K. J. SOMAIYA COLLEGE OF SCIENCE AND COMMERCE

Re-accredited with 'A' Grade by NAAC
Vidyanagar, Vidyavihar, Mumbai 400 077

DEPARTMENT OF INFORMATION TECHNOLOGY

PROJECT CERTIFICATE FOR B.Sc. (I.T.) STUDENTS

2020 – 2021

This is to certify- that the project entitled

'CareerIn'

(Job Reccommendation System)

Undertaken by

Aishwarya Jamdar

in fulfilment for B.Sc I.T. Degree(Semester 6) Examination has been completed by him/her. This project has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

This is to further certify that he/she has completed all required phases of the project.

Acknowledgement

Before we get into the thick of things we would like to add a few heartfelt words for the people who were part of this project in numerous ways, people who gave unending support right from the stage the project idea was conceived. A project report is such a comprehensive coverage; it would not have been materialized without the help of many.

I am privileged to express our sense of gratitude to our respected teacher Prof. PRAJAKTA KAMBLE whose unparalleled knowledge, moral fiber and judgment along with her know-how, was an immense support in completing the project.

A sincere quote of thanks to the non-teaching staff for providing us books and with all the information we needed for this project, without which the successful completion of this project would not have been possible. I appreciate the outstanding cooperation by the non-teaching staff, especially for the long lab timing that I could receive.

Last but not least I wish to avail myself of this opportunity, express a sense of gratitude and love to my friends and my beloved parents for their manual support, strength and help for everything.

Thanking You

Table of Contents

Chapter 1: Introduction

1.1BACKGROUND:

- CareerIn is a smart Job recommendation system based on Web Crawler or spider built on Python Django Framework
- When it comes to job/internship searches, we frequently subscribe and make accounts on various websites, but we lack a worldwide reach to all of the chances available on the internet. CareerIn gives you chances right to your doorstep and keeps you updated about employment that match your qualifications.
- The fundamental idea is to utilise web crawlers to index and search for perfect job possibilities for you based on your resume.

1.2 OBJECTIVE:

a)Permissive Job searches:

Our job /internship searching web crawler seeks to simplify the job/internship searching process. With this web application, gone are the days in which students spend hours scouring the internet . Instead, the searching process will be expedited because our application allows the user to simply upload their PDF/Docx resume and a ranked list of job/internship opportunities will be a click away.

b)Security:

Users are not required to create an account on any of the job-search websites, and their account information is not shared with third-party applications.

c)Keeps you updated :

Every time a user looks for employment possibilities, the most up-to-date information is retrieved, and the user may store the positions that are recommended to them for future use.

1.3 PURPOSE, SCOPE, AND APPLICABILITY

1. Purpose

- The main purpose of this webapp is the user can access the website from home or any other different place to search for employment possibilities.
- We plan to crawl popular job searching websites such linkedin,intershala,indeed.com etc in search of their internship opportunity listings based on the resume provided by the user.

2. Scope

- This webapp is created to keep users informed about employment possibilities based on their resumes without having to visit each separate website.
- Users do not need to fill out any information; all they need to do is upload their CV and they are set to begin their careers.

3. Applicability

- Implemented system has higher flexibility i.e. it can be used anytime and anywhere by the user.
- The webApp is accessible everywhere with internet connection and user can save the data.
- The application will be developed keeping in mind the applicability.The information that will be provided will be verified with the respective authorities. If there are any changes or the information needs to be updated,the team will first verify and then it will be updated accordingly.

Chapter 2:
Survey of Technologies

Resumes are a great example of unstructured data. Each resume has its unique style of formatting, has its own data blocks, and has many forms of data formatting. This makes reading resumes hard, programmatically. Recruiters spend ample amount of time going through the resumes and selecting the ones that are a good fit for their jobs. Tech giants like Google and Facebook receive thousands of resumes each day for various job positions and recruiters cannot go through each and every resume. This is why Resume Parsers are a great deal for people like them. Resume Parsers make it easy to select the perfect resume from the bunch of resumes received.

For making a resume parser we followed following steps:

First Step: Reading the Resume

Resumes do not have a fixed file format, and hence they can be in any file format such as .pdf or .doc or .docx. So our main challenge is to read the resume and convert it to plain text. For this we can use two Python modules: pdfminer and doc2text. These modules help extract text from .pdf and .doc, .docx file formats.

