Job Recommender System Project Report

Group 7 Member

1. Daniel Tan Hoong Xiang (A0074608B)
2. Leonard Loh Kin Yung (A0213553M)
3. Aaron Kueh Hee Kheng (A0213552N)

Table of Contents

1.1 Executive Summary

With the increasing usage of the internet, recruiters and job seekers alike are turning more towards job search websites such as LinkedIn, JobStreet or MyCareersFuture for job hunting. Users will be provided a list of possible job options based on their search criteria. From there, users must sort through the list and analyze each job listing by themselves to see if they are able to meet the requirements.

This process is very tedious and inefficient due to the amount of workload required from the applicants. In addition, a significant number of job listings included from the search result tend to be completely incompatible with the job seeker, causing them to spend hours finding ones that are suitable. Thus, a new system to automate most of the manual process would be appealing to job seekers. The system would be able to reduce the overall amount of workload required and inform them of which jobs are most suitable based on certain criteria.

1.2 Market Research

According to statistical data on LinkedIn, one of the most popular job search websites in the world, an average of 100 million job applications are submitted on a monthly basis[1]. This shows that a significant number of users rely on online job search websites in their job hunt, which would open the opportunity to tap upon this group of users by providing an alternative method to find suitable jobs efficiently.

As of now, the job market is not in a good shape, the unemployment rate has gradually climbed up since April 2018 even before the COVID-19 crisis. The unemployment rate during April 2018 is 2.1% and now it is 2.4% [2], this figure is expected to even increase further due to the forecast stating that Singapore will enter recession in this year 2020. According to the Monetary Authority of Singapore (MAS), Singapore will enter into recession this year 2020, the first quarter of the Singapore economy has contracted by 2.2%, forecast has also shown the full year will have a contraction from -4 to -1% in range[3]. This would not only result in an influx of job seekers, but also fiercer competition as less jobs are expected to be available for the growing pool of job seekers. These further stresses the need to provide a way for job seekers to find the most suitable jobs in order to boost their chances of getting the job.

2. Project Scope

2.1 Problem Background

Tedious Job Search Process

While the use of employment-oriented online services has allowed applicants to quickly obtain a list of jobs for them to consider, it is usually very long and contains numerous jobs that would be incompatible for them. This is most likely due to the limited and simple search filters made available to them. As such, job seekers often waste a large amount of time sorting through each job to determine if it is viable for them to apply for. The tediousness and inefficiency of the job search process thus reduces the amount of applications a job seeker can send out per day.

Unsure of Expected salary

In some cases, the job listing requires the candidate to provide an expected salary in their application documents. This is difficult for the candidate as they are unsure on how to weigh the value of their current skills against the job requirements, especially for those that are fresh graduates or planning to shift their career focus into a new area. If the candidate's expected salary placed is too high, employers might not consider them for the position. Conversely, if the candidate's stated expected salary is lower compared to the standard salary in the job market, they might be short selling themselves by accepting a lower pay compared to their colleagues in similar positions.

Job Suitability

For fresh graduates or recently retrenched individuals, there would be an urgency to quickly get a job as soon as possible. Driven by the sense of urgency, they might apply for job positions even though they do not possess the skills, experience or seniority as required. If these factors are not carefully considered, the applicant could end up accepting a position in which they are unqualified for. This would place the applicant in danger of not being able to clear their probation period, resulting in early termination. Thus, it is important to provide a method for applicants to make an informed decision of whether the job is suitable for them.

Career Prospects

In light of the digital age and Industry 4.0, it is not uncommon to see cases of working individuals shifting their career prospects to IT-related fields. However, these individuals might not be equipped with the necessary skills that are commonly required. In order to increase their chances of shifting successfully, it would be necessary for them to take the initiative and learn the skills required.

2.2 Project Objective

The Smart Job Recommender system aims to resolve issues faced by job seekers outlined in Section 2.1 through process automation and knowledge-based reasoning. Firstly, the system takes in information provided by the applicant such as their educational qualifications, years of work experience and technical skills. Based on the predefined rules, the system will filter out incompatible jobs and determine the level of suitability of the remaining jobs. The result is a list of recommended jobs provided to the applicant. The advantage of the Smart Job Recommender System compared to current job search websites is summarized below in Table 1.

Current Job Search Websites	Our System
Unable to efficiently filter out irrelevant jobs	System can filter out irrelevant jobs efficiently
Job seekers must examine job requirements and decide if they are suitable	System will score suitability of the job by comparing job requirements and job seeker's qualifications
Not Applicable	System provides job seekers an expected salary based on their qualifications, skills, years of experience & seniority
Not Applicable	System will recommend you what skills to learn and what academic qualification required base on job seeker's applications

Table 1. Advantage of Smart Job Recommender System over current job search websites

3. Project Solution

The solution for the system is a knowledge-based reasoning model. Knowledge modelling consists of the following parts:

1. Knowledge Acquisition
2. Knowledge Specification

3.1. Knowledge Acquisition

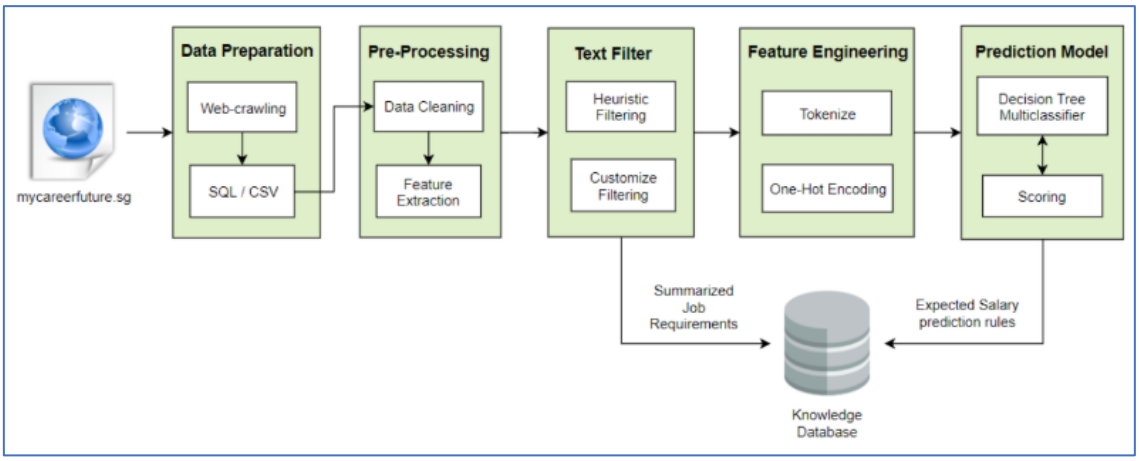


Fig 1. High level architecture diagram for data mining process

3.1.1 Data Preparation

Web-crawling with Beautiful Soup

- Beautiful Soup is a Python package for parsing HTML and XML documents. It is used to extract data from mycareerfuture.sg

Automate with Selenium Web driver

- It is an open source browser automation framework that accepts commands and controls the browser by directly communicating with it.
- It is used to perform web-crawling throughout thousands of job posting webpages automatically.

Data Ingestion

- The crawled job posting information are ingested into data table and transformed into csv file for KIE to consume.

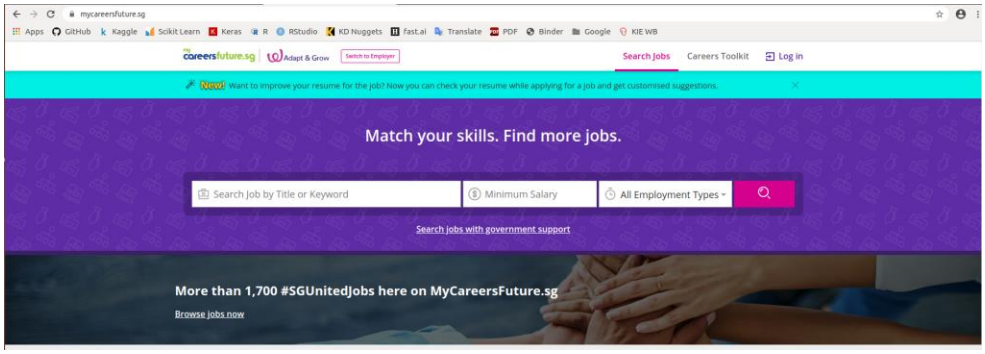


Fig 2. Job crawling website for this project: mycareersfuture.sg

Web Crawling Algorithm

- Web-crawling algorithm will crawl all the page(s) related to the search keyword

```
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=1
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=2
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=3
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=4
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=5
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=6
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=7
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=8
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=9
https://www.mycareersfuture.sg/search?search=Machine%20Learning&sortBy=new_posting_date&page=10
```

- Algorithm will access each page and look for individual job link as seen in Fig 3.

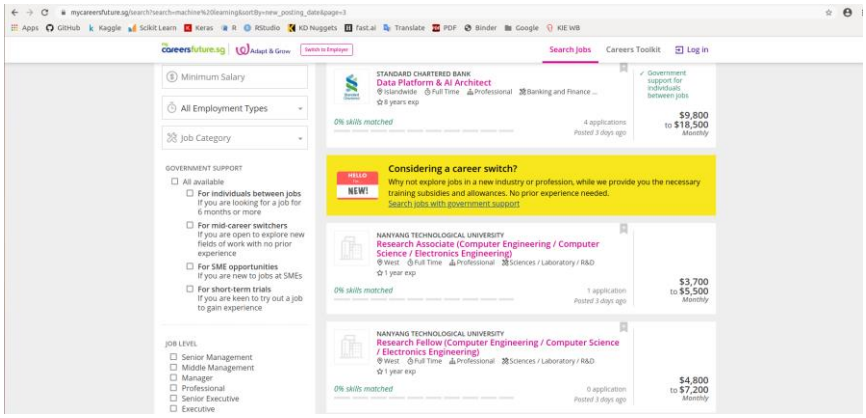


Fig 3. Example of mycareersfuture.sg job webpage

- Individual job link will be saved into a data

```
['/job/risk-technology-manager-nexus-standard-chartered-bank-e16cdc913c55d782eeb657880c1c640e',
 '/job/sgunitedjobs-software-developer-cybersecurity-defence-science-technology-agency-3edf76eadb4acf0d2eb48edac867e32d',
 '/job/information-technology/software-engineer-cashshield-d07c114a494095fb88201089f393b205',
 '/job/information-technology/senior-software-engineer-cashshield-c2b67ef948d0395d98d164bbffb5cc31',
 '/job/design/machine-learning-hardware-architect-nityo-infotech-services-336154f4802b718b9b7da7002bf225db',
 '/job/engineering/machine-learning-engineer-asus-global-3d8a0dd82e7937359c750089adad4e74',
 '/job/engineering/ai-engineer-continental-automotive-singapore-c79fafba50bcfb6432d92f2ed2df3fc7',
 '/job/insurance/assistant-manager-data-innovation-axa-insurance-86d10ff165ddd2c20ecfcae4790fc87d',
 '/job/senior-human-resource-officer-inland-revenue-authority-singapore-d45d462a08da9f495b796351358a0e01',
 '/job/information-technology/data-engineer-manpower-staffing-services-b9a8de8bfc329ca4cb18f1d9979b0604']
```

- Define type of information to be crawled

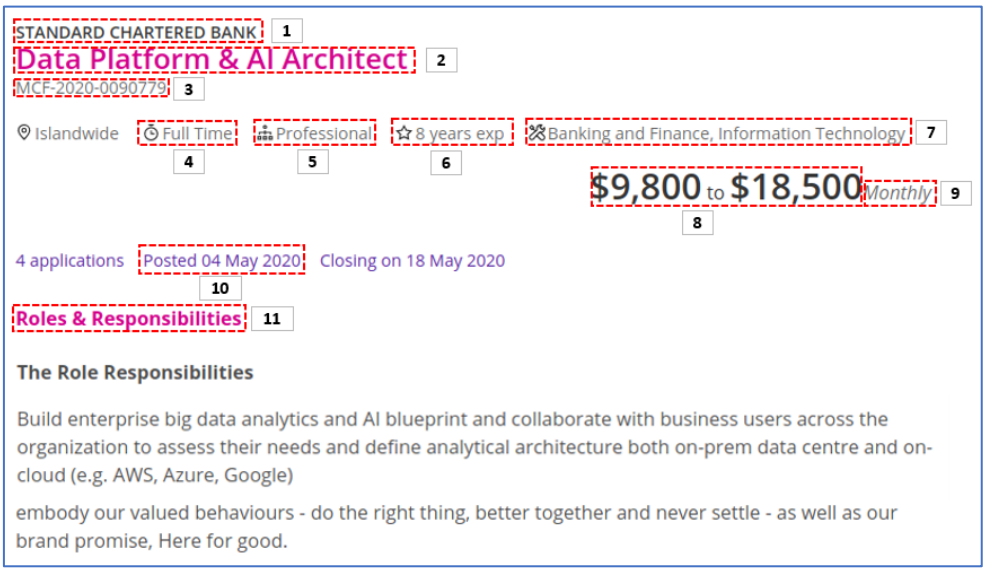


Fig 4. Example of Job detail from mycareersfuture.sg

- Job information crawled by the algorithm will be categorized into 11 schemas:

1. Company name
2. Job title
3. Job application number
4. Employment type
5. Seniority
6. Year of experience
7. Industry/Category
8. Salary range
9. Salary type
10. Posted date
11. Roles & Responsibilities (Requirements)

- Job information crawled by the algorithm will be stored into data table for data mining process.

Job_Id	Emp_Type	Job_Title	Company	Date_Posted	Salary_Range	Salary_Type	Year_Experience	Seniority	Category	Requirements
[b'MCF-2020-0091748']	[b'Contract, Full Time']	[b'Research Assistant (Computer Science / Comp...	[b'NANYANG TECHNOLOGICAL UNIVERSITY']	[b'Posted 06 May 2020']	[b' $3,...	[b'Monthly']	[b'1 year exp']	[b'Professional']	[b'Sciences / Laboratory / R&D']	[\nSoftware development to support research pr...
[b'MCF-2020-0092366']	[b'Full Time']	[b'Senior Data Scientist']	[b'TALENTVIS SINGAPORE PTE. LTD.']	[b'Posted 06 May 2020']	[b' $5,...	[b'Monthly']	[b'5 years exp']	[b'Senior Executive']	[b'Sciences / Laboratory / R&D']	[\nDevelop, refine and implement state-of-the-...
[b'MCF-2020-0091948']	[b'Permanent, Full Time']	[b'Software Engineer Lead, Banking Technology ...	[b'JPMORGAN CHASE BANK, N.A.']	[b'Posted 06 May 2020']	[b' $10...	[b'Monthly']	[b'10 years exp']	[b'Senior Executive']	[b'Information Technology']	[\nHands-on leadership of a local development ...
[b'MCF-2020-0091704']	[b'Contract, Full Time']	[b'Senior Engineer (IT) / Engineer (IT)']	[b'NATIONAL UNIVERSITY OF SINGAPORE']	[b'Posted 06 May 2020']	[b' $3,...	[b'Monthly']	[b'2 years exp']	[b'Professional']	[b'Engineering']	[\nDesign, develop and maintain software (mari...

3.1.2 Pre-Processing

Data Cleaning

- Remove non-ascii character from the data such as the HTML code prefix and suffix.
- Remove duplicated job posting based on Job ID which is unique.
- Remove missing data such as job without year of experience or job requirements.
- Remove outlier based on salary range. Example job with extremely low or high salary.

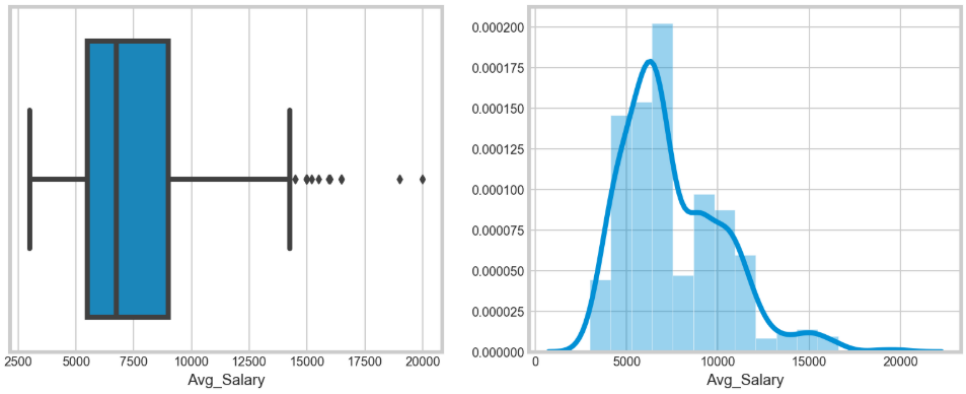


Fig 5. Example of average salary distribution from the dataset

Feature Extraction

- Categorize Seniority:
  [Fresh/entry level, Junior Executive, Executive, Senior Executive, Professional, Manager, Middle Management and Senior Management]

- Categorize Job Category:
  [R&D, Laboratory, Sciences, Information Technology, Engineering, Finance, Banking, Consulting, Civil Service, Public, Telecommunications, Others, Training, Education, Manufacturing, Pharmaceutical, Healthcare, Insurance, General Management, Supply Chain, Risk Management and Logistics]

- Categorize Employment type:
  [Full time, Part time, Contract, Permanent and Internship]

- Extract requirement base on domain knowledge:
  [Academic qualification and skills]

3.1.3 Text Filtering

Objective

- The purpose of this function is to reduce the number of words under requirements feature.

