worker cannot be assigned a job that occurs on a shift (day and time of day) that he/she is not available, as predetermined before planning initialized (e.g. applied for leave).

Worker grades are also determined by professional status and number of jobs assigned must adhere to this rule.

Workers cannot feasibly work more than one of the same shift as it would mean either being at two places at once or doing the work of multiple people.

If a worker is assigned a job at a location that is more convenient, efficiency would be better.

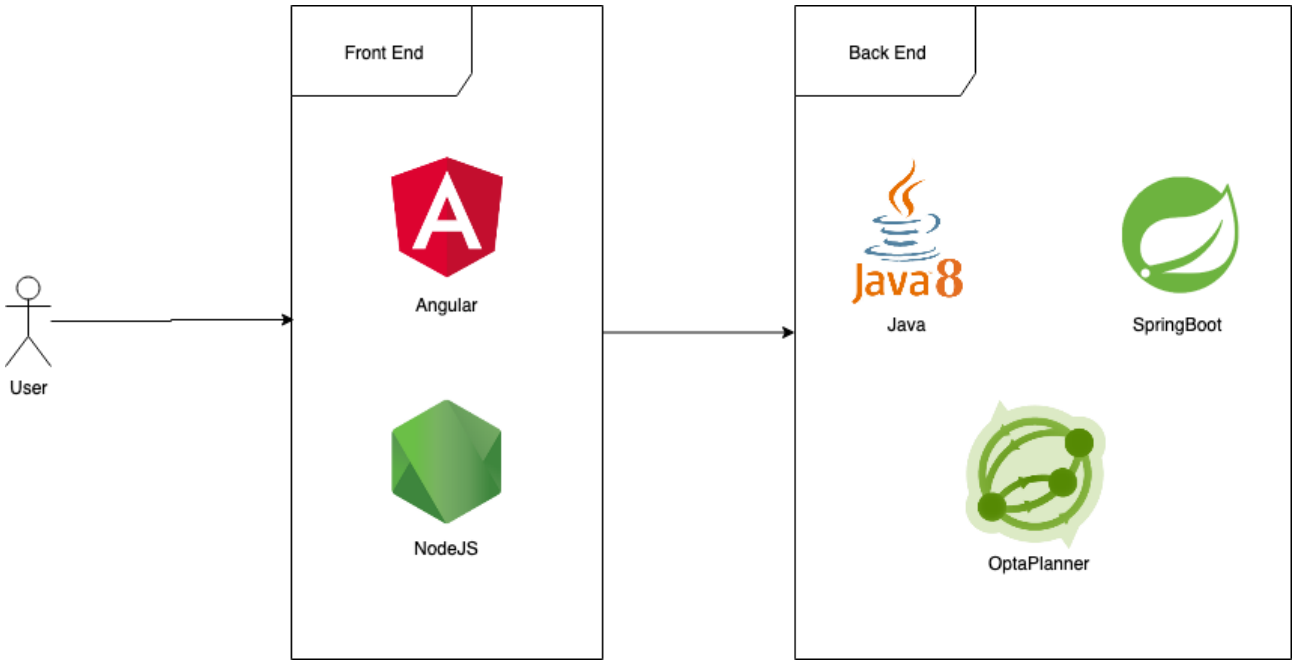
The scores are accumulated by iterating through each job assignment for each candidate job schedule. Schedules whose Hard Score is less than 0 are not valid solutions. Schedules with higher Soft Scores are preferred.

Chapter 4: System Implementation

We have used Springboot framework with OptaPlanner as backend containing the logic and Angular as the front end. The use of MVC Framework integrated with OptaPlanner was not an easy task but it is worth the effort considering the ease of use of the tool for anyone. The system has been designed keeping in mind the main aspects of **user-friendliness** and providing **best combinations of employee-shifts** based on user preferences.

4.1 System Architecture

After careful considerations, the system architecture was derived and implemented as shown in the figure below:



The application has been designed on the **Springboot** framework which is a light MVC framework and provides an easy to use web based front end integrated with OptaPlanner as Constraint Solver in the backend. It is a modern server-side Java template engine for both web and standalone environments.