Second Step: Extracting Name

For extracting names from resumes, we can make use of regular expressions. But we will use a more sophisticated tool called spaCy. Spacy is a Industrial-Strength Natural Language Processing module used for text and language processing. It comes with pre-trained models for tagging, parsing and entity recognition. Our main moto here is to use Entity Recognition for extracting names.

Third Step: Extracting Phone Numbers

For extracting phone numbers, we will be making use of regular expressions which means we will be using re library. Phone numbers also have multiple forms such as (+91) 1234567890 or +911234567890 or +91 123 456 7890 or +91 1234567890. Hence, we need to define a generic regular expression that can match all similar combinations of phone numbers.

Forth Step: Extracting Email

For extracting Email IDs from resume, we have used a similar approach that we used for extracting mobile numbers. Email IDs have a fixed form i.e. an alphanumeric string should follow a @ symbol, again followed by a string, followed by a . (dot) and a string at the end. We can use regular expression to extract such expression from text.

Fifth Step: Extracting Skills

. We can extract skills using a technique called tokenization. Tokenization simply is *breaking down* of text into paragraphs, paragraphs into sentences, sentences into words. Hence, there are two major techniques of tokenization: Sentence Tokenization and Word Tokenization.

Before implementing tokenization, we will have to create a dataset against which we can compare the skills in a particular resume. For this we will make a comma separated values file (.csv) with desired skillsets. For example, if I am the recruiter and I am looking for a candidate with skills including NLP, ML, AI then I can make a csv file with contents:

Assuming we gave the above file, a name as skills.csv, we can move further to tokenize our extracted text and compare the skills against the ones in skills.csv file. For reading csv file, we will be using the pandas module. After reading the file, we will removing all the stop words from our resume text.

Sixth Step: Extracting Education:

Now, moving towards the last step of our resume parser, we will be extracting the candidates education details. The details that we will be specifically extracting are the degree and the year of passing. For example, XYZ has completed MS in 2018, then we will be extracting a tuple like ('MS', '2018'). For this we will be requiring to discard all the stop words. We will be using nltk module to load an entire list of stopwords and later on discard those from our resume text.

For making web crawler :

We will build a web crawler in Python. We plan to crawl company pages in search of their internship opportunity listings. The libraries that we have considered are Scrapy and BeautifulSoup. Although Scrapy is a powerful web framework that allows for the extraction, processing, and storing of web data, it has a larger learning curve than BeautifulSoup which in comparison, is slower unless multiprocessing is used. As a result, we will consider testing both libraries to see which one is the best fit for our crawler.

Chapter 3 :
Requirements and Analysis

3.1 PROBLEM DEFINITION :

- When people looked for employment or internships on the internet, they had to go to different websites each time. There's also the possibility of missing out on next internships and career chances.
- We don't have an application that combines all of the job possibilities together in one place.
- Many websites ask you to manually enter your hobbies and experience. However, this issue is no longer an issue because users just need to upload their resumes.

3.2REQUIREMENTS SPECIFICATION:

3.2.1 FunctionalRequirements:

User:

- The user shall be able to login to the website.
- The user shall be able to navigate from one page to another.
- The user shall be able to locate different categories.
- The user shall be able to get the output for a search query.

3.2.2 Non-Functional Requirements:

- Availability

The website should be made available 24 hours per week.

- Security

The website should never disclose the user's personal information.

- Maintainability

The website should be closed once in 6 months for maintenance and elaboration.

- Reliability: The website should never crash more than 10 minutes per month.
- Design Constraints: The website design should give an attractive look.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

● Hardware requirements :
A functional laptop or mobile with at least 1 following browser available:

Sr no	Browser	Version
1)	chrome	4.3.6 and high
2)	Mozilla firefox	92.0.1
3)	Edge	91.0
4)	Opera	65.3.3 and high

●Software requirements
1. HTML,css
2. Python (ver-3.6 and high)
3. system pre installled with liberaries given below:

Sr.no	Library /Packages	version
1)	django	3.2.7
2)	pillow	8.3.2
3)	Scrapy	2.5.0
4)	beautifulSoup	4.9.0
5)	Pylucene	4.0
6)	pdfminer	1.1.2
7)	Spacy	v3.0
8)	nltk	3.6.3
9)	Other machine learning liberaries	

