SKILL AND JOB RECOMMENDER

A PROJECT REPORT

Submitted By

JAYASHREE R	210819205016
RANJITHA A	210819205039
KAVIYARASI R	210819205024
RAKSHINI DEVI M	210819205037

TEAM ID : PNT2022TMID25506

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

KINGS ENGINEERING COLLEGE, IRUNGATTUKOTAI

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that mini project report **"Skill and Job Recommender"** is bonafide work of **"JAYASHREE R, RANJITHA A, KAVIYARASI R, RAKSHINI DEVI M"** who carried out this Nalaiyathiran project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.G. MANIKANDAN	Mrs.K.SARANYA
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Associate Professor
Dept of Information Technology,	Dept of Information Technology
Kings Engineering College,	Kings Engineering College,
Irungattukottai,	Irungattukottai,
Chennai-602 117	Chennai-602 117.

ACKNOWLEDGEMENT

We thank God for his unstoppable blessings and also for giving as good knowledge and strength in enabling us to finish our project. Our deep gratitude goes to our founder late **Dr. D. SELVARAJ, M.A., M.Phil.,** for his patronage in the completion of our project. We like to take this opportunity to thank our honourable chairperson **Dr.S. NALINI SELVARAJ, M.COM., MPhil., Ph.D.** and honourable director, **MR.S. AMIRTHARAJ, M.Tech., M.B.A** for their support given to us to finish our project successfully. We wish to express our sincere thanks to our beloved principal. **Dr.T.JOHN ORAL BASKAR., M.E.,Ph.D** for his kind encouragement and his interest towards us.

We are extremely grateful and thanks to our professor **Dr.G.MANIKANDAN,** head of Information Technology, Kings EngineeringCollege, for his valuable suggestion, guidance and encouragement and to our SPOC **DR. MRS.PARAMASHWARI,** head of Artificial Intelligence and Data Science. We wish to express our sense of gratitude to our project supervisor **Mrs.K.SARANYA,** Assistant Professor of Information Technology Department, Kings Engineering College and project evaluator **Mr. B. MUTHAZAGHAN**, Associate Professor of Information Technology Department, Kings Engineering College whose idea and direction made our project a grand success, also our Industrial Mentor **KRISHNA CHAITANYA.** We express our sincere thanks to our parents, friends and staff members who have helped and encouraged us during the entire course of completing this project work successfully.

TABLE OF THE CONTENTS

CHAPTER	CONTENTS	PAGE NO
1	**INTRODUCTION**	
	1.1 Project overview	1
	1.2 Purpose	2
2	**LITERATURE SURVEY**	
	2.1 Existing problems	3
	2.2 References	4
3	**IDEATION AND PROPOSED SOLUTION**	
	3.1 Empathy map	6
	3.2 Ideation and Brainstorming	7
	3.3 Proposed solution	8
	3.4 Problem solution fit	9

4	**REQUIREMENT ANALYSIS**	
	4.1 Functional requirement	11
	4.2 Non-functional requirements	11
5	**PROJECT DESIGN**	
	5.1 Data flow diagrams	12
	5.2 Solution & technical architecture	14
	5.3 User stories	15
6	**PROJECT PLANNING & SCHEDULING**	
	6.1 Sprint planning & estimation	15
	6.2 Sprint delivery schedule	16
	6.3 Reports from jira	16
7	**CODING & SOLUTIONING**	
	7.1 Feature 1	17
	7.2 Feature 2	20
8	**TESTING**	
	8.1 Test cases	20
	8.2 User acceptance testing	21
9	**RESULTS**	
	9.1 Performance metrics	23
10	**ADVANTAGES & DISADVANTAGES**	24
11	**CONCLUSION**	25

12	**FUTURE SCOPE**	26
13	**APPENDIX**	29
	Source code & project demo link	

1. INTRODUCTION

The fast growth of the Internet caused a matching growth of the amount of available online information that increased the need to expand the ability of users to manage all this information. This encourages a substantial interest in specific research fields and technoˇlogies that could benefit the managing of this information overload. The most important fields are Information retrieval and Information filtering. Information retrieval deals with automatically matching user‟s information and Information filtering aims to assist users eliminating unwanted information (Hanani et al., 2001). The latest technology designed to fight information overload is the recommender systems that originated from cognitive science, approximation theory, information retrieval, forecasting theories and also related to management science and to consumer choice modeling in marketing (Adomavicius and Tuzhilin, 2005). The recommender systems used to determine the interested items for a specific user by employing a variety of information resources that is related to users and items. (Adomavicius and Tuzhilin, 2005).

1.1. PROJECT OVERVIEW

In the mid-1990s, the term recommender system was published for the first time in information system literature (Resnick and Varian, 1997). Many researches in industry and academic areas have been known to develop new approaches for recommender systems in the last decade. The interest in this area still remains high because it is composed of a problem-rich research area and has a wealth of practical applications (Adomavicius and Tuzhilin, 2005). Recommender systems are being broadly accepted in various applications to suggest products, services, and information items to latent customers. Many e-commerce applications join recommender systems in order to expand customer services, increase selling rates

and decrease customers search time (Schafer et al., 1999). For example, a wide range of companies such as the online book retailer Amazon.com (Linden et al., 2003), books (Mooney and Roy, 2000), and news articles (Das et al., 2007). Additionally, Microsoft provides users many recommendations such as the free download products, bug fixes and so forth (Shani and Gunawardana, 2011). All these companies have successfully set up commercial recommender systems and have increased web sales and improved customer fidelity. Moreover, many software developers provide stand-alone generic recommendation technologies. The top providers include Net Perceptions, Epiphany, Art Technology Group, Broad Vision, and Blue Martini Software (Huang et al., 2007). For many years, information system supports in human resource management have been mainly restricted in storing and tracking applicants‟ data through the appli cant management systems. These systems support the internal workflows and communication processes between the human resource management department and the other departments. Recently, the increased amount of digital information and the emergence of eˇbusiness reform the way companies conduct business in different aspects. Initially, simple solutions are applied such as posting the job ads on the career unit of the corporate website. Then, based on the experiences gained from these first implementations, the opportunities are realized, establishing other changes and hence, implementing enhanced e-recruitment platforms. The Internet-based online recruiting platform or eˇrecruitment platform is one of the most successfulˇbusiness changes, which changed the way companies employ candidates. These platforms spread in the recent years because the recruiting of the appropriate person is a challenge faced by most companies, as well as the unavailability of certain candidates in some skill areas has long been identified as a major obstacle to companies success (Laumer and Eckhardt, 2010). The online channels like Internet job portal, social media applications or a firm‟s career

website have driven this development. While the companies established job positions on these portals, job-seeker uses them to publish their profiles. For each posted job, thousands of resumes are received by companies. Consequently, a huge volume of job descriptions and candidate resumes are becoming available online. This vast volume of information gives a great opportunity for enhancing the matching quality; this potential is unused since search functionality in recruiting applications is mainly restricted to Boolean search method. The need increases for applying the recommender system technologies that can help recruiters to handle this information efficiently (Färber et al., 2003; Yi et al., 2007).

