SEMANTICS BASED INTELLIGENT JOB PORTAL

By

Muhammad Hamza Chaudhry

Hira Wahid

Muhammad Kamran Saeed

Submitted to the Faculty of Computer Software Engineering Department

National University of Sciences and Technology, Islamabad in partial fulfilment for the requirements of a B.E Degree in Computer Software Engineering

JUNE 2014

CERTIFICATE

Certified that the contents and form of project report entitled **"Semantics Based Intelligent Job Portal"** submitted by Muhammad Hamza Chaudhry, Hira Wahid, and Muhammad Kamran Saeed have been found satisfactory for the requirement of the degree.

Supervisor: _____

Dr. Hammad Afzal

Department of CS

Military College of Signals

ABSTRACT

Semantics based Intelligent Job Portal is a web application that will carry automated meaningful search on behalf of the user (Job Provider and Job Seeker) so that only precise and correct information is retrieved and both the users can find their right match, that is, Job Seekers find the right job and Employers find the right person for the job.

The job portal takes data of job seekers from LinkedIn and as their CVs. Data is extracted from LinkedIn using its profile API. CVs can be provided either by filling data into online forms provided by portal or by uploading in PDF format. PDF Parser is a PHP library that is used to extract data from CVs in pdf format. Skills are separately extracted from this data. These extracted skills are individually mapped over relational database. Every new added skill is sent to google to get its Uniform Resource Identifier using Ajax call to google API. Different representations of skill names are stored with common URI in relational database.

On the other hand, Job portal takes data of Job providers and their related companies. Job providers upload jobs on behalf of their companies. Skills are extracted from job and are mapped to relational database. Each skill is sent to google using AJAX call to get its URI.

URI of skills are used to perform skill based search. Each URI from google is searched over the database and skills with same URIs are grouped together to achieve semantics using google as base ontology. Users can perform skill hunt loosely means without restricting themselves to actual spells of words. Companies are efficiently maintained over the portal. Interface of website is adaptive to increase usability of portal.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

This thesis is dedicated to our parents, siblings and teachers for their guidance, support and prayers through thick and thin enabling us to achieve our goals.

ACKNOWLEDGMENT

We would like to thank Allah Almighty for His incessant blessings which have been bestowed upon us. We are grateful to our parents for their unwavering faith in us, their continuous support and love without which we would not have been able to succeed.

We are extremely grateful to our project supervisor Dr. Hammad Afzal who in addition to providing valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.

We are highly thankful to all of our teachers and staff who supported and guided us throughout our course and research work. Their knowledge, guidance and training enabled us to carry out this research work.

In the end we would like to acknowledge the support provided by all our friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Introduction

Most of the content today is designed for humans to read and not for computer programs to manipulate meaningfully. Computers can parse webpages for layout but computers have no reliable way to process semantics. Secondly, the search carried on different search engines is syntax based. According to this point of view our aim is to develop a semantic web application that shall act as an intelligent job portal for job seekers and job providers. Our system will establish an automated linkage between a job seeker and job provider. Thus making this need of searching for the right employee and searching for the right job easier, faster and reliable for both the job provider and seeker respectively.

1.2. Problem Statement

Keeping in view the concept established in the introduction. We have scanned through all the major job portals and found out that the searching criteria is manual i.e. the job seekers manually look for the jobs posted by job provider and similarly the job providers manually scan through the details to match their job requirement. This activity is tedious and time consuming thus making the process of hiring difficult.

Secondly, with manual scanning of resumes; data inconsistency is one major problem that is faced by the job providers. The third party which is manually scanning the resumes for job providers might not comprehend the job requirements well enough and may send resumes irrelevant for the particular job. On the other hand if a job seeker directly applies

for a job without meeting the requirements pertaining to the vacancy and applies, it again results in unwanted data at the job provider's side.

At job seeker end, they have to scan through each job criteria manually as a result it leads to slow retrieval of information and inefficient searching mechanism. This will result in job seeker being not able to get his desired job or the job of his potentials.

1.3. Statement of Goals and Scope of the Project

The basic idea behind the project is to make the search for job easier and intelligent from the perspective of job seeker and job provider. Another objective is to provide ease to the user by minimizing his intervention in searching and sorting the desired information. This can only be done by automating the search. As all the advancement in the field of science and technology aim at one main purpose i.e. "to provide humans as much convenience as possible". This project hence is an effort to contribute towards achieving this goal.

The scope of the object is to make the search in the context of job seeking meaningful by implementing a scenario where a job seeker wants to search for his ideal job and job provider wants to post a job to look for its ideal employee.

Following technical and academic goals are aimed to be achieved during the development of this project.

a. Learn to develop web application
b. Learn PHP, XML and JavaScript
c. Semantic Analysis of Data
d. Character reading by parsing the document file

e. Automating the linkage between seeker and provider

f. Learning RDBMS

g. Learning concepts of ontology.

1.4. Deliverables

Following are the deliverables associated with the project

a. Project Synopsis

b. Software Requirement Specification Document

c. Software Design Document

d. Implementation Code & Documents

e. Software Testing Document

f. Final Project Report

2. Literature Review

2.1. Understanding Terminologies

2.1.1 Semantic Web

The Semantic Web is a collaborative movement led by the international standards body, the World Wide Web Consortium (W3C). The standard promotes common data formats on the World Wide Web. By encouraging the inclusion of semantic content in web pages, the Semantic Web aims at converting the current web, dominated by unstructured and semi-structured documents into a "web of data".

According to the W3C, *"The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries."* The term was coined by Tim Berners-Lee for a web of data that can be processed by machines.

2.1.2 Linked Data

Linked data describes a method of publishing structured data so that it can be interlinked and become more useful. It uses standards of the Web such as HTTP and URIs, but instead of using them for serving human readers, it helps to share information in way that it can be understood by the computer automatically. This enables data from different sources and resources over the web to be connected and queried.

2.1.3 Ontology

Ontology is a data model that represent a set of concepts with a domain and the relationship between those concepts. It is used to reason about the objects with in that domain. Ontology is used in artificial intelligence, the sematic and information architecture as a form of knowledge representation about the world.

Ontology describes the individuals as basic objects which exists in class of similar objects along with attributes, relationship or parameter that objects can have. [3]

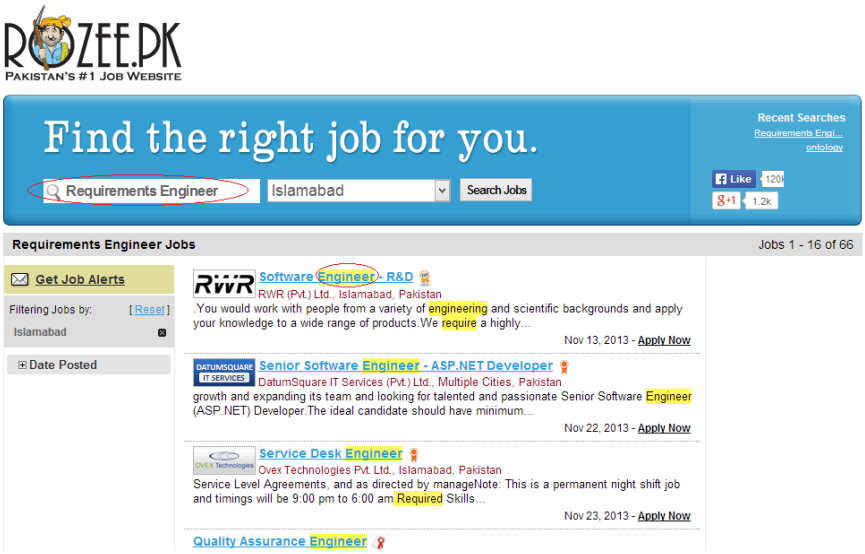
2.2. What has been done in the domain: Job Portal

2.2.1 Active Job Portals

There are a large number of commercial job portals, like Monster, CareerBuilder or StepStone, that charge a publication fee for posting job. The different portals intend to target different audiences and are specialized in certain geographic regions. Since the publication fee often add up to huge number, employers publish their postings only on a limited number of portals. For this reason often employers prefer to post jobs on their own website. Publishing postings on the corporate website is cheap but reaches only a very limited audience, because the indexing capabilities of current search engines like Google are too imprecise to support directed searches for open positions.

Even if the job posting are free of cost there still remains a huge gap between the job provider and the job seeker. These portals don’t form the strong bridge between both the users. The relevant jobs are sometimes missed out by the job seeker or in worst cases they even do not appear. These job portal have the incapacity to rightly address the needs of the

job seeker in comparison to the skill set he or she has. Similarly is the case for the other user of these portals, that is, Job Providers are unable to find the right match for the skill set they are looking for in their prospect employee. Here are some screenshots of the famous job sites:



The above screen shots shows the the job portal only performed keyword search and gave results that did not match the query of the user. They job portal was not intelligent enough to understand the meaning of the query done by the user.

2.3. Research Papers

Following are some of the papers reviewed during the process of identifying the need for our system and to make our system better in view of the research being carried out in the field.

2.3.1 Literature Survey on the Semantic Web and Search

The main idea of this paper was to discover existing state of the Semantic Web with the having its major focus on search. The paper reviewed papers on semantics and ontology based search. The conclusion drawn from this survey paper were:

Existing RDF(S) query languages are not complete, lacking expressivity. Absence of standards on query languages, rules and inferencing resulted in many different incompatible implementation which makes them difficult to compare, learn, exploit, and so forth. Existing systems are not mature and have limitations of performance, inefficient storage schemas and completeness of query results.

2.3.2 Improving the recruitment process through ontology-based Querying

The paper mentioned discussed cases of inconsistent or overly specific queries which would return no results still have to be dealt with in view of semantics based job portal. The author showed use of techniques in a prototype job portal in which both job offers and applicants are described according to ontologies. Preparation of ontologies from non-ontological sources was time consuming and tedious but enabled to improved search by showing relevant results. It also ranked them according to similarity.

Furthermore, we have shown that semantic matching alone does not allow for levels of similarity to be differentiated and for inconsistent or overly specific queries to be resolved. Hence we introduce another technique called query relaxation, in which queries can be rewritten to allow similarity to be ranked. At present, the semantic job portal demonstrates improved precision and recall in the semantic matching, finding relevant job offers or

applicants which would not be selected by a syntactic (text-based) query algorithm. This technique has really helped both the job seeker and job provider to find the right results and thus making recruitment process easier and less tedious.

2.3.3 Ontology based Recruitment Process

In this paper, the authors proposed an application of Semantic Web technologies in the recruitment process. The paper has used the approach of human resource ontology which provides means for semantic annotation of job postings and job applications. The use of controlled vocabularies against the free text description render better results in terms of machine processabililty, data interoperability and integration. Moreover, having job postings and user profiles semantically annotated, enables the system to perform semantic matching which significantly improves query results and gives a ranked list of best matching candidates for a given job position. The authors also described the system architecture of the recruitment portal based on Semantic Web technologies.

2.3.4 The Impact of Semantic Web Technologies on Job Recruitment Processes

The authors of this paper presented a scenario for supporting recruitment processes with Semantic Web technologies. They described the online recruitment process and analysed the state-of-the-art technologies in this domain. Suggestion for improvements for several steps by enhancing job postings and job applications with annotations using controlled vocabularies were made. The core components of the prototype are a human resource ontology derived and a retrieval engine using semantic match-making. The analysis showed that the technology will benefit both the job seekers and job provider and will bring

in transparency in the market. Employers would benefit more by reaching out to potential employee and would also save publication fee.

3. Software Requirement Specification

3.1. Introduction

This chapter requirements for Semantics Based Intelligent Job Portal. The purpose of our system, is to offer a web application that will carry automated meaningful search on behalf of the user (Job Provider and Job Seeker) so that only precise and correct information is retrieved and both the users can find their right match, that is, Job Seekers finds the right job and Employers find the right person for the job.

3.2. Product Scope

The project aims to cover three major modules that are as follows:

Web Application

1. Provides the basic interface for the clients to interact.
2. Controls the access to our portal via assigned credentials.
3. Stores user information.
4. Creates Queries and finds specific result for user preferences.
5. Notifies job seeker and job provider from time to time if any matching criteria is found

Data Collection and Parsing Module

1. Gathers job seekers information through three ways:

    a. Providing Online form for CV data collection

    b. Uploading CV by filling the template provided by the portal.

    c. Interacting with LinkedIn web service to gather information.

2. After data has been gathered skills are separated and stored.

URI Mapping

1. Google will be used as base ontology of the system.

2. Google API will be used to match keywords which refer to same set of skills.

3. Corresponding Wikipedia URI will be mapped to the skills in the database.

3.3. System Overview

Semantics Based Intelligent job portal will be hosted over the internet. A user connected to the internet will be able to access the system and sign up as a job seeker or a job provider and use the system according to the specified roles. The following diagrams shows the general context of the system.

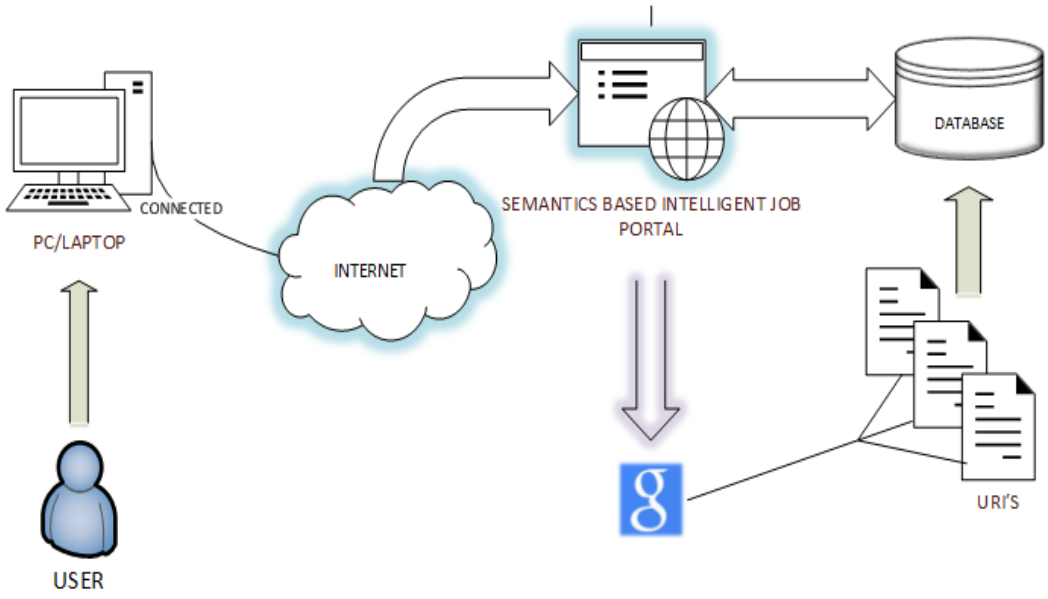


Figure 3-1: Context Diagram

3.4. Product Functions

Semantics Based Intelligent Job Portal is intended to offer the following functions:

Login/Sign-up: SBIJP will allow the users to sign up with our application in order to generate a profile page. After that every time an existing user enters a new session he/she user would be required to login to his/her page.