The front end is built using the Angular single page application framework and Node.js for handling communications with the back end. The combination of the two provide a seamless user experience and an uninterrupted integration with the backend.

The backend is built in Java 8, following the Springboot framework. The domain and logic are all standard Java classes, and OptaPlanner integrates seamlessly with it as well.

OptaPlanner provides the constraint solving capability. The problem space is defined in Java, then fed to OptaPlanner, where each candidate solution is scored and compared with the rest, until a satisfactory solution is obtained.

4.2 Constrain Solver Design Process

The steps followed for designing the constraint solver application are as below:

1. Finalized the class diagram
2. Identified the planning variables
3. Created the POJO Objects for facts and Solution
4. Defined HardSoftScore
5. Build the Solver
6. Input the Employee and Jobs from Front End
7. Get the Output on Front End

4.3 System Capabilities

The planning solution developed as Reasoning System Project is very well integrated web solution for shift roster planning problem. The planner considers all the various constraints as required for the planning the shift roster of the employees weekly. It is a great solution as it can very well take care of the following constraints:

1. Employee Unavailability
2. Equal loading on all the employees
3. Covering all the job locations
4. Shift Assignment based on Grades of Employees
5. Preferred Shift Location
6. Preferred Shift (Morning/Night)

4.4 Areas for Improvement

We would like to do few more things to provide final project as solution to LTA but due to time constraints we were not able to implement them as of now:

1. Map the locations with real postal codes

2. Provide the @PlanningPin for user to immovable the shifts which does not need to be changed in case of re-planning.

Chapter 5: **User Guide**

Here we will demonstrate how to use IES. This also documents the pre-requisites needed to run the system locally on your machine.

5.1 Software/Tools Pre-requisites

5.1.1 Web Browsers

Either of the following browser versions will be needed to display the web application.

- Firefox 38
- IE 11
- Google Chrome 54

5.1.2 Tools/Kits

In order to run the application locally on any machine you will need the following tools/kits.

- Apache Maven
- Java JDK 1.8
- NPM package manager
- Angular 6
- NodeJS

Apache Maven download link: https://maven.apache.org/download.cgi

Java JDK 1.8 download link: https://bit.ly/2NsJMMs

NPM install guide: https://www.npmjs.com/get-npm

Angular 6 guide: https://angular.io/guide/quickstart

Please follow the installation guide below to install maven and to ensure that JDK 1.8 is set to JAVA_HOME variable in your system.

https://maven.apache.org/install.html

5.2 Deploying the application locally

5.2.1 Backend

Open up your command prompt/terminal and go into the 'Services' project folder. Please ensure section 1.1.2 has been fulfilled before proceeding or you will end with build failures.

Enter the following in your command prompt/terminal:

mvn clean install

mvn spring-boot:run

The projects will take a long time to build for the first time as it will be downloading dependencies to your machine.

If you encounter this error, run the launch command one more time.

Once the app has finished building and deployed on *localhost:8082*. You can start to build and deploy the front end version.

