*191071902*

*Srushti Shah*

*T. Y. B. Tech. C. S.*

***Experiment No. 2***

***AIM*:** OOAD USING UML

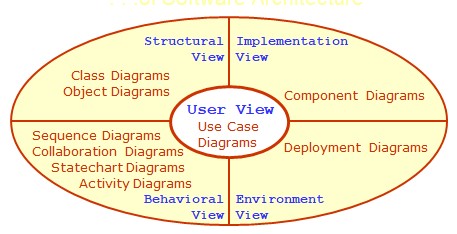
***THEORY:***

(i) Give structural view and Behavioral view.

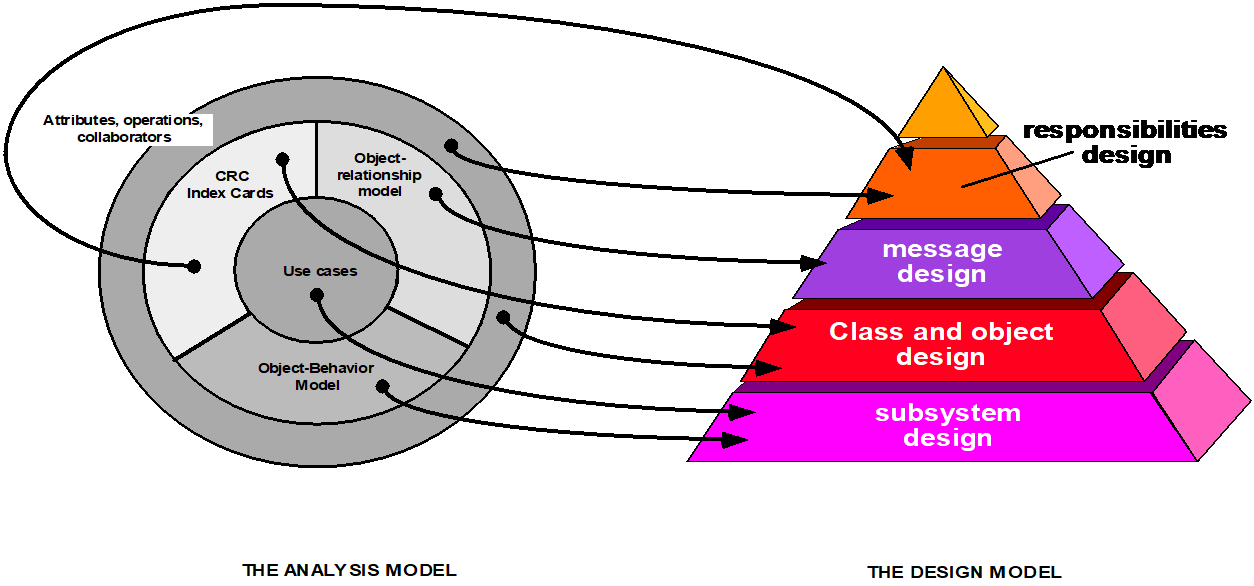
(iii) Implement the project and illustrate implementation view.

(iii) Illustrate the environmental view with specifications.

Use the following dimensions for answering your experiment.