1.2 PURPOSE

The significance of Information System (IS) support in the recruitment process can be observed when considering the phases of the recruitment such as the handling of candidates‟ applications and the pre-selection of candidates. However, a best fit between job and candidates depends on underlying aspects that are hard to measure. These underlying aspects are a significant reason why information systems have not been extensively used in the area of personnel selection so far. Mostly, IS technology is used to pre-select applicants based on Boolean search method. This method used queries contain a combination of key words that define skill requirements in order to determine those candidates that match with search criteria. Such type of skill matching is applied in numerous e-recruiting

applications. However, as mentioned above, the simple filter techniques such as Boolean search method cannot be sufficient to realize the complexity of a person-job fit as selection decisions often depend on underlying attributes such as personal characteristics or social skills that cannot be put into an operational way easily (Malinowski et al., 2006). Additionally, the need to understand the job requirements, in terms of the skills that are mandatory and those that are optional but preferable, the experience criteria if any, preference for the location of the candidate etc. Consequently, the major challenge faced e-recruiting applications as identified by the literature analysis is the large number of low qualification of applicants that match the search criteria (Singh et al., 2010). The recommender systems techniques can be used to address the problem of information overload by prioritize the delivery of information for individual users based on their learned preferences (Lee and Brusilovsky, 2007). Additionally, the success of personalization technologies depends critically on the existence of comprehensive user profiles that precisely capture users‟ interests (Rafter and Smyth, 2001) and the perfect matching method. Moreover, the recommender systems could use historical rating information to determine which type of job required which type of candidate characteristics in the past in order to be rated positively by the recruiter. This information could then be used to predict the match between job and previously not rated candidates.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Having lots of skills but wondering which job will best suit you? Don’t need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job.

2.2 REFERENCES

Abd Elaziz, M., Xiong, S., Jayasena, K., Li, L., 2019. Task scheduling in cloud

computing based on hybrid moth search algorithm and differential evolution.

Knowl.-Based Syst. 169, 39–52.Abdelmaboud, A., Jawawi, D.N., Ghani, I., Elsafi, A., Kitchenham, B., 2015. Quality of service approaches in cloud computing: A systematic mapping study. J. Syst.Softw. 101, 159–179.Abedi, M., Pourkiani, M., 2020. Resource allocation in combined fog-cloud scenarios by using artificial intelligence. In: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, pp. 218–222. Adhikari, M., Nandy, S., Amgoth, T., 2019. Meta heuristic-based task deployment mechanism for load balancing in iaas cloud. J. Netw. Comput. Appl. 128, 64–77. Al-maamari, A., Omara, F.A., 2015. Task scheduling using pso algorithm in cloud computing environments. Int. J. Grid Distrib. Comput. 8 (5), 245–256. Alemnesh, G., 2020. Time Optimized Hybrid Scheduling Algorithm for Cloud Computing Environment. Ph.D. thesis. ASTU.

Alhaidari, F., Balharith, T.Z., 2021. Enhanced round-robin algorithm in the cloud

computing environment for optimal task scheduling. Computers 10, 63.

Ali, J., Zafari, F., Khan, G.M., Mahmud, S.A., 2013. Future clients' requests estimation for dynamic resource allocation in cloud data center using cgpann. In: 2013 12th International Conference on Machine Learning and Applications. IEEE, pp.331–334.Alkayal, E.S., Jennings, N.R., Abulkhair, M.F., 2016. Efficienttaskscheduling multiobjective particle swarm optimization in cloud computing. In: 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops). IEEE, pp. 17–24. Alkhateeb, F., Abed-alguni, B.H., Al-rousan, M.H., 2021. Discrete hybrid cuckoo search and simulated annealing algorithm for solving the job shop scheduling problem. J. Supercomput. 78 (4), 4799–4826.Allahverdi, A., 2015. The third comprehensive survey on scheduling problems with setup times/costs. Eur. J. Oper. Res.

246 (2), 345–378.Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y., 2008. A survey of scheduling

problems with setup times or costs. Eur. J. Oper. Res. 187 (3), 985–1032.

Alworafi, M.A., Dhari, A., El-Booz, S.A., Nasr, A.A., Arpitha, A., Mallappa, S., 2019. An enhanced task scheduling in cloud computing based on hybrid approach. In:Data Analytics and Learning. Springer, pp. 11–25.Ananth, A., Chandrasekaran, K., 2015. Cooperative game theoretic approach for job scheduling in cloud computing. In: 2015 International Conference on Computing and Network Communications (CoCoNet). IEEE, pp. 147–156. Ardagna, D., Casale, G., Ciavotta, M., Pérez, J.F., Wang, W., 2014. Qualityof-service in cloud computing: modeling techniques and their applications. J. Internet Serv. Appl. 5, 1–17.

Aslam, S., Shah, M.A., 2015. Load balancing algorithms in cloud computing: A survey of modern techniques. In: 2015 National software engineering conference

(NSEC). IEEE, pp. 30–35.Bagheri, M.H., Bagherizadeh, M., Moradi, M., Moaiyeri, M.H., 2021. Design of cntfetbased current-mode multi-input m: 3 (4m 7) counters. IETE J. Res. 67, 322–332. Belgacem, A., Beghdad-Bey, K., 2021. Multi-objective workflow scheduling in cloud computing: trade-off between makespan and cost. Clust. Comput. 25 (1), 579–595. Benny, R., Wirawan, I., 2022. Comparison analysis of round robin algorithm with highest response ratio next algorithm for job scheduling problems. Int. J. Open Inf. Technol. 10, 21–26. Bezdan, T.,Zivkovic, M., Bacanin, N., Strumberger, I., Tuba, E., Tuba, M., 2022. Multiobjective task scheduling in cloud computing environment by hybridized bat algorithm. J. Intell. Fuzzy Syst. 42 (1), 411–423. Chen, X., Long, D., 2019. Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm. Clust. Comput. 22 (S2),2761–2769. Cheng, F., Huang, Y., Tanpure, B., Sawalani, P., Cheng, L., Liu, C., 2022. Cost-aware job scheduling for cloud instances using deep reinforcement learning. Clust. Comput. 25 (1), 619–631.Chien, W.-C., Lai, C.-F., Chao, H.-C., 2019. Dynamic resource prediction and

allocation in c-ran with edge artificial intelligence. IEEE Trans. Ind. Inf. 15 (7),

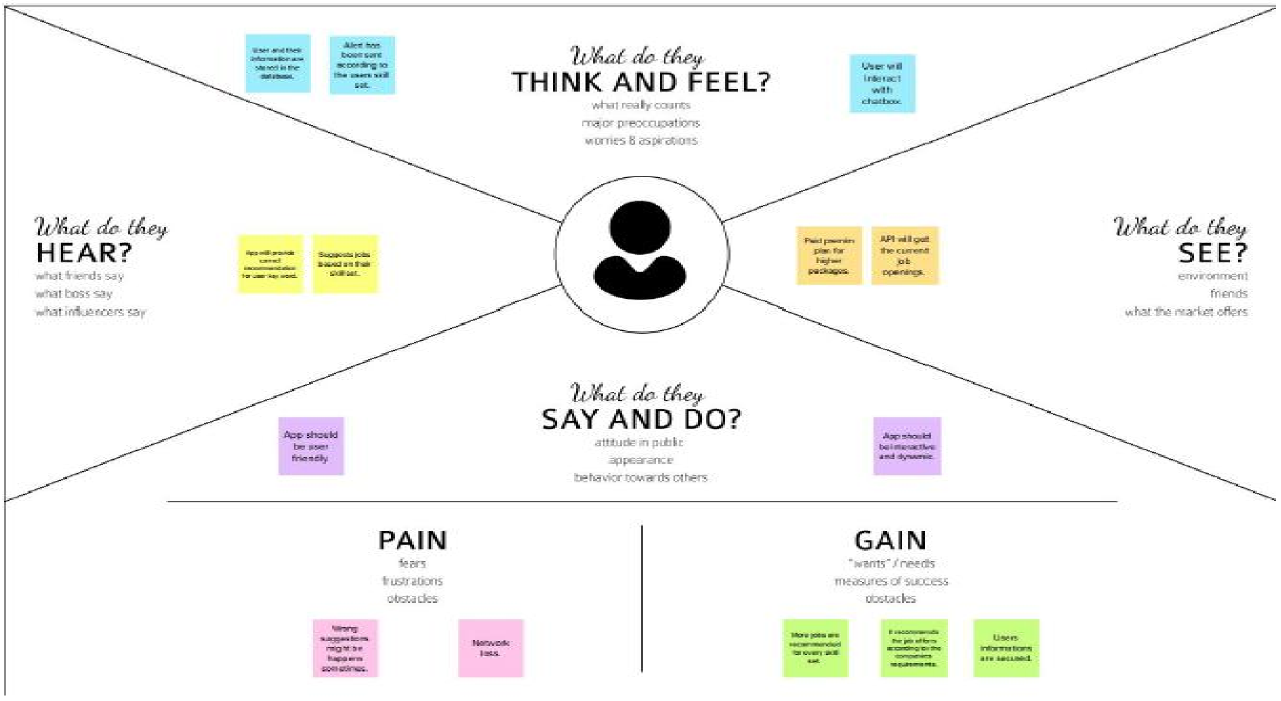
4306–4314. Cobham, A., 1954. Priority assignment in waiting line problems. J. Oper. Res. Soc. Am. 2 (1), 70–76. Saydul Akbar Murad, Abu Jafar Md Muzahid, Zafril Rizal M Azmi et al. Journal of King Saud University – Computer and Information Sciences 34 (2022) 2309–2331