Heuristic Filter

- Stop words with additional of pre-defined words to remove common words.
- Use CountVectorizer to remove word which are unique instead of common words

```
cvt = CountVectorizer(lowercase=True, strip_accents='unicode',max_features=10000, min_df=1, max_df=0.01,
                      stop_words=stop, ngram_range=(1,1))
vect_word = cvt.fit_transform(df['Requirements'])
features = np.array(cvt.get_feature_names())

key_word = freq_words(vect_word, features)
print(key_word)

#update stop_word with unique words
new_stop = key_word[key_word<2].index
print(new_stop)
stop.extend(new_stop)
```

- Example of unique words extraction using countvectorizer.

```
Index(['abbyy', 'tracks', 'dives', 'dl', 'yield', 'dl4j', 'dm', 'dmp', 'dnn',
       'doc', 'dockers', 'yii', 'doctorate', 'distribute', 'trainer',
       'tracing', 'yarn', 'disclosure', 'discount', 'acceleration', 'discrete',
       'discretion', 'discriminant', 'discriminate', 'discuss', 'transferring',
       'yet', 'academics', 'tranformation', 'displays', 'disposal', 'disprove',
       'academically', 'disrupt', 'trajectories', '7years', 'dollar',
       'workers', 'zealand', 'eai', 'early', '2nd', '25', 'earth', 'ease',
       '24x7', 'yrs', 'zemax', 'dynochem', '2013', 'easymiles', 'toprogramme',
       '18', '16', 'eclipseexperience', '12', 'eco', '2years', 'dynatrace',
       'dose'],
      dtype='object')
```

Customize Filter

- Extract only relevant job tile based on industry knowledge
- Example of AI related keyword inside job title.
  [Data, Machine Learning, Analyst, Scientist, Deep Learning, Research, NLP, Artificial Intelligent, AI, IoT, Industry 4.0, Fintech, Engineer, Developer, Solution, Architect, Manager, VP, Lead, Technology, Consultant, Software]

3.1.4 Feature Engineering

Classify salary range into Low, Medium, and High for prediction.
Bin salary class into 3 groups:

- $3000 to $4500 as low
- $4500 to $6000 as Medium
- $6000 and above as High

One-Hot Encoding on categorical parameter as predictors.

- Employment type
- Seniority
- Salary range

Tokenize and CountVectorize on requirements column which has a lot of words to use as predictors.

- Job Requirements
- Job Skills
- Academic Qualification

...	optimization	c#	opencv	computer vision	api	jira	unix	docker	keras	qlik	gcp	scrum	airflow	.net	d3.js
...	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Fig 6. Example one-hot-encoding and bag-of-word from the feature engineering

3.1.5 Machine Learning Modelling

Prediction Target:

- Salary range in 3 different class: Low, Medium High

Predictors:

- Years of experience
- Seniority
- Job Category
- Skills
- Academic Qualification
- Requirements

Model Selection:

- Using Decision Tree Multi-classifier to predict Salary range
- Classification report and confusion matrix:

	precision	recall	f1-score	support
Low	0.92	0.82	0.87	226
Med	0.53	0.68	0.60	37
High	0.61	0.70	0.65	89
accuracy			0.78	352
macro avg	0.69	0.73	0.70	352
weighted avg	0.80	0.78	0.78	352

	Pred Low	Pred Med	Pred High
Actual Low	186	10	30
Actual Med	2	25	10
Actual High	15	12	62

- Top 10 importance feature from decision tree:

	coef	abs coef
Year_Experience	0.405679	0.405679
Sciences	0.098422	0.098422
java scala	0.083142	0.083142
Non-executive	0.066195	0.066195
phd	0.061634	0.061634
python java	0.046882	0.046882
algorithms	0.043642	0.043642
science engineering	0.043628	0.043628
opencv	0.037369	0.037369
science computer	0.033332	0.033332
software	0.030956	0.030956
Junior Executive	0.026203	0.026203

- Decision tree rule generated, will be transferred to KIE

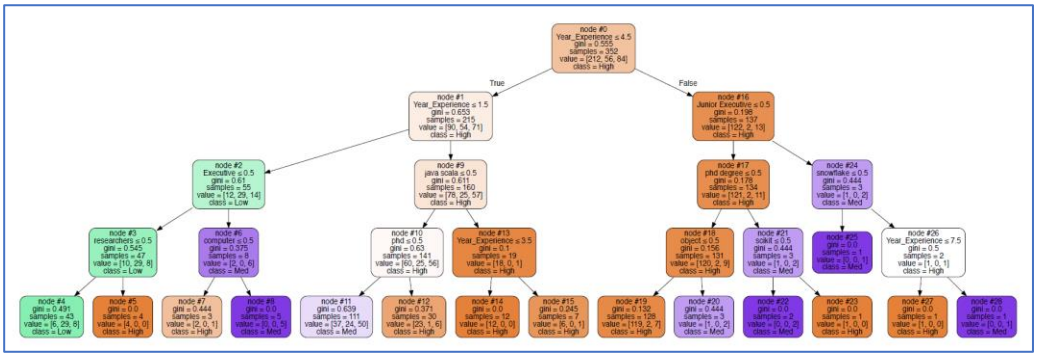


Fig 7. Example of decision tree diagram generated from Graphviz and Pydot

```
|--- Year_Experience <= 3.50
|   |--- java scala <= 0.50
|   |   |--- phd <= 0.50
|   |   |   |--- Year_Experience <= 1.50
|   |   |   |   |--- Executive <= 0.50
|   |   |   |   |   |--- class: Low
|   |   |   |   |--- Executive >  0.50
|   |   |   |   |   |--- class: Med
|   |   |   |--- Year_Experience >  1.50
|   |   |   |   |--- architecture <= 0.50
|   |   |   |   |   |--- class: High
|   |   |   |   |--- architecture >  0.50
|   |   |   |   |   |--- class: Med
|   |   |--- phd >  0.50
|   |   |   |--- engineering <= 0.50
|   |   |   |   |--- statistical <= 0.50
|   |   |   |   |   |--- class: Med
|   |   |   |   |--- statistical >  0.50
|   |   |   |   |   |--- class: High
|   |   |   |--- engineering >  0.50
|   |   |   |   |--- Year_Experience <= 1.50
|   |   |   |   |   |--- class: High
|   |   |   |   |--- Year_Experience >  1.50
|   |   |   |   |   |--- class: High
|   |--- java scala >  0.50
|   |   |--- class: High
```

Fig 8. Example of decision tree rule generated from scikit-learn

3.2. Knowledge Specification

After knowledge acquisition, knowledge specification involves representing the kinds of knowledge and reasoning processes used to perform a task, in this case is to recommend a list of jobs to job seekers. As such, it is necessary to specify the job requirements factors, the rules used to determine expected salary and job suitability score.

3.2.1 Job Requirement Factors

Based on the data collected from job listings, the following factors that affect the suitability of a job are:

- Educational qualifications
- Years of work experience
- Technical skills
- Salary range

However, it is also important to consider the applicant's own preferences to determine whether the job is suitable for them. This is because job matching is a bi-directional process, where the preferences of both the recruiter and the applicant needs to be taken into account[4]. Identified factors that would affect the applicant's preference:

- Preferred type of employment
- Preferred job scope/area
- Preferred seniority level
- Required travelling time to the job location

Based on the identified job requirement factors, the results are presented using a dependency diagram as seen below in Fig 9. The dependency diagram arranges the factors that determine whether a job is recommended to the applicant in a hierarchical tree structure. The topmost level node represents the decision of the recommender system, i.e. to output a list of recommended jobs.

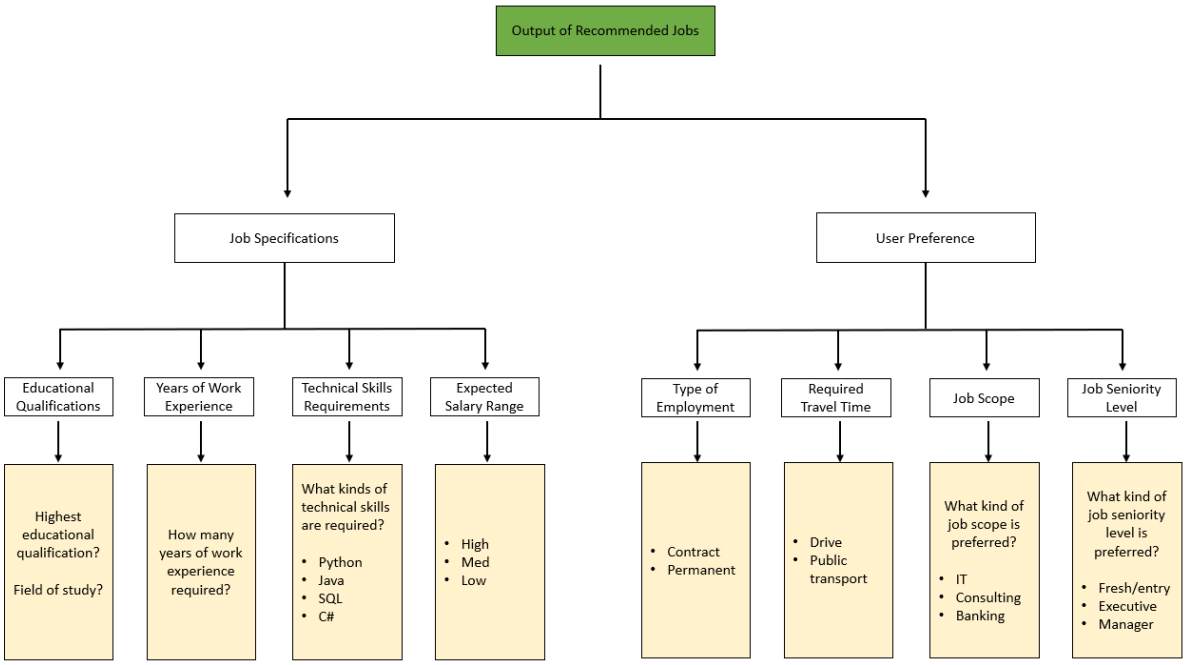


Fig 9. Dependency diagram of Smart Job Recommender System

3.2.2 Job Suitability Scoring

The first phase of the scoring logic is to filter out jobs that are not relevant or applicable to the applicant based on their provided information. The job requirement factors used in this phase are (1) educational qualification, (2) years of work experience and (3) type of employment. In order for a job listing to pass through the first phase, all 3 factors have to be fulfilled. The details of the rule used for the first phase is:

IF (applicant.EducationalQualification = job.EducationalQualifications)

AND (applicant.YearsOfExperience >= job.YearsOfExperience)

AND (applicant.PreferredTypeOfEmployment = job.TypeOfEmployment)

THEN (passThroughFirstPhase = true)

Two different scenarios of whether a job listing passes through the first phase is shown below in Table 2 and Table 3.

Factor	Applicant	Job Listing	Factor Fulfilled?	Result
Educational Qualification	Degree, Material Science	PhD, Computer Science	No	Failed first phase
Years of work experience	10	5	Yes	
Type of Employment	Contract	Contract	Yes	

Table 2. Scenario of job failing to pass through first phase

Factor	Applicant	Job Listing	Factor Fulfilled?	Result
Educational Qualification	PhD, Computer Science	PhD, Computer Science	Yes	Passed first phase
Years of work experience	10	5	Yes	
Type of Employment	Contract	Contract	Yes	

Table 3. Scenario of job passing through first phase

After passing through the first phase, the second phase involves the recommender system analyzing and assigning a suitability score for the remaining jobs based on certain job requirement factors. The job requirements factors used for the suitability scoring system are:

- Job scope
- Job seniority level
- Expected salary range
- Required travelling time
- Technical Skills

Each factor is assigned a weight value, which contributes to the overall suitability score. All the listed factors except for technical skills have a weight score of 1, where the latter has a weight score of 2. The reason why "technical skills" have a higher weight is because it directly affects the employer's decision to consider the suitability of the candidate, while the rest do not and caters more towards the applicant's preferences. This means that the ceiling value for the score would be 6. The higher the score, the more suitable the job is deemed for the applicant. The job suitability score is calculated through summation of the factor values from each job requirement factor (F) using Equation 1:

$$jobSuitabilityScore = \sum_{i=1}^{n} Score(F_i), where\ i = (1, 2, \ldots, n) \quad (1)$$

If the requirement factor is fulfilled, it will contribute the maximum of their weight score to the suitability score. Alternatively, if it is not fulfilled, it will contribute a score of 0 or a penalized score. The factors that would contribute a penalized score if they are not fulfilled are “Required Travel Time” and “Technical Skills”. This is because unlike the other factors, the two factors might come close to hitting the target value. For example, consider the scenario where the applicant’s preferred travelling time is 45 minutes, while the travelling time for a job location is 50 minutes. Although the job travelling time of 50 minutes does not meet the applicant’s target value of 45 minutes, it still comes quite close. As such, it would be more appropriate to penalize the factor’s contributed value according to how close it comes to meeting the target value.

Since each job requires a different number of technical skills, the system calculates a factor score for the “technical skills” based on the proportion of skills the applicant is able to match. The score is determined based on the variable, *techincalSkillsScore,* which calculates the proportion of skills met using Equation 2 as seen below:

$$technicalSkillScore = \frac{number\ of\ skills\ matched}{total\ number\ of\ required\ skills} \qquad (2)$$

According to MyCareersFuture website, applicants that have a skill match of 70% and above are more likely to be considered as seen below in Fig 10. Using this as a guideline, skill match proportion can be separated into three areas:

- High: *technicalSkillScore >= 0.7 (70% and above match)*
- *Med: 0.3 <= technicalSkillScore < 0.7 (between 30% and 70% match)*
- *Low: technicalSkillScore < 0.3 (less than 30% match)*

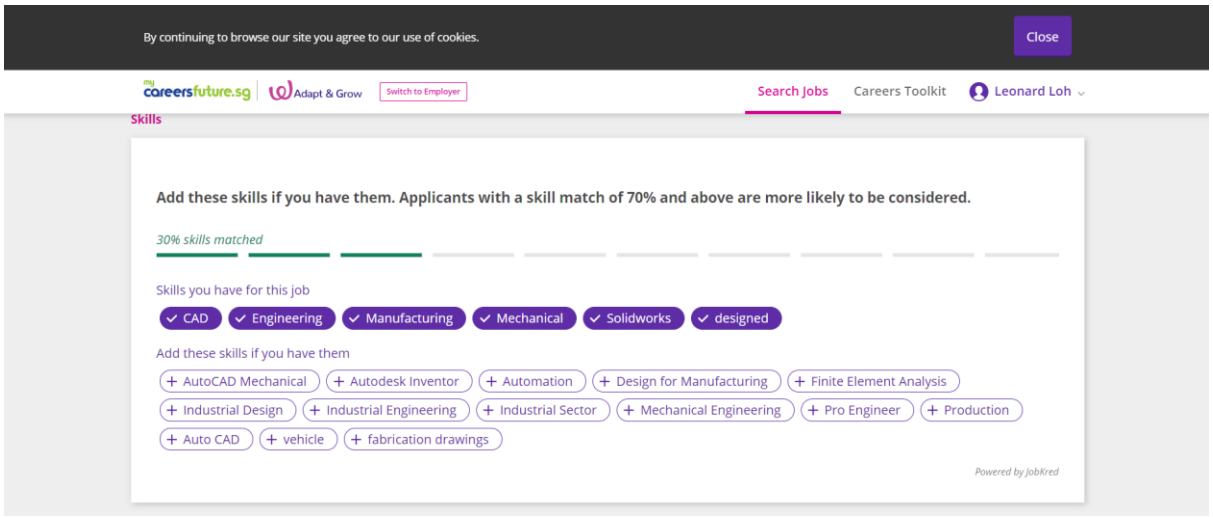


Fig 10. MyCareersFuture skill match scoring

If the technicalSkillScore corresponds to “High”, the factor score would be 2. If it is “Med”, the factor score will be penalized to be 1. If it is considered “Low”, the factor score would be 0 as the applicant is unable to meet a large majority of the skills required.

The rules logic used for the job suitability score calculation is as summarized below in Table 4.

S/N	Factor	Rules
F1	Job Scope	**IF** (applicant.PreferredJobScope = job.JobScope) **THEN** factorScore = 1
		IF (applicant.PreferredJobScope != job.JobScope) **THEN** factorScore = 0
F2	Job Seniority Level	**IF** (applicant.PreferredJobScope = job.JobScope) **THEN** factorScore = 1
		IF (applicant.PreferredJobScope != job.JobScope) **THEN** factorScore = 0
F3	Expected Salary Range	**IF** (applicant.ExpectedSalaryRange <= job.salaryRange) **THEN** factorScore = 1
		IF (applicant.ExpectedSalaryRange > job.salaryRange) **THEN** factorScore = 0
F4	Required Travel Time	**IF** (applicant.PreferredTravelTime >= job.RequiredTravelTime) **THEN** factorScore = 1
		IF (applicant.PreferredTravelTime < job.RequiredTravelTime) **THEN** factorScore = 1 - (job.RequiredTravelTime - applicant.PreferredTravelTime) / job.RequiredTravelTime
F5	Technical Skills	**IF** (technicalSkillScore >= 0.7) **THEN** factorScore = 2
		IF (0.3 <= technicalSkillScore < 0.7) **THEN** factorScore = 1
		IF (technicalSkillScore < 0.3) **THEN** factorScore = 0

Table 4. Summary of score calculation rules for each factor

4. System Architecture

The system architecture of the Smart Job Recommender System is illustrated below in Fig 11. jBPM is selected to serve as the front-end interface for users, which is integrated with the back-end Google Maps API, rule-based system and job website database.