Profile page creation: Users would be able to view details about either vacant jobs or applicants only on their respective profile page.

Resume submission: This feature will allow job seekers to submit resume through any option suitable to their choice of interest. For instance they can sync resume from LinkedIn, download available template for resume fill up or they can fill CV online.

Parsing Skills: Skills will be extracted from the pdf uploaded and will be mapped on the database

URI Mapping: URI's for the related skills will be extracted and mapped on to the database.

Searching: SBIJP will allow users to carry out semantically intelligent search, so that they can view jobs on the basis of matching criteria first.

Notifying users: This feature will help the job seekers as well as job providers to get updates on any matching job or candidate from time to time. It will be done by Emails.

Setting status: Once a candidate has been selected and vacant job positions have been filled up then a status of employed would be set against the respective profile page.

3.5. User Classes and Characteristics

Job Seeker: They will sign up on our web application and provide their resume to our portal so that jobs can be recommended to them as well as resume can be dropped on employer's profile.

Job Providers: They will sign up on our web application to search for right professionals to be hired by their firms.

System Administrator: This would be the person who would manage our web portal. His responsibilities would include maintaining record, ensuring confidentiality and policy making.

3.6. Operating Environment

The Web Application has two sides which include Client side and Server side. For the portal to run efficiently and effectively require follow operational environment.

a. Client

For clients to communicate through our portal it is necessary to have:

a. A Browser

b. An Internet Connection (recommended 256 kbps or higher)

b. Server

Since all the processing of information is to be done on server side therefore it needs to have the following requirements full filled:

a. Fast internet connection for fast transactions.

b. 1 tera byte of storage space since user data will grow over a period of time.

c. Minimum 2GB of RAM for fast processing.

3.7. Assumptions and Dependencies

Skills set can be defined in any manner by the job provider, to cater the problem the system depends on the Google API to map a Wikipedia URI to the database. LinkedIn API will also be used to extract skills if a user wants to sign up on the basis of his LinkedIn account. The system is developed keeping in mind the CV's of the graduating batch of last year and have accounted for the skills they have mentioned in their CV's.

The system will be operating in a client server environment. Due to this various protocols would be used in our system like for communication between client and server HTTP

channel would be used. SMTP protocol would be deployed for notifying users with email. Communication will be made secure using internet security conventions.

3.8. Constraints

The system will be designed and developed under the Following constraints:

1. The system will target only the skills of the job seeker.
2. Unexpected results would occur If Google Search API is unable to map logically same set of skills to the same target Wikipedia URI.
3. Internet connection is a requirement to access the job portal.

3.9. Stimulus/Response Sequences

3.9.1 Job Portal Sign Up

Feature Name:	Job Portal Sign-Up
Description:	This is the base feature of our portal that will allow access to profile pages to only registered members. (Priority: Medium)
Stimulus/Response (primary scenario)	1. User Clicks account creation page. 2. System responds by presenting sign up form to user. 3. User enters specific details and submits the form. 4. System verifies the fields entered by user. 5. System after verification stores data of user in repository. 6. User receives a verification email in order to ensure secrecy of credentials set by user.

Alternative	1. User enters data which is not verifiable. 2. Sign up is denied to user by the system.

Functional Requirements	
Req#1	The system shall ensure that one user can only create one account.
Req#2	The system shall validate user fields on the run time.
Req#3	The system shall also validate entered fields from the data base in order to avoid redundancy of data.
Req#4	The system shall send verification email to user on the entered email address in order to ensure that user is authentic.

3.9.2 Sign In

Feature Name:	Job Portal Sign-In
Description:	To use the services provided by our application user must sign in. (Priority: Medium)
Stimulus/Response (primary scenario):	1. User clicks on the Sign-In button on our web page. 2. System asks for user credentials. 3. User is prompted to enter his/her login id. 4. After entering login id and password user clicks on the login button to enter his/her profile page. 5. The system takes the credentials of user and matches with stored values in database. As a result user is allowed access to his/her profile page.

Alternative	1. User clicks Forget password option or create a new account option. 2. The operation is canceled.

Functional Requirements	
Req#1	The web application shall not allow any guest user to have access to profile pages.
Req#2	The web application shall allow the use of job portal services to only verified users.
Req#3	The system shall handle the forget password field for the local accounts.

3.9.3 Resume Submission

Feature Name:	Resume Submission
Description:	The Job seeker shall be able to upload his/her resume through three options available to the user. Getting the resume from user is an essential aspect of our portal upon which further features would be catered. (Priority: Medium)
Stimulus/Response (primary scenario):	1. When user has logged in, the user would click upload resume option 2. System provides a drop down list to user for selecting the method of resume submission. 3. User selects syncing resume details from LinkedIn.

	4. Our application would ask for LinkedIn credentials of user 5. User enters LinkedIn credentials and presses login. 6. If the login is successful system syncs user information from LinkedIn to his/her profile page.
Alternative	1. User selects Online CV fill up option. 2. Application opens multiple pages containing online form to be filled up by user. 3. User clicks submit button. 4. System validates the form and notifies user if all fields were correctly filled up. **OR** 1. User selects template download option. 2. Application downloads template file on the user system. 3. Once user has filled the form he submits on his profile page.

Functional Requirements	
Req#1	The system shall provide to upload resume in pdf format.
Req#2	The system shall validate the resume by checking all necessary fields are provided.
Req#3	The online mode to fill CV shall allow user to save the form to come back on complete.
Req#4	The system shall store key phrases from the CV against the user id.

3.9.4 Job Posting and Notification

Feature Name:	Job Posting and Notifying
Description:	This feature is responsible to select matching job for the job seeker and to post it on user's profile. Once the job has been posted both the job seeker and job provider are informed through email and internal messaging process of our web application. (Priority: High)
Stimulus/Response (primary scenario):	1. System matches the criteria set by job provider with the URI's and corresponding skills from the database 2. System on selection of job informs job seeker through internal messaging process. 3. Job seeker views the jobs posted and selects drop CV option. 4. System drops user CV on that job provider's profile.
Alternative	5. System can't find any matching criteria for job seeker as well as job provider.

Functional Requirements	
Req#1	The system shall recommend jobs to all the users matching the criteria.
Req#3	In case of no matching criteria found the system shall inform user instantly.
Req#4	Once the data has been parsed, the system will extract the URI's of the entered skills

Req#5	Once the jobs have been posted and candidates are selected, the system should notify the users.

3.9.5 Search

Feature Name:	Search
Description:	This features allow the users to manually search for the job or employers by entering respective skill query. (Priority: High)
Stimulus/Response (primary scenario):	1. User enters a search query. 2. The query is passed on to the Google API. 3. API returns the URI 4. The URI is sent to the database to be matched against the skills. 5. The resulting skills are matched against the entries containing the respective skills 6. The user is displayed with the results of the
Alternative	3. No matching result is found.

Functional Requirements	
Req#1	The system shall recommend jobs to the job seeker depending upon the skills entered.
Req#2	The system shall recommend prospect employees to the job provider depending upon the skills entered

Req#3	In case of no matching criteria found the system shall inform user instantly.
Req#4	The queried skills will be mapped to the database for future queries.

3.10. External Interface Requirements

3.10.1 User Interfaces

The system will have a graphical user interface (GUI).It will be designed according to web design principles. Client side languages such as HTML, CSS, and JAVASCRIPT and JQUERY would be used to provide user friendly and attracting appearance to our portal. Final design is open to suggestions ofthe developer and the capabilities of the system.

3.10.2 Software Interfaces

a. Application shall connect to LinkedIn API to fetch user profiles and sign in through it.

b. Application shall connect to Famous Job Portals for job retrievals.

c. Connection to databases and ontology.

3.10.3 Communications Interfaces

a. Internet connections through which the user accesses our Application.

b. HTTP standards will be the basis of web application.

c. REST will be used to communicate with LinkedIn API.

3.11. Other Nonfunctional Requirements

3.11.1 Performance Requirements

3.11.1.1: The system shall respond any search query within 3-5 seconds with the minimum internet connection speed specified.

3.11.1.2: The login shall not take more than 2-3 seconds with the minimum internet connection speed specified.

3.11.2 Security Requirements

Security threats are always a major worry for any web based application therefore any security threat must catered for.

3.11.2.1 Login Mechanism

User with a valid Local/LinkedIn account is allowed to login and access only his/her details and make changes if required. If username/password of Local/LinkedIn is compromised the user's profile on the portal is also compromised as a result of that.

3.11.2.1 User's Information Security

User have no direct communication with user data to prevent compromising user personal information. Only admins will be allowed to access and manipulate the stored data.

3.11.3 Software Quality Attributes

1. Availability:

The system shall be made available for the user 24/7, 365 days per year as this holds the key to efficient use of the system. If there is a failure, recovery time must be no greater than 1 hour, and total yearly uptime must be at least 98% (2% downtime, or approximately 7 days per year).

2. Adaptability:

The system developed is a web application and web standards and technologies change quite frequently with. Therefore system should be adaptable to the changes that occur over the web.

3. Flexibility:

It shall be ensured that the web application is flexible enough so that any future changes could be accommodated easily. Change in standards of the web or addition of another module and feature shall not require the system to be re-written altogether. Any changes would not take more than 12% of the existing code.

4. Maintainability:

The system is being divided into modules so its maintainability becomes an easy job as the defects in one module would not render defects in other modules.

5. Correctness:

It shall be ensured that the system conforms to the required specification during implementation and once the product is finally delivered.

6. Interoperability:

The web application shall be accessed through web browsers, smart phone browser therefore the server shall the ability to respond to all such mediums of access.

7. Reliability:

This attribute can be mapped to ‘Availability’ since we intend to make the system available for the maximum time makes the system reliable as well.

9. Usability:

The interface of the system shall be kept simple keeping in view the basics of improving user-system interaction. The user will not take more than 2-5 minutes to get use to the system and work around efficiently.

10. Robustness:

The system shall cater all the errors that may occur during processing and shall prevent any wrong response to be generated by the system.

4. System Design Specification

4.1. System Architecture

SBIJP system is designed using **Layered Architecture** approach. Layered architecture focuses on the grouping of related functionality within an application into distinct layers that are stacked vertically on top of each other. Benefits of layers:

- Simplicity: easy to design, once layers and their interaction are defined clearly
- Flexibility: easy to modify and develop networks by separate layers modifications
- Incremental changes: add new layers, add new functions to a layer

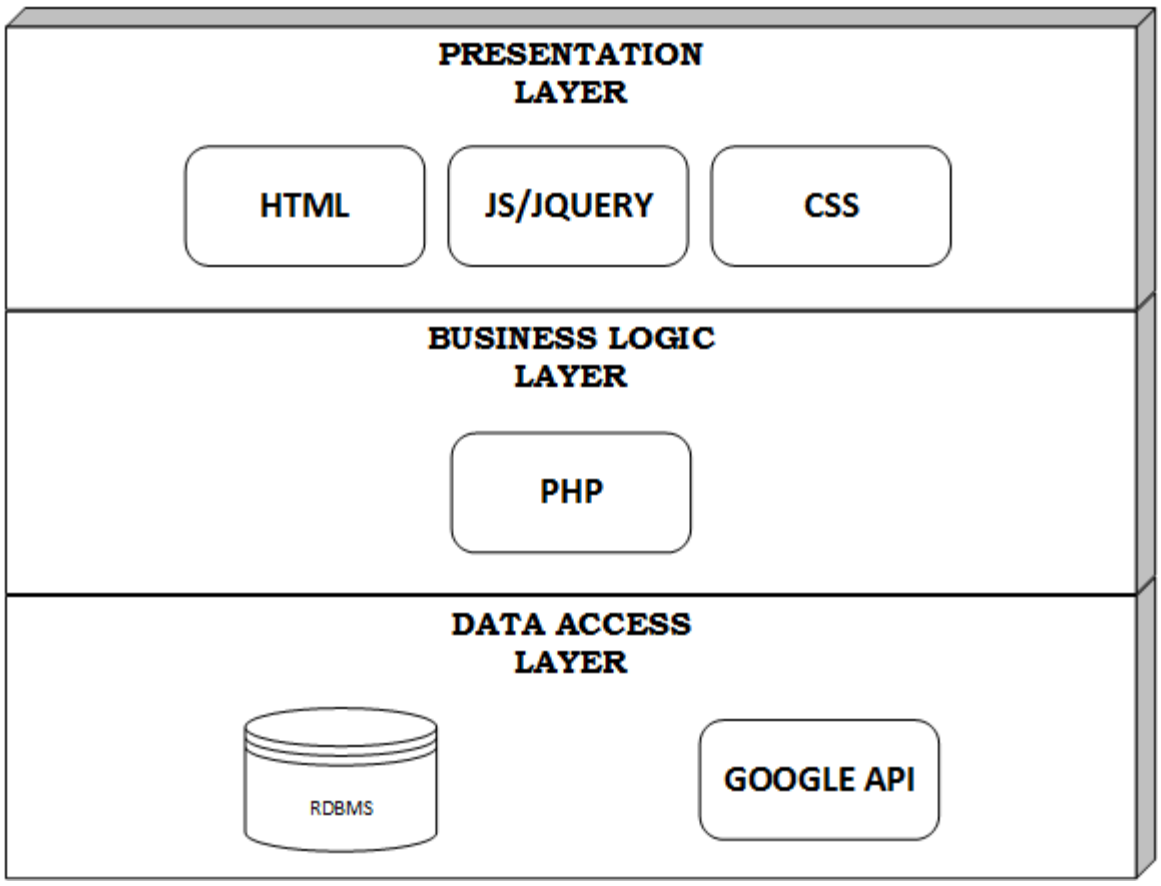


Figure 4-1: System Architecture

4.2. Design Details

4.2.1 Design Pattern

The design pattern that we are using for our system is **Façade**. Since we are providing a front end that the presentation a simpler interface to interact with hiding all the resources and function being carried out at the back end. The design pattern hides the usage of Google API to the user interacting and also limits the access to the database directly. Thus ensuring the integrity of the data by the controlled access to the database management system.