2.3 PROBLEM STATEMENT DEFINITION

To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.

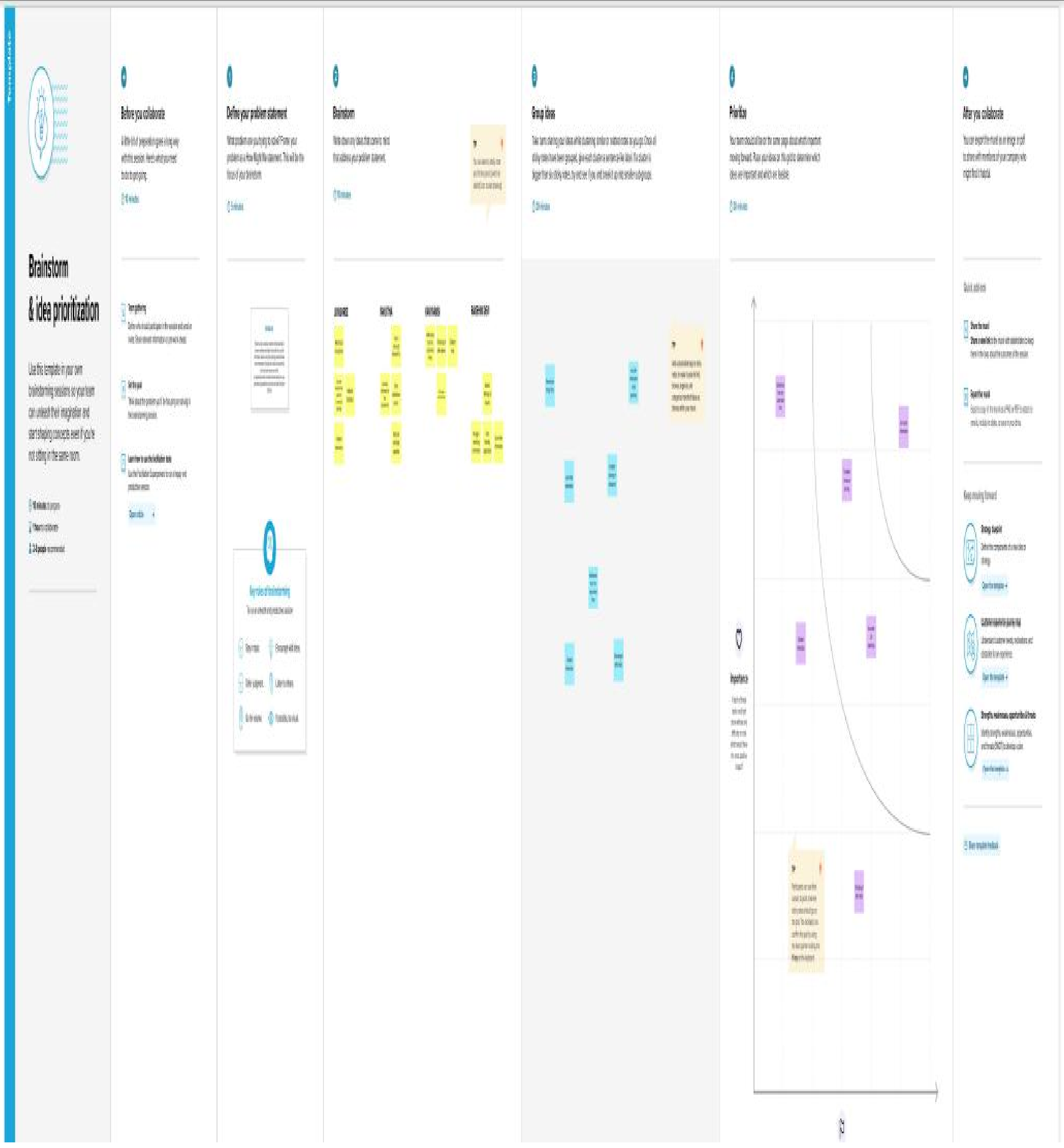
3. IDEATION &PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
User and their information are stored in the database.
Alert has been sent according to the users skill set.
User will interact with chatbox
What do they HEAR?
what friends say
what boss say
what influencers say
App will provide correct recommendation for user key word.
Suggests jobs based on their skillset.
What do they SEE?
environment
friends
what the market offers
Paid premium plan for higher packages.
API will get the current job openings.
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
App should be user friendly.
App should be interactive and dynamic.
PAIN
fears
frustrations
obstacles
Wrong suggestions might be happens sometimes.
Network loss.
GAIN
"wants" / needs
measures of success
obstacles
More jobs are recommended for every skill set.
It recommends the job offers according to the companies requirements.
Users informations are secured.

3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

Description

1.Problem Statement (Problem to be solved)

There may be a massive collection of job openings in various websites and might not provide the accurate information about current job openings and skill based recommendations.Having lots of skills but wondering which job suits as per your skills.

2.Idea / Solution description

This application will give accurate results as possible and provides instant

solution for the customer problems in addition to that , customers can interact

with the chatbot which gives relevant solution to the customer problems

Furthermore, this skill/job recommender application provides up to date

information of job openings of the firms.

3.Novelty / Uniqueness

Our uniqueness is all users can find the jobs according to their skills sets which may have any kind of qualifications.Customers will be treated same for every skills there is no partiality by their skills or qualifications , jobs according to their every kind of skills will be recommended by our app.And also we don't have any premium or subscription plans in our app it is same for all the users.

4.Social Impact / Customer Satisfaction

After launching of this application many peoples burden will reduce to seek job according to their skills. This will create huge benefit for customers to find the correct job which also suits for their skill sets and also according to their interests.

5.Business Model (Revenue Model)

The Internet-based recruiting platforms become a primary recruitment channel in most companies. While such platforms decrease the recruitment time and advertisement cost, they suffer from an inappropriateness of traditional information retrieval techniques like the Boolean search methods. Consequently, a vast amount of candidates missed the opportunity of recruiting. The recommender system

technology aims to help users in finding items that match their personnel interests; it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently. In order to improve the e-recruiting functionality, many recommender system approaches have been proposed.