**Map the analysis Model with design Model. Use the following dimensions for the design of object oriented software.**

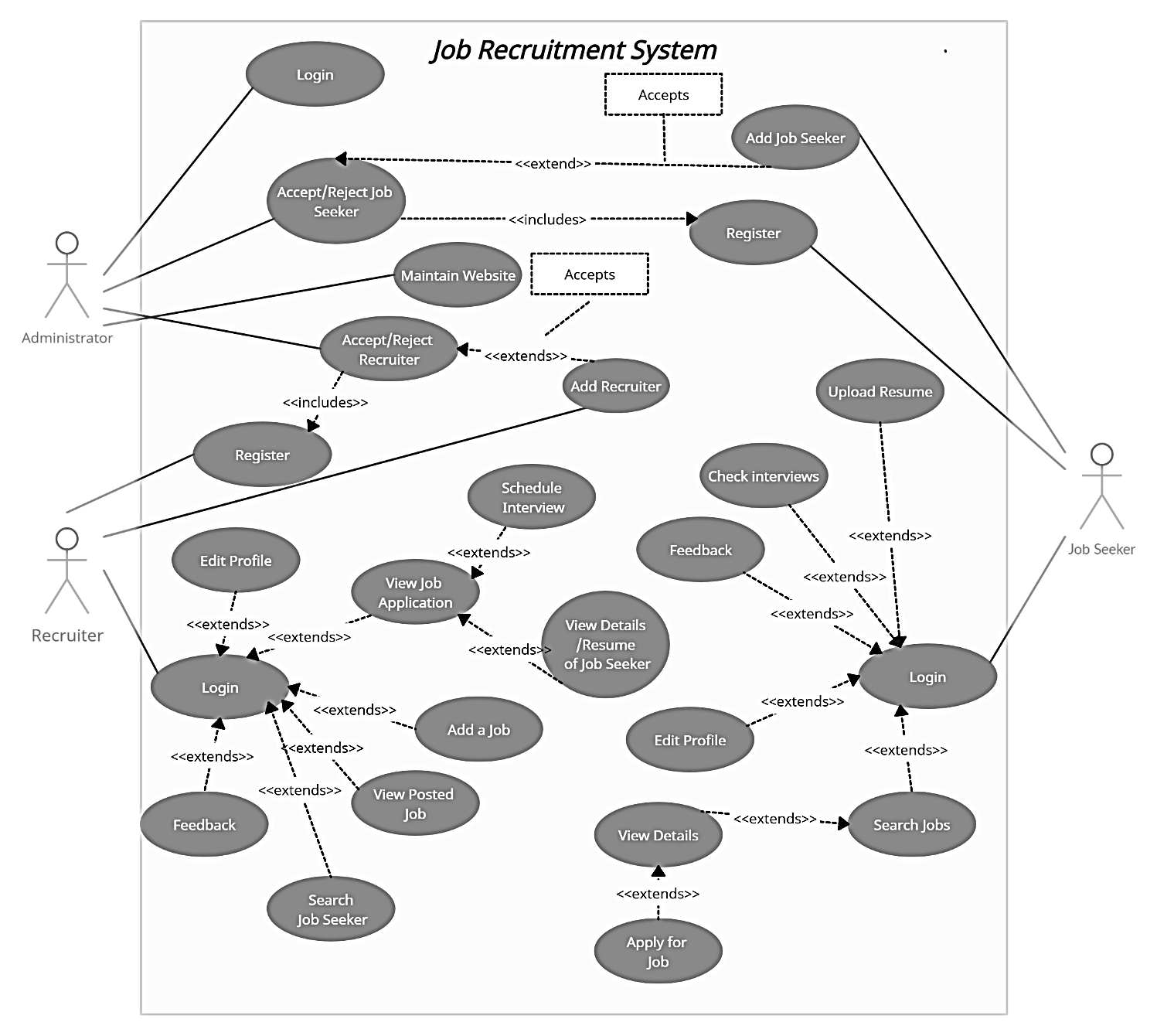


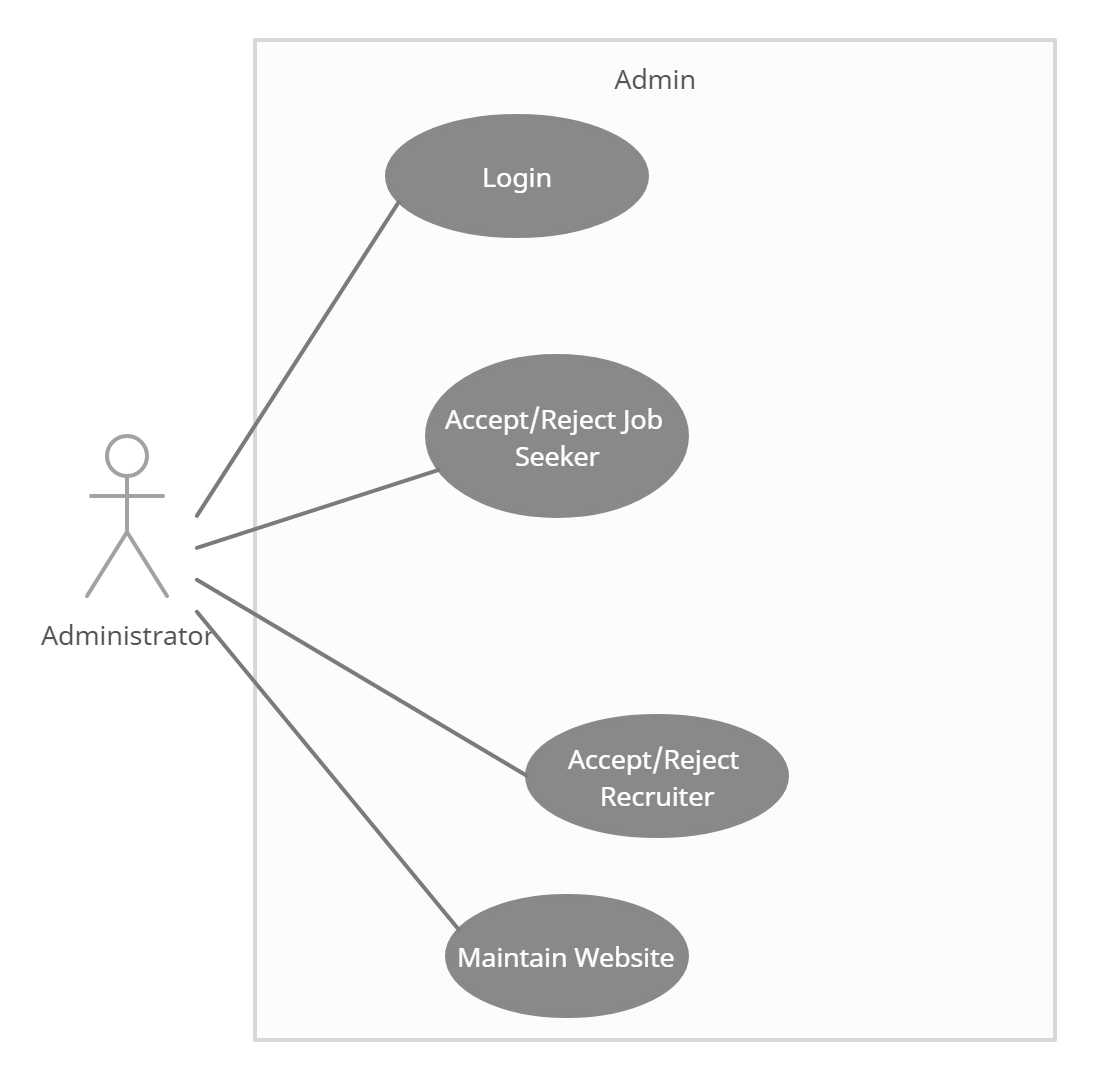
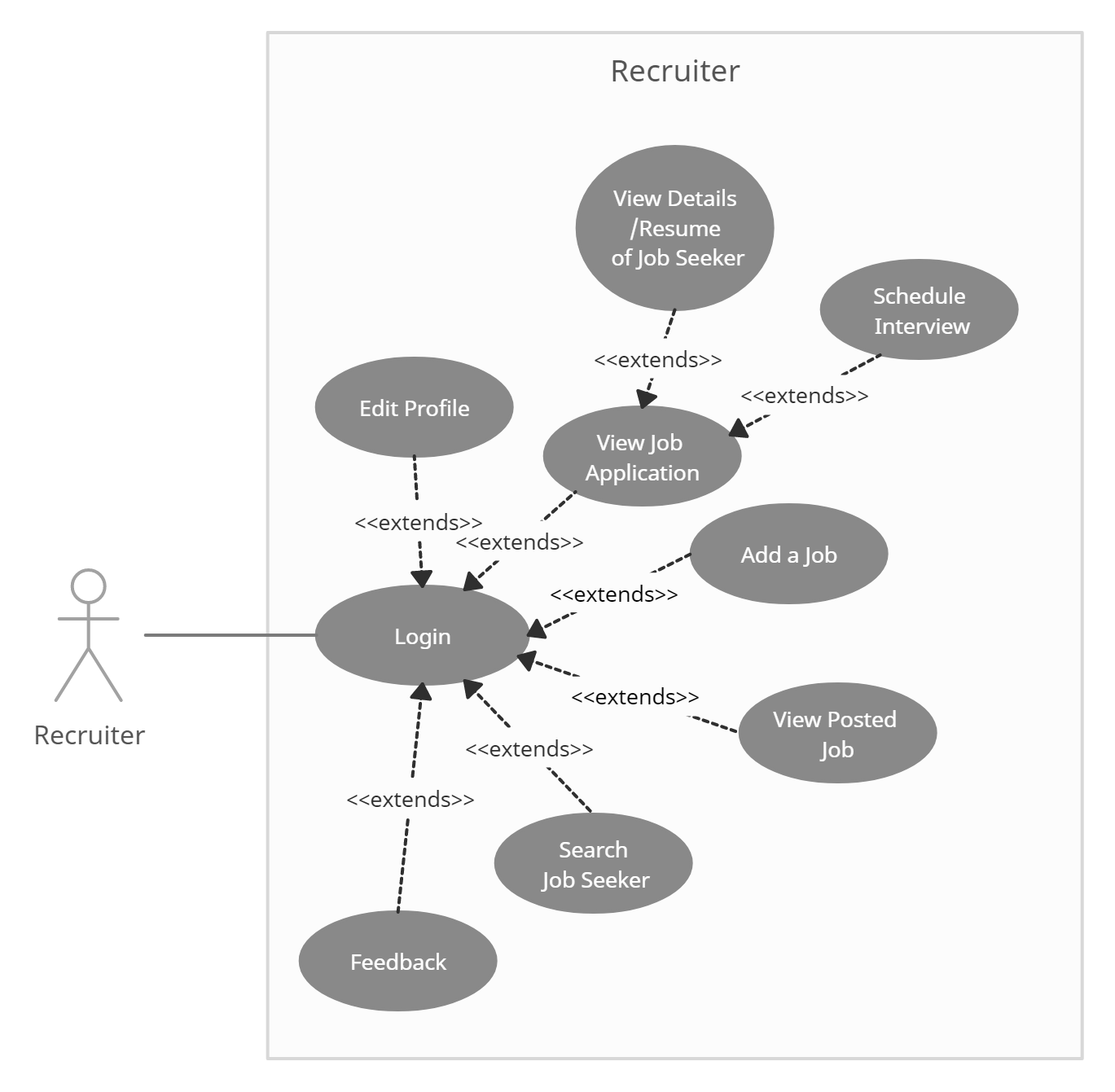
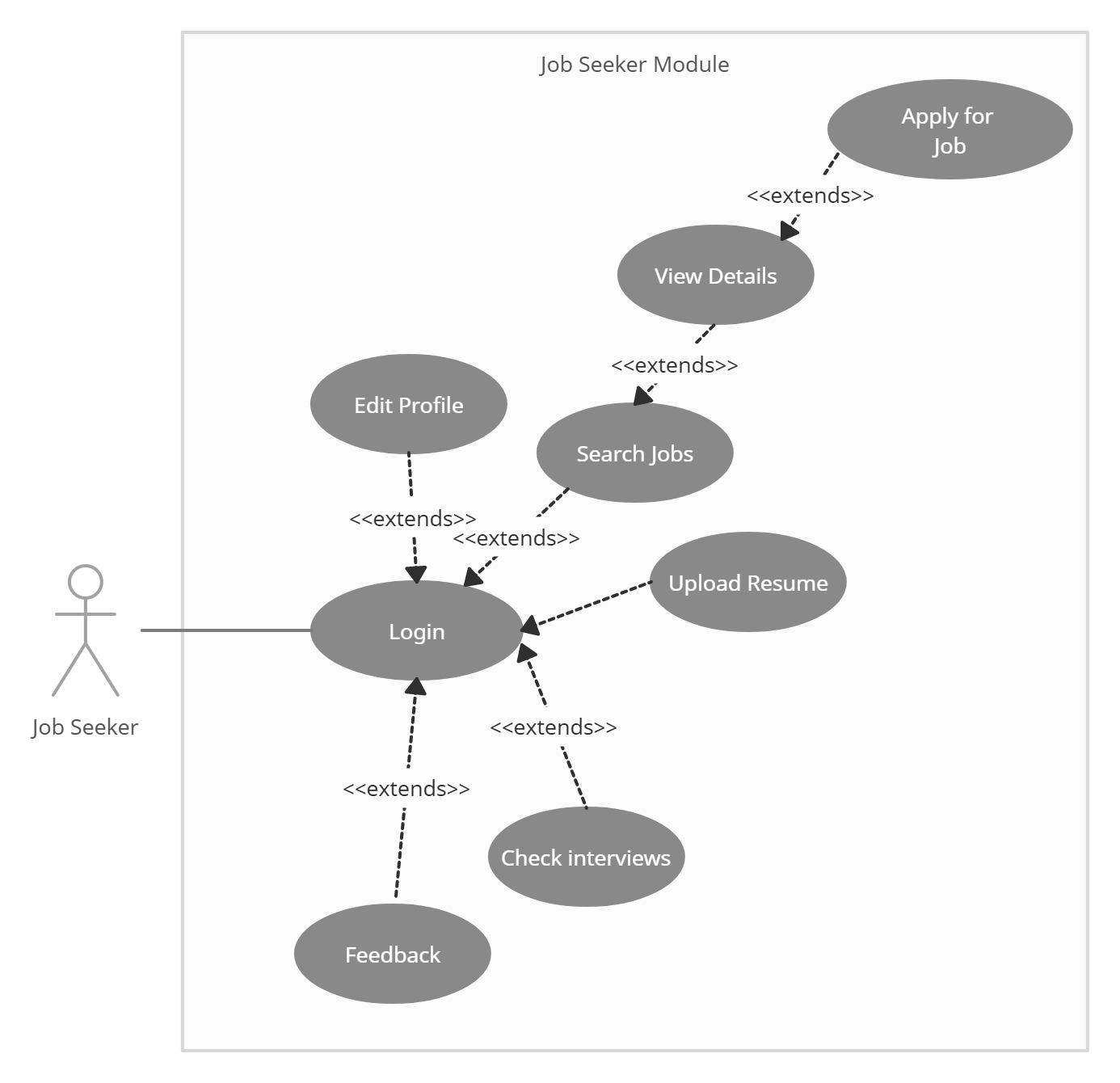
Design the software architecture, algorithm and data structure design and authoring system for the given project.