4.3. Use Case Diagrams

4.3.1 Job Portal Sign Up

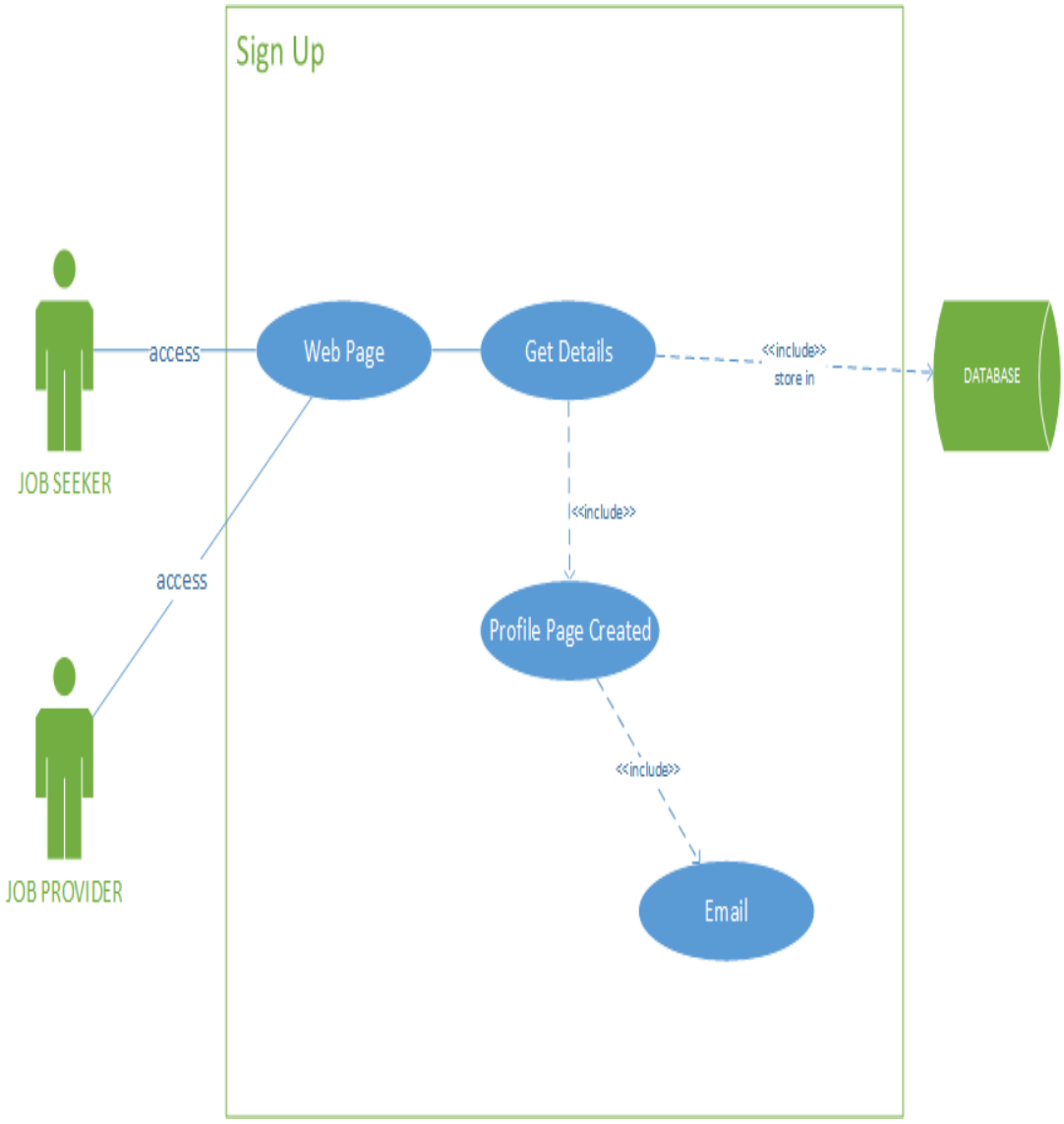


Figure 4-2: Job Portal Sign Up

Use Case Name:	Job Portal Sign-Up
Actors:	**Primary:** Job Seeker, Job Provider **Secondary:** Database
Description:	This is the base feature of our portal that will allow access to profile pages to only registered members. (Priority: Medium)
Preconditions:	User has access to job portal website.
Post conditions:	User gets registered with our portal. Profile page is created.
Normal Flow (primary scenario):	1. User Clicks account creation page. 2. System responds by presenting sign up form to user. 3. User enters specific details and submits the form. 4. System verifies the fields entered by user. 5. System after verification stores data of user in repository. 6. User receives a verification email in order to ensure secrecy of credentials set by user.
Alternative	4. User enters data which is not verifiable. 5. Sign up is denied to user by the system.

Table 4-1: Use Case-Sign Up

4.3.2 Sign In Use Case Diagram

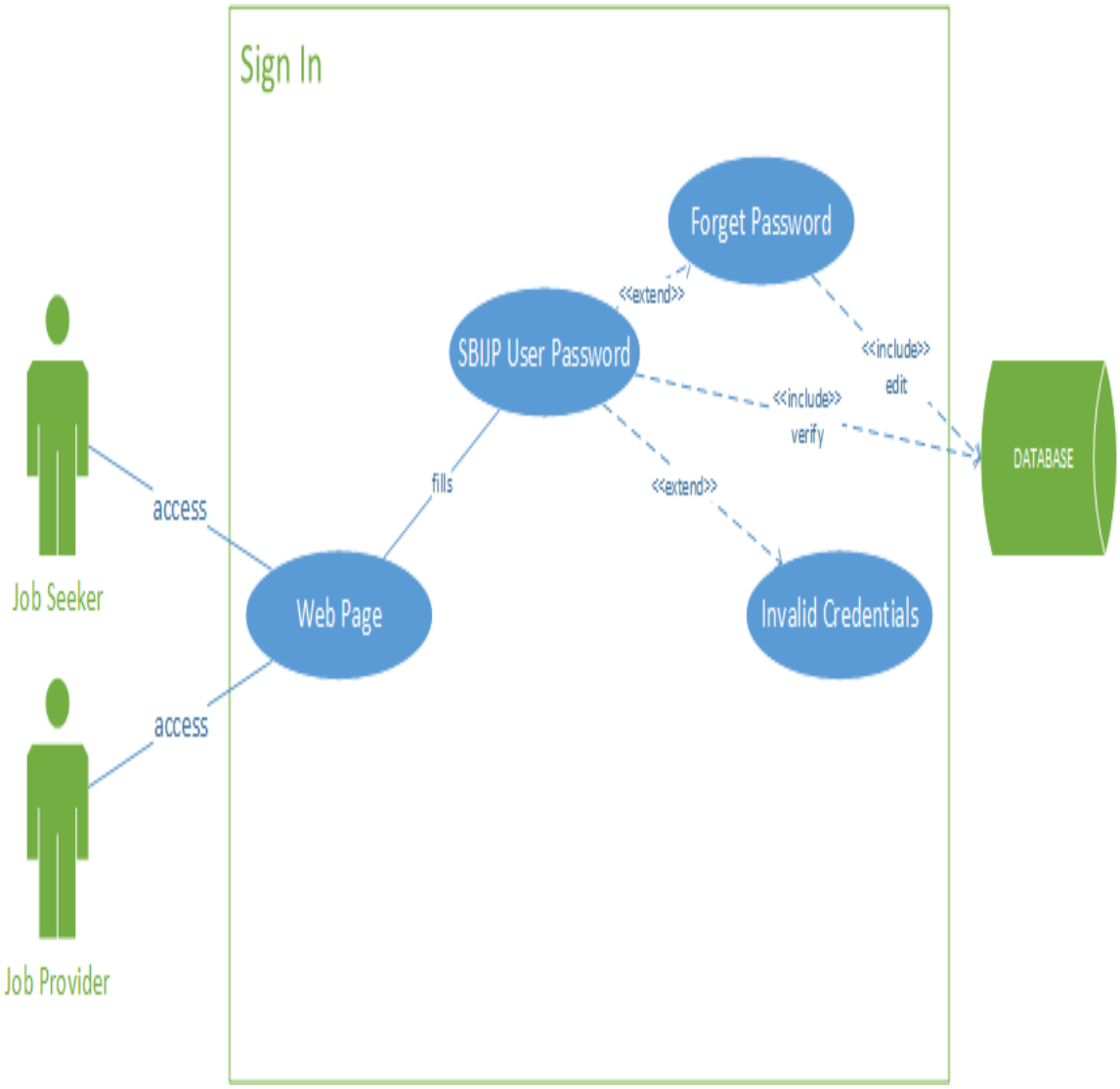


Figure 4-3: Sign In

Use Case Name:	Job Portal Sign-In
Actors:	**Primary:** Job Seeker, Job Provider **Secondary:** Database
Description:	To use the services provided by our application user must sign in. They must either have (Priority: Medium)
Preconditions:	User has access to Job Portal website
Post conditions:	The user has signed into the account
Normal Flow (primary scenario):	1. User clicks on the Sign-In icon on our web page. (Stimulus) 2. System asks for user credentials. (Response) 3. User is prompted to enter his/her login id. 4. After entering login id and password user clicks on the login button to enter his/her profile page. 5. The system takes the credentials of user and matches with stored values in database. As a result user is allowed access to his/her profile page.
Alternative	1. User clicks Forget password option or create a new account option. 2. The operation is canceled.

Table 4-2: Use Case-Sign In

4.3.3 Resume Submission Use Case Diagram

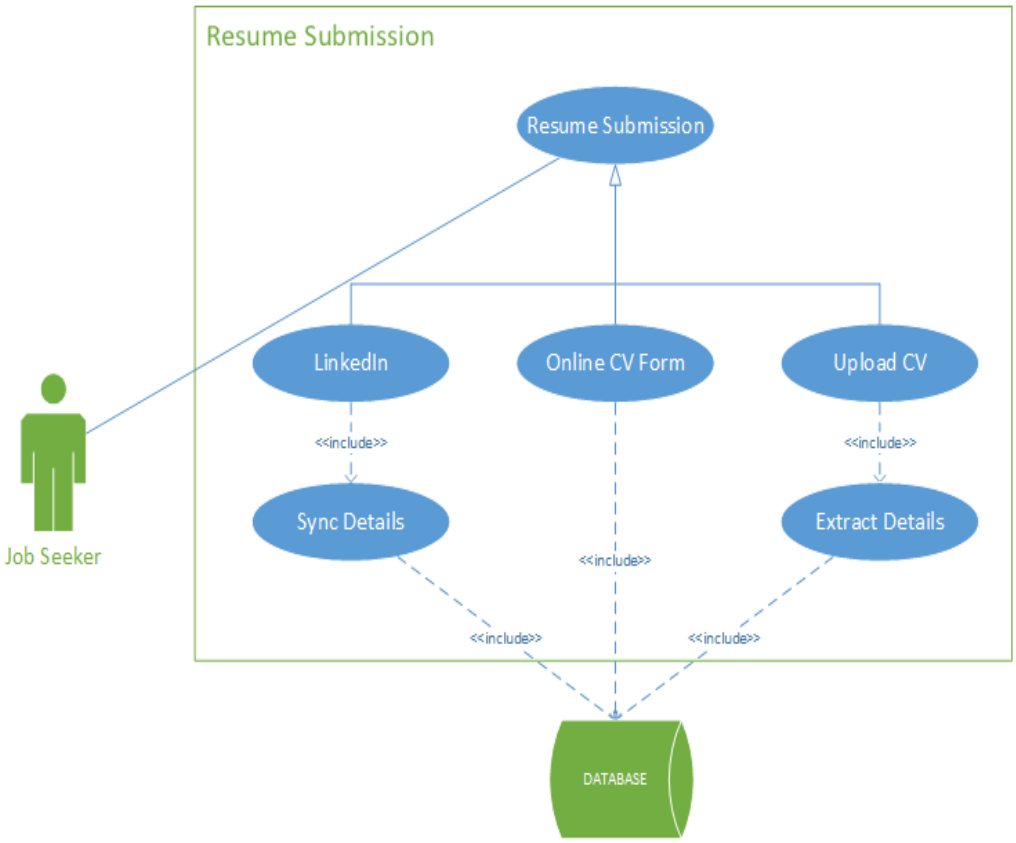


Figure 4-4: Resume Submission

Use Case Name:	Resume Submission
Actors:	**Primary:** Job Seeker **Secondary:** Database
Description:	The Job seeker shall be able to upload his/her resume through three options available to the user. Getting the resume from user is an essential aspect of our portal upon which further features would be catered. (Priority: Medium)
Preconditions:	User has access to Job Portal website.

	User must be signed
Post conditions:	Resume is submitted
Normal Flow (primary scenario):	7. When user has logged in, the user would click upload resume option 8. System provides a drop down list to user for selecting the method of resume submission. 9. User selects syncing resume details from LinkedIn. 10. Our application would ask for LinkedIn credentials of user 11. User enters LinkedIn credentials and presses login. 12. If the login is successful system syncs user information from LinkedIn to his/her profile page.
Alternative	1. User selects Online CV fill up option. 2. Application opens multiple pages containing online form to be filled up by user. 3. User clicks submit button. 4. System validates the form and notifies user if all fields were correctly filled up. **OR** 1. User selects template download option. 2. Application downloads template file on the user system. 3. Once user has filled the form he submits on his profile page.