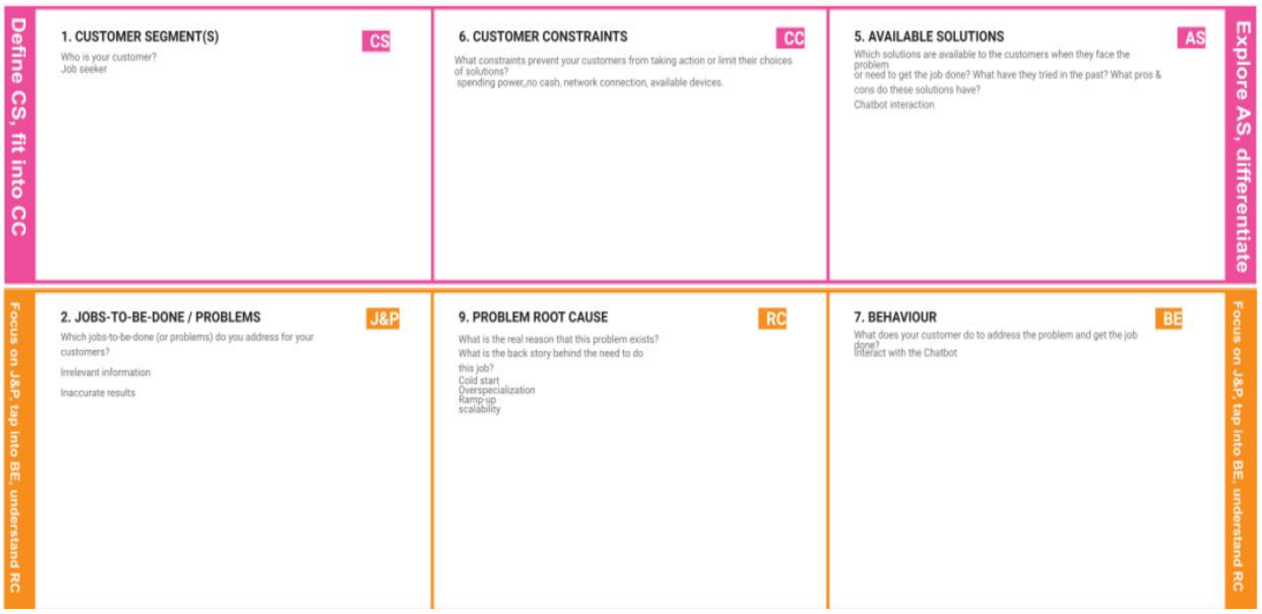
6.Scalability of the Solution

• Our application takes less time to respond.

• Our application will response many requests simultaneously without making any server crashes.

• So this will ensure our application will scalable.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
Job seeker

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?
spending power,,no cash, network connection, available devices.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have?
Chatbot interaction

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?

Irrelevant information

Inaccurate results

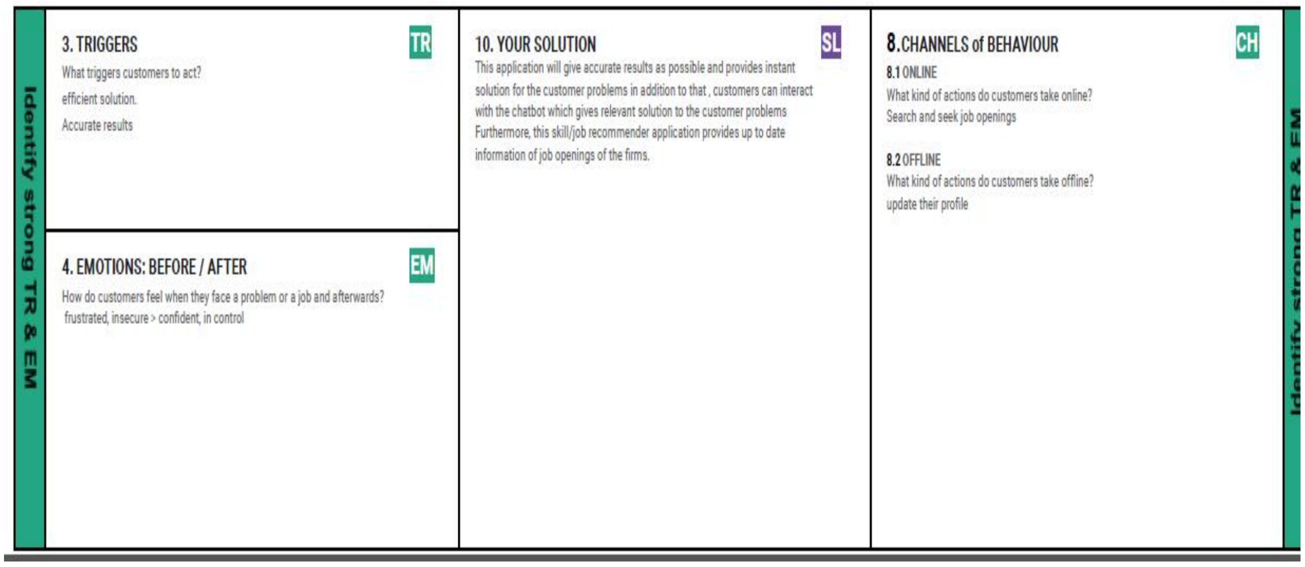
9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
Cold start
Overspecialization
Ramp-up
scalability

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
Interact with the Chatbot

Focus on J&P, tap into BE, understand RC



Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act?

efficient solution.

Accurate results

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
frustrated, insecure > confident, in control

10. YOUR SOLUTION SL

This application will give accurate results as possible and provides instant solution for the customer problems in addition to that , customers can interact with the chatbot which gives relevant solution to the customer problems Furthermore, this skill/job recommender application provides up to date information of job openings of the firms.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online?
Search and seek job openings

8.2 OFFLINE

What kind of actions do customers take offline?
update their profile

Identify strong TR & EM

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR-1 User Registration -Registration through Form Registration through Gmail Registration through LinkedIN

FR-2 User Confirmation- Confirmation via Email Confirmation via OTP

FR-3 User login - User should login the with the user name and password

FR-4 User installation - User can install the app from Google play store or Apple store or directly from the website.

NON-FUNCTIONAL REQUIREMENT

NFR-1 Usability - Everyone can easily install and use from the play store by using the instructions in the app.

NFR-2 Security - The application is very secure and confidential.

NFR-3 Performance - The app can work fast and more reliable.

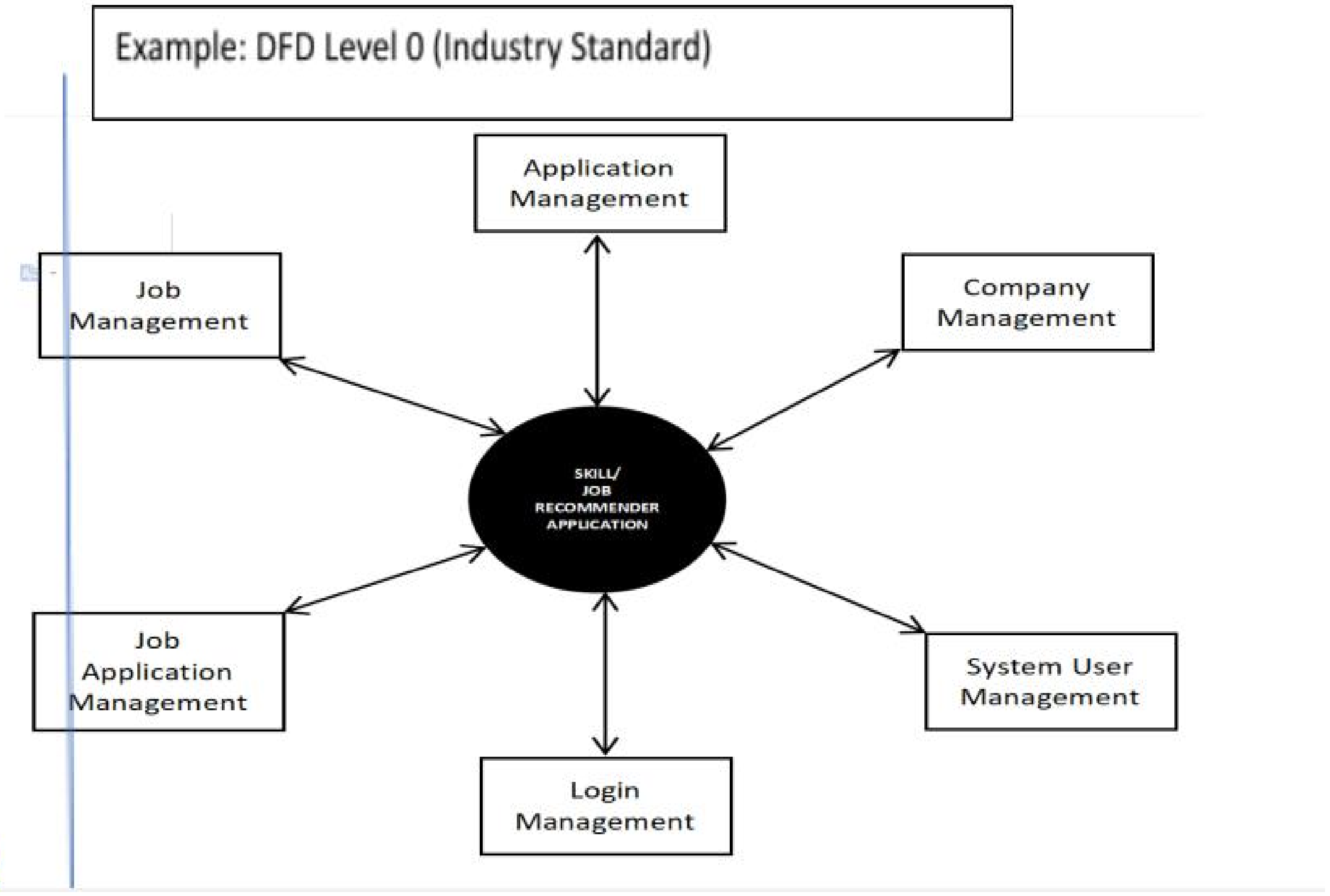
NFR-4 Availability - It is available in all kind of play stores and more categories and jobs are recommended.