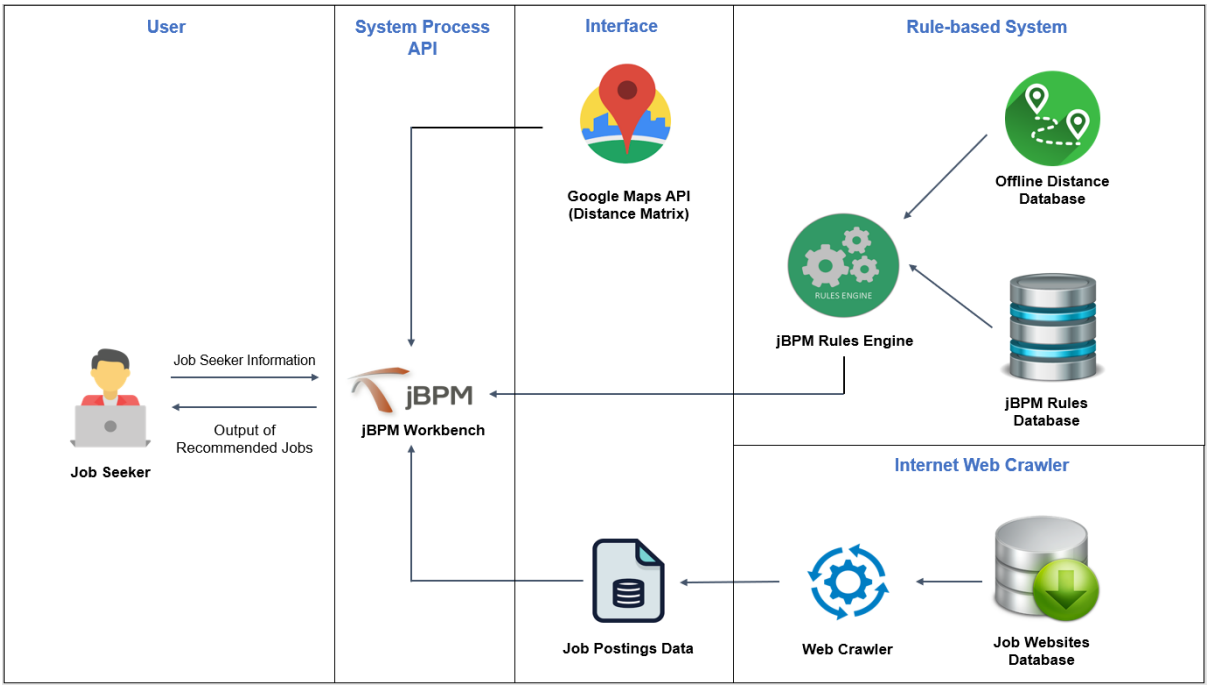


Fig 11. Smart Job Recommender System Architecture

4.1. jBPM/KIE Business Process

Collection of job seeker's information

When the system is started up, it will provide a form to the user to fill in. Information that is obtained from the form are:

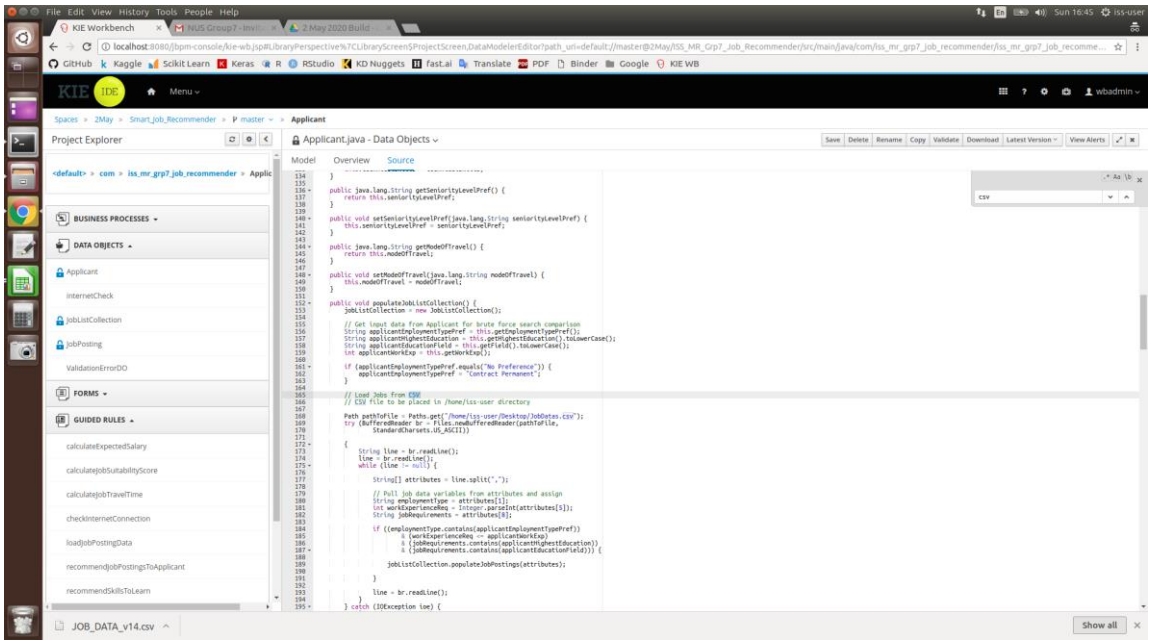
- Educational qualifications
  - Highest education attained
  - Field of study
- Postal address
- Years of working experience
- Technical skills
- Preferred job scope/area
- Preferred job seniority level
- Preferred type of employment
- Desired travelling time to work
  - Drive
  - Public transport

Parsing of job posting data from CSV file

After the user form has been filled in and verified, the system will parse the job database CSV file. Each job entry in the file will be passed through filters to determine its suitability for the applicant.

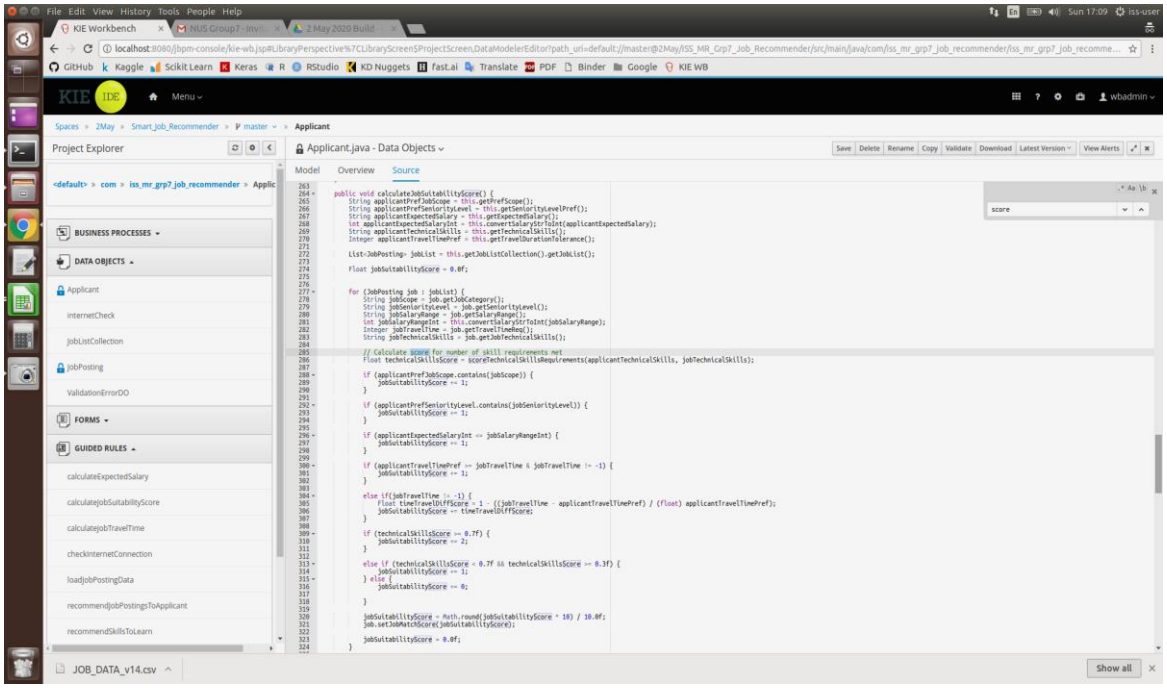
First Phase - Hard Constraint Filter

Using the predefined rules for the first phase, jobs that meet the requirements of the rules will be allowed to pass through. This removes any job listings that would be completely incompatible for the applicant.



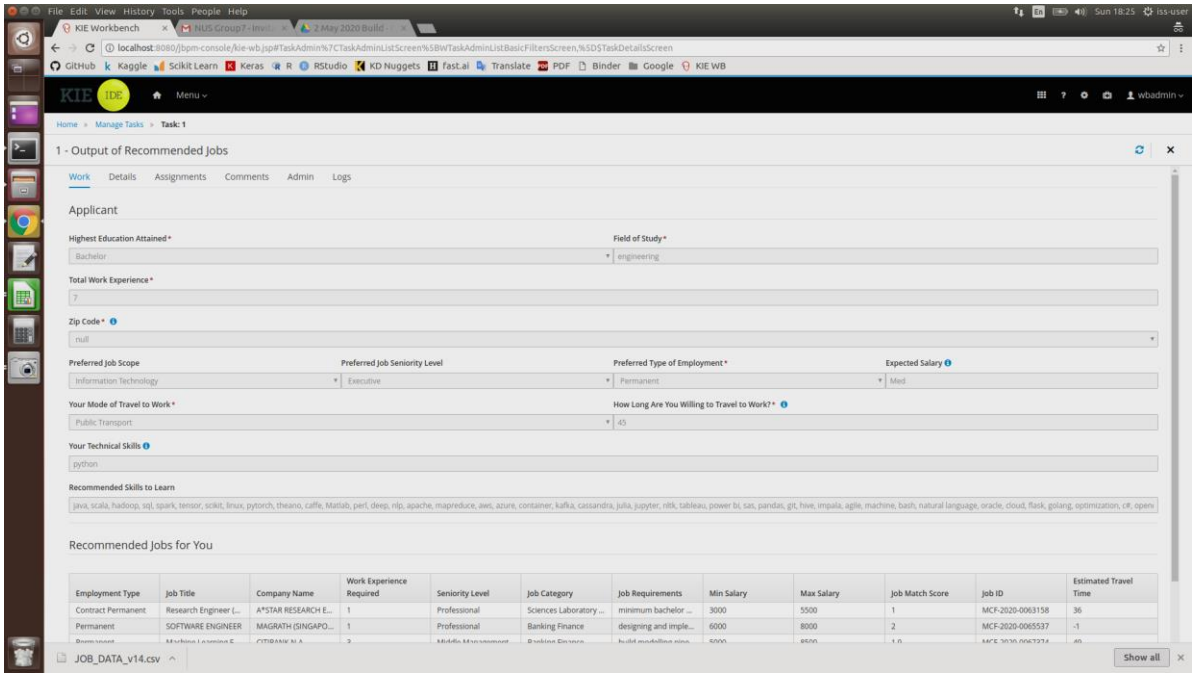
Second Phase - Job Suitability Scoring

For the jobs that have passed through the first phase, the second phase involves the job suitability scoring system, where the score is used to inform the applicant how close or suitable the job is for them.



Output of the recommended jobs

At the end of the process, the system will output a list containing all the recommended jobs with a corresponding job suitability score for the applicant to review. Jobs with higher suitability scores would indicate better matching of the job to the applicant.



5. Marketing and Monetization Strategy

5.1 Consumer awareness/Marketing

The first step is to raise awareness of the system to users, this would help to build a user-base and further advertise the system. Marketing strategies that can be employed involve the use of social media platforms such as Facebook, Instagram and Google Ads. Email marketing is also another possible marketing method, with the target group being fresh graduates and active job seekers.

5.2 Monetization Strategy

To sustain this as a business plan, the system is subdivided into 2 different models.

A. **Free-to-use model:** features such as job suitability scoring, expected salary range calculation and skills recommendation will not be available
B. **Subscription model**: has full access to all features, including job suitability scoring, expected salary range calculation and skills recommendation

6. System Limitations & Future Improvements

Due to limited time and resources, the current dataset is only able to recommend Information Technology based jobs in Singapore. This can be expanded in the future to cover more major industries such as manufacturing and retail.

The future improvement involves the need of allowing the applicant to have the control of how many jobs they should be recommended per day and even set a job scoring filter which will be helpful for the applicants.

Another possible area of improvement for the system is to implement a feature that can recommend possible job courses based on the recommended skills to learn for the applicant. This could potentially open up business opportunities to work with tertiary course providers such as Tertiary Courses Singapore or SkillsFuture, where their courses would be advertised and recommended to users of the system.

7. Conclusion

In this project, a Smart Job Recommender System is presented, which can provide a list of recommended jobs for users through effective filtering of incompatible jobs and suitability scoring.

Firstly, a web crawler is used to generate a job listing database, which is outputted to a CSV file for the system to consume. Concepts such as knowledge-based reasoning and optimization techniques are employed to build the system, allowing it to effectively filter out incompatible jobs and provide a suitability score for the remaining jobs. Users are then able to use the suitability score as a gauge to determine which jobs are most suitable for them at a glance. Thus, the system can address the problems faced by job seekers that continue to use current job search websites.

The system also has high commercial potential with its subscription-based model and potentially large user-base to be tapped upon. Business opportunities with external course providers have also been identified, further adding to the business value of the system.

Appendices

Appendix A: Mapped System Functionalities

- **Knowledge-based reasoning techniques:** KIE business rule
- **Optimization techniques:** Score-based heuristic evaluation
- **Knowledge Discovery:** data mining through web crawling and rule induction with decision tree

Appendix B: Installation & User Guide

ISS-VM Installation

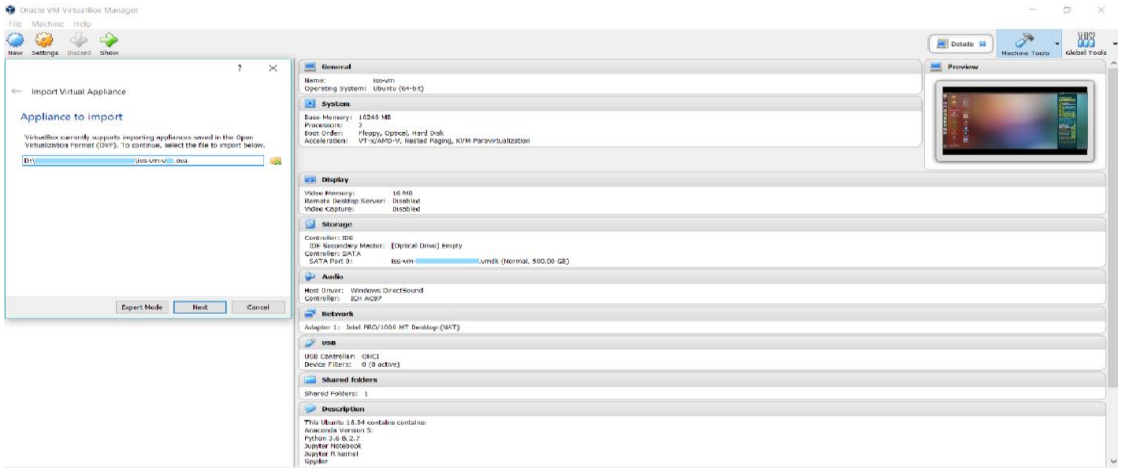
Requirements:

- ISS-VM Ubuntu 16.04
- Tool KIE 7.12
- Google Chrome

Procedure to install ISS-VM:

1. Download and install Virtualbox software (recommended version 5.2.20): https://www.virtualbox.org/wiki/Downloads
2. Download iss-vm virtual machine (an Appliance) from:
    1. http://bit.ly/iss-vm-v20a (part 1 about 11 GB in file size)
    2. http://bit.ly/iss-vm-v20b (part 2 about 11 GB in file size)
    3. http://bit.ly/iss-vm-v20c (part 3 about 10 GB in file size)
3. **[Note] Please check/ensure the 'virtualization' option is enabled in your computer's BIOS/hardware**
4. Put all three zip files in the same folder; select the first file iss-vm-vNN.zip.001. Use tools like 7-zip to unzip the folder. (https://www.7-zip.org/download.html)
5. Start Virtualbox software