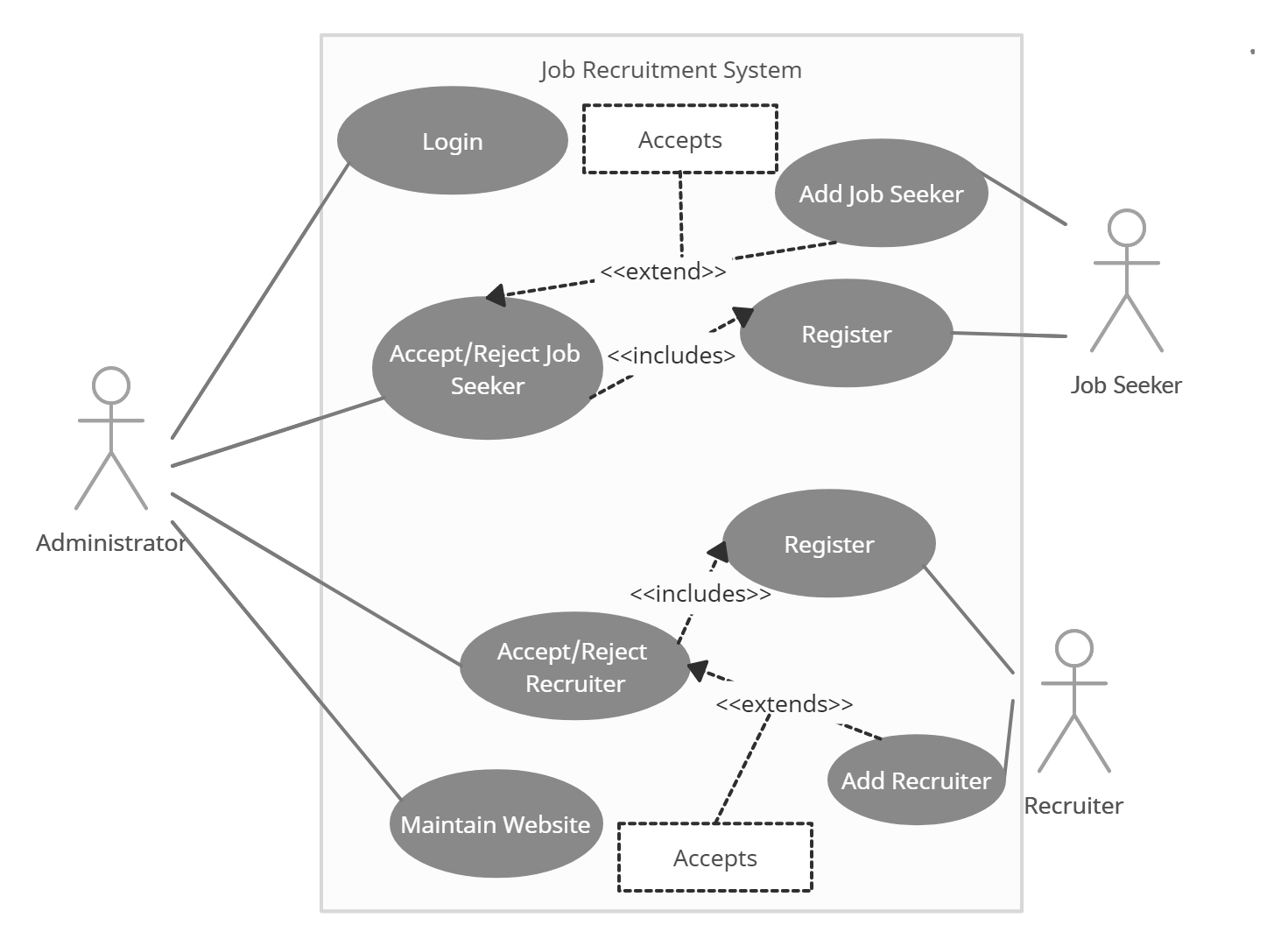
***User View***

***USE CASE DIAGRAM***

1. ***Complete System***

******

1. ***Admin***
2. ***Recruiter***
3. ***Job Seeker***
4. ***Admin Interaction***

******

***USE CASE TEMPLATE***

## Use Case: Registration

|  |  |
| --- | --- |
| Name | Register |
| Summary | A new user (a user which doesn’t have an account) should first register into the system to use it. This feature will allow the user to enroll into the system if the user is a new user and doesn’t already have an existing account. |
| Rationale | The user can access the system |
| Actor | Job Seeker and Recruiter |
| Pre-Condition | None |
| Basic course of Event | 1. User opens the desired module of the system. 2. The system displays GUI the registration. 3. User enters the valid details required to join the system 4. Submits the form |
| Post- Condition | User is registered |
| Alternate Flow | * + - 1. User goes back to the Login Page |
| Exception | Username already taken |

## Use Case: Login

|  |  |
| --- | --- |
| Name | Login |
| Summary | Job seeker can change information into their profiles which include profile edit, delete, and update. |
| Rationale | After registration of the user, the valid user can login into the system |
| Actor | All Users |
| Pre-Condition | The login name and password should match with the login name and password provided while registering.  If the username and or password do not match, the user cannot login successfully into the system. |
| Basic course of Event | * 1. User opens the desired module of the system.   2. The system displays GUI the login form   3. The user writes its username, password and type   4. If valid, user can login into the system. If the username and or password do not match, the user cannot login successfully into the system. |
| Post- Condition | User goes to his/her dashboard |
| Alternate Flow | * + 1. The username and password is blank     2. The type of user is not specified     3. The username and password do not match |

## Use Case: Manage Profile

|  |  |
| --- | --- |
| Name | Manage Profile |
| Summary | User can change information into their profiles which include profile edit, delete, and update. |
| Rationale | If there is unnecessary date and fake profiles that are running on the site then Admin will have rights to remove or delete the profile. |
| Actor | All Users |
| Pre-Condition | Profile and record must exist |
| Basic course of Event | 1. User opens the desired module of the system. 2. The system displays GUI of different pages. 3. User will input the necessary data into the fields. 4. System will update the record as per requirements. |
| Post- Condition | User receives the notification that profile has been approved or profile has been updated. |
| Alternate Condition | * + - 1. The details enter to not validate, e.g. age is <18, date of bitrth is out of range. |

## Use Case: Upload CV

|  |  |
| --- | --- |
| Name | Upload CV |
| Summary | Job seeker can post his/her CV on job portal |
| Actor | Job seeker |
| Pre-Condition | New Jobs and other features should be displayed. |
| Basic course of Event | 1. Job seeker search for a specific job category. 2. The system displays GUI for the new jobs and features. 3. Job seeker clicks on the new jobs option and upload his/her CV there. 4. System will show all the CV on the admin and job seeker end. |
| Post- Condition | CV successfully uploaded. |
| Alternate Flow | * + - 1. The type of file uploaded does not match the required file type       2. The size of file is more than the maximum available size |

## Use Case: Apply/Search for Job

|  |  |
| --- | --- |
| Name | Search/Apply Jobs |
| Summary | Job seeker can search for jobs related to his interest using filters, if interested he can apply for the job |
| Rationale | Job Seeker is searching for a job |
| Actor | Job Seeker |
| Pre-Condition | Job Seeker must have an account |
| Basic course of Event | 1. Jobseeker access the required page to search the job. 2. System will show the appropriate GUI for searching and filtering the jobs 3. Job Seeker will fill all the required fields and describe the category and type of a job. 4. If, he/she is interested he will Apply for the job |
| Post- Condition | Apply for the job and wait for Interview notification. |
| Alternate Flow | The search criteria entered has no jobs listed |

## Use Case: Job Application Status

|  |  |
| --- | --- |
| Name | Job application status |
| Summary | Job seeker can check its status either their application is accepted. |
| Rationale | If their application is accepted and rejected, then they must know about the status, so that they can further proceed. |
| Actor | Job seeker |
| Pre-Condition | Job should have been applied for |
| Basic course of Event | 1. The job seeker login to system and check status of his application 2. The system displays GUI for the job application page. 3. Job seeker clicks on the page and check its status. 4. System will show all the results regarding their post on the jobs. |
| Post- Condition | Result successfully shown to the job seeker |

## Use Case: Post jobs

|  |  |
| --- | --- |
| Name | Post Jobs |
| Summary | Recruiter can post the job |
| Rationale | Recruiter wants to find the right employees for specific job. |
| Actor | Recruiter |
| Pre-Condition | Recruiter must have a task and reasonable price for that task |
| Basic course of Event | 1. Recruiter access the required page to post the job. 2. System will show the appropriate GUI for posting the project. 3. Recruiter will fill all the required fields and describe the category and type of a project. 4. System will post the job. |
| Post- Condition | Anyone can see the posted job and apply for it after registration. |
| Alternate Flow | * + - 1. The job details are not sufficient enough to be posted |

## User Case: View Job seeker profile

|  |  |
| --- | --- |
| Name | View Job seeker profile |
| Summary | Recruiter can search the job seeker by name and can view the whole portfolio of job seeker. |
| Rationale | If a Recruiter has a job then he can search the job seeker and see entire profile of all employees or jobseekers. |
| Actor | Recruiter |
| Pre-Condition | Recruiter must be existing in a list and Job Seeker has applied for the Job |
| Basic course of Event | 1. The user indicates that the system is to perform a search function. 2. System responds by the requesting the search term and shows the results. 3. System will show the entire job seeker in a relevant field in which Recruiter wants. |
| Post- Condition | Recruiter can see whole information about the job seeker. |

## Use Case: Hire Jobseeker

|  |  |
| --- | --- |
| Name | Hire Job seeker |
| Summary | Recruiter can hire the employee for a specific job required |
| Rationale | Recruiter has a job and wants to search the employee for the project if found, Recruiter will hire schedule the interview. |
| Actor | Recruiter |
| Pre-Condition | Recruiter should have posted a job. |
| Basic course of Event | 1. The user indicates that the system is to perform a search function. 2. System responds by the requesting the search term and shows the results 3. System will show the entire list of job seeker. 4. System will send the interview notification to the job seeker. |
| Post- Condition | Job seeker will receive the notification that he or she has been asked for interview. |

## User Case: Handle Users

|  |  |
| --- | --- |
| Name | Handle Users |
| Summary | Admin can view all the user’s proceedings. |
| Rationale | If admin wants to job seeker and recruiter to be added to the system or denied. |
| Actor | Admin |
| Pre-Condition | Job Seeker or Recruiter has registered through the website |
| Basic course of Event | 1. Job Seeker or Recruiter has registered through the website 2. Admin verifies the details and credentials of the user 3. If valid, he admits them into the system, else removes them |
| Post- Condition | Job Seeker and Recruiter’s account will be created |