NFR-5 Scalability - Our application takes less time to response many requests simultaneously without making any server crashes,so this will ensure our application will scalable.

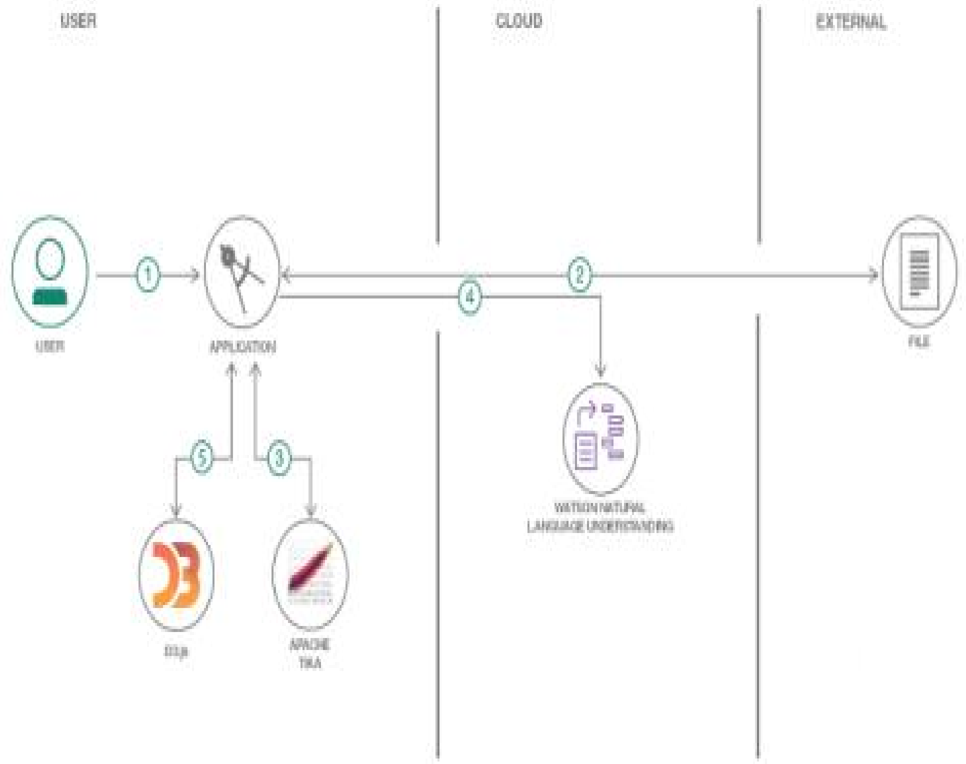
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

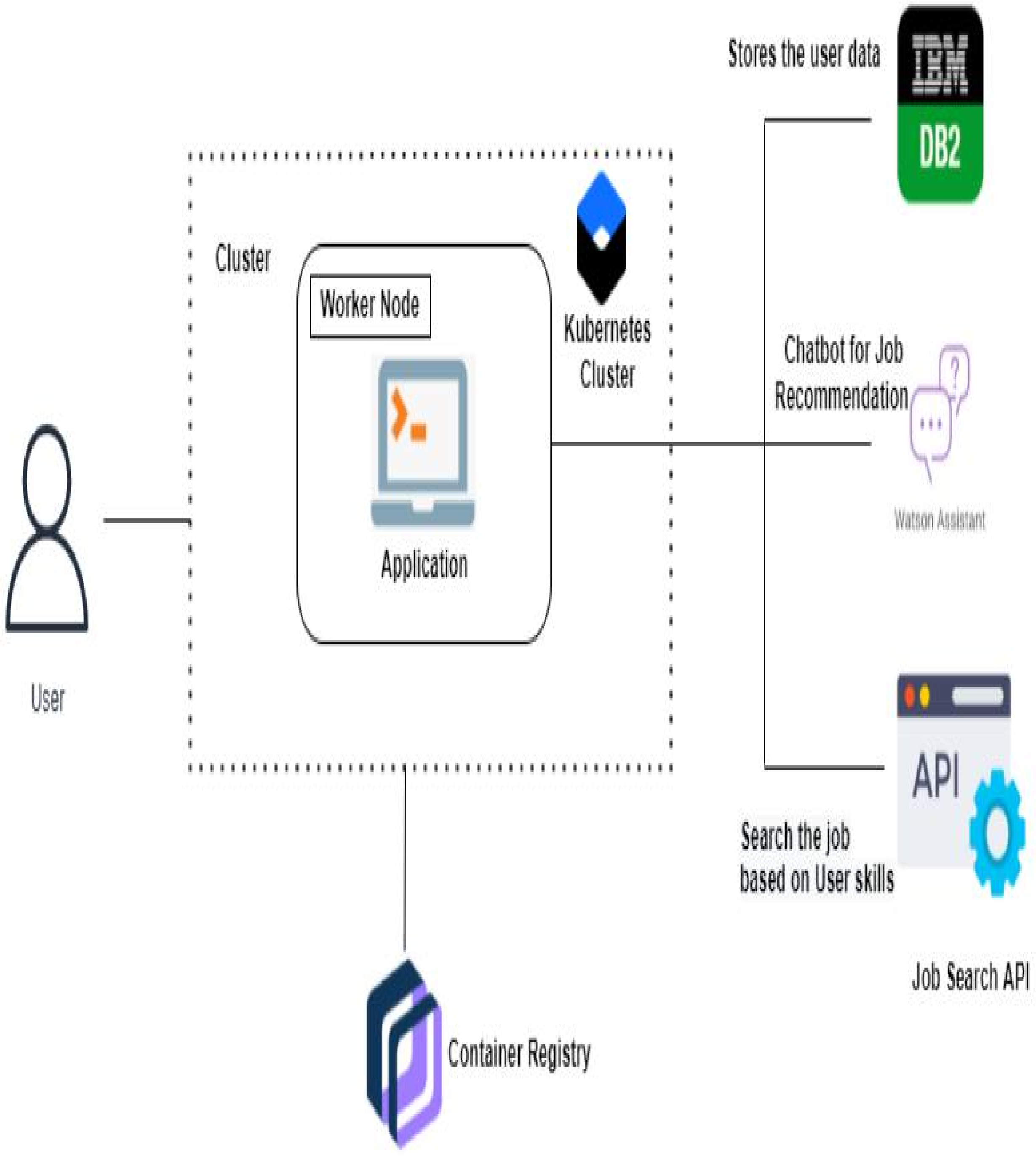
Data Flow Diagrams: A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