Table 4-3: Use Case-Resume Submission

4.3.4 Job Posting and Notification Use Case Diagram

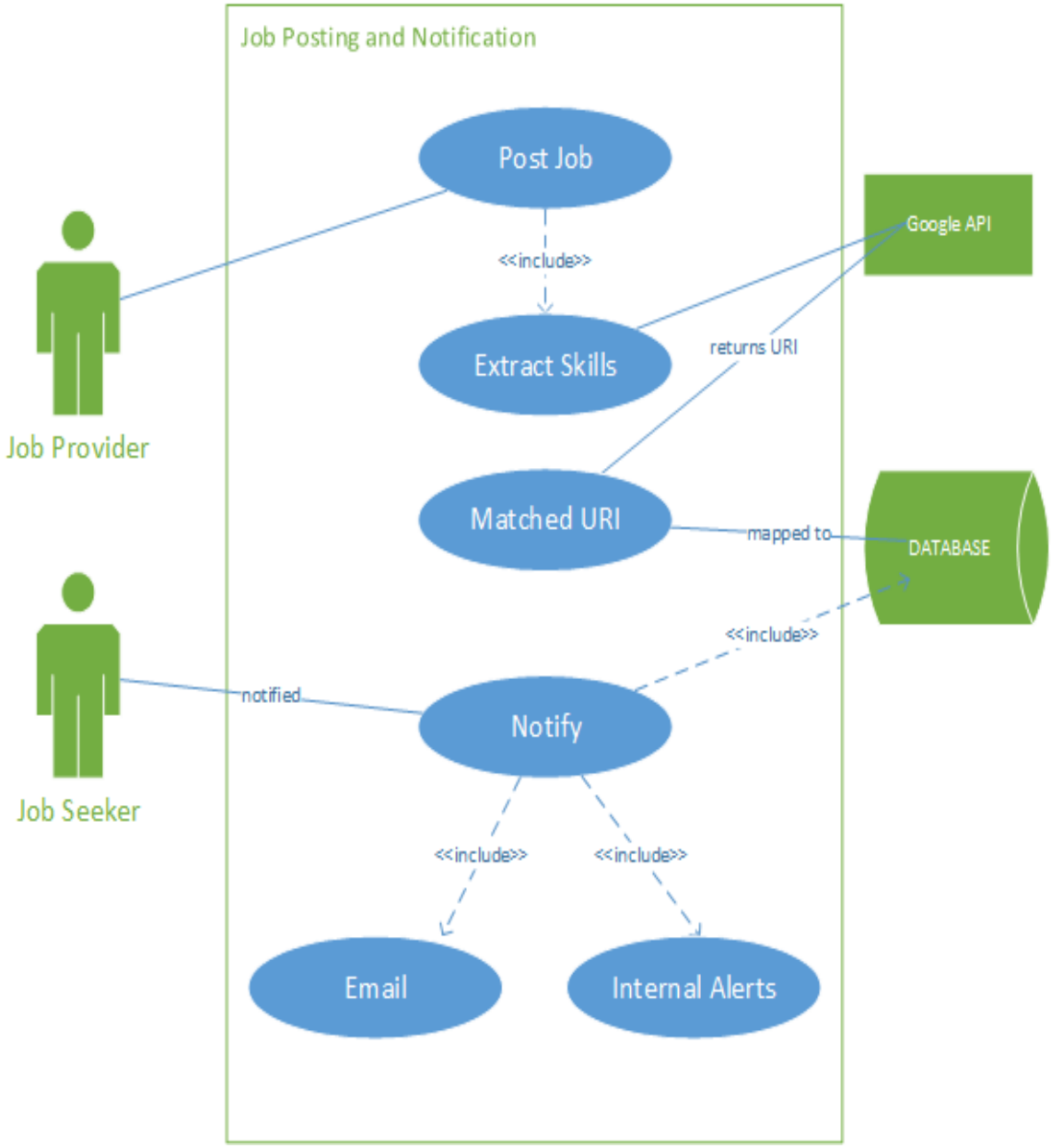


Figure 4-5: Job Posting and Notification

Use Case Name:	Job Posting and Notifying
Actors:	**Primary:** Job Seeker, Job Provider **Secondary:** Database, Google API
Description:	This feature is responsible to select matching job for the job seeker and to post it on user's profile. Once the job has been posted both the job seeker and job provider are informed through email and internal messaging process of our web application. (Priority: High)
Preconditions:	User data has been collected. Preferences have been set. Data has been matched with query.
Post conditions:	User is notified.
Normal Flow (primary scenario):	1. System matches the criteria set by job provider with the URI's and corresponding skills from the database 2. System on selection of job informs job seeker through internal messaging process. 3. Job seeker views the jobs posted and selects drop CV option. 4. System drops user CV on that job provider's profile.
Alternative	6. System can't find any matching criteria for job seeker as well as job provider.

Table 4-4: Use Case-Job Posting and Notifying

4.3.5 Search Use Case Diagram

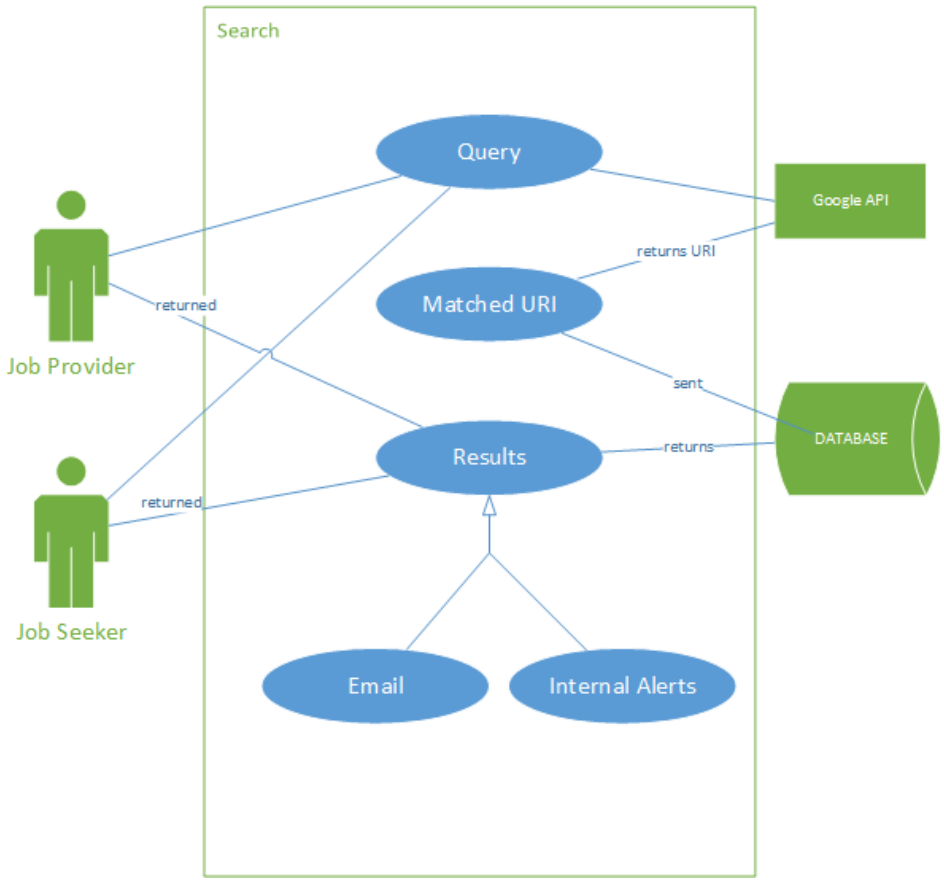


Figure 4-6: Search

Use Case Name:	Search
Actors:	**Primary:** Job Seeker, Job Provider **Secondary:** Database, Google API
Description:	This features allow the users to manually search for the job or employers by entering respective skill query. (Priority: High)
Preconditions:	User has access to Job Portal website. User must be signed

Post conditions:	Queried Results are returned
Normal Flow (primary scenario):	7. User enters a search query. 8. The query is passed on to the Google API. 9. API returns the URI 10. The URI is sent to the database to be matched against the skills. 11. The resulting skills are matched against the entries containing the respective skills 12. The user is displayed with the results of the
Alternative	6. No matching result is found.