## User Case: Feedback

|  |  |
| --- | --- |
| Name | Feedback |
| Summary | To get a feedback from the user of the system |
| Rationale | To improve the system and understand the bugs |
| Actor | Recruiter and Job Seeker |
| Pre-Condition | Job Seeker or Recruiter has registered through the website |
| Basic course of Event | 1. Job Seeker or Recruiter has registered through the website 2. GO to the feedback tab from dashboard 3. Enter and submit the feedback |
| Post- Condition | The developer receives the feedback |

***Structural View***

***CLASS DIAGRAM***

1. Identify the Classes
2. User
   * 1. jobseeker
     2. Recruiter
     3. Admin
3. Feedback
4. Job
5. Interview
6. Result
7. Resume
8. Identify Attributes
   1. User

+ Full\_Name: String

+ username: String

# password: String

* + 1. Jobseeker

+JobSeekerID:int

-designation: String

-location:String

-experience:String

-email:String

-mobile:int

-keyskills: Array

-projects: Array

-ITSkills: Array

-Education:String

-Certification:String

-DOB: Date

/age:int

-gender: Gender

-pincode: Int

-LanguagesKnown: String

* + 1. Recruiter

+RecruiterID: int

-Company: String

-Designation: String

-Location:String

-email:String

-mobile:Int

-gender: Gender

-DOB : Date

/age:int

* + 1. Admin
  1. Feedback

- message: String

- name: String

- rating: String