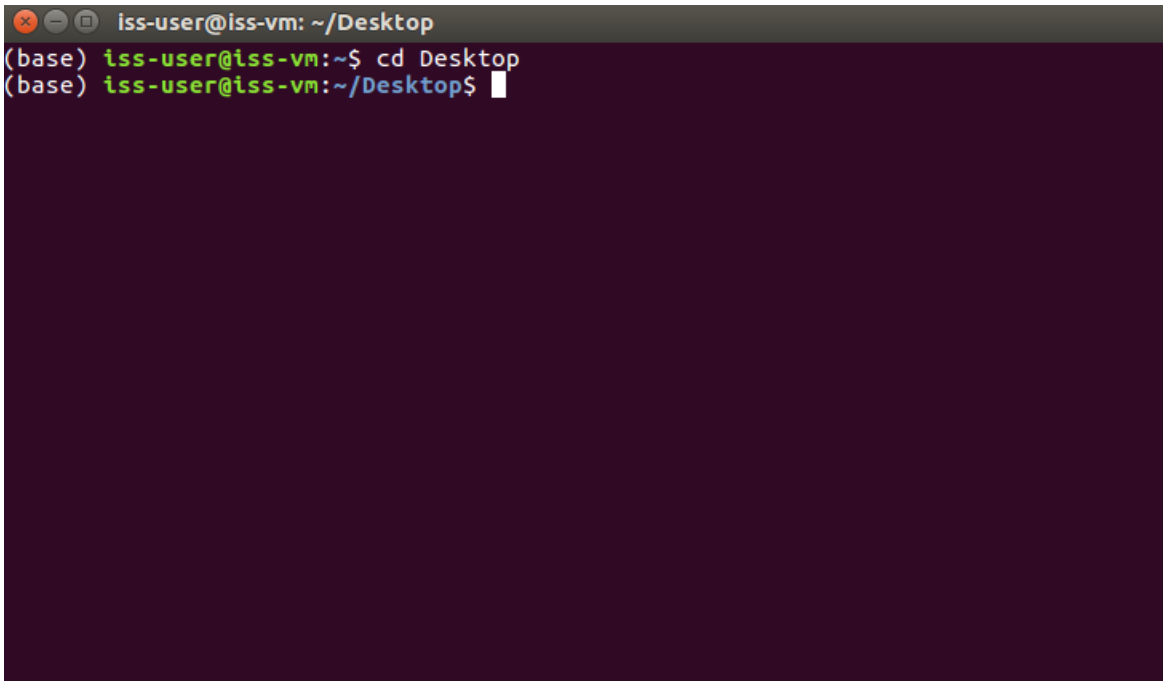
6. Click File ->Import Appliance



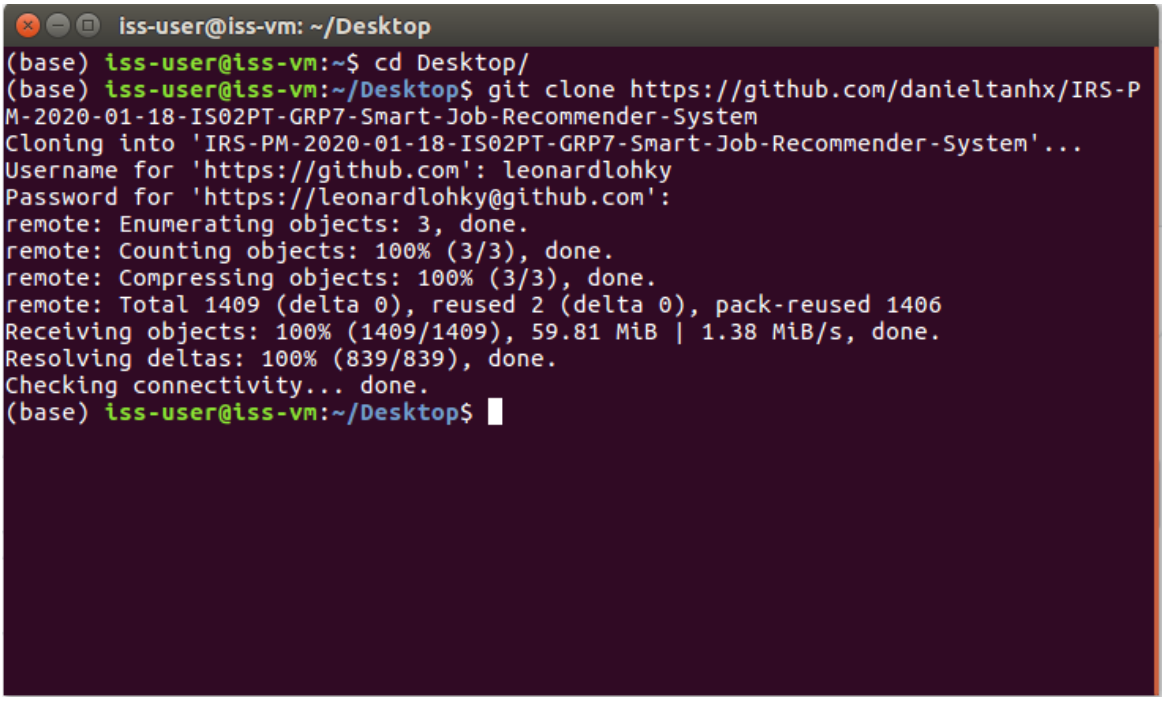
7. Click “Start” to launch iss-vm

Deploying in KIE jBPM 7.12

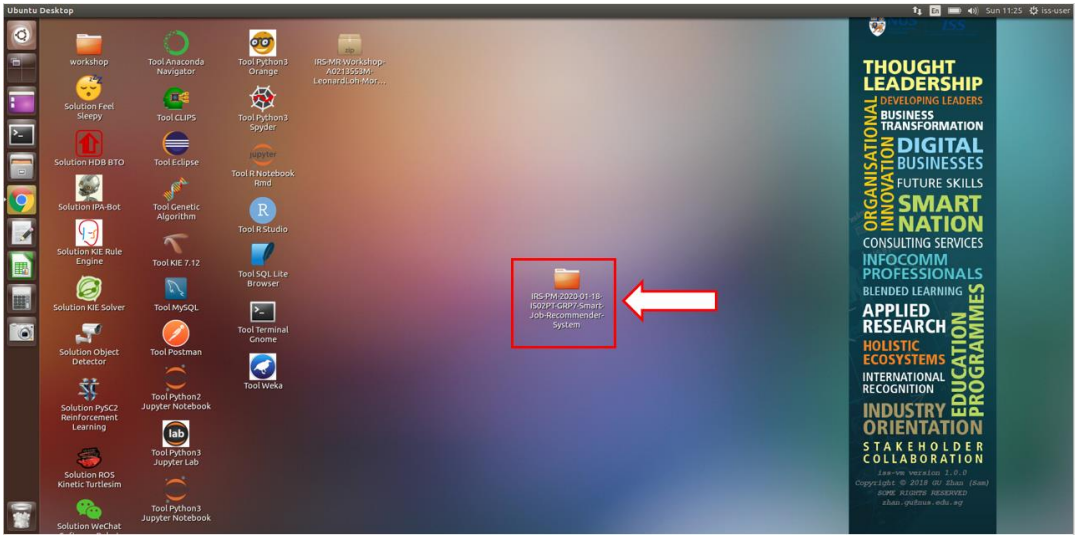
1. Open a new terminal. Navigate to the Desktop by typing “cd Desktop” in the terminal



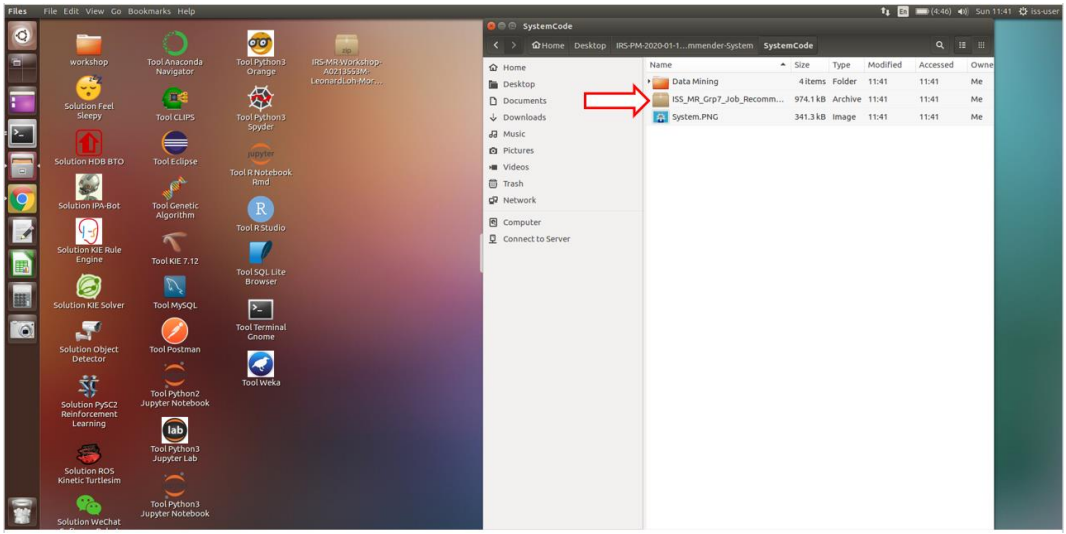
2. Clone the Git repository by typing “git clone https://github.com/danieltanhx/IRS-PM-2020-01-18-IS02PT-GRP7-Smart-Job-Recommender-System” in the terminal



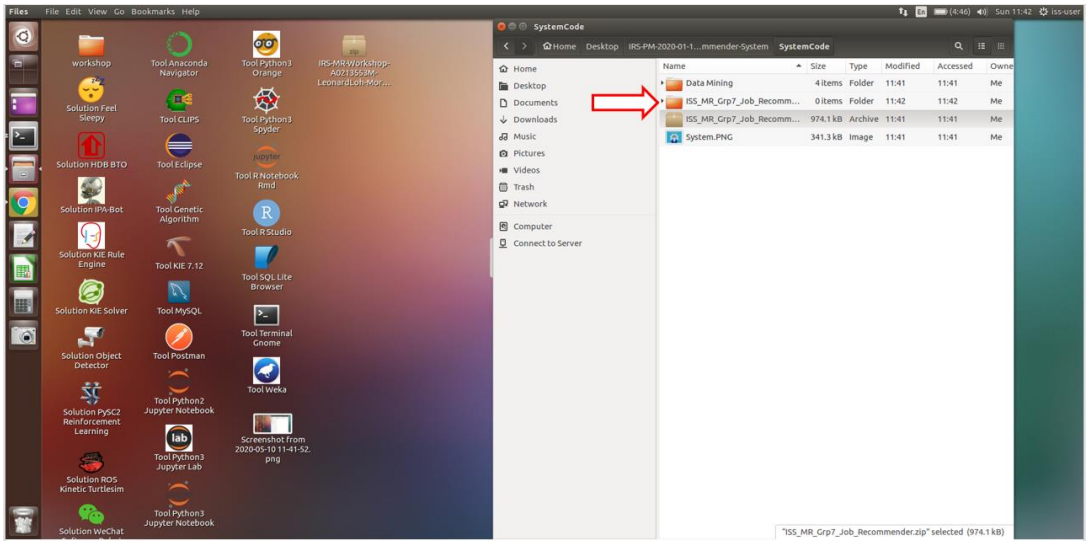
3. The cloned repository will appear on the Desktop screen as a folder named “IRS-PM-2020-01-18-IS02PT-GRP7-Smart-Job-Recommender-System”



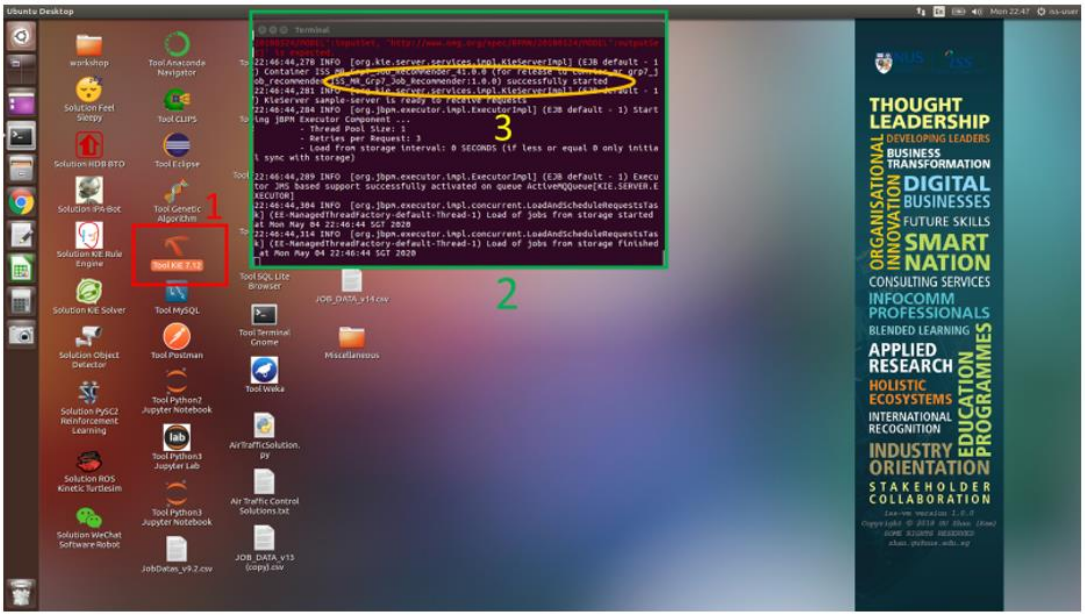
4. Enter the “IRS-PM-2020-01-18-IS02PT-GRP7-Smart-Job-Recommender-System” folder and navigate to the “SystemCode” subfolder. You will find a ZIP folder named “ISS_MR_Grp7_Job_Recommender.zip”



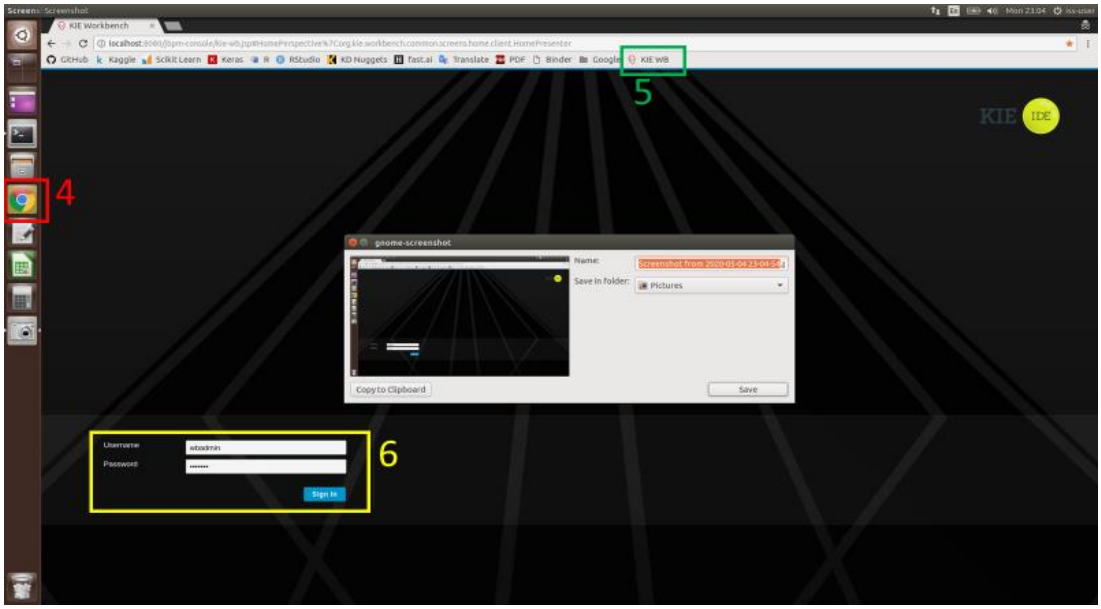
5. Extract the contents of the folder to the same location where the ZIP folder is located



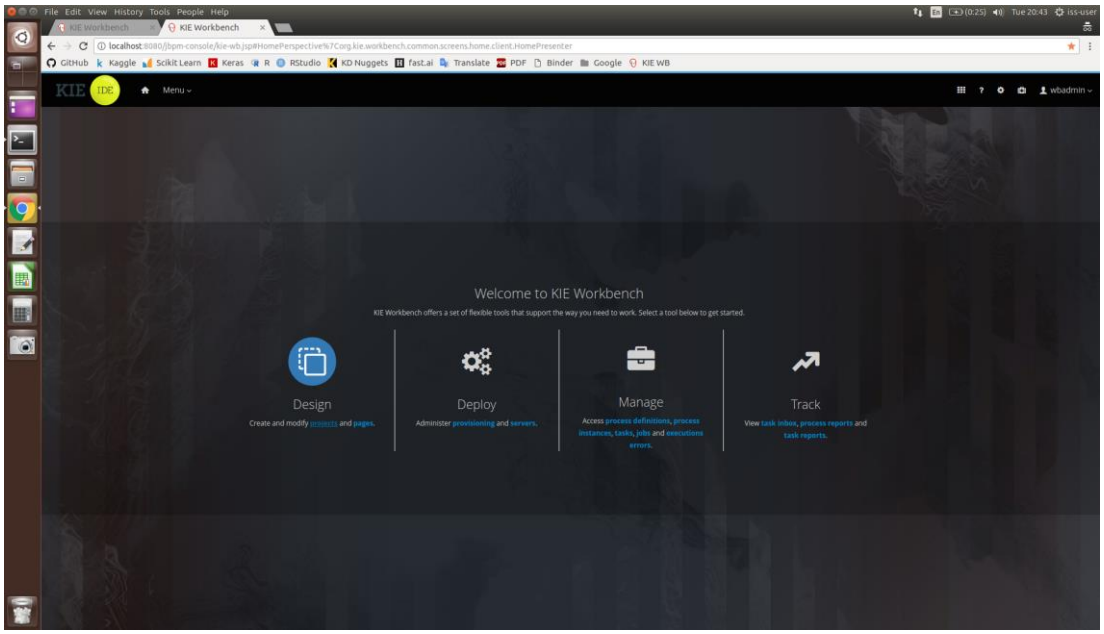
6. Click on Tool KIE 7.12, then wait for the terminal to pop out “successfully started” or “localhost:8080 successfully registered kind of words”.



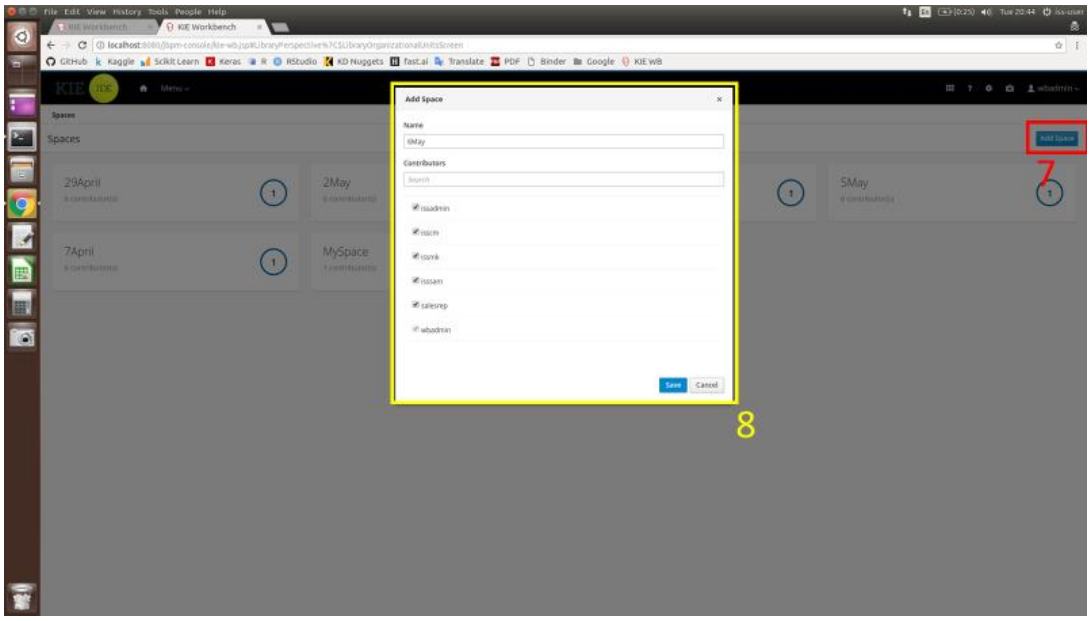
7. Launch Google Chrome and click on the KIE WB bookmark (labelled in Box 5). Once loaded, the below screen will appear. Sign in using username and password “wbadmin” and “wbadmin” respectively.