5.2.2 Frontend

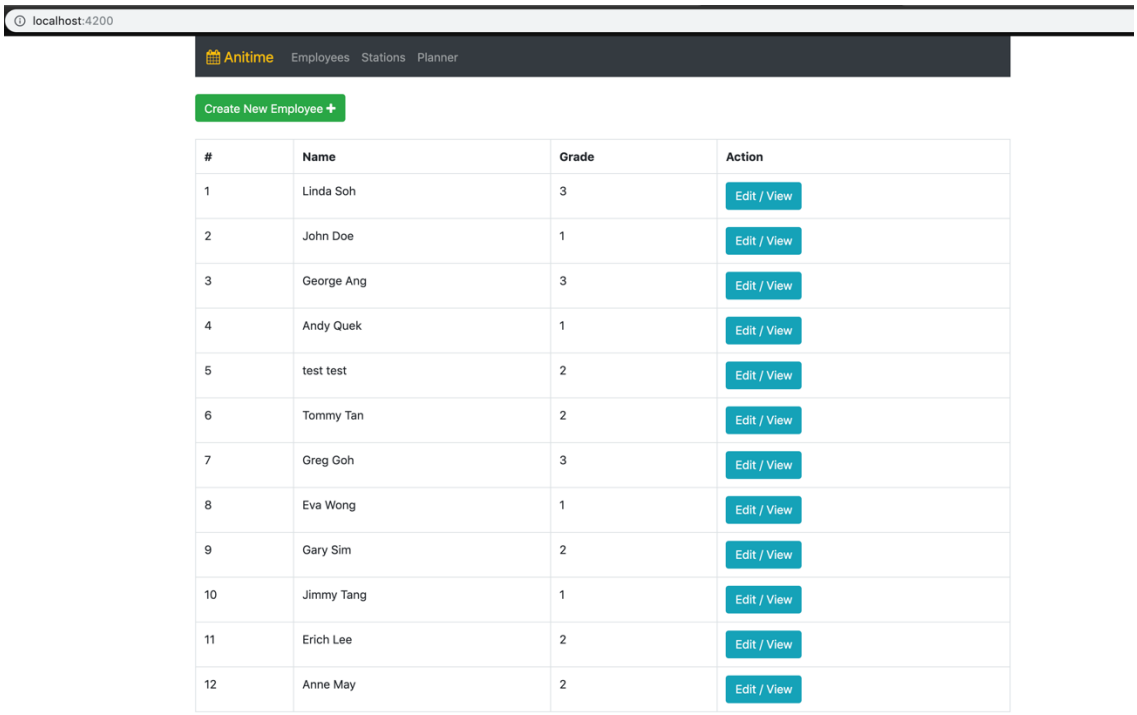
Open up your command prompt/terminal and go into the 'Frontend' project folder. Please ensure section 1.1.2 has been fulfilled before proceeding or you will end with build failures.

Enter the following in your command prompt/terminal:

npm install

ng serve

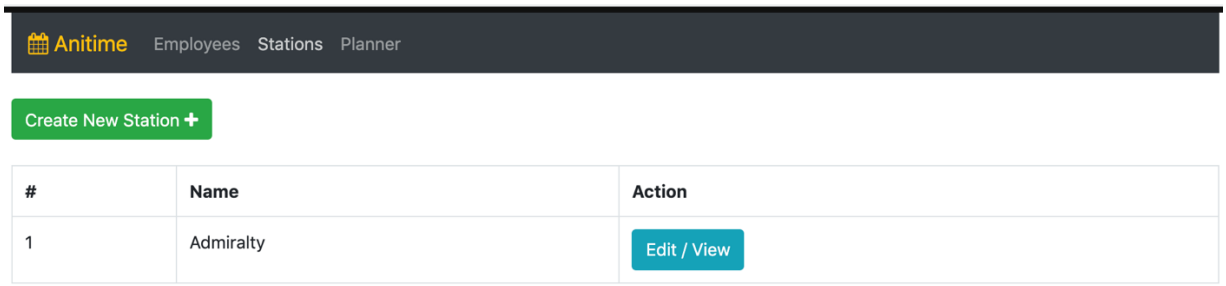
When the frontend has finished building and deployed successfully, you can navigate to localhost:4200 to view the frontend.