* 1. Job

+JobID:int

-Title:String

-Location:String

-Experience:Int

-Salary:Float

-Description: String

-Role: String

-KeySkills: String

-Vacancy: Boolean

-Recruiter: String

-DateofPosting: Date

* 1. Interview

+InterviewID:int

-InterviewDate: Date

-Interviewer: String

-Interviewee: Job

-IntervewLcoation

* 1. Result

+JobID

+interviewID

* 1. Resume

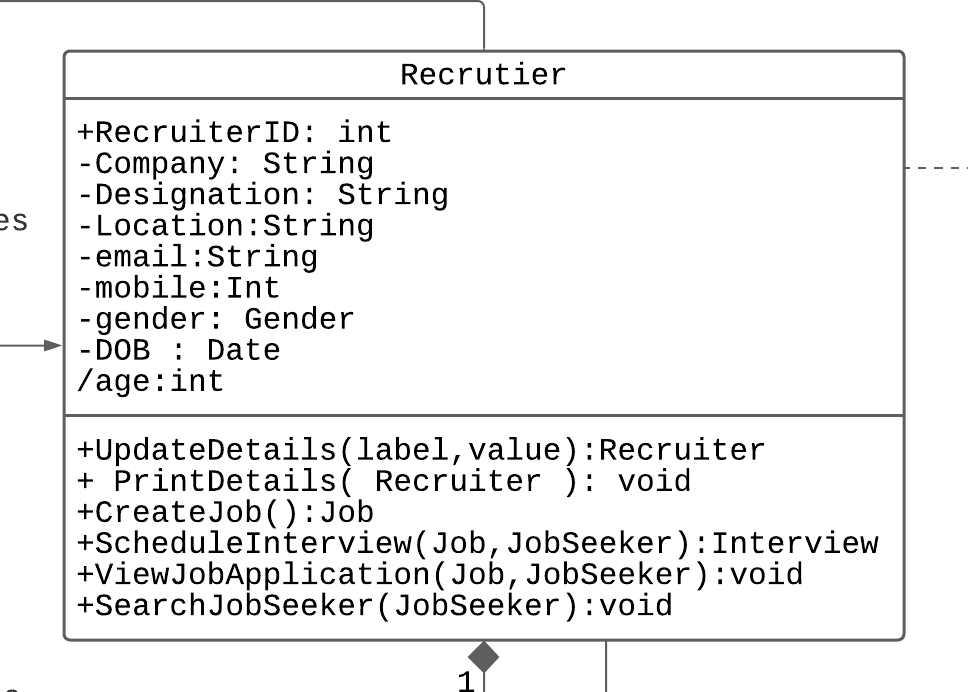
+JobSeekerID: int

-ResumeFIle: file

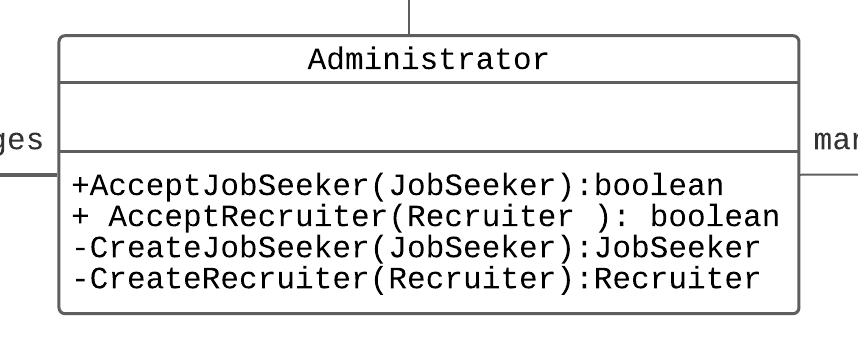
1. Identify functions
2. User



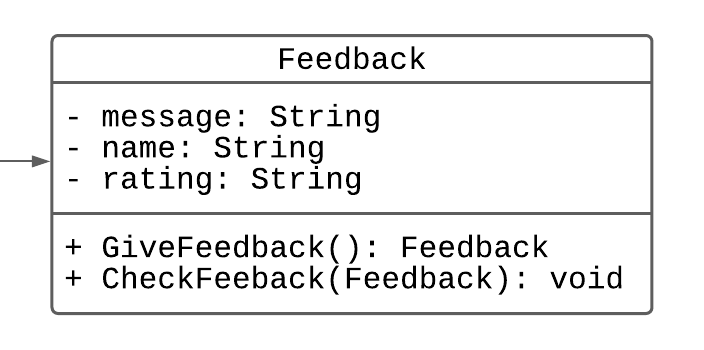
* + 1. jobseeker 
    2. Recruiter



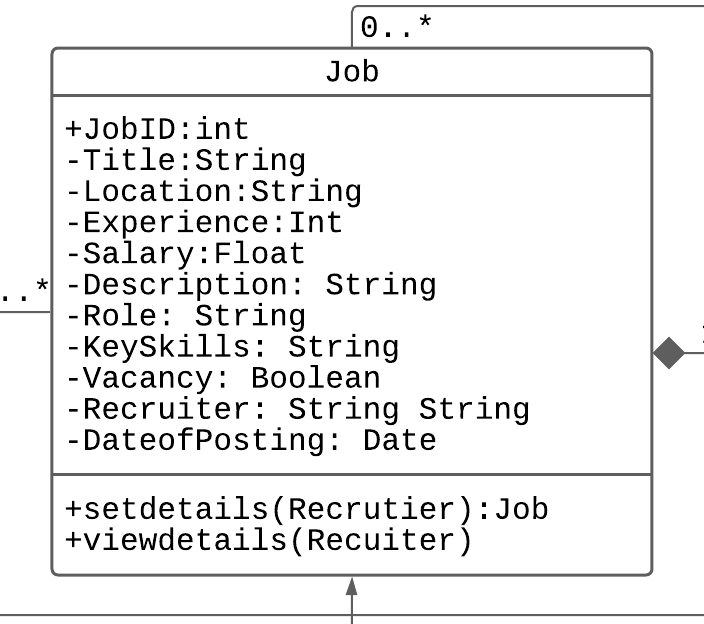
* + 1. Admin



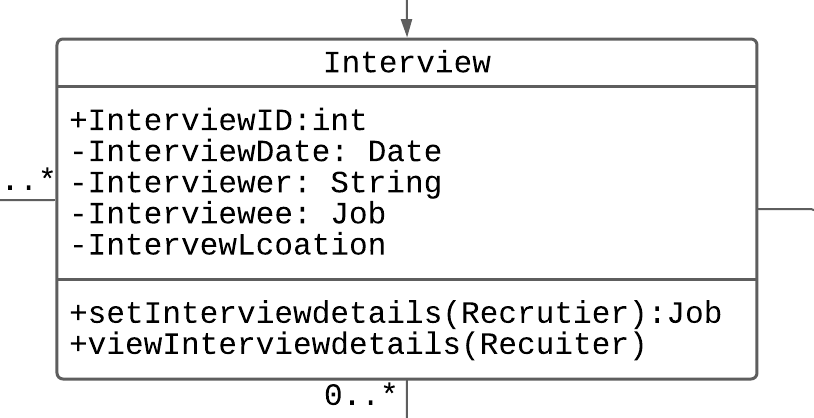
1. Feedback



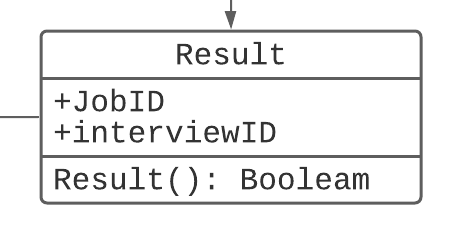
1. Job



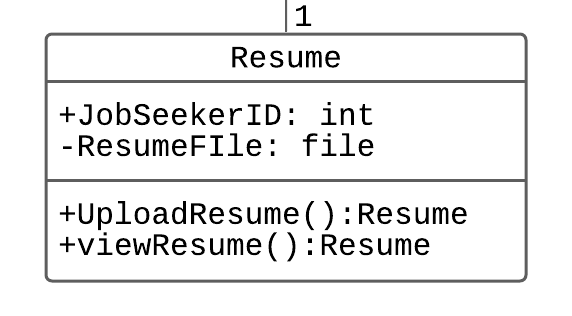
1. Interview



1. Result



1. Resume



1. Identify relationships

1:1

User gives feedback

Resume belongs to a jobseeker

One interview has a result

1:n

A recruiter posts many jobs

A recruiter scehdules many interview

Admin manages many job seekers

Admin manages many recruiters

A job has many interviews

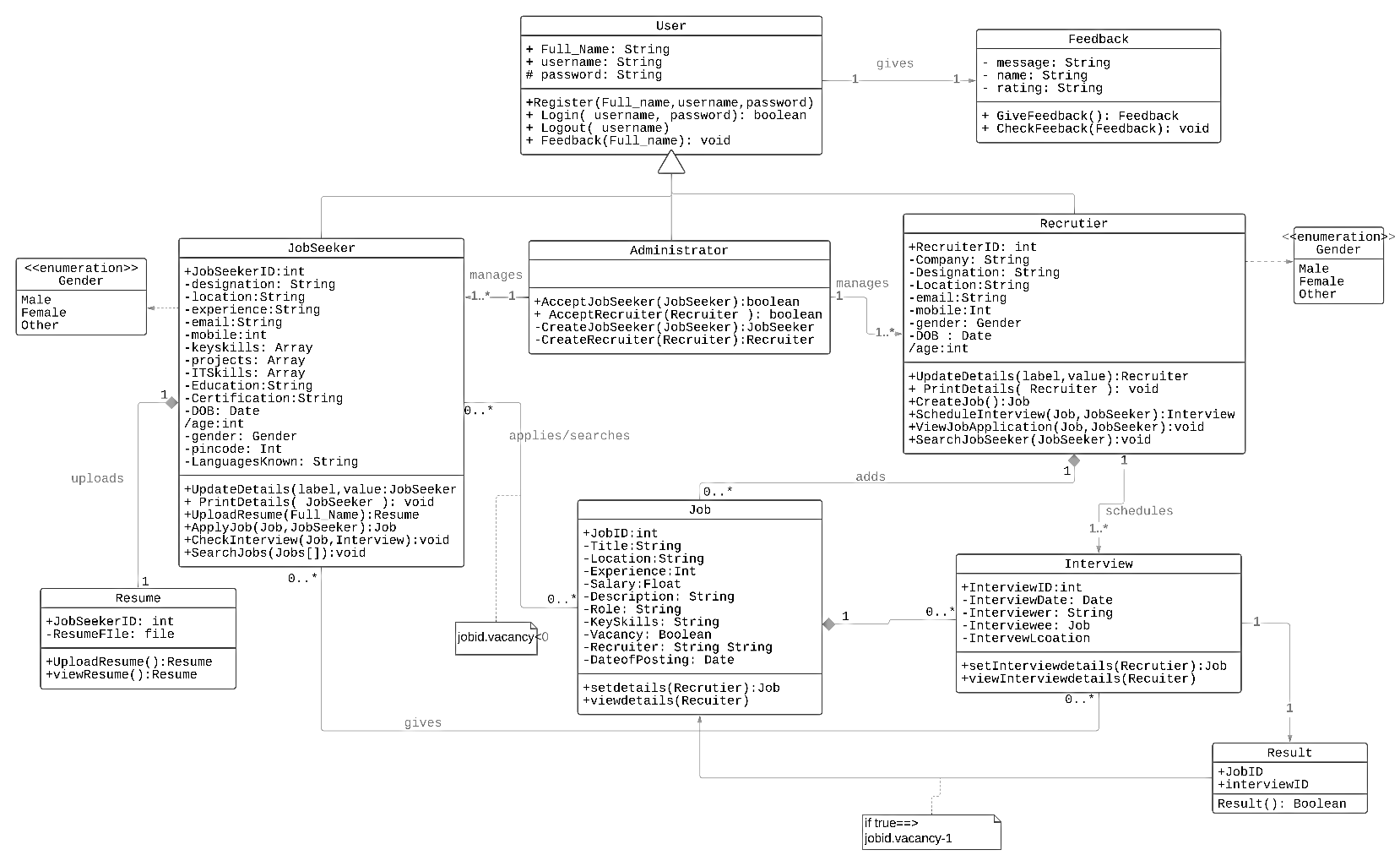
N:n

Jobseeker applies/searches for jobs

1. Constrain

Apply Job: precondition: vacancy>0

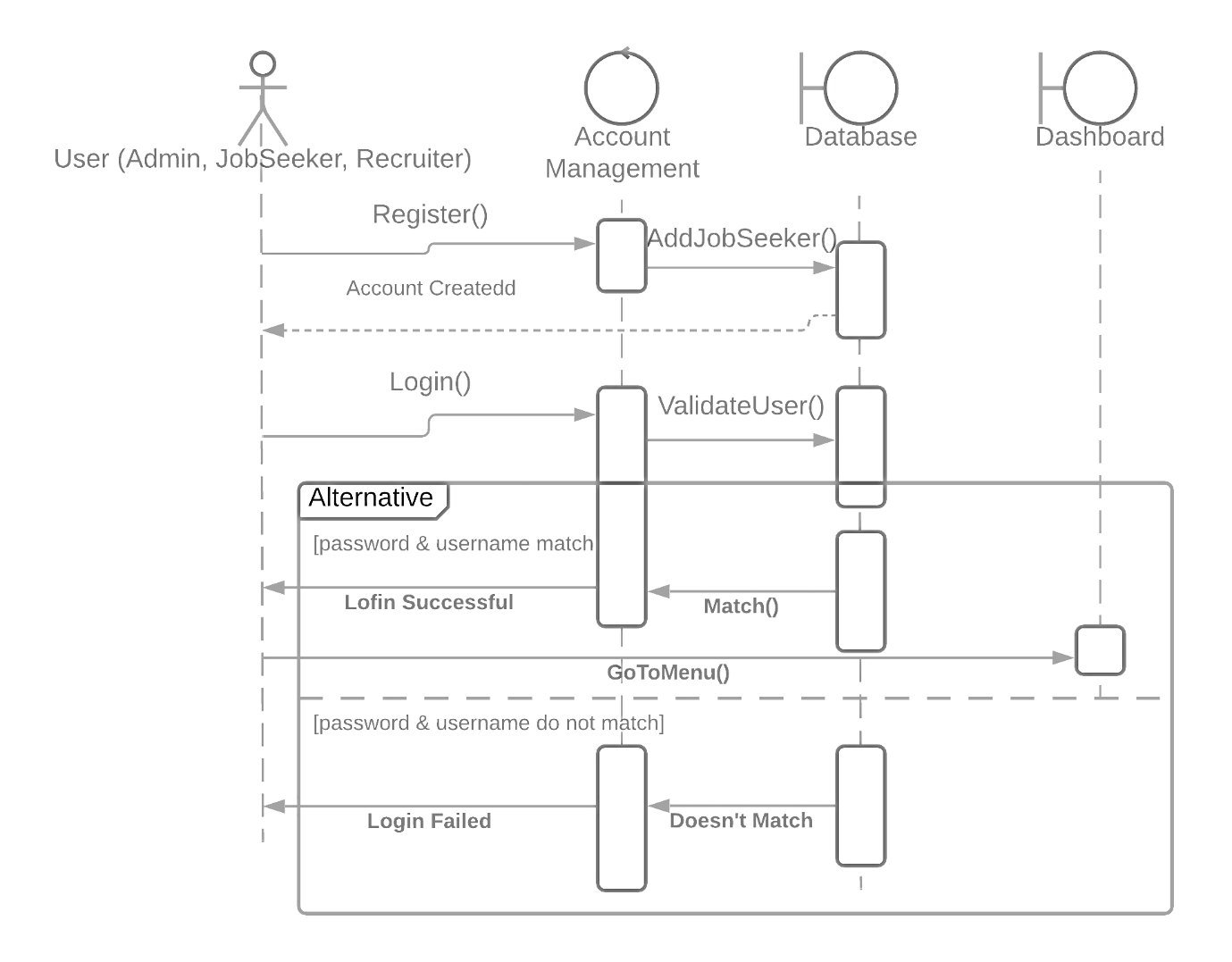
After interview if result==true, then c vacancy--;