4.4. System Class Diagram

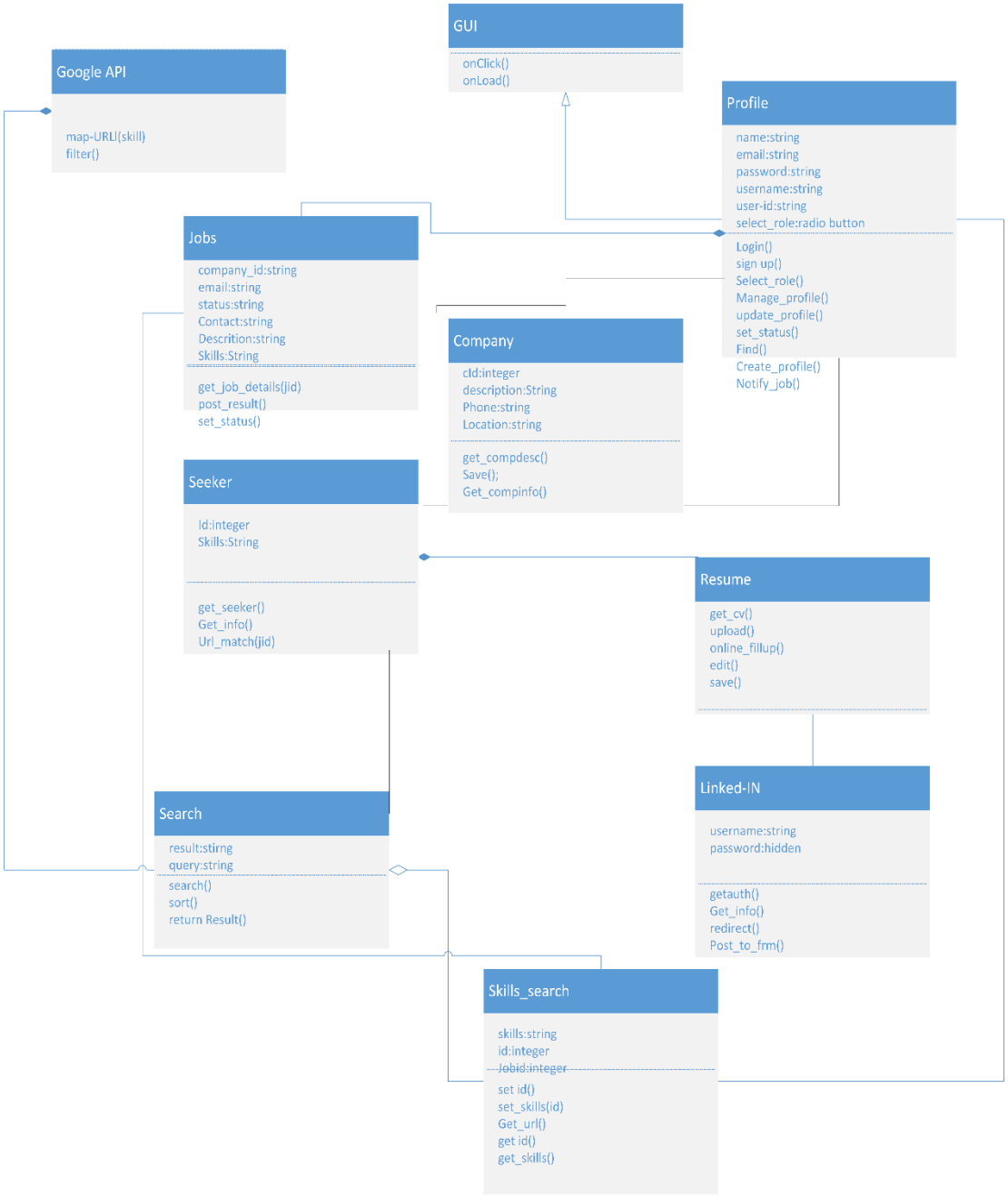


Figure 4-7: System Class Diagram

4.5. Dynamic View

4.5.1 Sequence Diagram

4.5.1.1 Sign Up (Job Seeker: Uploading CV) Sequence Diagram

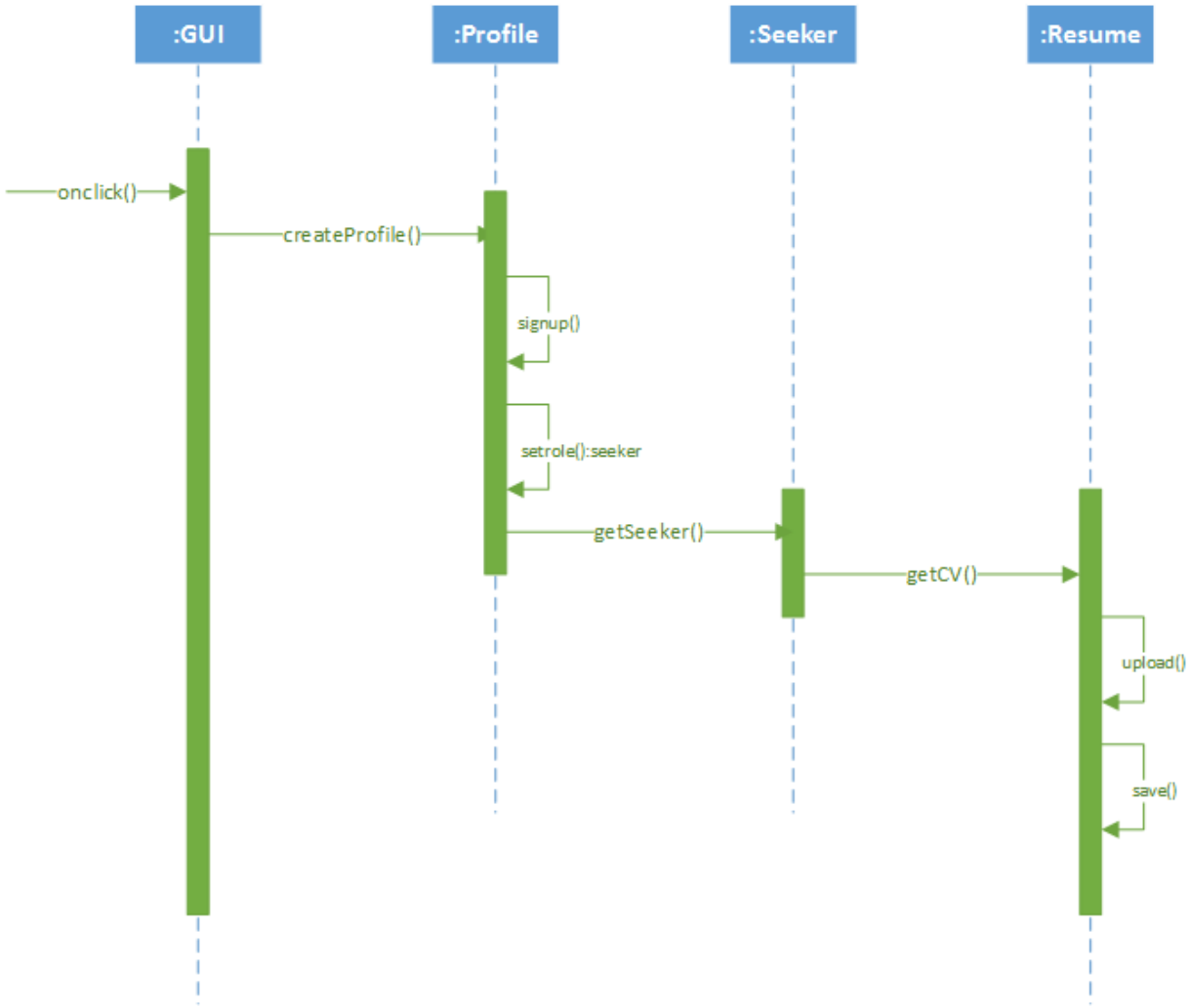


Figure 4-8: Sign Up (Job Seeker: Uploading CV) Sequence Diagram

4.5.1.2 Sign Up (Job Seeker: LinkedIn/Online Form) Sequence Diagram

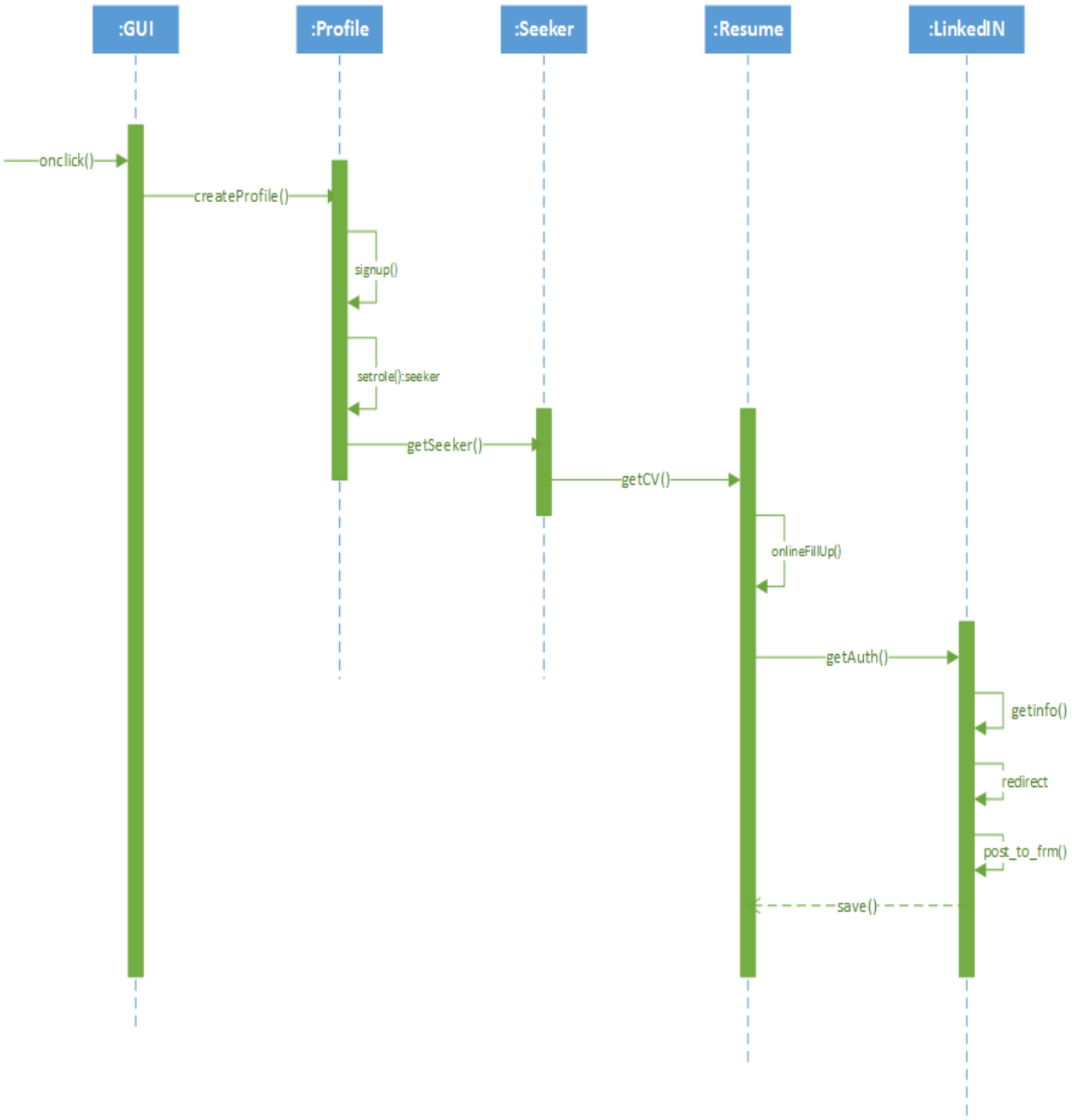


Figure 4-9: Sign Up (Job Seeker: LinkedIn/Online Form) Sequence Diagram

4.5.1.3 Sign Up (Job Provider) Sequence Diagram

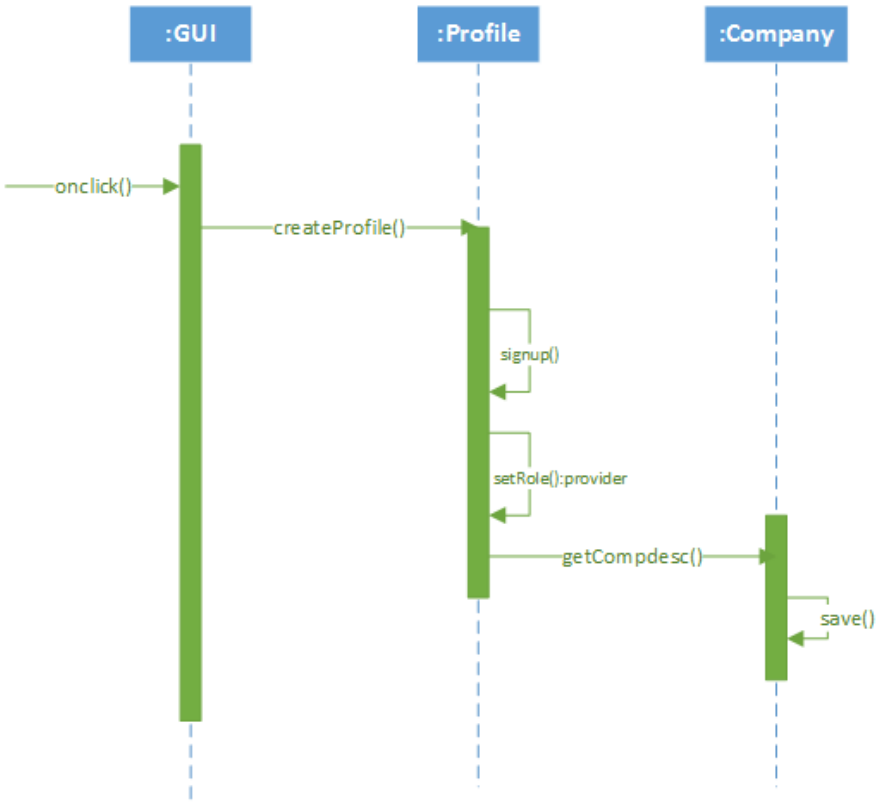


Figure 4-10: Sign Up (Job Provider) Sequence Diagram

4.5.1.4 Search Diagram

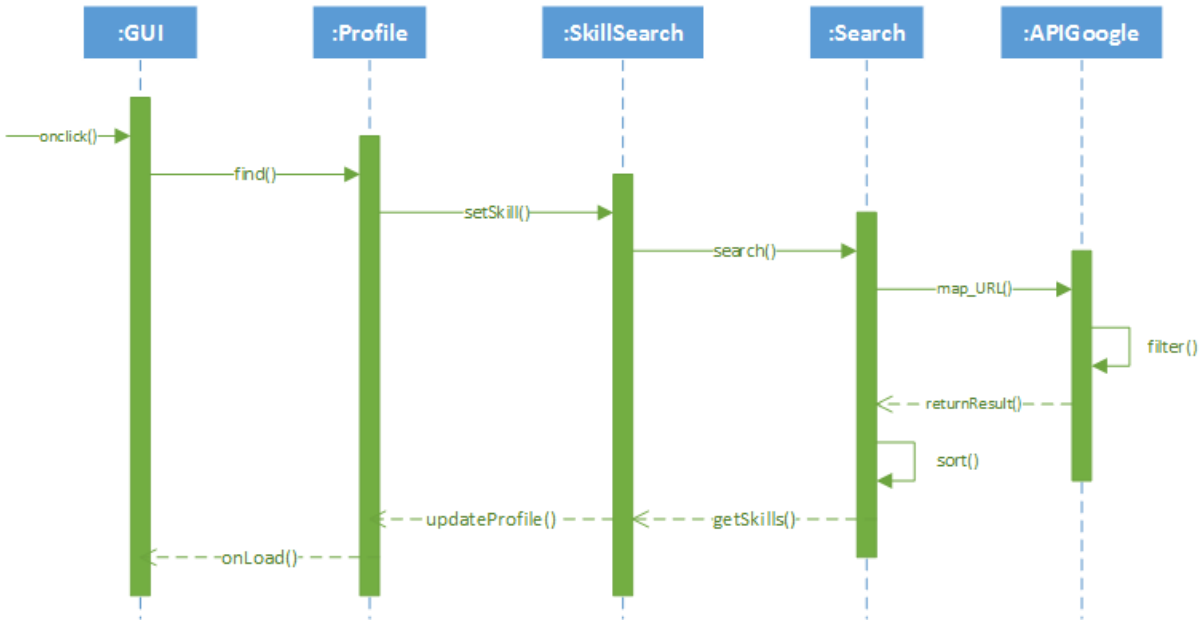


Figure 4-11: Search Sequence Diagram

4.5.1.5 Job Posting and Notification Diagram

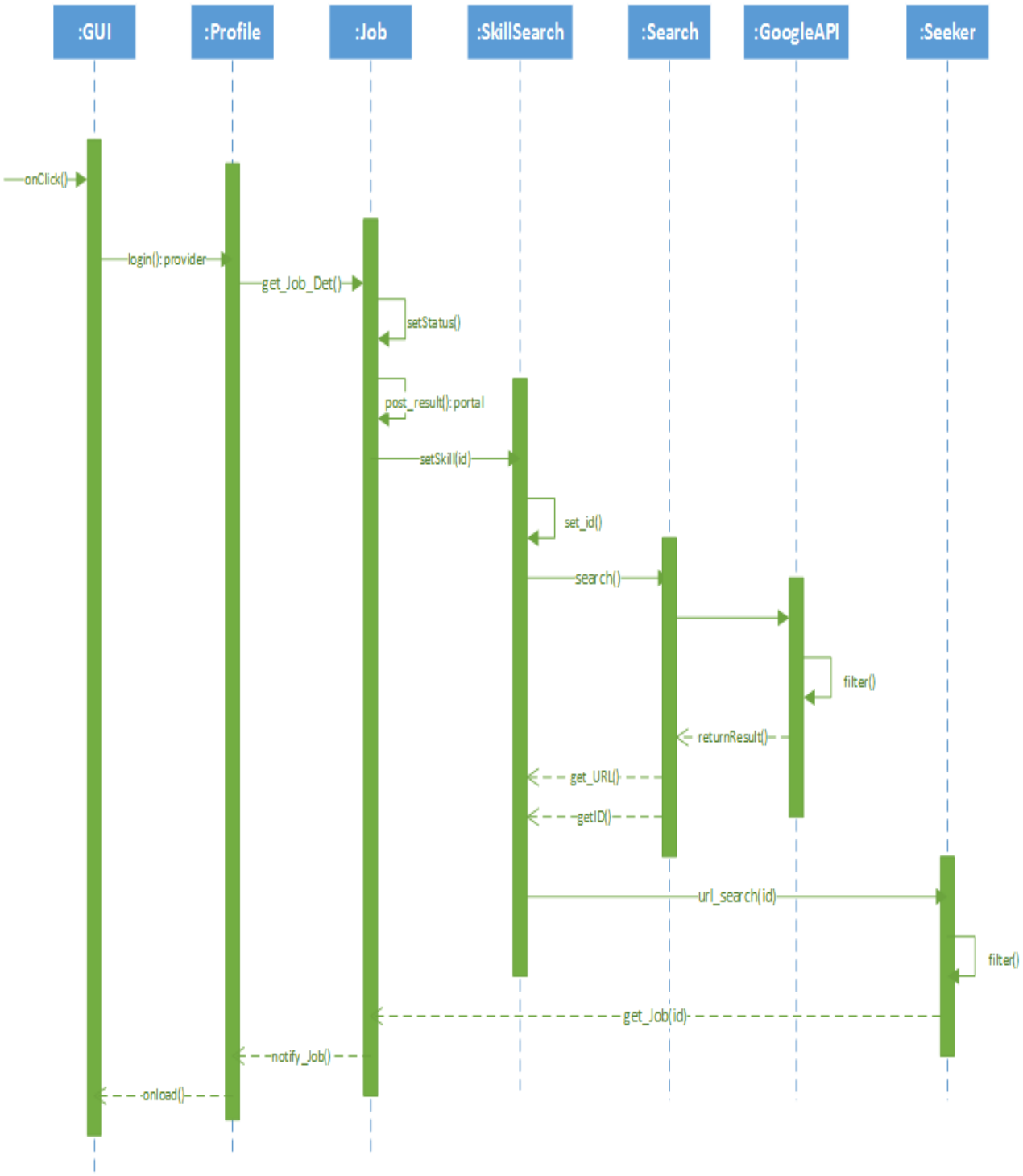


Figure 4-12: Job Posting and Notification Sequence Diagram

4.5.2 Activity Diagrams Sign Up Activity

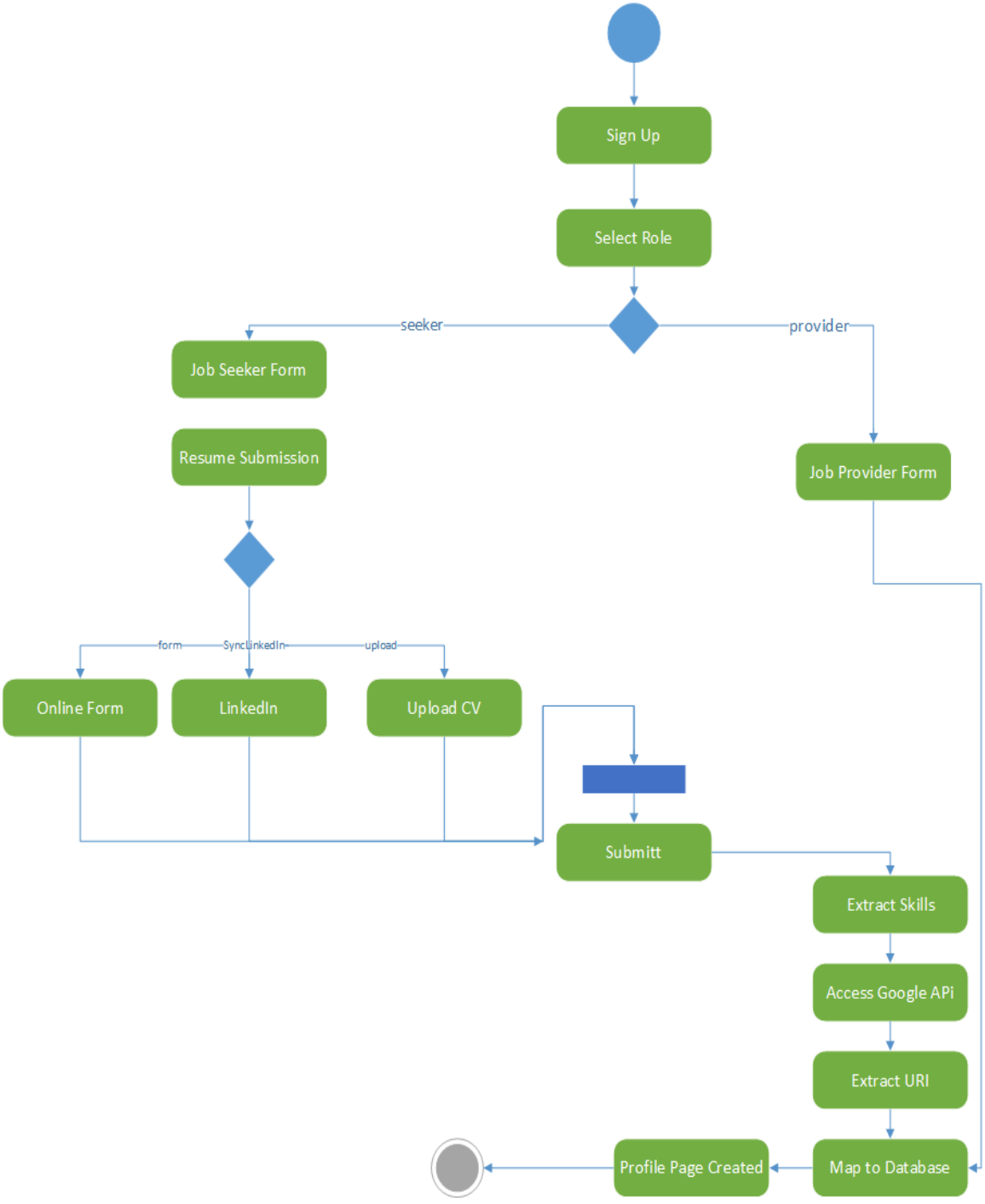


Figure 4-13: Sign Up Activity

4.5.2.1 Post Job and Notification Activity

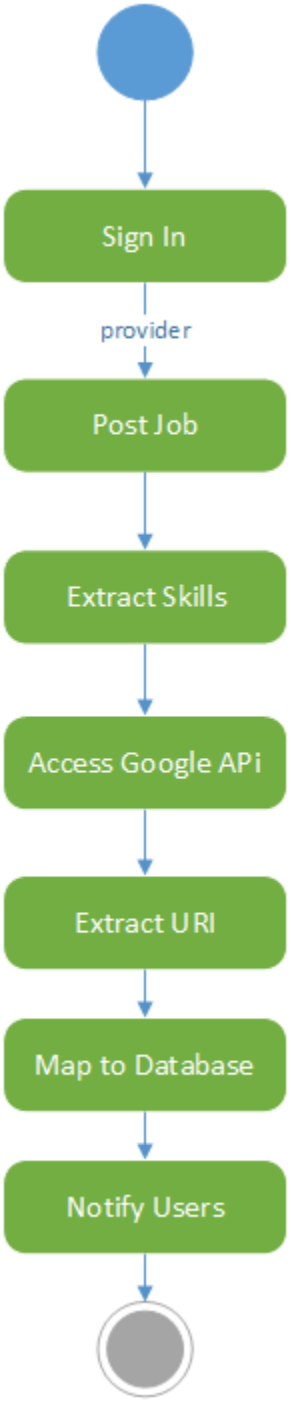


Figure 4-14: Post Job and Notification

4.5.2.2 Search Activity

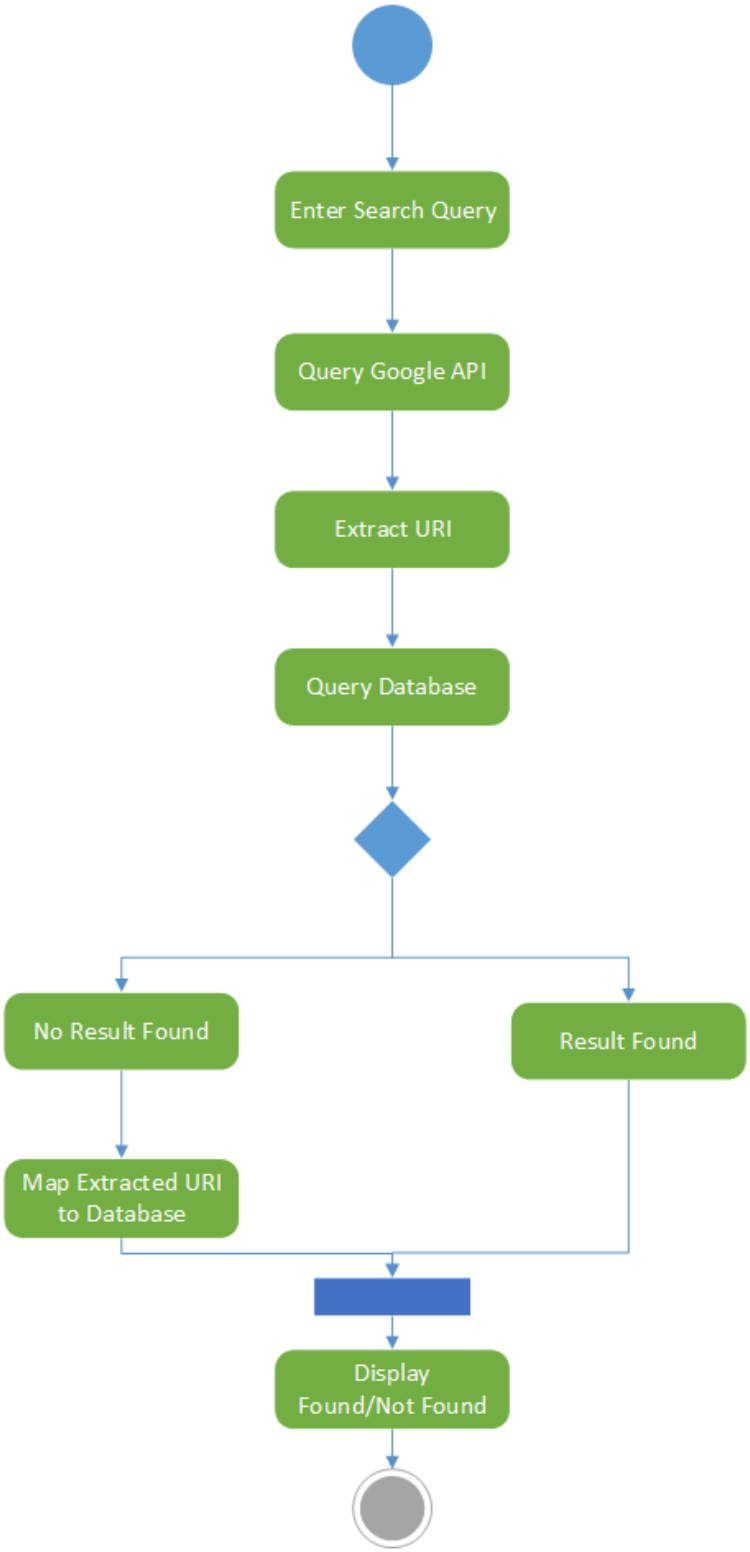


Figure 4-15: Search Activity Diagram

4.6. Data Design View

4.6.1 Database Diagram

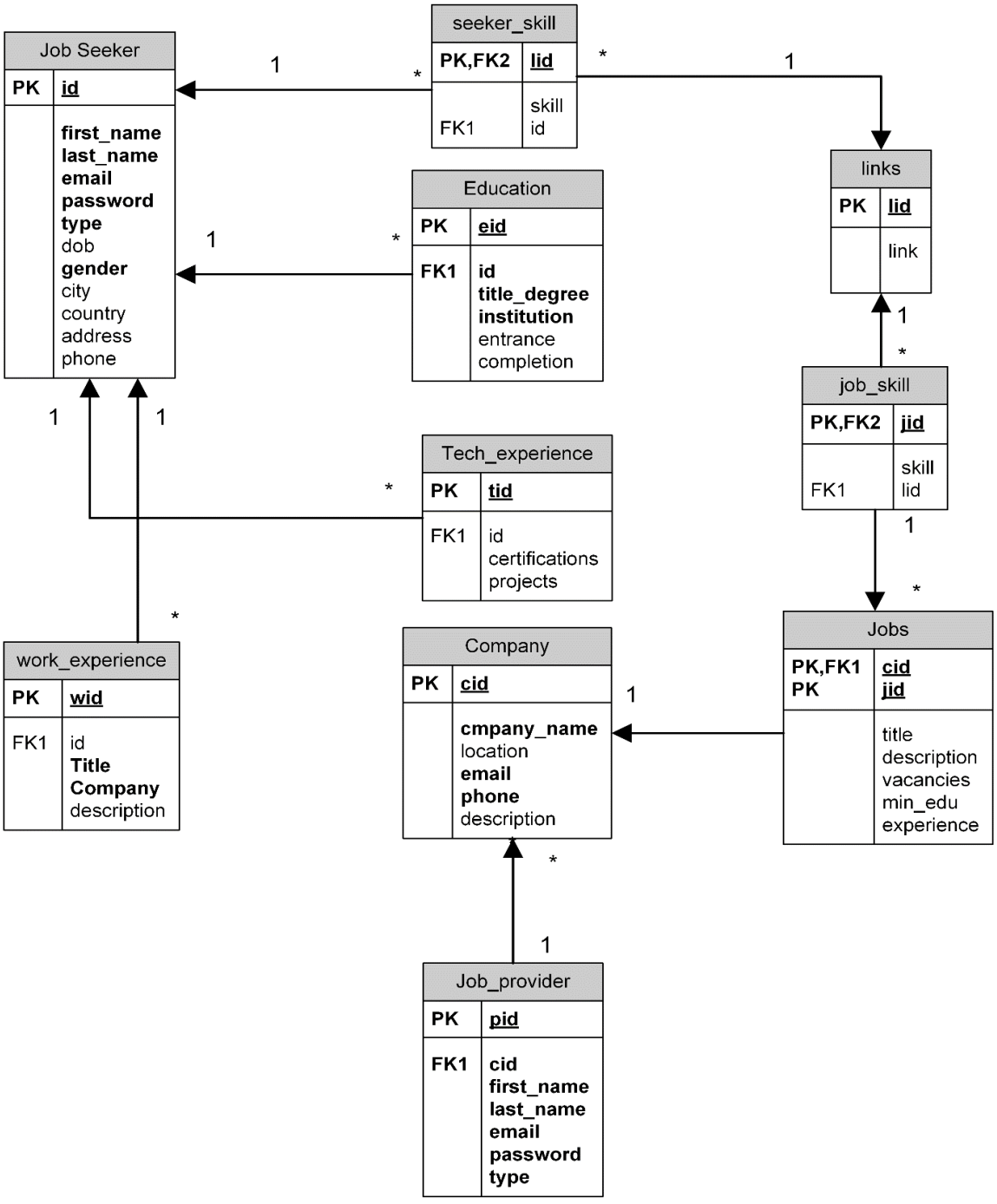


Figure 4-16: Database Diagram

4.7. User Interface Design

4.7.1 Main Window

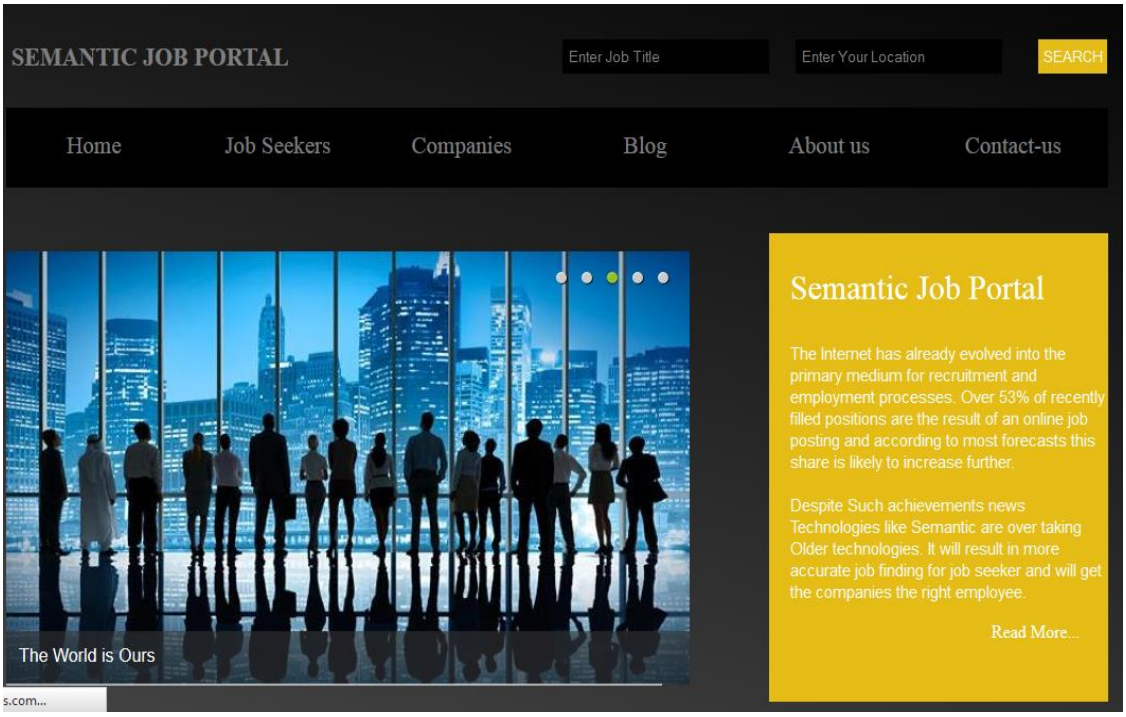


Figure 4-17: Main Window

4.7.2 Sign Up Window

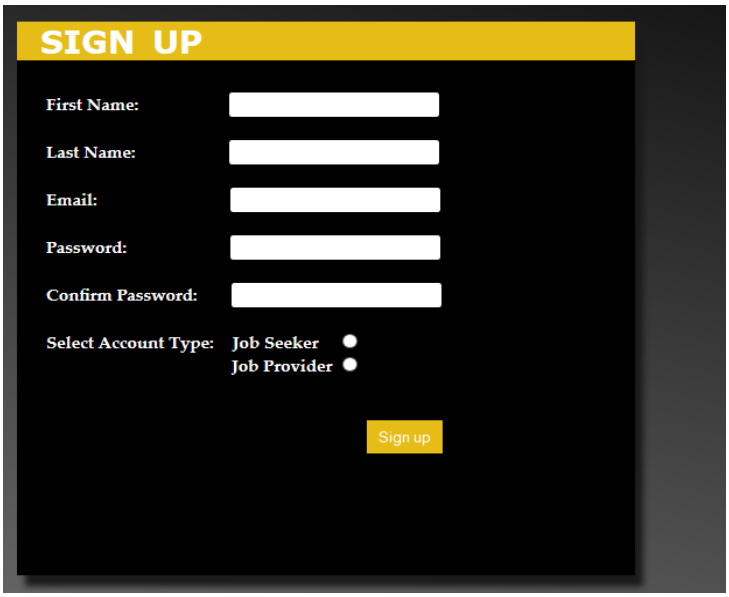


Figure 4-18: Sign Up

4.7.3 Getting Information

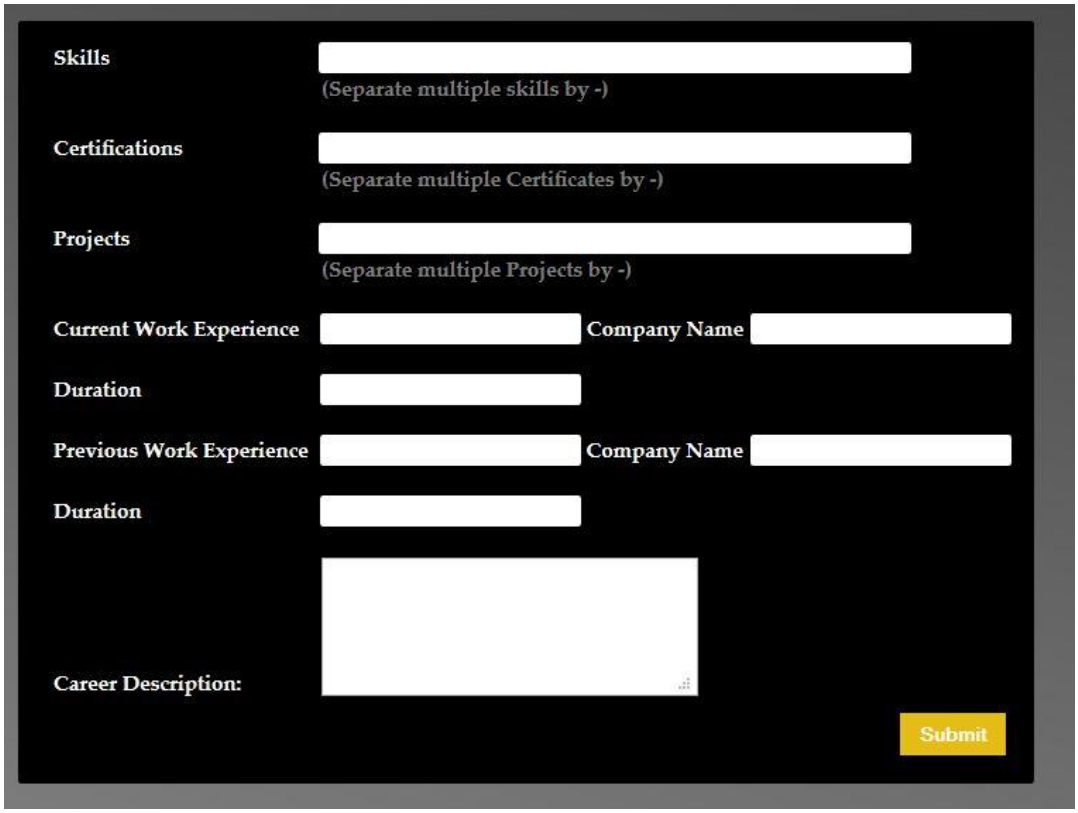


Figure 4-19: Gathering Information

4.7.4 Setting Company Profile and Job Posting

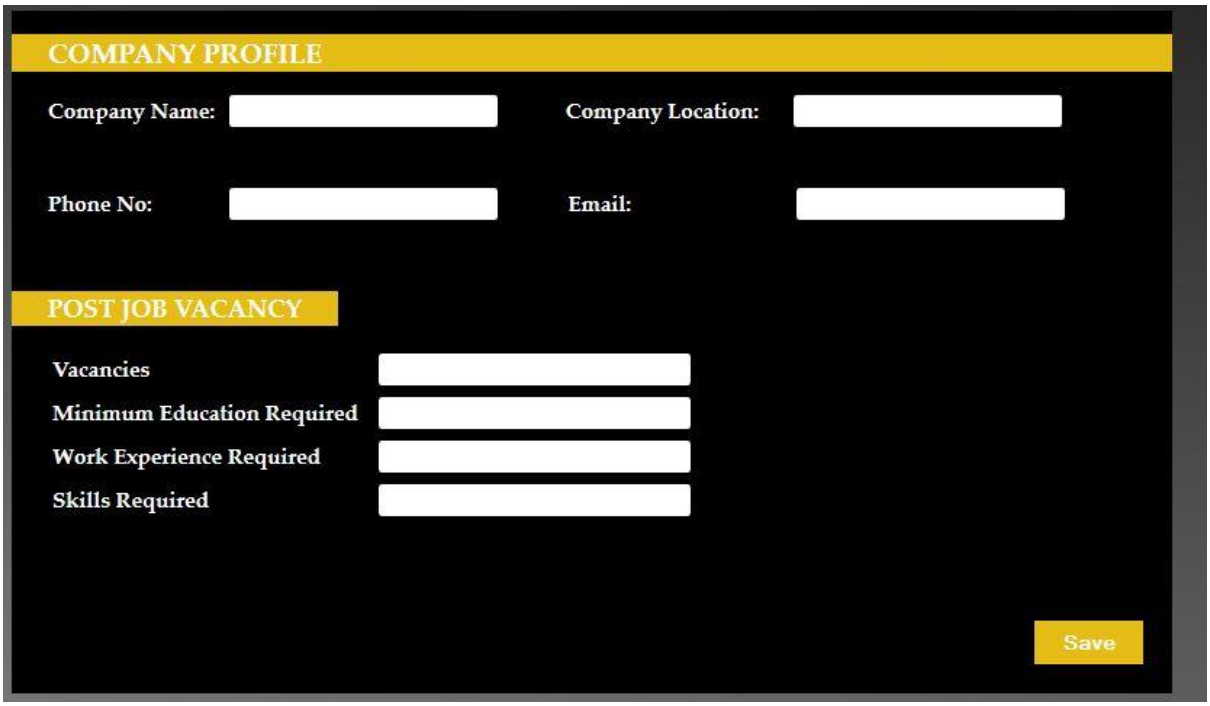


Figure 4-20: Job Posting and Notifying

5. System Implementation

5.1. Tools and Technologies

5.1.1 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data.

We have opted to develop the business logic in PHP as it has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

5.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. We have used CSS extensively throughout our project to create an appealing web interface.

5.1.3 JQuery

JQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. JQuery has been used to create animations in the web interface such as animated menu bars.

5.1.4 JavaScript

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It has been used for form validation as well as other tasks throughout the course of the project.

5.1.5 HTML

Short for HyperText Markup Language, the authoring language used to create documents on the World Wide Web. HTML is similar to SGML, although it is not a strict subset. HTML defines the structure and layout of a Web document by using a variety of tags and attributes.

5.1.6 Google API

Google APIs (or Google AJAX APIs) is a set of JavaScript APIs developed by Google that allows interaction with Google Services and integration of rich, multimedia, search or feed-based Internet content into web applications. Google API helps us in extracting URI's of skills that are further mapped to database for querying.

5.1.7 Linkedin Profile API

The Profile API returns a member's LinkedIn profile. You can use this call to return one of two versions of profile:

Standard: Displays the profile the requestor is allowed to see. The specific content will depend on the privacy settings of the profile owner, the relationship (degree separation or groups in common) between the owner and requestor, and in rare cases, the privacy settings of the requestor.