5.3 Using the system

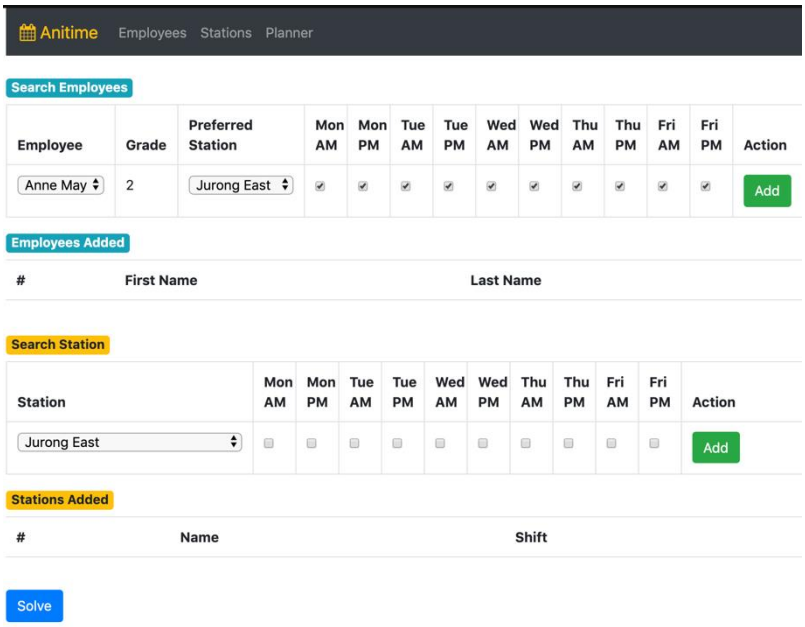
Upon accessing localhost:4200 you can choose to navigate between Employees, Stations, and Planner.

Upon starting, you may want to add a few stations into the system before starting.

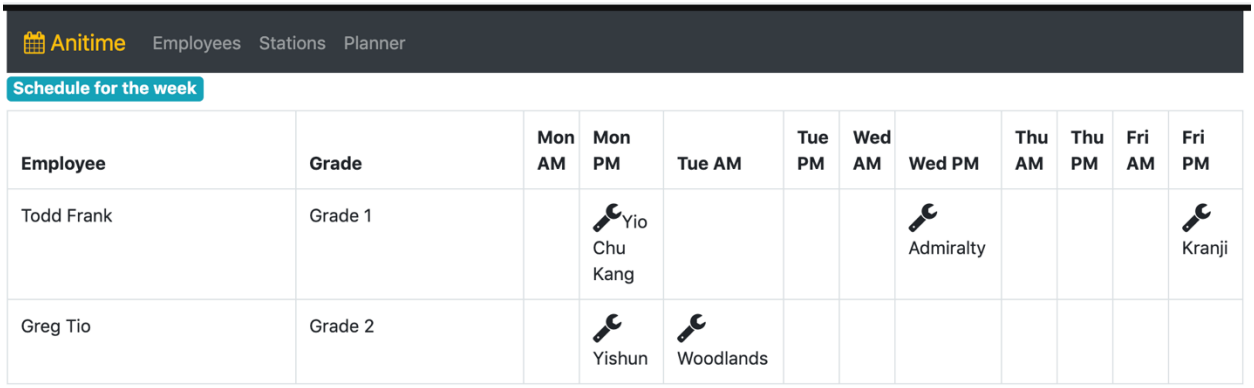


To begin using the planner, click on the Planner tab and begin adding in stations and employees with the corresponding shifts.

1) Select the desired employee and uncheck the shifts if the employee is unavailable.
2) Once done with the employees, select the desired stations and check the desired shifts that maintenance work is scheduled for.
3) Once done, click Solve.



Your results will be displayed.



Chapter 6: **Conclusion**

The project is a very good example of Intelligent Reasoning System as it incorporates all the important aspects of the Reasoning System.

We have also demonstrated how relatively simplistic search algorithms can be used to tackle highly combinatorically complex problems, such as employee rostering. As long as the problem and solution domain are modelled carefully, taking into account the nuances of real-life scenarios, search algorithms can take advantage of the speed of modern computers to quickly provide the best solutions.

While statistically-based machine learning may be in vogue now, we believe that humble search algorithms and optimization is still relevant in solving modern problems.

Bibliography

1. OptaPlanner User Guide - https://docs.optaplanner.org/7.20.0.Final/optaplanner-docs/html_single/index.html

2. A Guide to OptaPlanner - https://www.baeldung.com/opta-planner