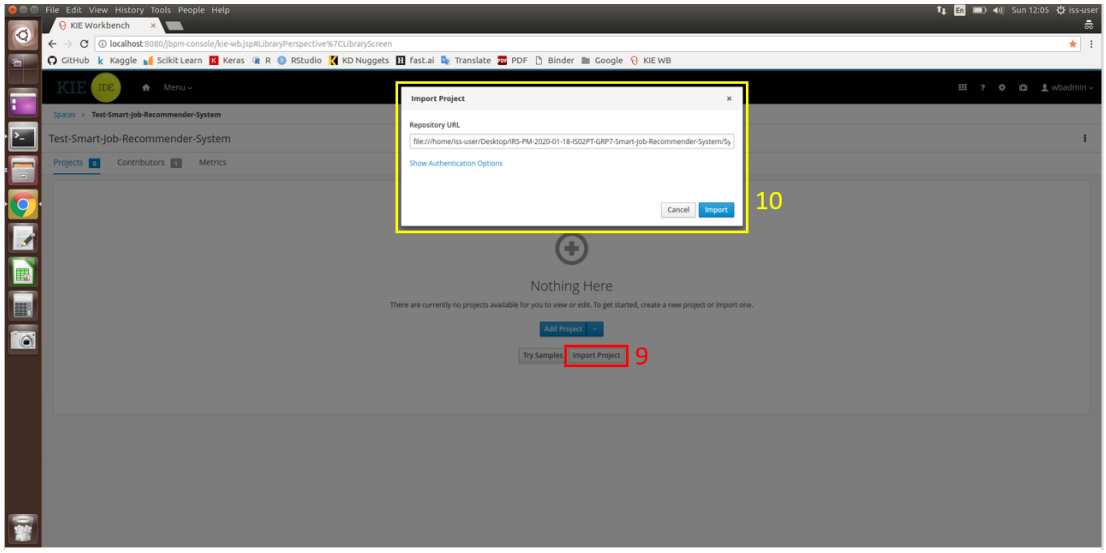
8. Click on Design



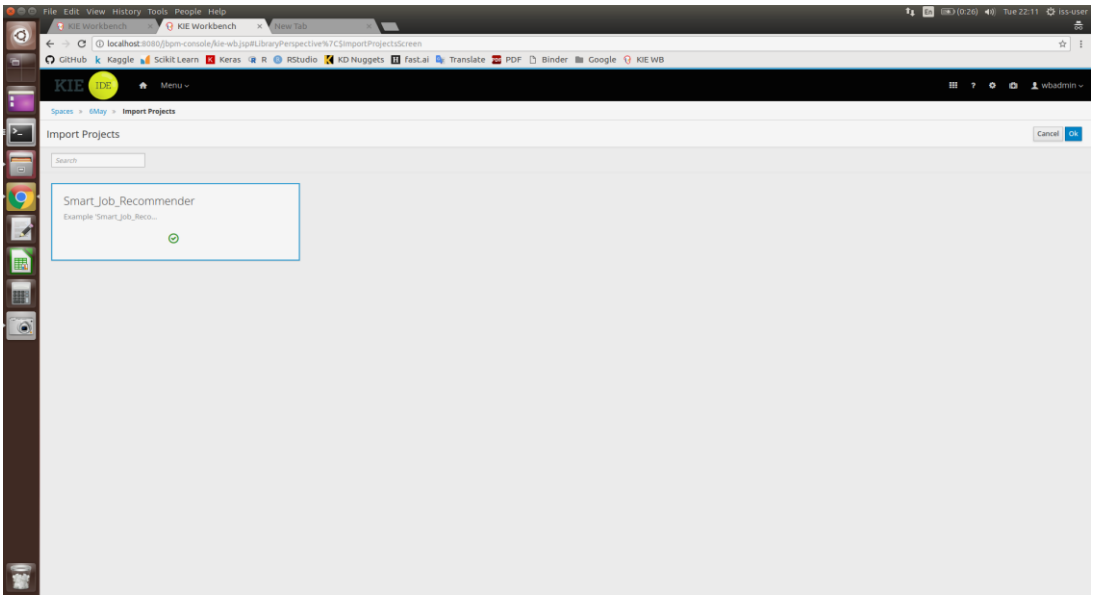
9. Press on the “Add Space” button (labelled in Box 7), enter a name for the space, e.g. “Smart-Job-Recommender”. Press “Save” once done



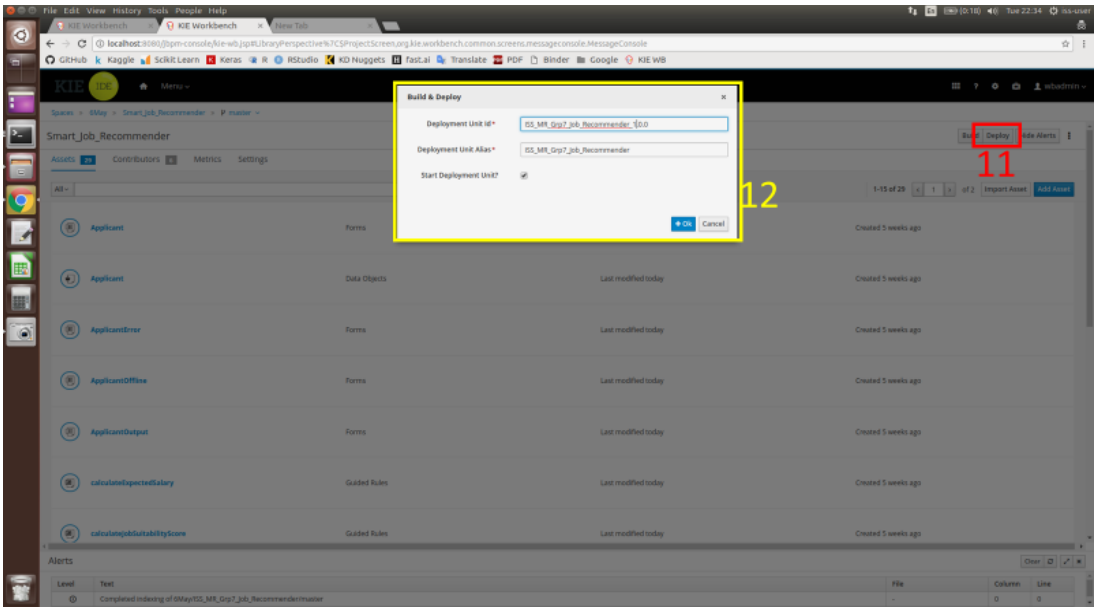
10. In the newly created space in KIE, click on “Import Project” (labelled in Box 9). Type “file:///home/iss-user/Desktop/IRS-PM-2020-01-18-IS02PT-GRP7-Smart-Job-Recommender-System/SystemCode/ISS_MR_Grp7_Job_Recommender” into the repository URL as shown in Box 10. Press “Import” after that



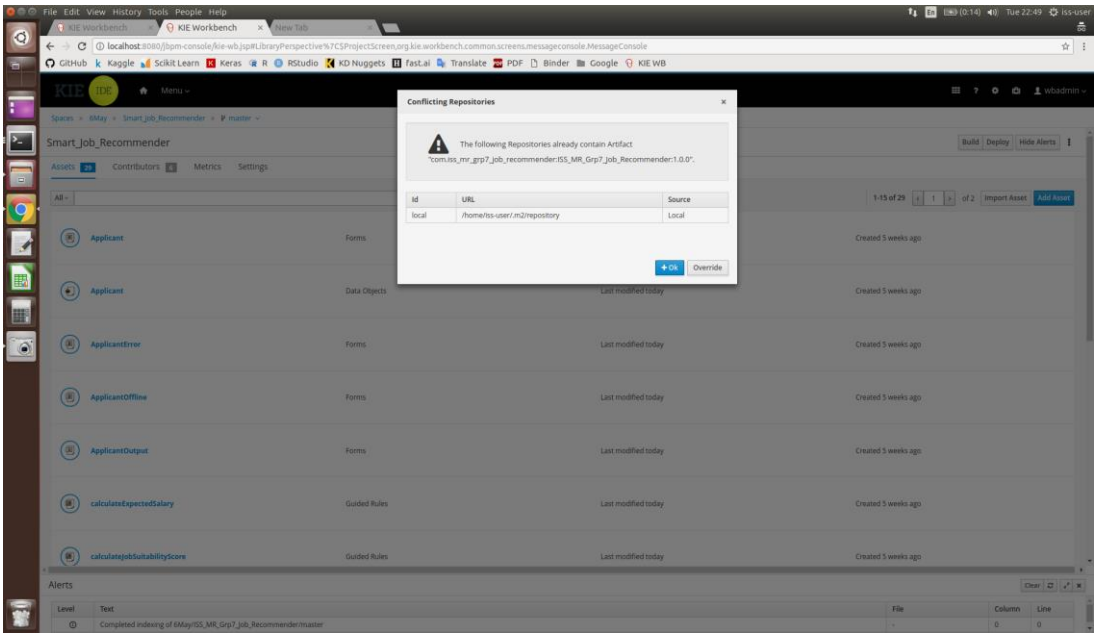
11. Once imported, click on the project “Smart_Job_Recommender” to select it. Then press “Ok” on the top right corner.



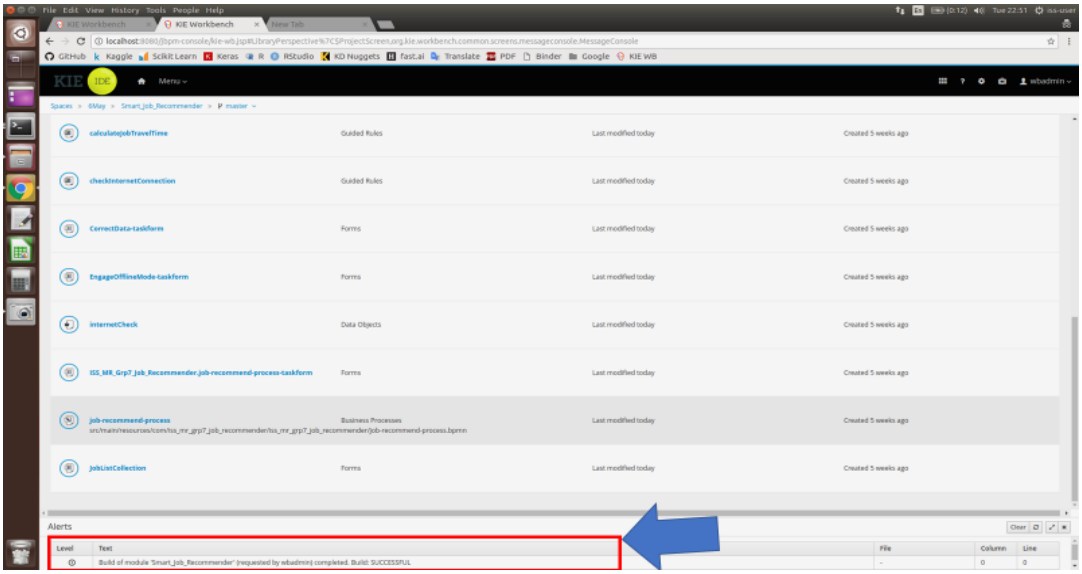
12. After importing successfully, click on the “Deploy” button (labelled in Box 11), which will bring up a popup window as seen in Box 12. Type in a Deployment Unit ID accordingly. After that, press “+Ok”



13. After pressing “+Ok”, the below screen will pop out. To continue, press “Override”.



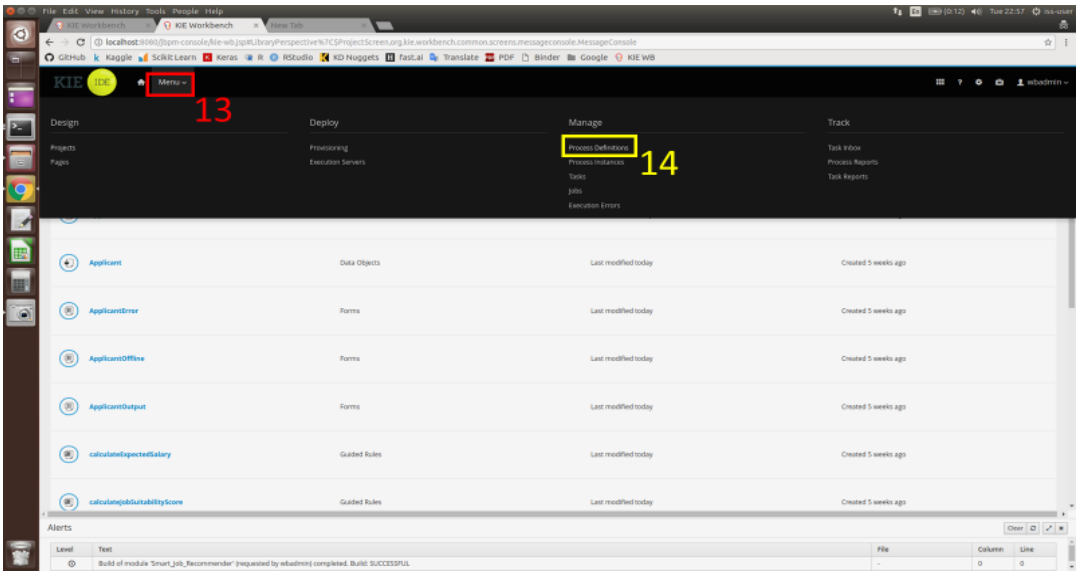
14. Once the deployment/build is completed, the text in level shown below will indicate Build: SUCCESSFUL.



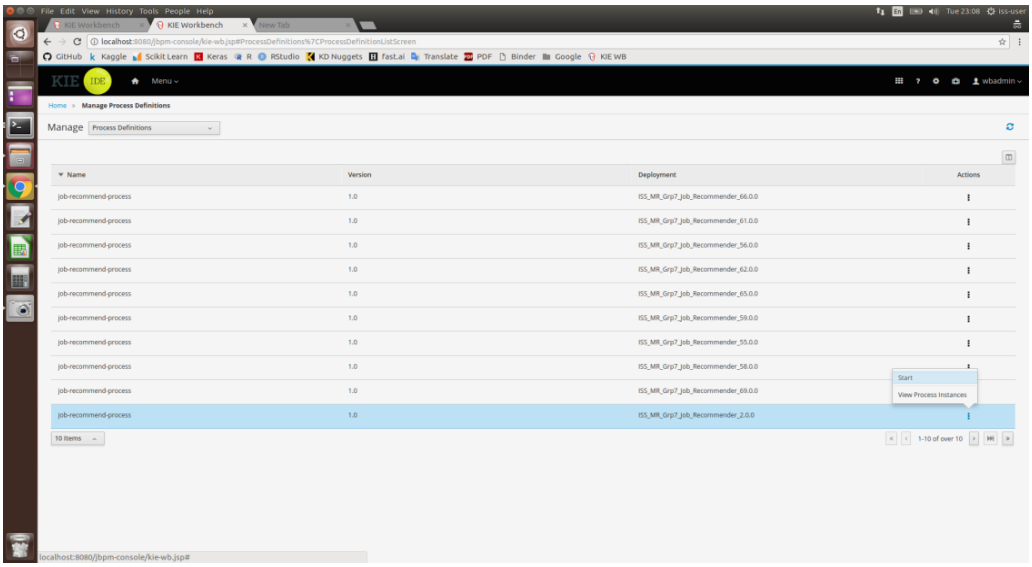
Running the System in KIE (Online Mode)

Online mode requires an internet connection. This is for the system to calculate an estimated travel time required for each job location depending on the user's ZIP code.