Public: Returns the public profile. The fields returned are only determined by the privacy settings of the profile owner.

5.1.8 PDF Parser

PDF parser is a free PHP library that helps to parse PDF files and extract text of the pdf file, metadata and other useful attributes. This library helps us in extracting our required data from the CV's uploaded by the job providers.

5.1.9 Bootstrap Framework

Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It has been used in the project to create a responsive web i.e. the interface can adapt to different platforms such as mobile phones and tablets.

5.2. Implementation

5.2.1 Getting Information User Information

Following are some of the ways the system offers to get user's information

5.2.1.1 LinkedIn

This method allows the user (Job Seeker only) to synchronize their LinkedIn profiles with the system to fill up the forms. The system can extract seeker's data from LinkedIn., after authentication is achieved, the received data includes skills, job experiences and educations.

Authentication is performed by LinkedIn API itself. Authorization process starts at

```
// Authentication request
$params = array('response_type' => 'code',
                'client_id' => API_KEY,
                'scope' => SCOPE,
                'state' => uniqid('', true), // unique long string
                'redirect_uri' => REDIRECT_URI,
            );
$url = 'https://www.linkedin.com/uas/oauth2/authorization?' . http_build_query($params);
// Redirect user to authenticate
header("Location: $url");
```

This request returns authorization code. after getting code we get access token as:

```
if (isset($_GET['code'])) then:
$params = array('grant_type' => 'authorization_code',
                                'client_id' => API_KEY,
                                'client_secret' => API_SECRET,
                                'code' => $_GET['code'],
                                'redirect_uri' => REDIRECT_URI,
                        );

        // Access Token request
        $url = 'https://www.linkedin.com/uas/oauth2/accessToken?' . http_build_query($params);
// Retrieve access token information
        $response = file_get_contents($url, false, $context);

        // Native PHP object, please
        $token = json_decode($response);
```

after getting access token we set session as:

```
// Store access token and expiration time
        $_SESSION['access_token'] = $token->access_token; // guard this!
        $_SESSION['expires_in']   = $token->expires_in; // relative time (in seconds)
```

$_SESSION['expires_at'] = time() + $_SESSION['expires_in']; // absolute time

After that we extract data as:

$user = fetch('GET', '/v1/people/~:(id,first-name,last-name,headline,industry,skills,educations,courses,positions)');

Now we can extract data from response returned as:

```
foreach($user->educations->values as $value) {
        $title=$value->degree;
        $institution=$value->schoolName;
        $start=$value->startDate->year;
        $end=$value->endDate->year;
        $degree=$value->fieldOfStudy;
```

5.2.1.2 PDF Parser

Once the user has selected to upload CV from the defined template available. The following code helps in extracting skills from the CV.

```
// Include Composer autoloader if not already done.
include 'pdfparser-master/vendor/autoload.php';
//include 'pdfparser-master\src\Smalot\PdfParser\Parser.php';