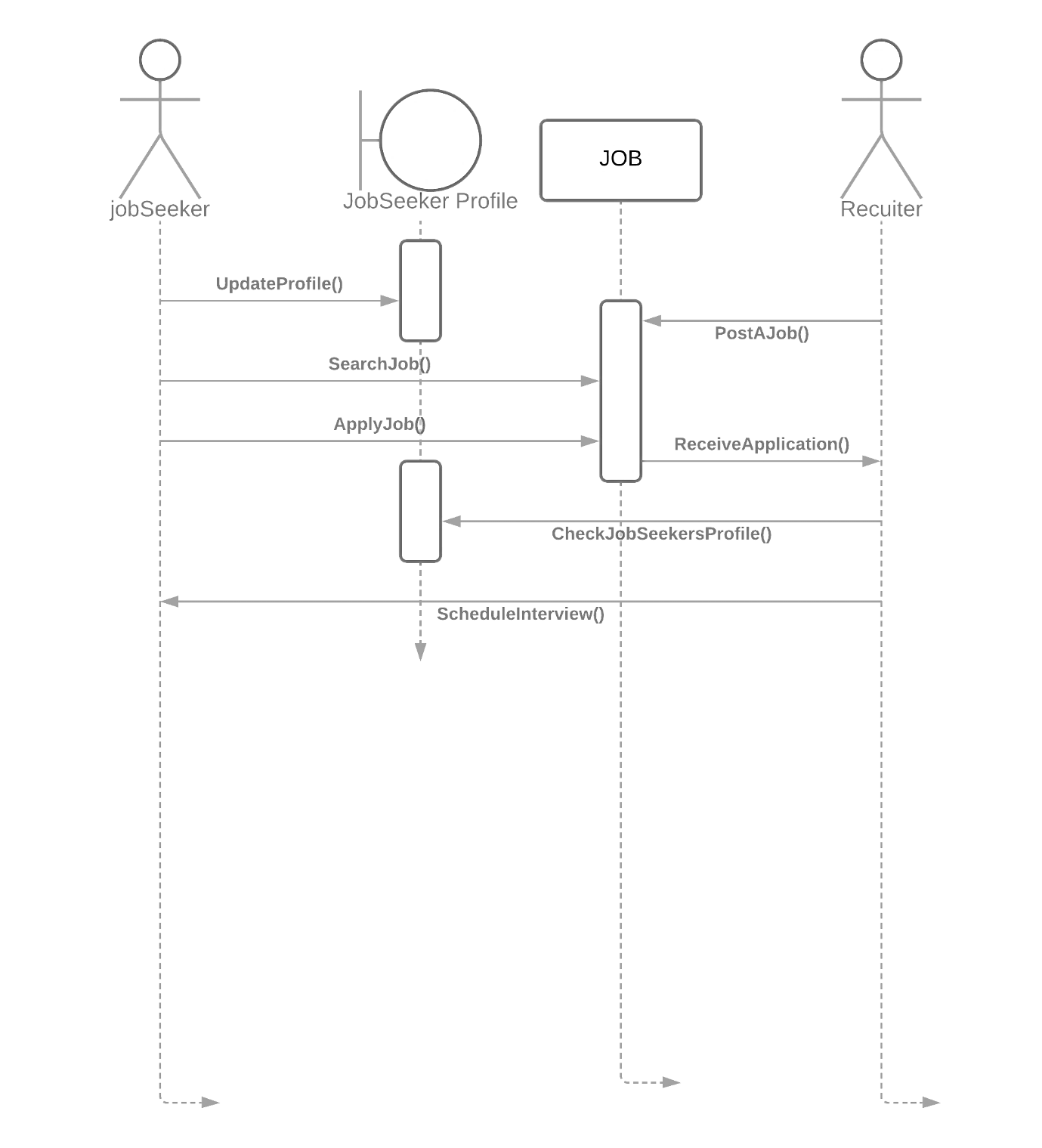
******

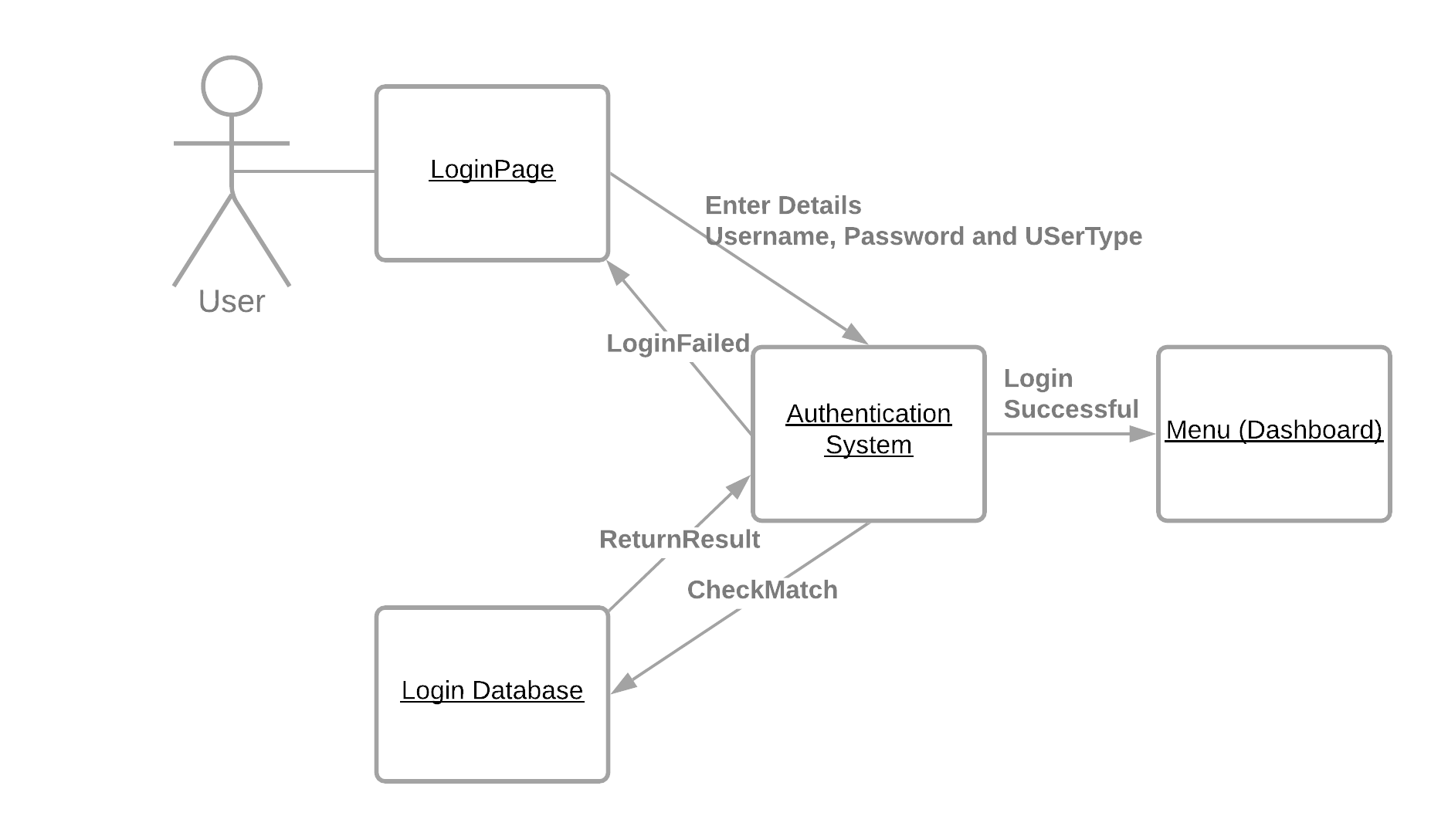
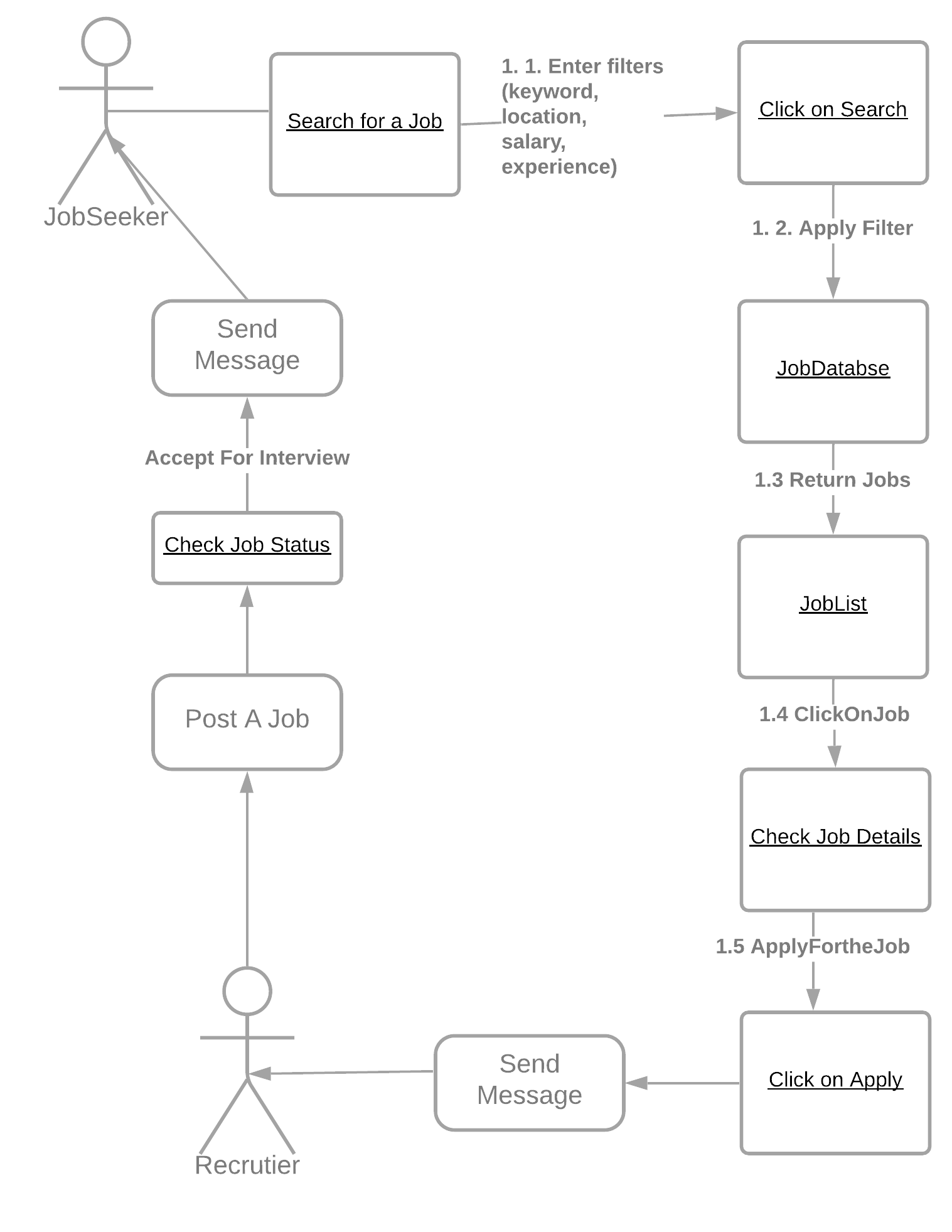
***Behavioural View***