Example: DFD Level 0 (Industry Standard)
Application Management
Job Management
Company Management
SKILL/ JOB RECOMMENDER APPLICATION
Job Application Management
System User Management
Login Management

Flow


USER
CLOUD
EXTERNAL
1
2
3
4
5
USER
APPLICATION
FILE
WATSON NATURAL
LANGUAGE UNDERSTANDING
D3.js
APACHE
TIKA

5.2 SOLUTION & TECHNICAL ARCHITECTURE

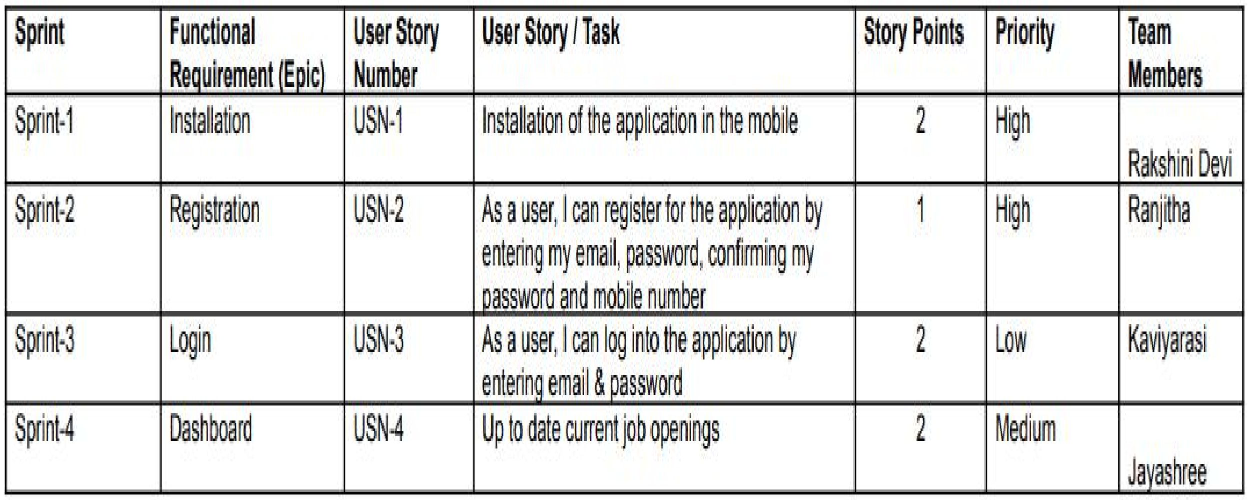


5.3 USER STORIES

- Installation of the application in the mobile
- As a user, I can register for the application by entering my email, password, confirming my password and mobile number
- As a user, I can log into the application by entering email & password
- Up to date current job openings

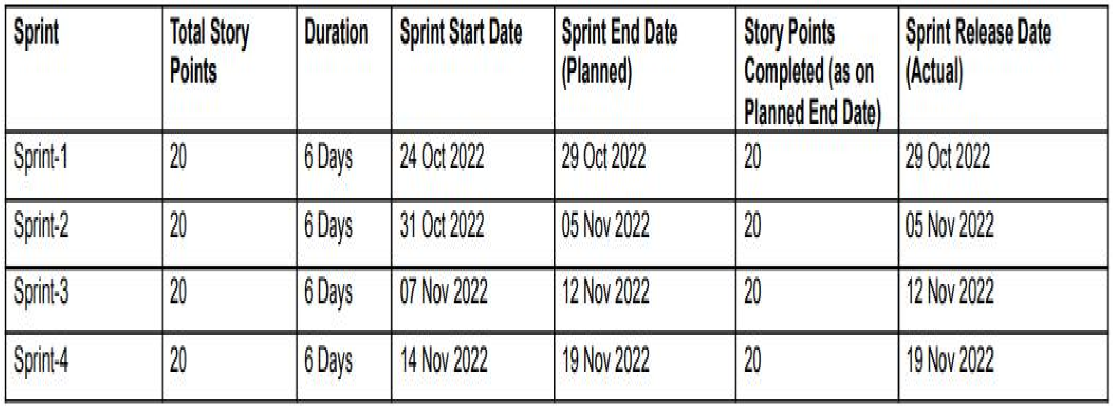
6.PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION



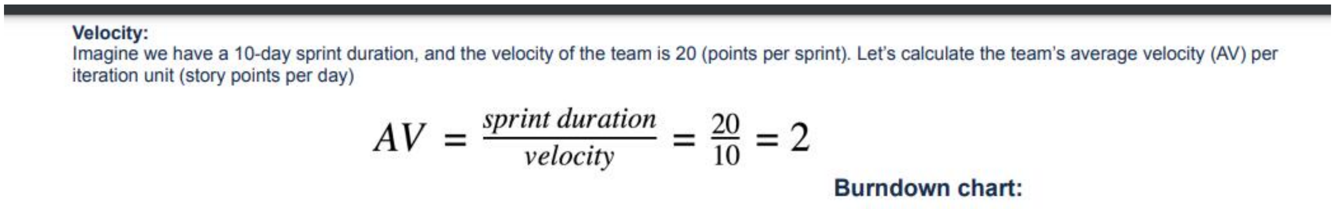
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Installation	USN-1	Installation of the application in the mobile	2	High	Rakshini Devi
Sprint-2	Registration	USN-2	As a user, I can register for the application by entering my email, password, confirming my password and mobile number	1	High	Ranjitha
Sprint-3	Login	USN-3	As a user, I can log into the application by entering email & password	2	Low	Kaviyarasi
Sprint-4	Dashboard	USN-4	Up to date current job openings	2	Medium	Jayashree

6.2 SPRINT DELIVERY SCHEDULE



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

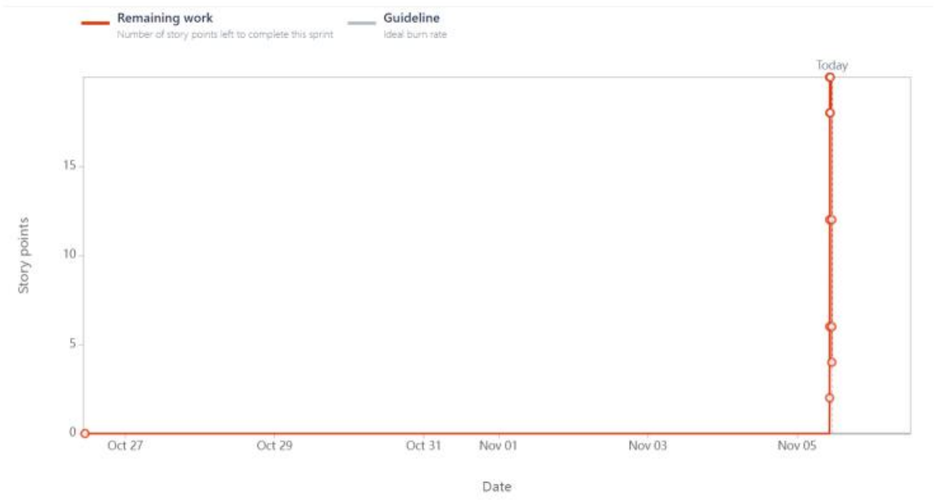
6.3 REPORTS FROM JIRA



Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown chart:



7. CODIND AND SOLUTIONING

7.1 FEATURE 1

```
{% extends 'main.html' %}

{% load static%}

{% block content %}

<div class="container card">

<h1>Available openings</h1>

<div class="card-body">

<table class="table table-hover">

<tr>

<th>Name</th>

<th>Position</th>

<th>Description</th>

<th>Salary</th>

<th>Experience</th>

<th>Location</th>

<th>Join</th>

</tr>

{% for c in companies %}

<tr>
```

```
<td>{{c.name}}</td>

<td>{{c.position}}</td>

<td>{{c.description}}</td>

<td>{{c.salary}}</td>

<td>{{c.experience}}</td>

<td>{{c.Location}}</td>

<td><a href="{% url 'apply' %}" class="btn btn-info btn-sm"
type="submit">Apply</a></td>

</tr>

{% endfor %}

</table>

</div>

</div>

{% for c in companies %}

{{c.name}}

{% endfor %}

{% endblock %}
```

7.2 FEATURE 2

```
from    django.shortcuts         import render,redirect
from.models   import*
from    django.contrib.auth.forms     import UserCreationForm
from    django.contrib.auth    import login,logout,authenticate
from.forms      import*
#Createyourviewshere.
Def     home(request):
   if     request.user.is_authenticated:

candidates=Candidates.objects.filter(company__name=request.user.userna
me)
         context={
            'candidates':candidates,
         }
         return

            render(request,'hr.html',context)
   else:
         companies=Company.objects.all()

            context={
            'companies':companies,
         }
         return

            render(request,'Jobseeker.html',context)
def      logoutUser(request):
   logout(request)
   return
```

```
         redirect('login')

def      loginUser(request):
   if     request.user.is_authenticated:
        return

         redirect('home')
   else:
       if  request.method=="POST":
        name=request.POST.get('username')

         pwd=request.POST.get('password')
        user=authenticate(request,username=name,password=pwd)


        if user     is        not       None:
           login(request,user)
           return

         redirect('home')
       return

         render(request,'login.html')
def      registerUser(request):
   if     request.user.is_authenticated:
        return

         redirect('home')
   else:
        Form=UserCreationForm()
        if request.method=='POST':
           Form=UserCreationForm(request.POST)
```

```
          if         Form.is_valid():
             currUser=Form.save()

         Company.objects.create(user=currUser,name=currUser.username)
             return          redirect('login')
        context={
          'form':Form

          }
        return

         render(request,'register.html',context)
def      applyPage(request):
    form=ApplyForm()
    ifrequest.method=='POST':
        form=ApplyForm(request.POST,request.FILES)


        if  form.is_valid():
          form.save()
          return

         redirect('home')
    context={'form':form}
    return render(request,'apply.html',context)
```

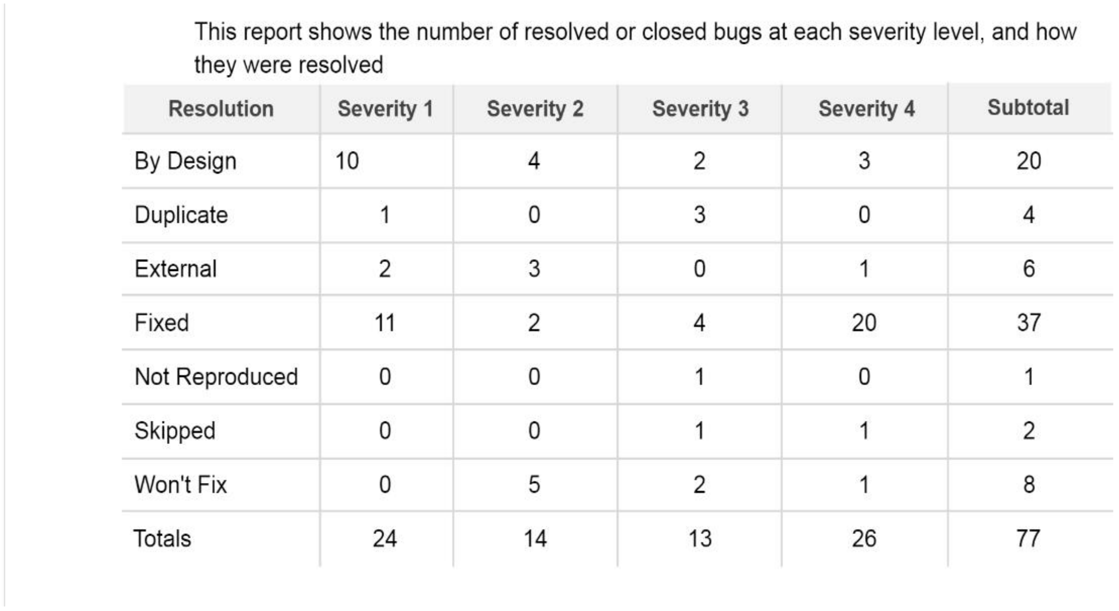
8. TESTING

8.1 TEST CASES

- Logged in and verified
- Storing web pages(web app)
- Sent grid app is tested for response

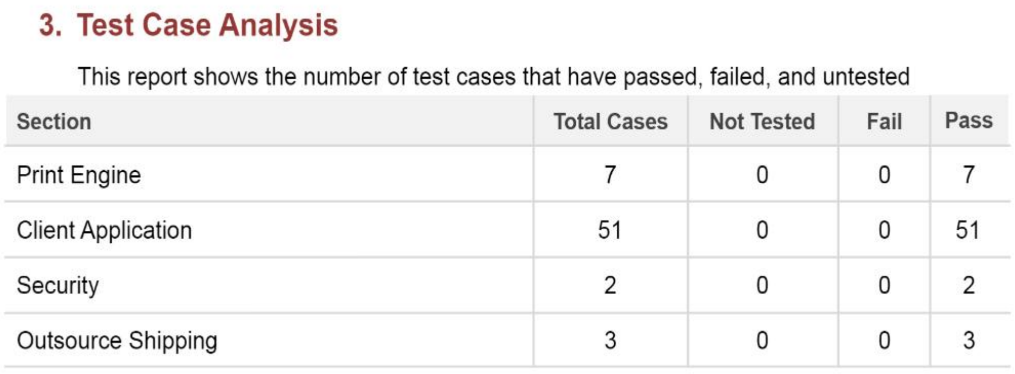
- Data access rate is calculated in DB2
- DB connection is ensured
- Cluster is generated as multiple copies
- Submitting the required details
- Submitting the empoyee details

8.2 USER ACCEPTANCE TESTING



This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

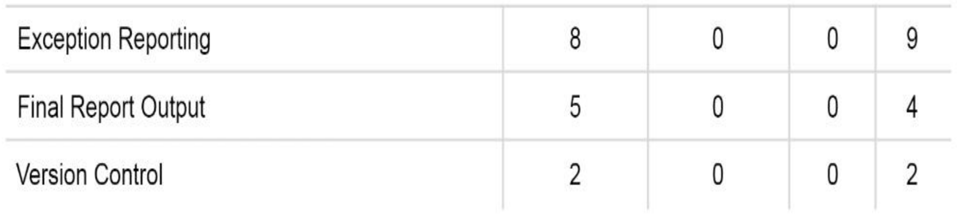
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77