 // Parse pdf file and build necessary objects.
   $parser = new \Smalot\PdfParser\Parser();
   $pdf    = $parser->parseFile('cv.pdf');
```

$text = $pdf->getText();

$regex_pattern = "/(Key Skills)([a-zA-Z0-9, \n\r:+@#%()\[\]])(Education)/";*

preg_match_all($regex_pattern,$text,$matches);

echo $matches[2][0];

5.2.1.3 Online Forms

Job seekers and Job providers can upload their data online through forms. Profiles are maintained for both job seekers and companies.

5.2.2 Skill extraction

Skills are separately extracted from job seeker's data and are stored separately in database.

5.2.3 Skill Linking

Skills when searched are passed to Google API to get unique URLs as, Google serves as our base ontology to achieve semantics. Google works through crawling and linking related pages to each other. We have used this feature of google for implementing semantics.

$url = "http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=".$skill;

$body = file_get_contents($url);

$json = json_decode($body);

$regex_pattern = "#(http://en\.wikipedia\.org)#";

if(preg_match($regex_pattern,$json->responseData>results[$x]->url))

$link=$json->responseData->results[$x]->url;

This link is stored as a reference to every new skill entered by user irrespective of spells.

5.2.4 Search

When user enters skills for the purpose of either find a job (incase of job seeker) or to find a prospect employee (incase of job provider) then it makes use of Google API which returns the URL of the queried skill. After the URL is retrieved, the database is queried for all skills related to that URL. The skills corresponding to the URL are then searched in profiles of user and are returned.

C-Sharp, C Sharp and C# are returned with same URL *en.wikipedia.org/wiki/C_Sharp_(programming_language* thus the result produced for each of these search queries is same.

5.2.5 Responsive Web

```
<section class="edu">
                    <form method="post" action="insert_edu.php">
                    <div class="row" >
                        <div class="col col-lg-2">

                                <label class="lab1"
for="myname">Education</label>

                        </div>
```

```
                    </div>

                    <div class="row"> <!--Defines a row in the grid Layout -->


                            <div class="logo col col-lg-10"> <!--Within that row of
each grid columns are defined -->


                                    <label class="col col-lg-2 control-label
lab2" for="myname">Degree</label>


                                    <input class="col col-lg-4" type="text"
name="degree" id="myname"  placeholder="Enter Degree" required>




                                    <label class="col col-lg-2 control-label
lab2" for="myname">Institution</label>


                                    <input class="col col-lg-4" type="text"
name="uni" id="myname"  placeholder="Institution Name" required>


                            </div>



                    </div>

                    </form>

</section>
```

6. Testing and Result Analysis

6.1. Introduction to test Plan

This software test plan is to provide the description of test cases for Semantics Based Intelligent Job Portal describing the scope and approach of intended test activities. This document will describe the test cases for different features of the. Software Requirement and Specification document of Semantics Based Intelligent Job Portal supports this Software Test Plan.