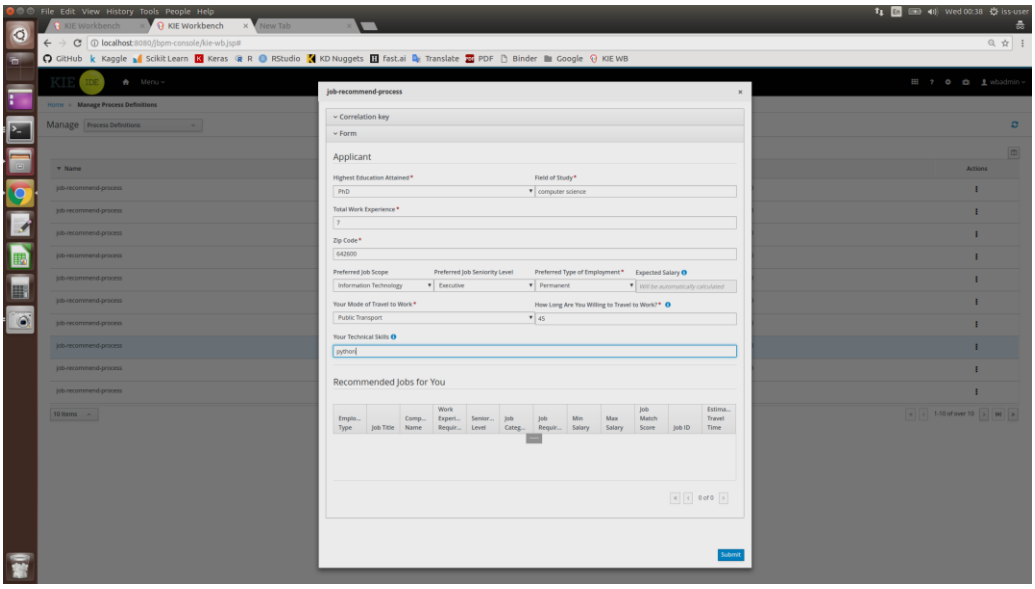
1. Press "Menu" (labelled in Box 13), and then click on "Process Definition" (labelled in Box 14)



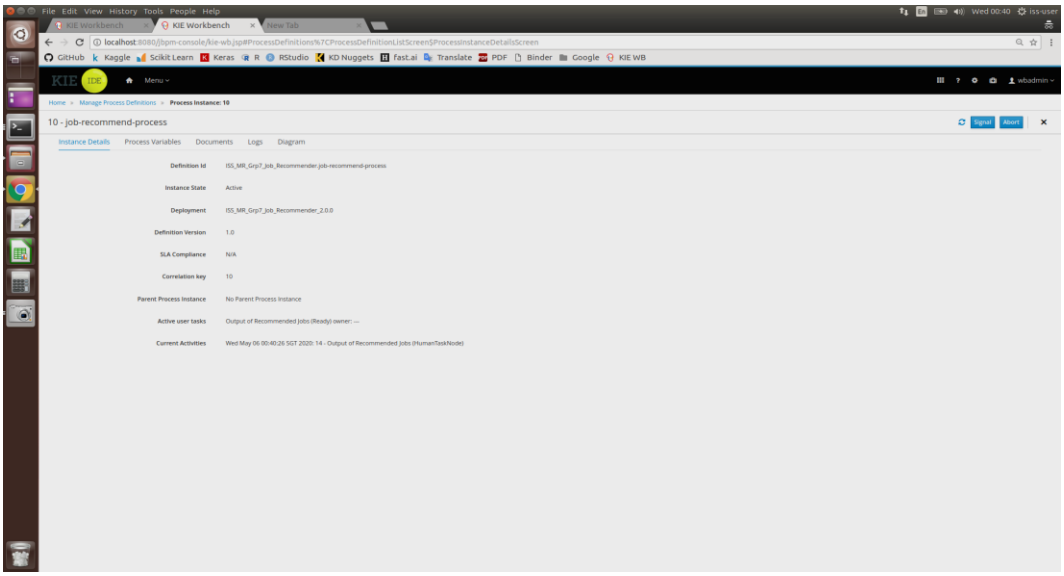
2. Select the relevant server that was deployed earlier to start the system



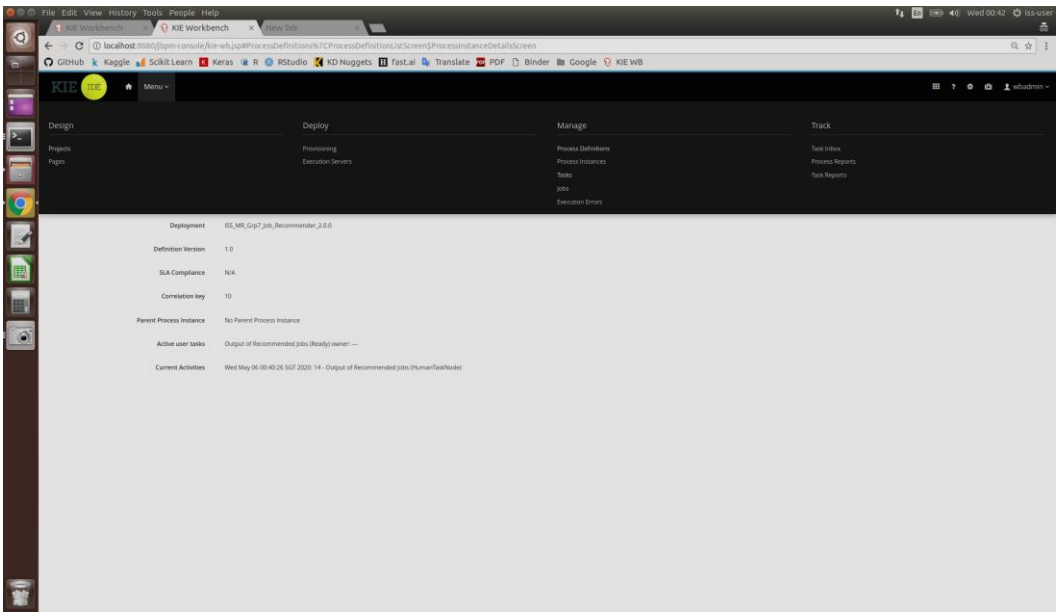
3. Once started, an applicant form for the user to fill in their details will appear. After filling up the form, press the "Submit" button ONCE. This is to avoid accidentally submitting multiple copies of the same task to the system. Please wait patiently as the system is processing the submitted information.



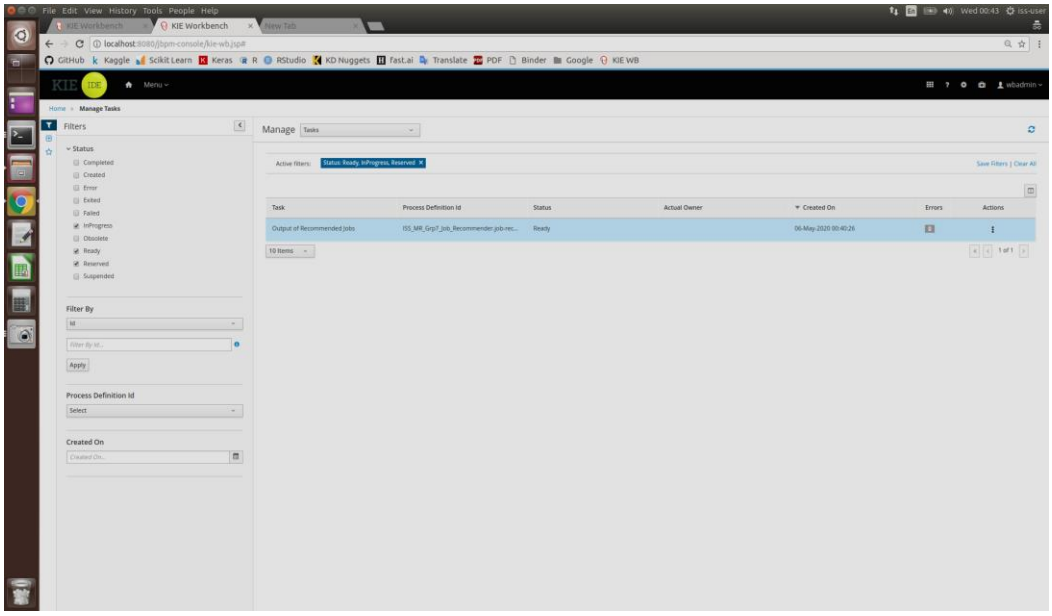
4. Once the user's information has been processed, the below page will appear.



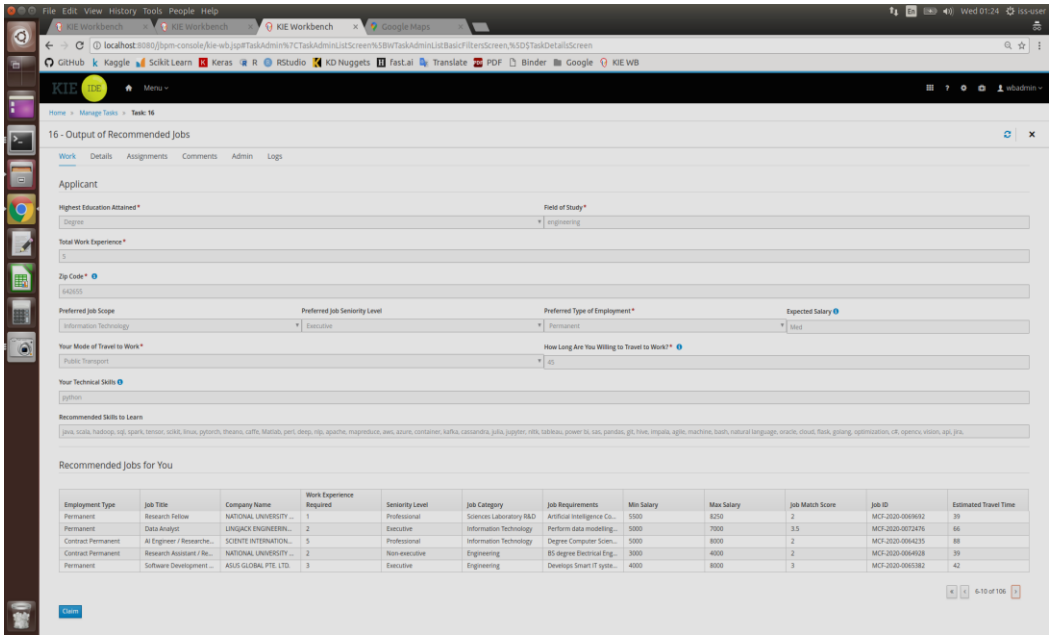
5. Click on "Menu", then click on "Task".



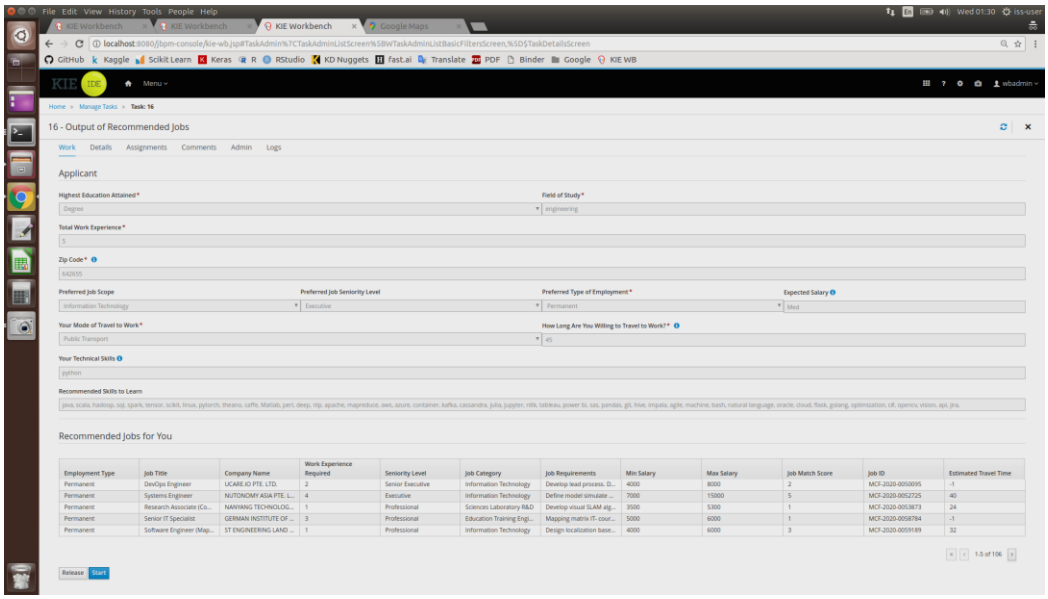
6. After completing Step 5, the below page should appear. To continue, click on on the task “Output of Recommended Jobs”



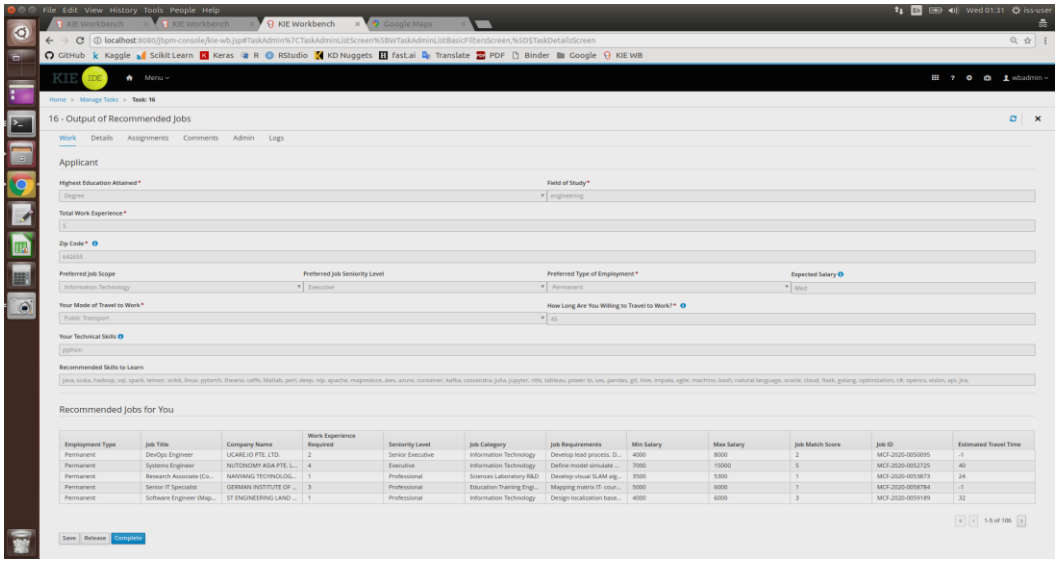
7. Press the “Claim” button at the below of the page



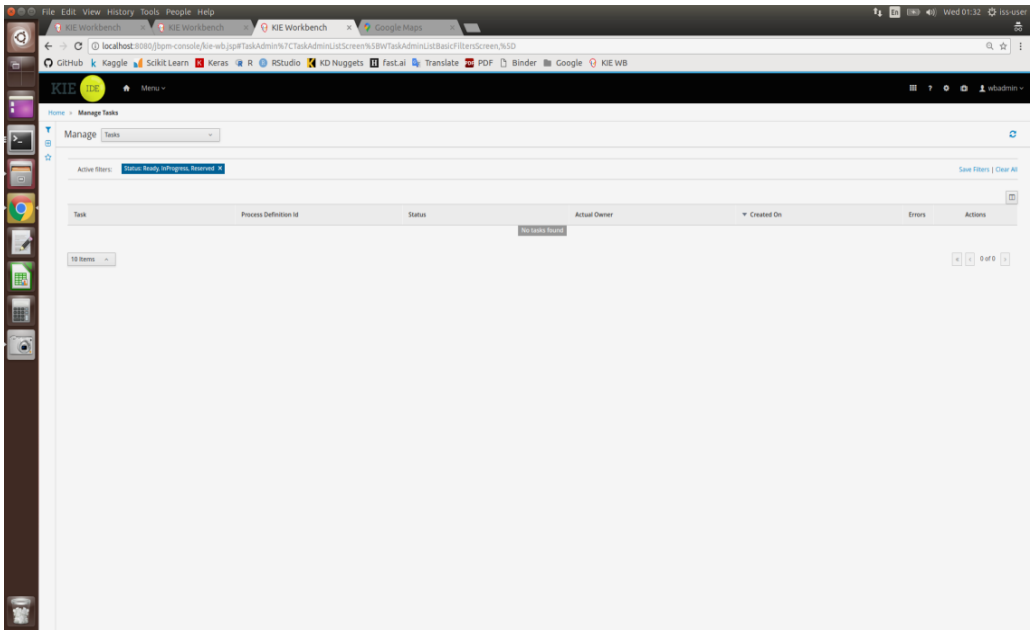
8. Press “Start”



9. After reviewing the output of recommended jobs, press “Complete” to end the task instance.



10. Process completed

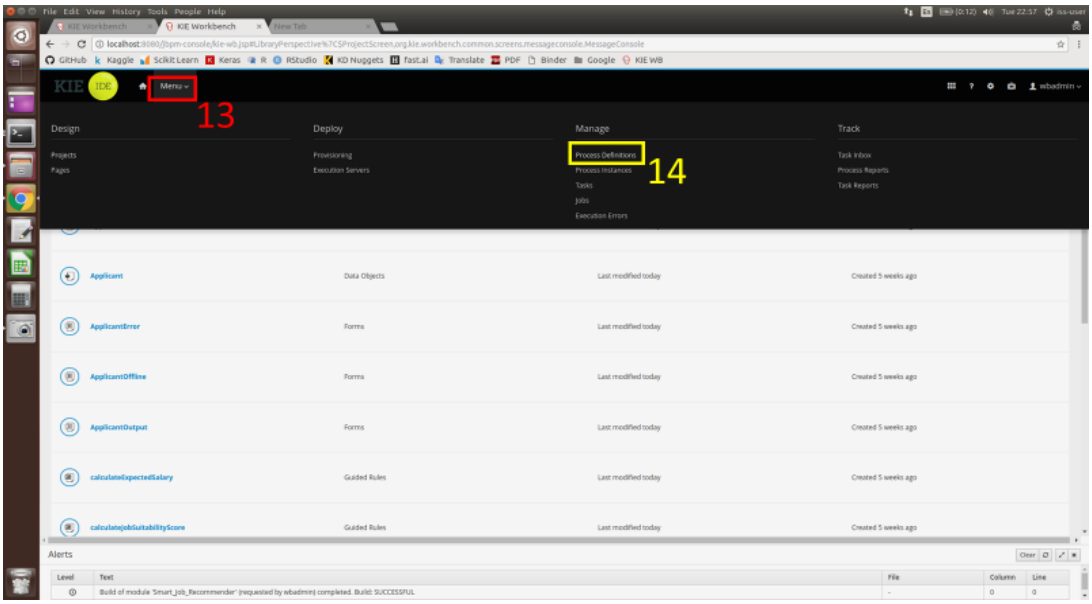


Running the System in KIE (Offline Mode)

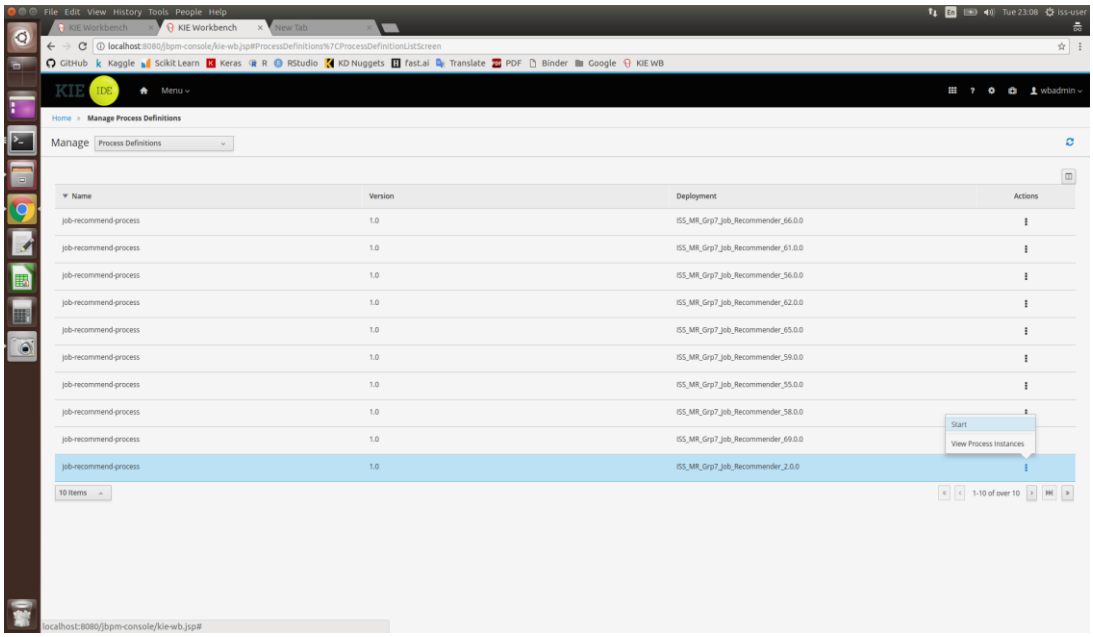
In the event an internet connection cannot be established, the system will run in Offline Mode. In this mode, the system will not be able to calculate an estimated travel time required for each job location according to the user's chosen ZIP code. Instead, users will be prompted to select one of three possible ZIP codes available in the system's database.