***Sequence Diagram***

***Register and Login***

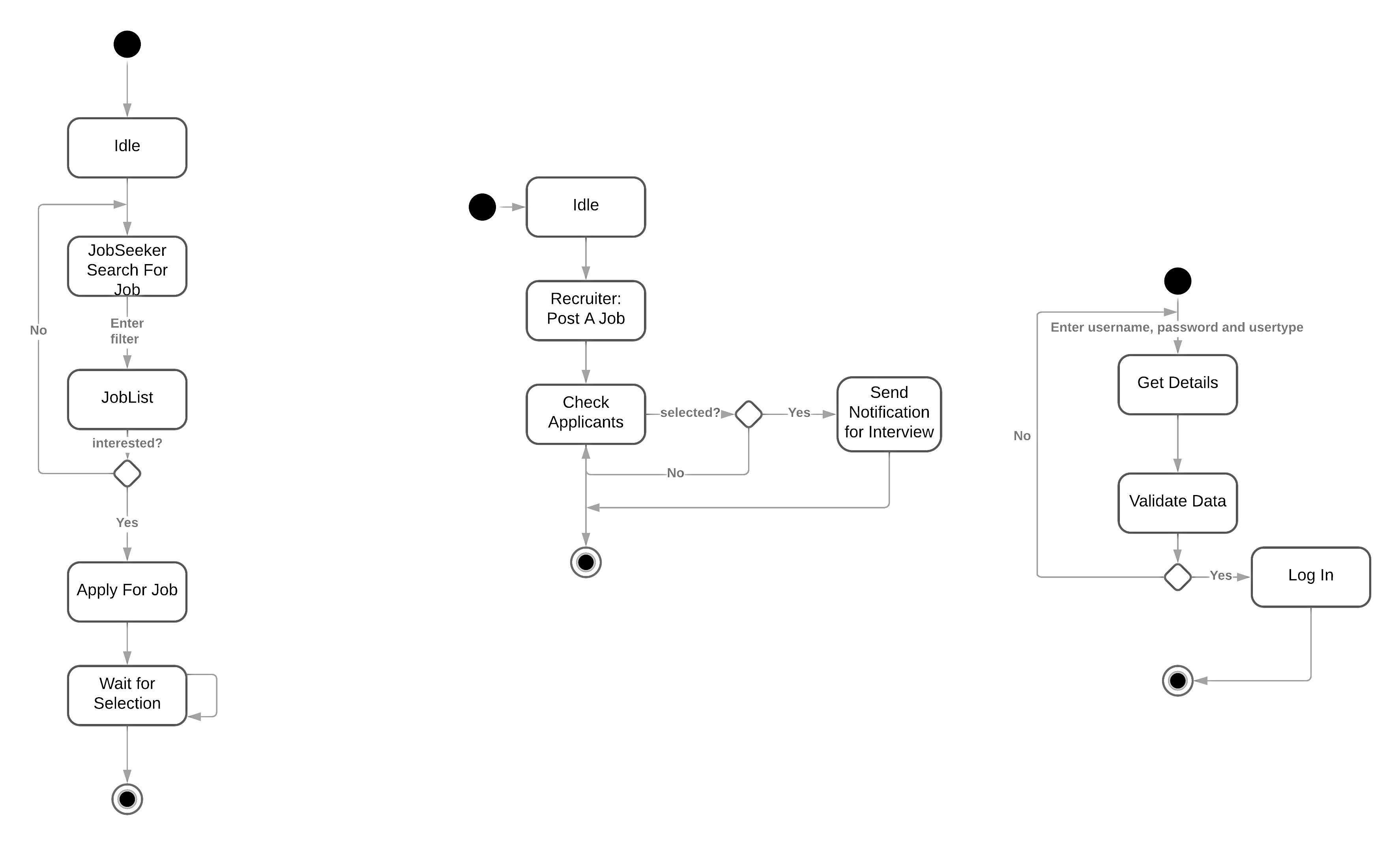
******

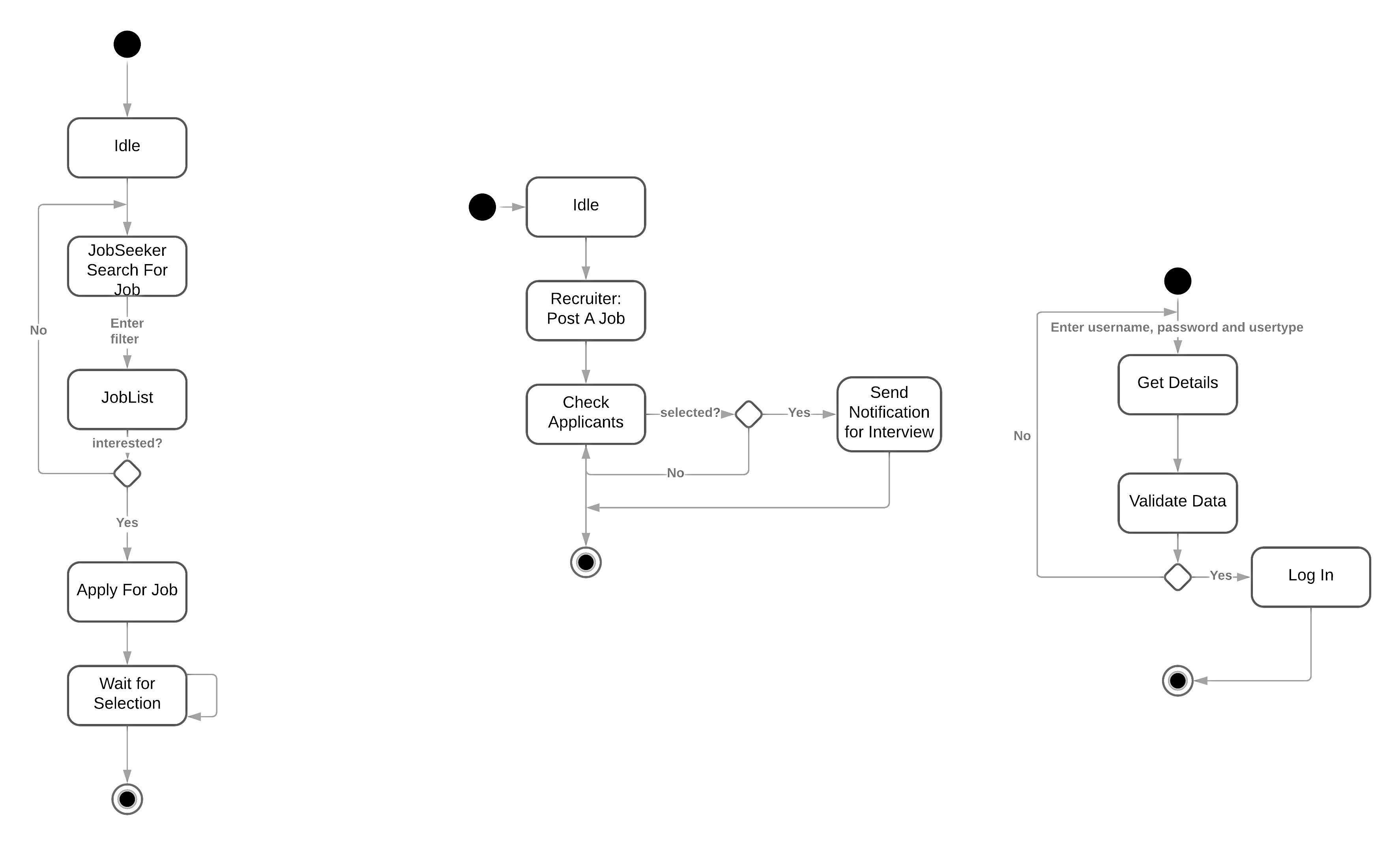
***Jobseeker and Recruiter ***

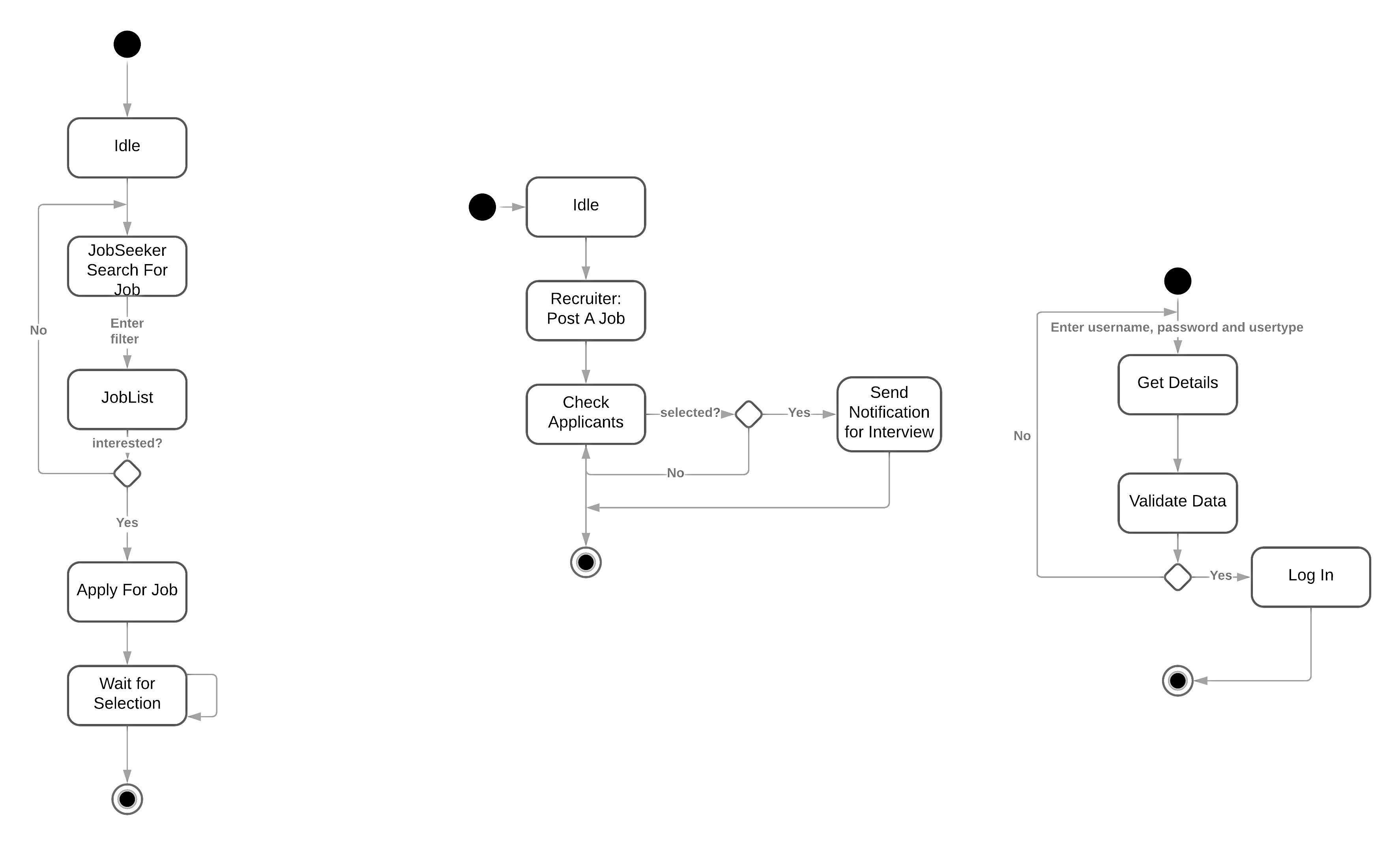