4. Windows 7 or more

3.4 PRELIMINARY PRODUCT DESCRIPTION :

- users have to create the profile on our webapplication by entering some basic information such as email,name.password and contact number.
- The first major requirement user have to fulfil in order to access all the functionalities is to simply upload their resume and the industry they are in
- Users can access all the functionality after login.
- after uploading the resume user have to review the extracted data which will further be used in employment search process.
- once everthing is upto date user can simply click on the search button and the ranked list of internship will be visible to their screen.
- The users can save the desired internships for future use on their account and access the internships from companies websites itself .

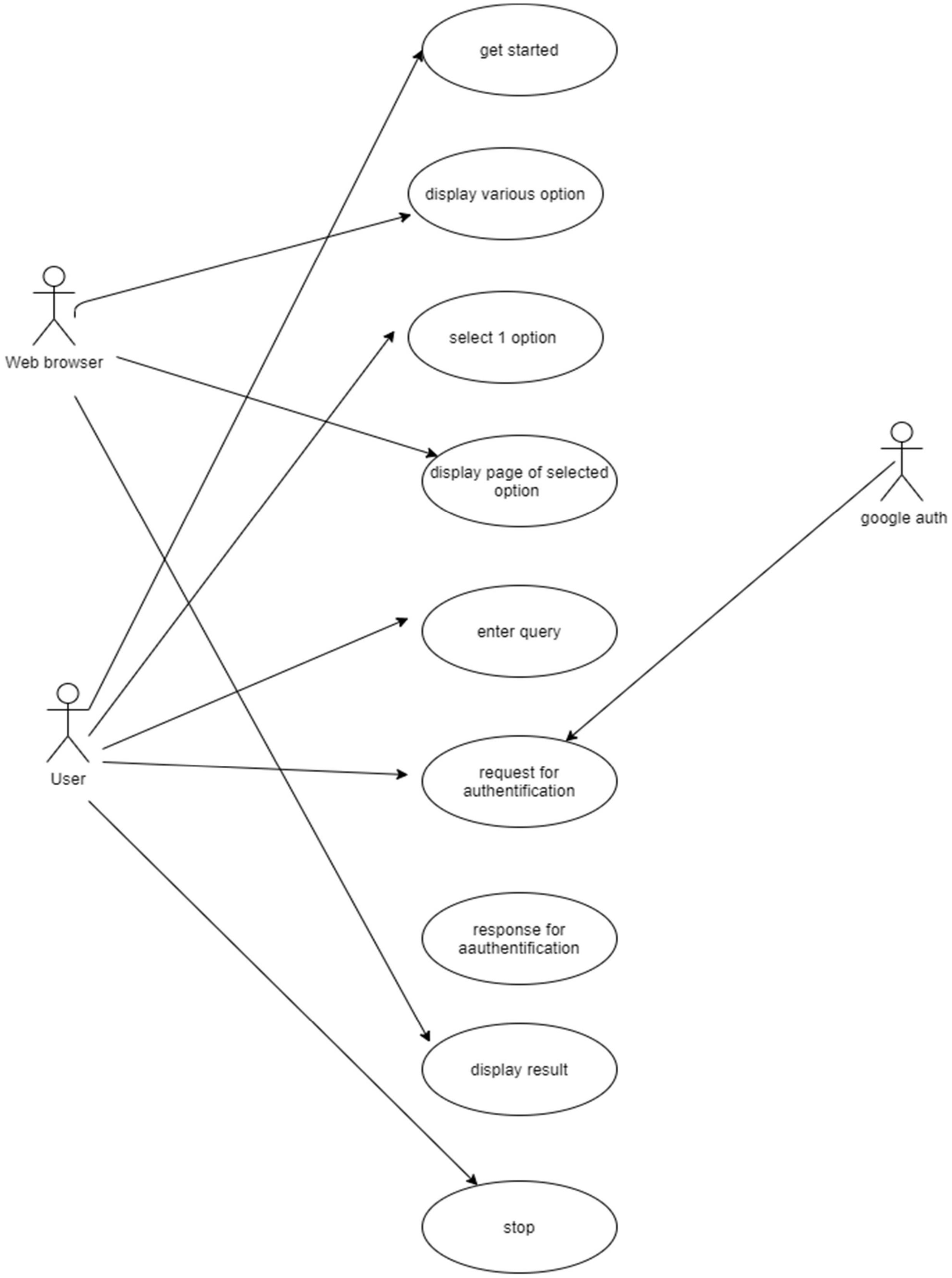
3.5 PLANNING AND SCHEDULING

Fig 3.5.1 Gantt Chart

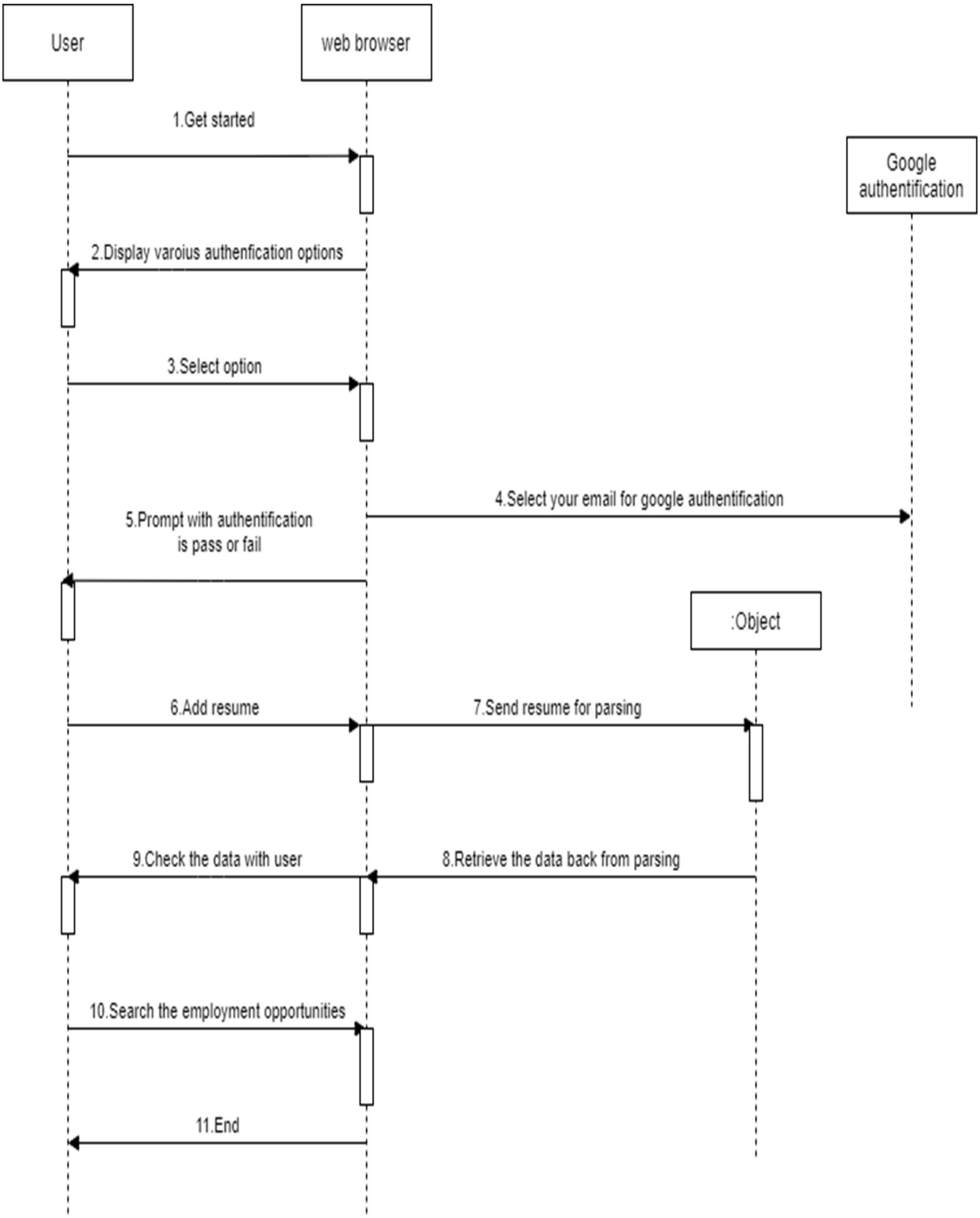
3.6 Conceptual Models:

A conceptual model is a presentation of a system, made of the composition of concepts that are used to help people know, understand, or simulate a subject the model represents. It is also a set of concepts. Some models are physical objects; for example, a toy model may be assembled, and maybe made to work like the object it represents.