The procedure to operate the system in Offline Mode is as follows:

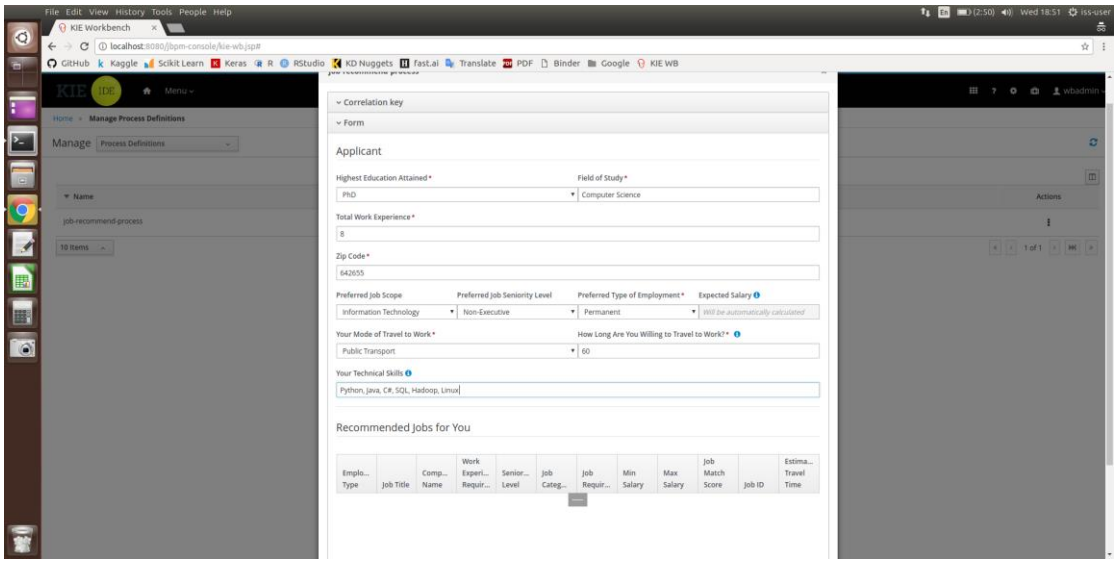
1. Press "Menu" (labelled in Box 13), and then click on "Process Definition" (labelled in Box14).



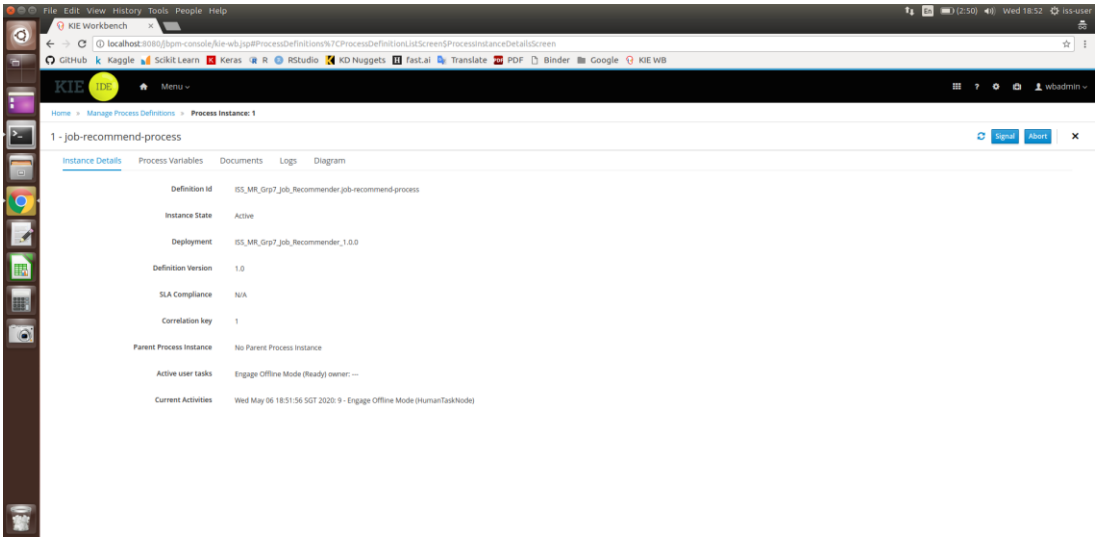
2. Select the relevant server that was deployed earlier to start the system



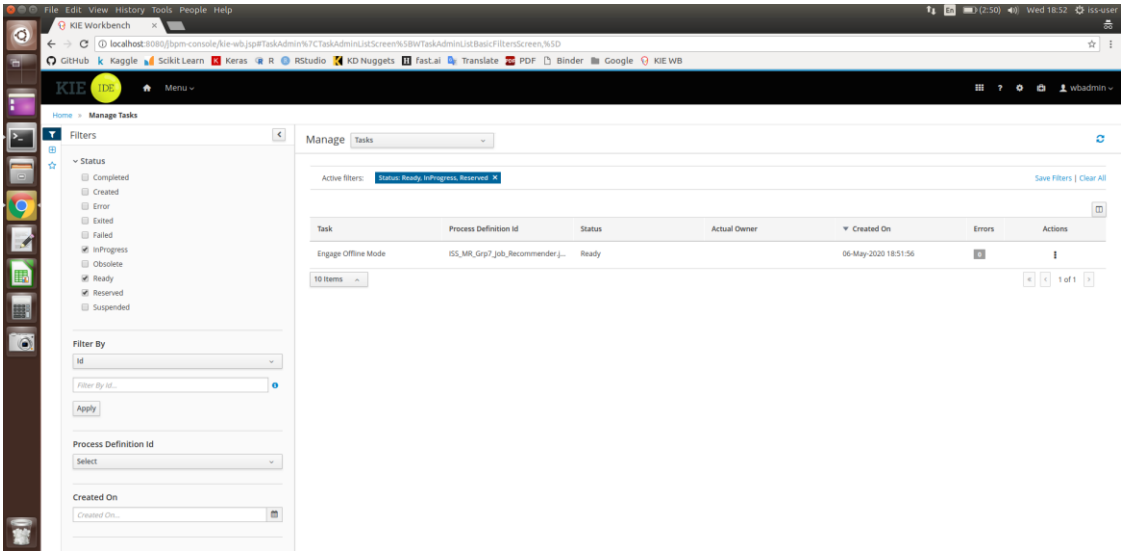
3. Once started, an applicant form for the user to fill in their details will appear. After filling up the form, press the "Submit" button ONCE. This is to avoid accidentally submitting multiple copies of the same task to the system. Please wait patiently as the system is processing the submitted information



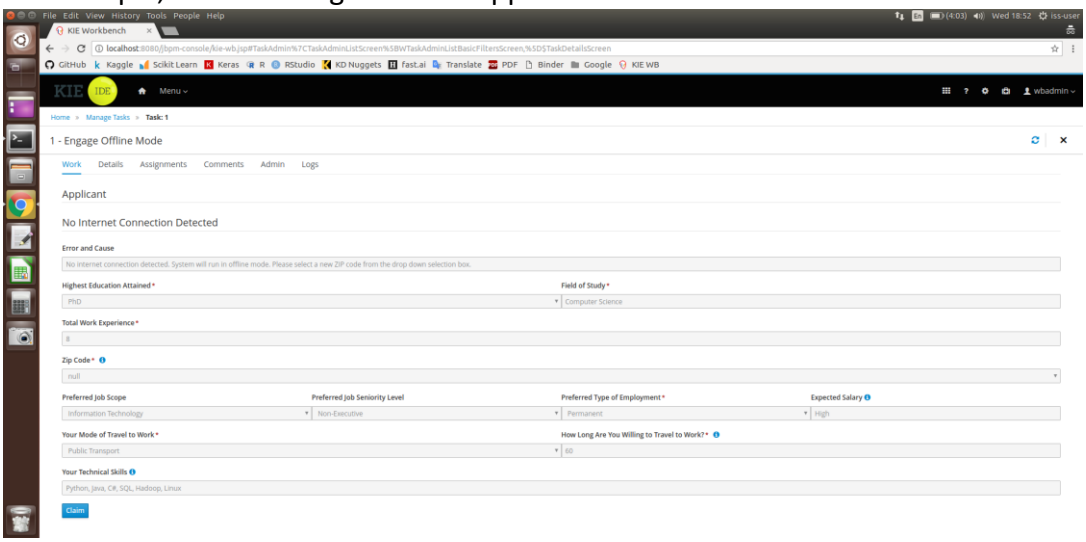
4. As no internet connection is established, the system will notify the user that it will be running in Offline Mode as shown in the page below



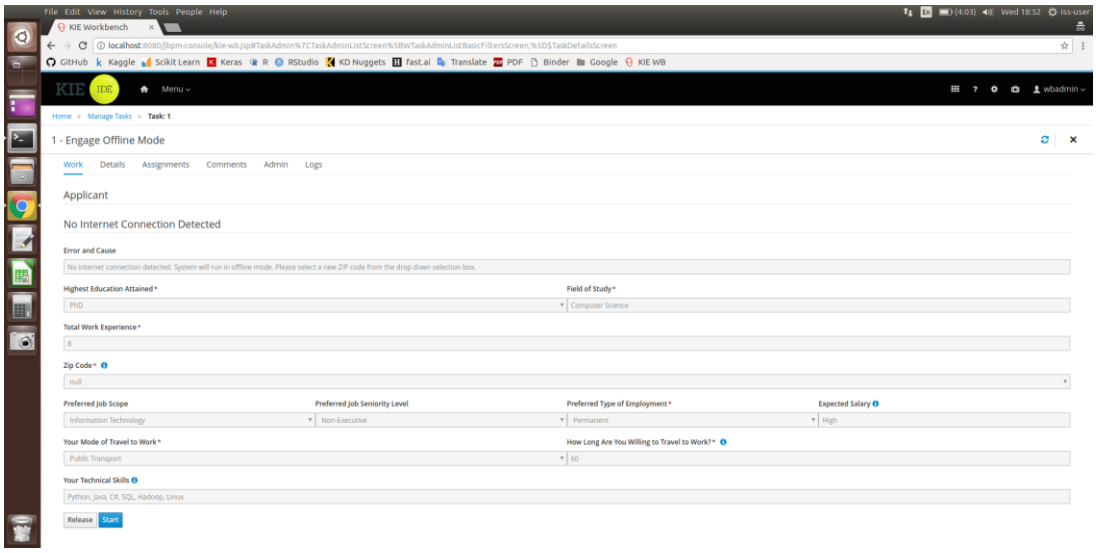
5. Click on “Menu”, then click on “Task”. This will cause the following page to appear. To continue, click on the task “Engage Offline Mode”



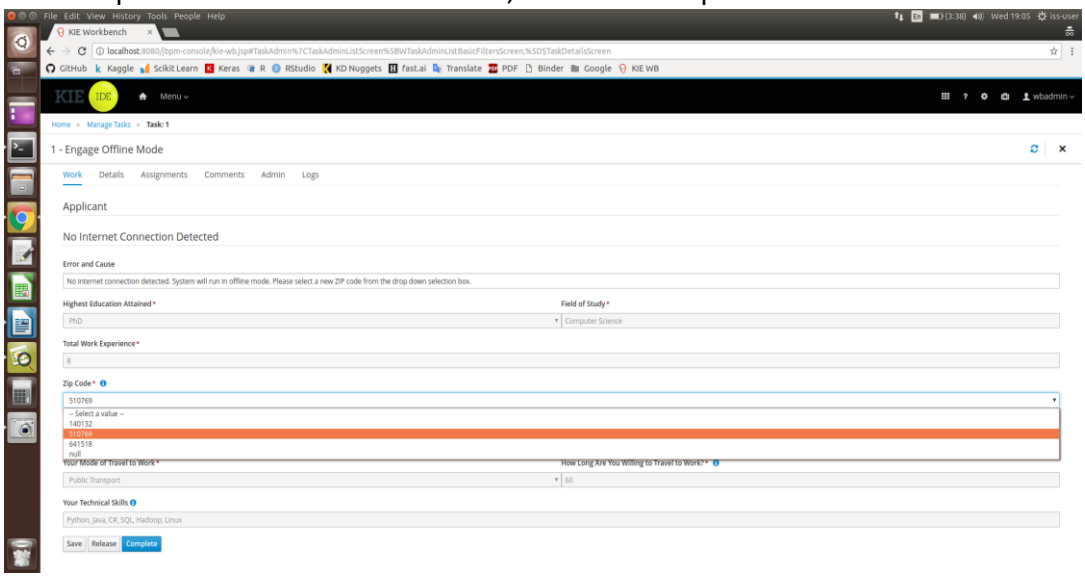
6. After Step 5, the following form will appear. Click on “Claim”



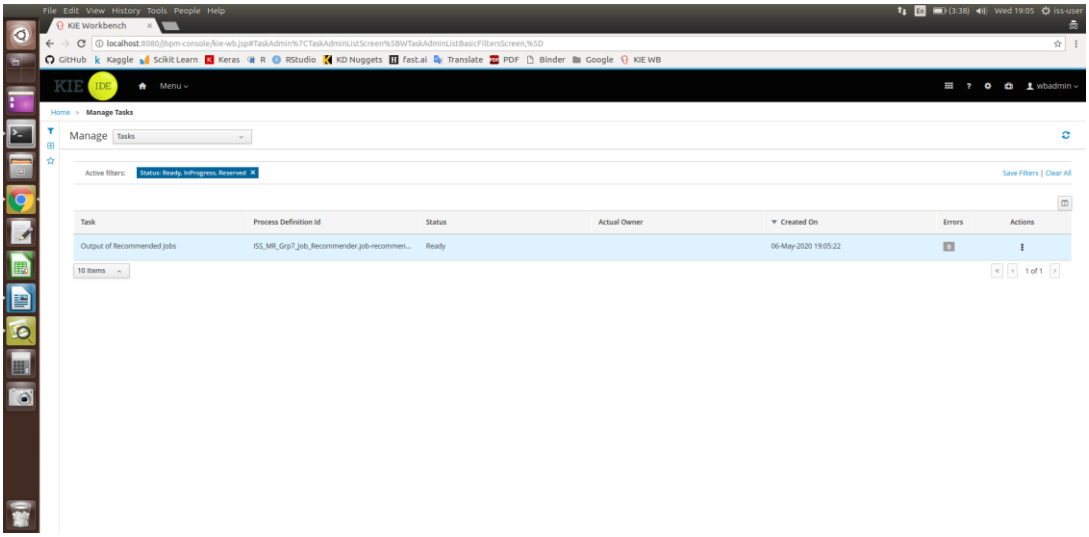
7. Click on “Start”



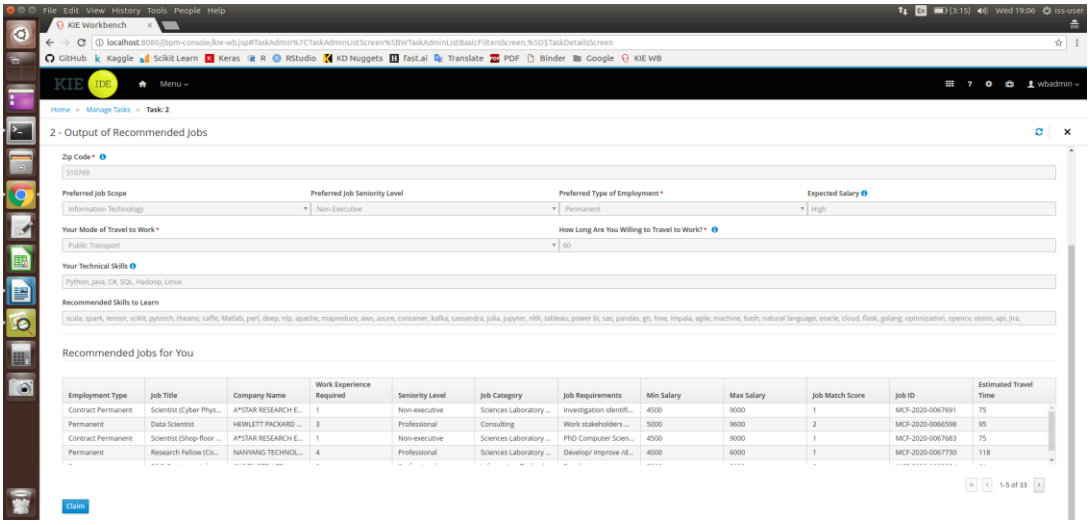
8. Click on the “Zip Code” box, which will cause a drop-down menu to appear. Select one of three possible ZIP codes. Once done, click on “Complete”