6.1.1 Test Items

Following are test items and their version:

Test Item Name	Test Item Version #	Test Type
Overall System	Ver.1	Black-box
Sign Up	Ver.1	Black-box
Sign In	Ver.1	Black-box
Search Module	Ver.1	Black-box
Responsive Web	Ver.1	Black-box
LinkedIn Synchronization	Ver.1	Black-box
Google API	Ver.1	White-box
Skill Extractor	Ver.1	White-box
Skill Mapper	Ver.1	White-box
PHP Mailer	Ver.1	White-box
PDF Parser	Ver.1	White-box

6.1.2 Features to be tested

Feature	Overview
Overall System	a. Access website with URL b. Register as job seeker or provider c. Search for desired skills
Sign Up	a. Register as Job Seeker or provider b. Create Profile Page
Sign In	a. Verify credentials b. Redirected to Profile Page
Search Module	a. Login as Job Seeker b. Search Jobs based on Skills c. Login as Job Provider d. Search prospect employees based on Skills
Responsive Web	a. Resize browser window b. Open on mobile phones c. Open on tablets
LinkedIn Synchronization	a. Select Sync with LinkedIn option during registration b. Enter credentials to be verified by LinkedIn

	c. Check field for retrieved data
Google API	a. Enter Search Query b. Check Links database of Retrieved URL
Skill Extractor	a. Register as seeker b. Upload data c. Check database for entered skills
Skill Mapper	a. Upload Skills b. Check Links table and Skill table in database
PHP Mailer	a. Select Forget Password b. Check SMTP server response
PDF Parser	a. Download PDF template b. Upload the PDF file (CV) c. Check database for changes in the tables

6.1.3 Features Not to Be Tested

All the features of Semantics Based Intelligent Job Portal need to be tested.

6.2. Approach

The overall approach of this test plan is Grey Box testing i.e. hybrid of Black Box and White Box testing. Testing approach will be same for all features of the system. System

will be tested in different perspectives such as GUI testing etc. No separate tool will be used to test the system. Testing will be comprehensive so that to ensure high quality of the system. To judge the comprehensiveness a record will be maintained of the features which have been tested completely.

6.2.1 Item Pass/Fail Criteria

The entrance criteria's for each phase of testing must be met before the next phase can commence. Any test item will be declared pass if it conforms to the requirements specified in the SRS and fail if it does not.

6.2.2 Suspension Criteria and Resumption Requirements

The only case in which test activity needs to be suspended is failure of a feature. In that case the development team will be reported to fix the error of that feature after which the testing of the whole system will start over again.

6.2.3 Environmental Needs

The system should be up and running on the server.

6.2.4 Schedule

Testing process has taken more than 1 calendar month.

6.3. Risks and Contingencies

Risk lies in incomplete execution of test cases and ignoring little bugs of the systems which may result in big faults and failures. Also the testing process needs to be done in

the specified time. If not, the whole schedule of development team may get disturbed resulting in lack of customer's trust on development team.

6.4. Functional Testing

6.4.1 Black-box

The software program or system under test is viewed as a "black box". The selection of test cases for functional testing is based on the requirement or design specification of the software entity under test. Functional testing emphasizes on the external behaviour of the software entity.

6.4.1.1 Test Case #A01

Name	Overall System Testing
Description	It will test the overall functionality of the system and the working of the system.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Enter the URL of the system to access the system and user details for registration
Steps	1. Enter website URL to gain access 2. Choose the option to Register. 3. Register as Job Provider once and Job Provider for the other by providing details in text boxes and selecting necessary option and check for invalid inputs

	4. Search the portal on the basis of skills
Expected Output	The portal should be accessed with the user being registered by providing valid details and selecting valid option. The portal should allow the user to search on the basis of skills.
Output	The results conformed to the expected results. The user was unable to access the web by the given URL and was unable to register and search if the details provided were invalid.
Result	Pass

Table 6-1: Test Case- Overall System Testing

6.4.1.2 Test Case #A02

Name	Sign Up
Description	It will test the sign up procedure for the system
Precondition	System is correctly deployed, bug free and working properly.
Inputs	The user fills the form.
Steps	1. Access the sign up page. 2. Enter Invalid details 3. Complete Sign up with valid details 4. Sign up again with same credentials
Expected Output	The system should not allow sign up with invalid credentials such as invalid email address, phone number date of birth etc. It should also not allow the user to register with same credentials again
Output	The results conformed to the expected results.
Result	Pass

Table 6-2: Test Case-Sign Up

6.4.1.3 Test Case#A03

Name	Sign In
Description	It will test the sign ip procedure for the system
Precondition	System is correctly deployed, bug free and working properly.
Inputs	User enters email id and corresponding password.
Steps	1. Access the sign in page. 2. Enter Invalid details. 3. Enter valid credentials.
Expected Output	Error Message shall be shown for step 2. For step 3 the user will directed to the profile page.
Output	Error was shown for invalid combination of email and password. The user was directed to his profile when correct credentials were entered.
Result	Pass

Table 6-3: Test Case-Sign In

6.4.1.4 Test Case#A04

Name	Search
Description	It will search module for the system
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Enter skills in search bar.
Steps	1. Login as Job Provider or Seeker 2. For Job Provider the system should return candidates seeking job having skill same as queried by the job provider. 3. For Job Seeker the system should return companies looking for a candidate having queried skill. 4. Repeat steps 2 and 3 to check consistency of results 5. Also repeat steps for semantically similar skills.
Expected Output	System should return same results every time queried. The results for semantically same words should not differ.
Output	The output conformed to the expected results
Result	Pass

Table 6-4: Test Case-Search

6.4.1.5 Test Case#A05

Name	Responsive Web
Description	This test case will test that the system perform well on other devices and platforms as well
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Resize the windows. Use on other devices.
Steps	1. Open the portal in browser. 2. Resize the browser to different dimension. 3. Open the portal on other devices such as cellphones and tablets. 4. Repeat the step for all the pages
Expected Output	All the pages must re-adjust themselves according to the dimensions of the screen.
Output	The output conformed to the expected results
Result	Pass

Table 6-5: Test Case-Responsive Web

6.4.1.6 Test Case#A06

Name	LinkedIn Synchronization
Description	This test case will check the user details are synchronized with the LinkedIn profile of the user.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Choose *Sync with LinkedIn* option
Steps	1. Click Register 2. Enter Details and sign up to get directed to insert more details page. 3. Choose option to synchronize with LinkedIn. 4. Enter Valid LinkedIn password to get authenticated by LinkedIn
Expected Output	The form should automatically fill up once the data is received.
Output	The session timed out.
Result	Failed

Table 6-6: Test Case-LinkedIn Synchronization (a)

6.4.1.7 Test Case#A07

Name	LinkedIn Synchronization
Description	This test case will check the user details are synchronized with the LinkedIn profile of the user.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Choose *Sync with LinkedIn* option
Steps	1. Click Register 2. Enter Details and sign up to get directed to insert more details page. 3. Choose option to synchronize with LinkedIn. 4. Enter Valid LinkedIn password to get authenticated by LinkedIn
Expected Output	The form should automatically fill up once the data is received.
Output	The personal data retrieved is visible in the form
Result	Pass

Table 6-7: Test Case-LinkedIn Synchronization (b)

6.4.2 White-box Testing

White-box testing is a method of testing software that tests internal structures or workings of an application. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The selection of test cases is based on the implementation of the software entity, on module by module basis.

6.4.2.1 Test Case#B01

Name	Google API
Description	It will test whether the API is integrated with the system and is producing valid result.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Enter Search Query
Steps	1. Click on Search bar 2. Enter search query 3. Check the database Links table.
Expected Output	URL corresponding to entered skill is found in database
Output	The URL is not found
Result	Fail

Table 6-8: Test Case-Google API (a)

6.4.2.2 Test Case#B02

Name	Google API
Description	It will test whether the API is integrated with the system and is producing valid result.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Enter Search Query
Steps	4. Click on Search bar 5. Enter search query 6. Check the database Links table.
Expected Output	URL corresponding to entered skill is found in database
Output	The output corresponds to the expected output
Result	Pass

Table 6-9: Test Case-Google API (b)

6.4.2.3 Test Case#B03

Name	Skill Extractor
Description	This test case will test if skills are correctly mapped on to the database
Precondition	System is correctly deployed, bug free and working properly.
Inputs	User enters skills
Steps	1. User registers 2. Inserts personal details 3. Saves when complete
Expected Output	Each skills should be individually mapped onto the database.
Output	In some cases two or more skills got mapped onto the same row.
Result	Fail

Table 6-10: Skill Extractor (a)

6.4.2.4 Test Case#B04

Name	Skill Extractor
Description	This test case will test if skills are correctly mapped on to the database
Precondition	System is correctly deployed, bug free and working properly.
Inputs	User enters skills
Steps	1. User registers 2. Inserts personal details 3. Saves when complete
Expected Output	Each skills should be individually mapped onto the database.
Output	The output conforms to the expected output.
Result	Pass

Table 6-11: Test Case-Skill Extractor (b)

6.4.2.5 Test Case#B05

Name	Skills Mapper
Description	This test case will ensure that skills are mapped correctly to the corresponding URL.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	Skills entered by user. URL retrieved by Google API
Steps	1. Enter Skills 2. Google API returns URL of the entered skills.
Expected Output	Skills are mapped to correct corresponding URL retrieved by Google API in the Links table of database
Output	The output conforms to the expected output
Result	Pass

Table 6-12: Test Case-Skill Mapper

6.4.2.6 Test Case#B06

Name	PHP Mailer
Description	This test case will test the working of mailing module.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	User email address
Steps	1. Register for the portal 2. Choose forget password option when logging in 3. Shortlist a candidate for a job position
Expected Output	For each of the above task and email should be generated to the user. For step 2 an email containing new password of the user should be generated with the change of password in database. For step 3 the job for which a job seeker is shortlisted should be generated to the registered email of job seeker.
Output	No change of password in the database for step 2.
Result	Fail

Table 6-13: Test Case-PHP Mailer (a)

6.4.2.7 Test Case#B07

Name	PHP Mailer
Description	This test case will test the working of mailing module.
Precondition	System is correctly deployed, bug free and working properly.
Inputs	User email address.
Steps	1. Register for the portal 2. Choose forget password option when logging in 3. Shortlist a candidate for a job position
Expected Output	For each of the above task and email should be generated to the user. For step 2 an email containing new password of the user should be generated with the change of password in database. For step 3 the job for which a job seeker is shortlisted should be generated to the registered email of job seeker.
Output	All steps produced desired output.
Result	Pass

Table 6-14: Test Case-PHP Mailer (b)

6.4.2.8 Test Case#B08

Name	PDF Parser
Description	It will test the working for PDF Parser
Precondition	System is correctly deployed, bug free and working properly.
Inputs	A PDF format CV.
Steps	1. Download template 2. Fill it the template 3. Save it in PDF format 4. Upload the template.
Expected Output	All the required information should be extracted from the CV specially skills.
Output	The output conforms to expected output
Result	Pass

Table 6-15: Test Case-PDF Parser

7. Conclusion and Future Work

The goal of Semantics Based Intelligent Job Portal is to create a platform where the abilities of the users to search for something is not restricted by the way they write the query.

The system instead of searching the conventional way, that is, by doing a keyword search on the skills, took another road to produce better results. Semantics Based Intelligent Job Portal allowed the user to query for their intended jobs or employees by searching the database irrespective of the way the type a particular skill. The project is a great step forward to help companies search for relevant human resource and employee the best out of the filtered results produced.

The scope of this project was aimed cater skills based search, However in the domain of Job Portals this scope can be extended to the education, certification and various other attributes associated with a user. The concept can be taken to a whole new level if the domain is generalized from Job Hunting to Talent Hunting where many other criteria and attributes could be brought into consideration enabling us to generate a more meaningful result.

APPENDIX A: GLOSSARY

Activity diagram: An analysis model that shows a dynamic view of a system by depicting the flow from one activity to another. Similar to a flowchart.

Assumption: A statement that is believed to be true in the absence of proof or definitive knowledge.

Class: A description of a set of objects having common properties and behaviours, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

Class diagram: An analysis model that shows a set of system or problem domain classes and their relationships.

Constraint: A restriction that is imposed on the choices available to the developer for the design and construction of a product.

API (application programming interface): specifies how some software components should interact with each other. In addition to accessing databases or computer hardware, such as hard disk drives or video cards, an API can be used to ease the work of programming graphical user interface components.

Feature: A set of logically related functional requirements that provides a capability to the user and enables the satisfaction of a business objective.

Implementation: Execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

Interface: A point where two systems, subjects, organizations, etc., meet and interact.

Precondition: A Condition that must be satisfied before a use case may begin.

Process: A sequence of activities performed for a given purpose. A process description is a documented definition of those activities.

Response: A reaction, as that of an organism or a mechanism, to a specific stimulus.

Scope: The portion of the ultimate product vision that the current project will address. The scope draws the boundary between what's in and what's out for the project.

Stimulus: Something causing or regarded as causing a response.

Use case: A description of an interaction between an actor and a system that results in an outcome that provides value to the actor.

Use case diagram: An analysis model that identifies the actors who can interact with a system to accomplish valuable goals and the various use cases that each actor will perform.

User: A customer who will interact with a system either directly or indirectly (for example, using outputs from the system but not generating those outputs personally). Also called end user.

APPENDIX B: REFERENCES

1. Artem Chebotko, Wayne State University. Literature Survey on the Semantic Web and Search. In March 2005.
2. Malgorzata Mochol, Holger Wache, Lyndon Nixon. Improving the recruitment process through ontology-based querying. In Workshop on *Applications and Business Aspects of the Semantic Web* SEBIZ 2006 November 6, 2006 Athens, Georgia, USA.
3. Malgorzata Mochol, Radoslaw Oldakowski and Ralf Heese. Ontology based Recruitment Process. *dbis.informatik.huberlin.de/fileadmin/research/papers/conferences/2004-semtech-mochol.pdf*
4. Christian Bizer, Ralf Heese, Malgorzata Mochol, Radoslaw Oldakowski, Robert Tolksdorf, Rainer Eckstein. The Impact of Semantic Web Technologies on Job Recruitment Processes. *www.inf.fu-berlin.de/users/mochol/papers/wi2005.pdf*
5. Mochol, M. und Paslaru Bontas. Practical Guidelines for Building Semantic eRecruitment Applications. In *International Conference on Knowledge Management,* 2006.
6. Semantic Web. *http://www.w3.org/standards/semanticweb/*
7. Linked Data. *http://www.w3.org/standards/semanticweb/data*