3.6.1 Use-case diagram



3.6.2 Sequence diagram

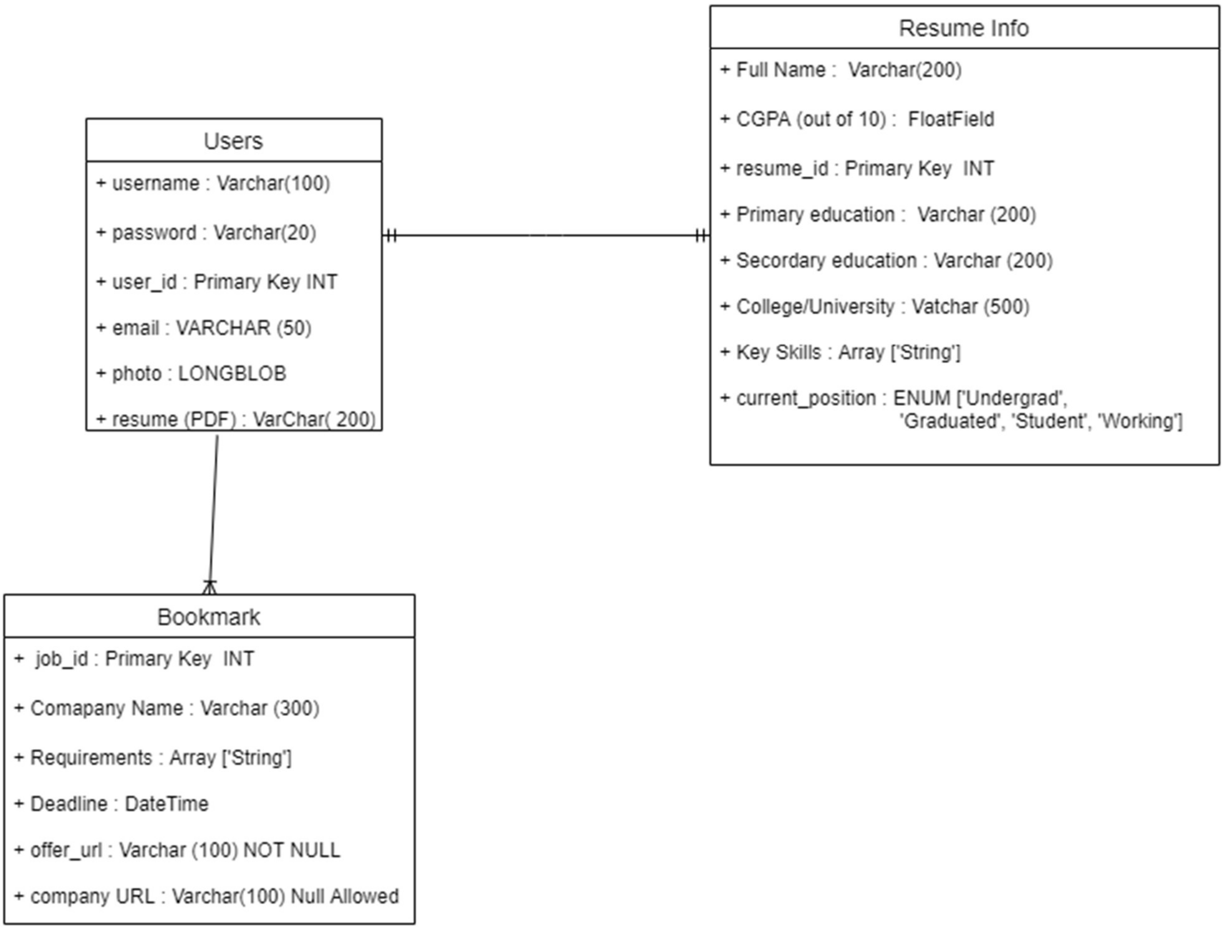

User
web browser
Google authentification
1.Get started
2.Display varoius authenfication options
3.Select option
4.Select your email for google authentification
5.Prompt with authentification is pass or fail
:Object
6.Add resume
7.Send resume for parsing
9.Check the data with user
8.Retrieve the data back from parsing
10.Search the employment opportunities
11.End