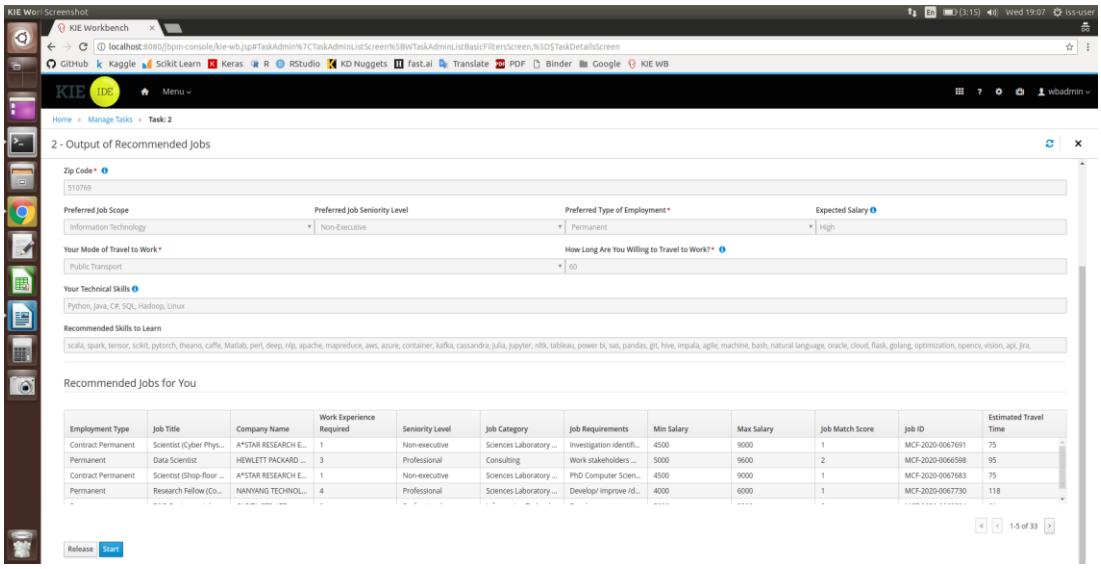
9. Click on “Menu”, then click on “Task”. This will cause the following page to appear. To continue, click on the task “Output of Recommended Jobs”



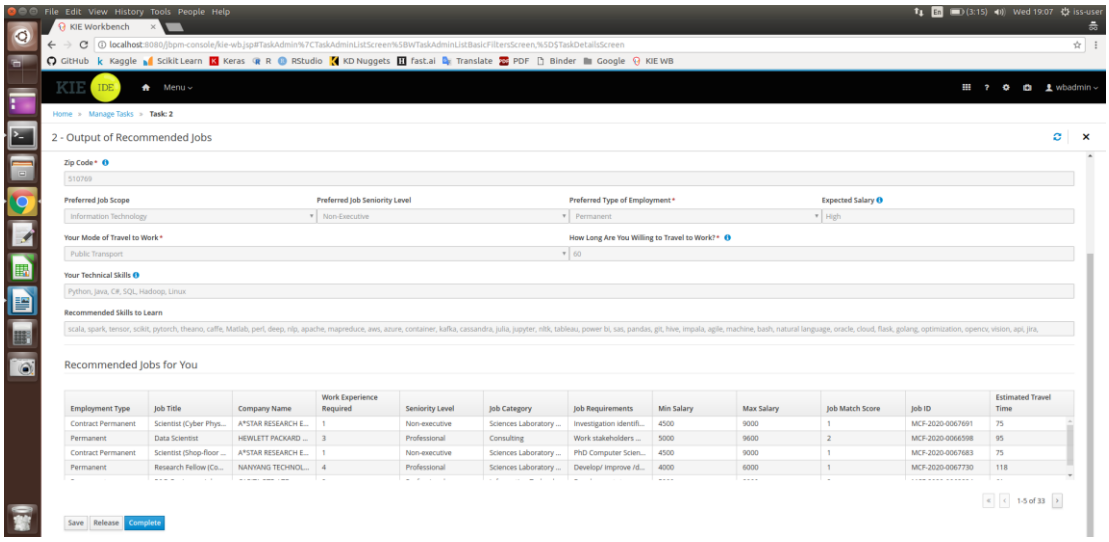
10. Press the “Claim” button at the below of the page



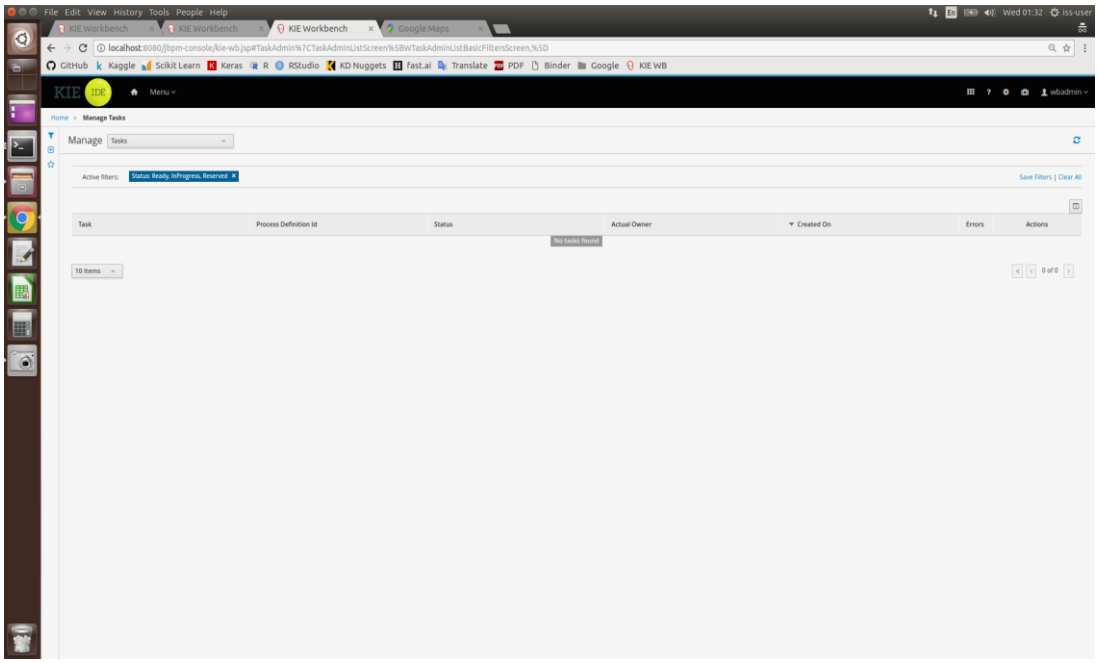
11. Press “Start”



12. After reviewing the output of recommended jobs, press “Complete” to end the task instance.



13. Process completed



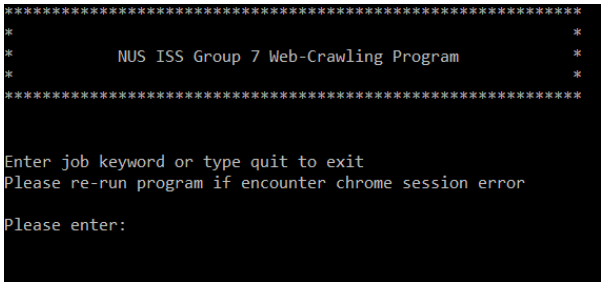
Running Data Mining algorithm in Python

Requirements:

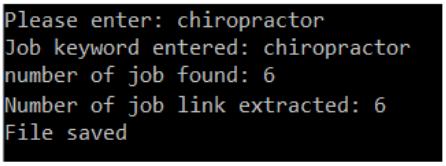
- Anaconda3
- Jupyter Lab
- Google Chrome browser version 81.0.4044
- Chromedriver version 81.0.4044
    - Saved to (\SystemCode\Data Mining\code\chromedriver)
- Anaconda Python3 environment and Library
    - Selenium
    - BeautifulSoup4
    - Scikit-learn

Web Crawling (only online mode)

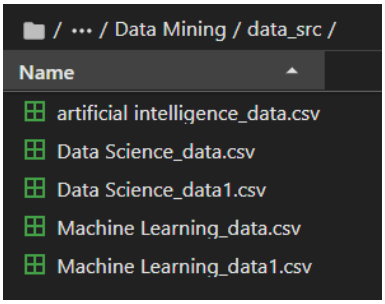
1. Open anaconda3 prompt
2. Navigate to project source code folder (..\SystemCode\Data Mining\code)
3. Type python “Web Crawling.py” to run program



4. Enter job search keyword or type quit to exit the program
5. Job crawled will be saved inside data_src folder



6. If encounter chrome session error, please re-run the program
7. Web-crawled data will be used for data mining and machine learning part.



8. Please take note popular job keyword might have few hundreds of records. Web-crawling will need few hours to crawl all records.

9. If Job not found, user will have to key in with another keyword.

```
Enter job keyword or type quit to exit
Please re-run program if encounter chrome session error

Please enter: Astronaut
Job keyword entered: Astronaut
Job not found, please try with other keyword
Job not found, please re-run
```

Data Mining (Offline)

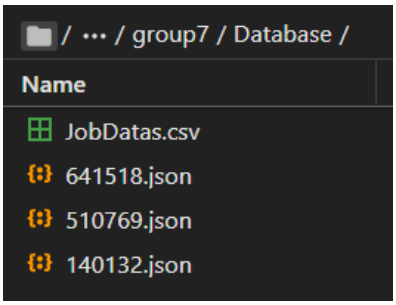
1. Open anaconda3 prompt
2. Navigate to project source code folder (..\SystemCode\Data Mining\code)
3. Type python “Data Mining.py” to run program
4. Program will ingest all crawled file inside the data_src folder to perform data mining process.

```
************************************************************
*                                                          *
*            NUS ISS Group 7 Data Mining Program           *
*                                                          *
************************************************************


Source file(s) loaded: ['../data_src\\artificial intelligence_data.csv', '../data_src\\Data
Science_data.csv', '../data_src\\Data Science_data1.csv', '../data_src\\Machine Learning_dat
a.csv', '../data_src\\Machine Learning_data1.csv']


Data ingestion and cleaning: Complete
Salary feature: Complete
Skills and qualification extraction: Complete
Number of words before filter: 68149
Number of words after filter: 45953
File saved
```

5. The program will generate an csv file named “JobDatas.csv” to the Database folder



6. The “JobDatas.csv” file will be consumed by the KIE program

Machine Learning (Offline)

1. Open anaconda3 prompt.
2. Navigate to project source code folder (..\SystemCode\Data Mining\code)
3. Type python “Machine Learning.py” to run program.
4. Program will consume the crawled data from data_src folder, process and predict the salary range.

```
**********************************************************
*                                                        *
*            NUS ISS Group 7 Salary Prediction ML        *
*                                                        *
**********************************************************


Data ingestion and cleaning: Complete
Feature engineering: Complete


Salary prediction: Complete
```

5. Model prediction report (classification report, confusion matrix and feature importance) will be generated.

```
classficication report:

              precision    recall  f1-score   support

         Low       0.81      0.97      0.88       231
         Med       0.72      0.62      0.67        37
        High       0.80      0.44      0.57        89

    accuracy                           0.80       357
   macro avg       0.78      0.68      0.71       357
weighted avg       0.80      0.80      0.78       357



confusion matrix:

            Pred Low  Pred Med  Pred High
Actual Low       224         3          4
Actual Med         8        23          6
Actual High       44         6         39
```

```
Feature Importances

                        coef  abs coef
Year_Experience     0.433057  0.433057
java scala          0.093971  0.093971
engineering         0.076283  0.076283
phd                 0.067935  0.067935
design              0.050304  0.050304
statistical         0.047727  0.047727
architecture        0.047501  0.047501
Executive           0.043291  0.043291
master              0.043003  0.043003
perl                0.023512  0.023512
implementations     0.021084  0.021084
optimization        0.019473  0.019473
automation          0.017714  0.017714
python              0.015145  0.015145
```

6. Decision tree rule will be generated and saved as .txt file inside output folder (\Data Mining\output). Figure below shows the example of tree rules generated.

```
Decision tree rules:
|--- Year_Experience <= 3.50
|   |--- java scala <= 0.50
|   |   |--- phd <= 0.50
|   |   |   |--- Year_Experience <= 1.50
|   |   |   |   |--- Executive <= 0.50
|   |   |   |   |   |--- class: Low
|   |   |   |   |--- Executive >  0.50
|   |   |   |   |   |--- class: Med
|   |   |   |--- Year_Experience >  1.50
|   |   |   |   |--- architecture <= 0.50
|   |   |   |   |   |--- class: High
|   |   |   |   |--- architecture >  0.50
|   |   |   |   |   |--- class: Med
|   |   |--- phd >  0.50
|   |   |   |--- engineering <= 0.50
|   |   |   |   |--- statistical <= 0.50
|   |   |   |   |   |--- class: Med
|   |   |   |   |--- statistical >  0.50
|   |   |   |   |   |--- class: High
|   |   |   |--- engineering >  0.50
|   |   |   |   |--- Year_Experience <= 1.50
|   |   |   |   |   |--- class: High
|   |   |   |   |--- Year_Experience >  1.50
|   |   |   |   |   |--- class: High
|   |--- java scala >  0.50
|   |   |--- class: High
```

Appendix C: Individual Member Report

Leonard Loh Kin Yung (A0213553M)

My personal contribution to the project mainly involved building the Smart Job Recommender System in the KIE Workbench. This included developing the applicant forms, rules as well as the job suitability scoring algorithm. I was also responsible for integrating the job database CSV file produced by the web crawler and Google Maps API with the system.

The most useful takeaway from the project is that it taught me how to integrate the back-end system with a front-end API. In the past, most of my projects simply involved developing back-end programming, with the code being run using an IDE. Thus, it was a great opportunity to learn front-end API tools such as KIE that can be used to develop an integrated process system. The knowledge and techniques imparted from the course also played an important role to allow me to develop the system, these included decision rules and score-based heuristic evaluation.

Moving forward, the knowledge and experience I have gained during the course of this project can be applied in situations where A.I can be used to replace manual and tedious tasks to make faster and smarter decisions. One such case would be a job scheduling system, where knowledge-based reasoning and optimization techniques can automatically schedule jobs in an efficient way that fully utilizes the available resources. This is important as Industry 4.0 is increasingly seeping into the industry, where process automation is starting to replace human decision making.

Daniel Tan Hoong Xiang (A0074608B)

My contribution to the team is to help to validate the KIE workbench, coding logic in a bigger picture, the data's and to check the logic of the codes whether it is tallied towards our goal of executing this project, business side of the project. Since our data mining and applicant input and output forms are a different system, the consistency between the two systems must be monitored closely. The KIE workbench codes must be validated with various situations and logic also, to make sure the codes are well designed. Therefore, my role is more like system integration, administrative and system validation side.

The techniques, knowledge and skills that I have learned are the integration part of the system. KIE workbench is a new environment for me to use, for normally I will use an IDE environment to debug the codes line by line, but because KIE integrated various parts in the system, it has posed a challenge for me to read and even execute the codes. However, I have started to get a hang of it and am beginning to troubleshoot the codes by segment with the assistance of Eclipse. Another part of it will be the business side of it, there is who should be our main customer and consumer, who should we develop a logic which is realistically useful for the user and executable codes given our circumstances of limited time.

This project has basically exposed me to how to integrate the knowledge or rules-based system, resource optimization techniques like brute force search and knowledge discovery using python. At the end of the day, I believe that automation such as our job recommender system or other automation should become the next trend and the next industrial revolution which will impact our lives in a not very distant future.

Aaron Kueh Hee Kheng (A0213552N)

My part is mainly on knowledge discovery and building knowledge database. The database was obtained through web-crawling process. By utilizing web automation tool such as selenium, thousands of job vacancy can be crawled and stored automatically.

I also responsible for designing the data mining flow and process. Few algorithms have been developed to perform data ingestion, data cleaning and data mining using python programming language. I also need to make sure the data mined and processed by the algorithms are clean for KIE workbench to consume without any error.

Another main part of my contribution is using machine learning to predict salary range. NLP technique was used to turn job requirement column into predictor by using tokenizer and countvectorizer. I also applied feature engineering technique to extract and engineered useful information from the job vacancy detail as predictors. I also spent time in using scikit-learn decision tree library to build prediction model and few rounds of iteration to make sure the model can obtain high accuracy performance before output the prediction into decision tree rules.

This project has taught me the challenging part of real-life data, where developing data cleaning algorithm can consume half of my time in this project. The noise present in real life data is unpredictable and can be in any forms, example typo error, wrong information given inside the job detail or missing information. This has motivated me to design a scalable or generalize algorithm which can be re-used for different condition. By parameterize most of the changeable parameters, can ease the code development iteration frequency.

References

1. Craig S (2020). 22 Interesting LinkedIn Recruiting Statistics (2020) | By the Numbers
2. Singapore Unemployment Rate. CEIC Data. Retrieved from: https://www.ceicdata.com/en/indicator/singapore/unemployment-rate
3. Ovais S (2020). Singapore economy could be headed for its worst-ever contraction this year. The Straits Times. Retrieved from: https://www.straitstimes.com/business/economy/singapore-economy-shrinks-22-in-q1-full-year-growth-forecast-slashed-to-4-to-1
4. Yu H, Liu C, Zhang F (2011). Reciprocal Recommendation Algorithm for the Field of Recruitment. J. Inform. Comput. Sci. 8(16):4061-4068.