3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

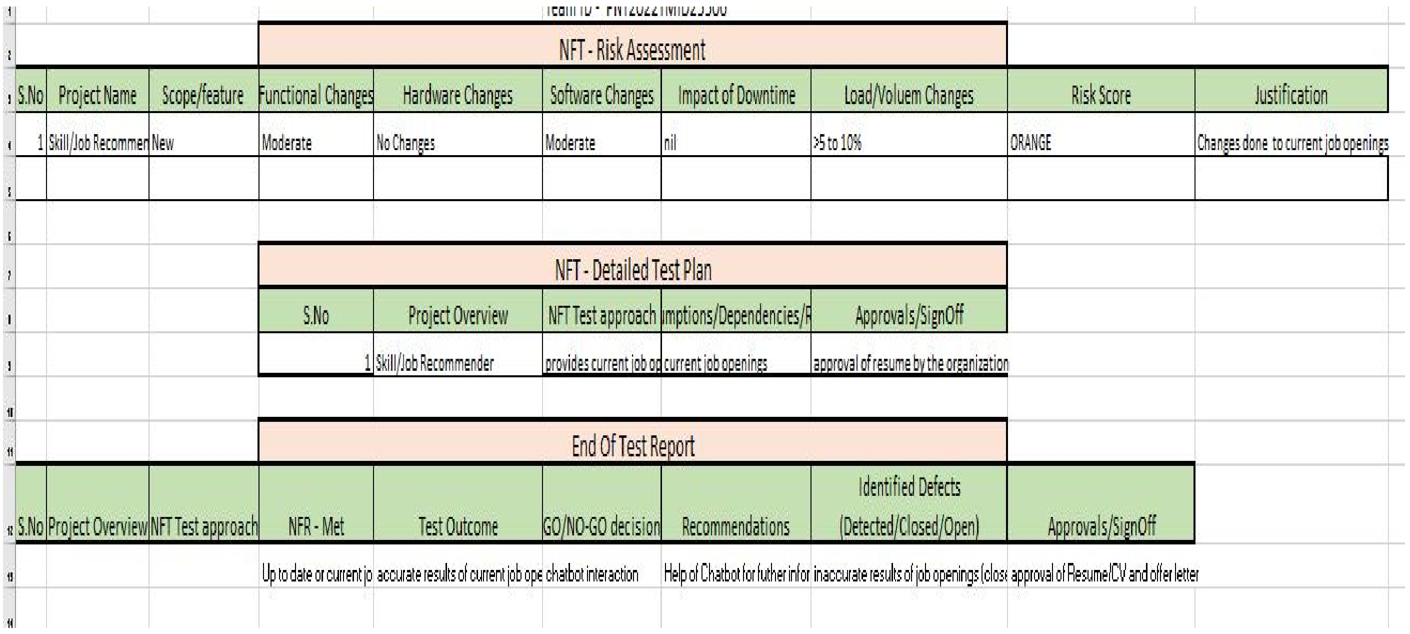
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3



Exception Reporting	8	0	0	9
Final Report Output	5	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS



NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Skill/Job Recommen	New	Moderate	No Changes	Moderate	nil	>5 to 10%	ORANGE	Changes done to current job openings

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	mptions/Dependencies/R	Approvals/SignOff
1	Skill/Job Recommender	provides current job op	current job openings	approval of resume by the organization

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
			Up to date or current jo	accurate results of current job ope	chatbot interaction	Help of Chatbot for futher infor	inaccurate results of job openings (close	approval of Resume/CV and offer letter

10.ADVANTAGES & DISADVANTAGES

ADVANATGES

- Bidirectional recommendation.
- Relational aspects are included.
- Adaptive system.
- Use many attributes.
- Use ontology to categorize jobs and as a knowledge base to define features (attenuate cold-start problem).
- Bidirectional recommendation.
- Effective matching methods.
- Includes many attributes.
- Relational aspects are included.
- Qualitative and quantity representation (proficiency level for skills is included).
- Use two levels in skills matching (constrains and preferences).
- Use many attributes.
- Relational aspects are included.
- Effective matching methods.
- Use linguistic variables to determine skill levels.
- Use many attributes.
- Various information retrieval techniques are used.
- Constrains used to eliminate candidates before ranking

DISADVANTAGES

- Binary representation only.
- Less attributes used.

- No perfect measures.
- Key words search method.
- One way recommendation.
- Knowledge acquisition and knowledge engineering problems.
- No relational aspects are included
- Knowledge acquisition and Knowledge engineering problems.
- Tools and technologies skills excluded
- One way recommendation.
- One way recommendation.
- No relational aspects are included.
- Scalability, ramp-up, and data sparsity problems.

11.CONCLUSION

In this paper, Content-Based Filtering and Collaborative Filtering of recommendations have been compared. Additionally,an aggregation plus recommender system has been devised. Content-Based Filtering recommends the results based on matching the personal preferences of the user with the given document whereas collaborative filtering recommends based on the preferences of fellow users. On evaluating both of these methods, it was concluded that a hybrid system of both of these overcomes the limitations of both of them and increases the efficiency of ranking. Problems of cold start, sparse database, scalability, and lack of trend recommendation [5] have been eliminated. +e proposal is to design a Job Recommender system that prioritizes quality over quantity. While there are websites and job listing portals already recommending jobs to job seekers based on their profiles, this research on aggregate quality recommendations has been achieved by crawling selectively, overcoming the limitations of [1, 4, 14]. A fully functioning user interface was developed to combine everything together to give the user a seamless experience.

12. FUTURE SCOPE

For this system to be hybrid, content-based filtering is required, which can only recommend jobs based on the user's current profile. It cannot deliver anything surprising based on the user's past searches. +is paper also uses collaborative filtering which faces well-known problems of privacy breaches and cold start. +e system has a broad scope that can be used to make it more robust and foolproof.Firstly, automating the crawling process is required, when a new company is added to the database. In other words, removing the one-time configuration step/process to fetch jobs of a particular new company can be done. +ese models can implement techniques such as KNN in collaborative filtering.

Implementing NLP in content-based filtering for better and more accurate search matching can be done. Along with this, testing and collecting more user data for better performance of the collaborative filtering module is required. Lastly, improving the cleansing process of the job description and using natural language processing are required. While using collaborative filtering, this work can be improved by giving different weights to different users based on their LinkedIn skills.

13. APPENDIX

SOURCE CODE

```
from    django.shortcuts        import render,redirect
from.models    import*
from    django.contrib.auth.forms

import UserCreationForm
from    django.contrib.auth     import login,logout,authenticate
from.forms      import*

# Create your views here.
```

```
defhome(request):
   if     request.user.is_authenticated:

candidates=Candidates.objects.filter(company__name=request.user.userna
me)
      context={
         'candidates':candidates,
      }
   return

         render(request,'hr.html',context)
   else:
      companies=Company.objects.all()
      context={
         'companies':companies,
      }
      return

         render(request,'Jobseeker.html',context)

def      logoutUser(request):
   logout(request)
   return

         redirect('login')



def

         loginUser(request):
   if     request.user.is_authenticated:
      return
```

```
            redirect('home')
    else:
       if  request.method=="POST":
        name=request.POST.get('username')
        pwd=request.POST.get('password')
        user=authenticate(request,username=name,password=pwd)

        if user     is        not       None:
            login(request,user)
            return

            redirect('home')
       return

            render(request,'login.html')
def       registerUser(request):
    if     request.user.is_authenticated:
        return redirect('home')

else:
        Form=UserCreationForm()
        if request.method=='POST':
            Form=UserCreationForm(request.POST)

            if         Form.is_valid():
                currUser=Form.save()

Company.objects.create(user=currUser,name=currUser.username)
                return

            redirect('login')
        context={
            'form':Form
    }
    return

        render(request,'register.html',context)
defapplyPage(request):
  form=ApplyForm()
  if     request.method=='POST':
    form=ApplyForm(request.POST,request.FILES)

    if form.is_valid():
      form.save()
      return

            redirect('home')
        context={'form':form}
        return render(request,'apply.html',context)
```

APPENDIX

GITHUB & PROJECT DEMO LINK

GITHUB LINK : https://github.com/IBM-EPBL/IBM-Project-28005-1660105565

PROJECT DEMO LINK : https://youtu.be/5qMhllvFgV0