3.6.3 E-R Diagram:

Features

1. Personality Tests
2. Job Prefference (According to Resume)
3. Web Crawlars
4. News Letter (Email Automation)

ER Diagram of Database



Users
+ username : Varchar(100)
+ password : Varchar(20)
+ user_id : Primary Key INT
+ email : VARCHAR (50)
+ photo : LONGBLOB
+ resume (PDF) : VarChar(200)

Resume Info
+ Full Name : Varchar(200)
+ CGPA (out of 10) : FloatField
+ resume_id : Primary Key INT
+ Primary education : Varchar (200)
+ Secordary education : Varchar (200)
+ College/University : Vatchar (500)
+ Key Skills : Array ['String']
+ current_position : ENUM ['Undergrad', 'Graduated', 'Student', 'Working']

Bookmark
+ job_id : Primary Key INT
+ Comapany Name : Varchar (300)
+ Requirements : Array ['String']
+ Deadline : DateTime
+ offer_url : Varchar (100) NOT NULL
+ company URL : Varchar(100) Null Allowed

3.6 Project SDLC Model:

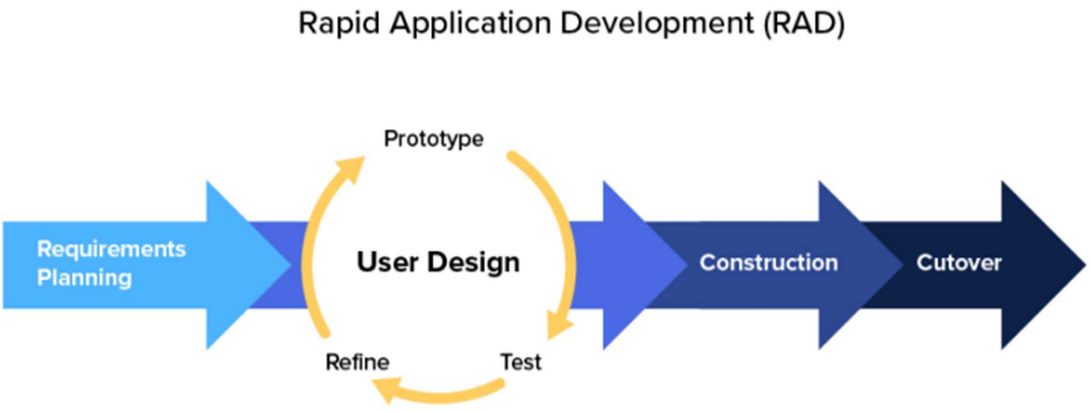
RAD SDLC Model:

- RAD model is the Rapid Application Development model. It is a type of incremental model. In the RAD model, the components or functions are developed in parallel as if they were mini-projects.

- The progress and development of the project can be measured through various stages.

- It is easier to accommodate changing requirements due to the short iteration periods.

- Thus we prefer the "RAD model" as requirements are clear and require short development time.



This model consists of 4 basic phases:

1. RequirementsPlanning:
It involves the use of various techniques used in requirements elicitation like brainstorming, task analysis, form analysis, user scenarios, etc. It also consists of the entire structured plan describing the critical data, methods to obtain it, and then processing it to form the final refined model.

2. User Description:
This phase consists of taking user feedback and building the prototype using developer tools. In other words, it includes re-examination and validation of the data collected in the first phase. The dataset attributes are also identified and elucidated in this phase.

3. Construction ;
In this phase, refinement of the prototype and delivery takes place. It includes the actual use of powerful automated tools to transform process and data models into the final working product. All the required modifications and enhancements are too done in this phase.

4. Cutover :
All the interfaces between the independent modules developed by separate teams have to be tested properly. The use of powerfully automated tools and subparts makes testing easier. This is followed by acceptance testing by the user

Chapter 4:
System design

4.1Basic Modules in project

4.2 Data Design

4.3 User Interface Design screenshots

4.4 Security Issue

4.5Test Case Design