***Collaboration DiagramJobSeeker and Recruiter Activity***

***State Chart Diagram***

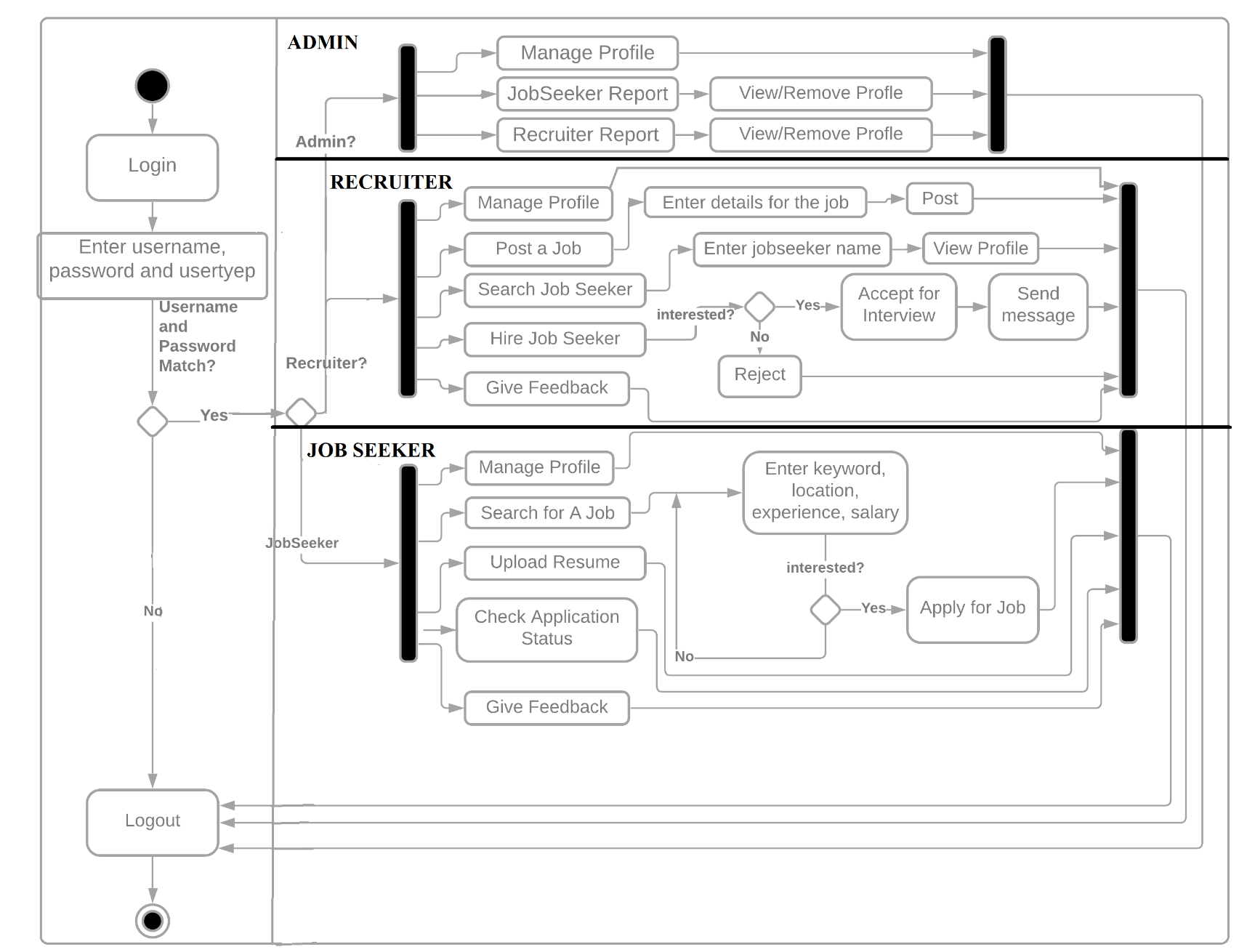
* + - 1. Login



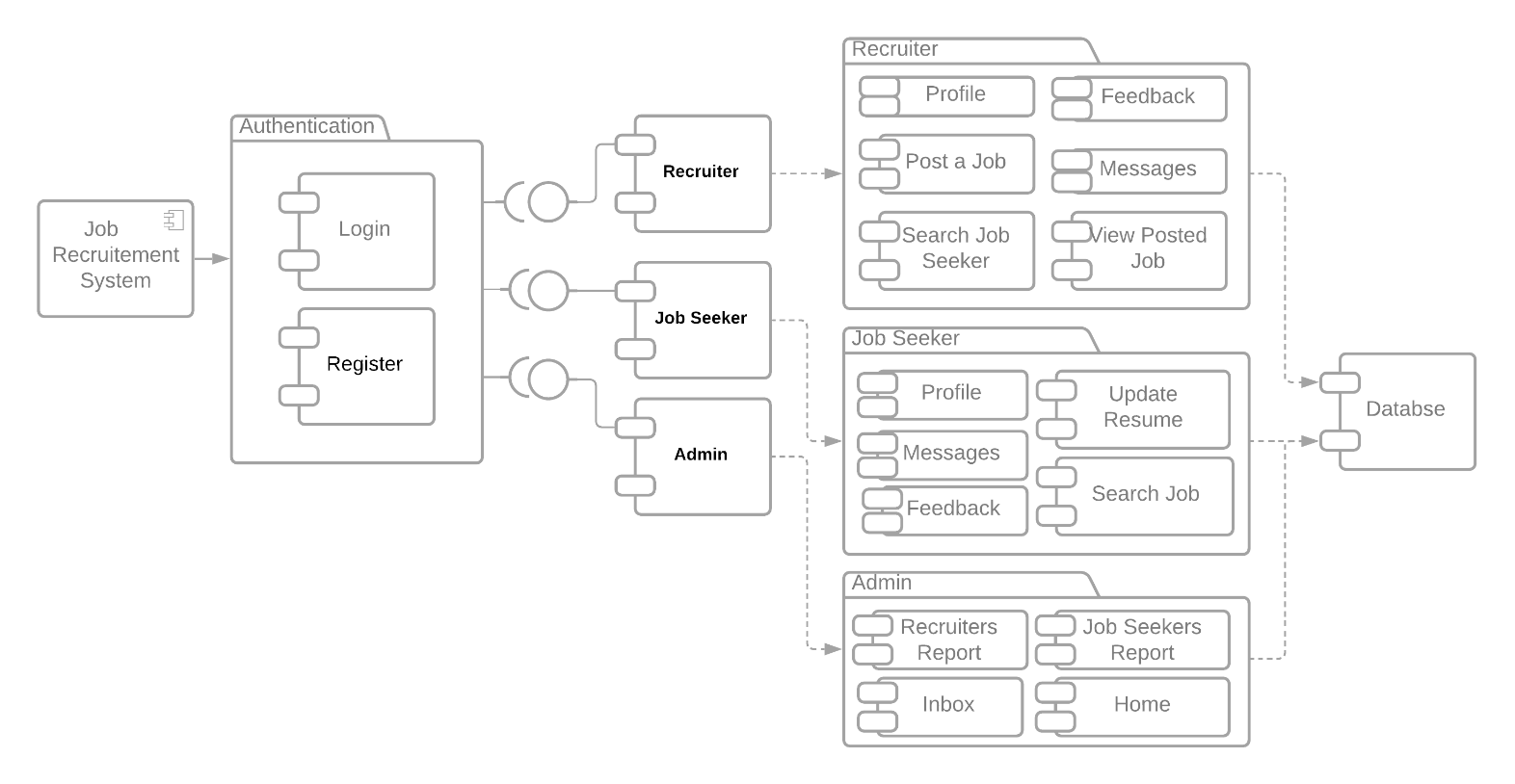
* + - 1. Recruiter Activity
      2. JobSeeker Activity



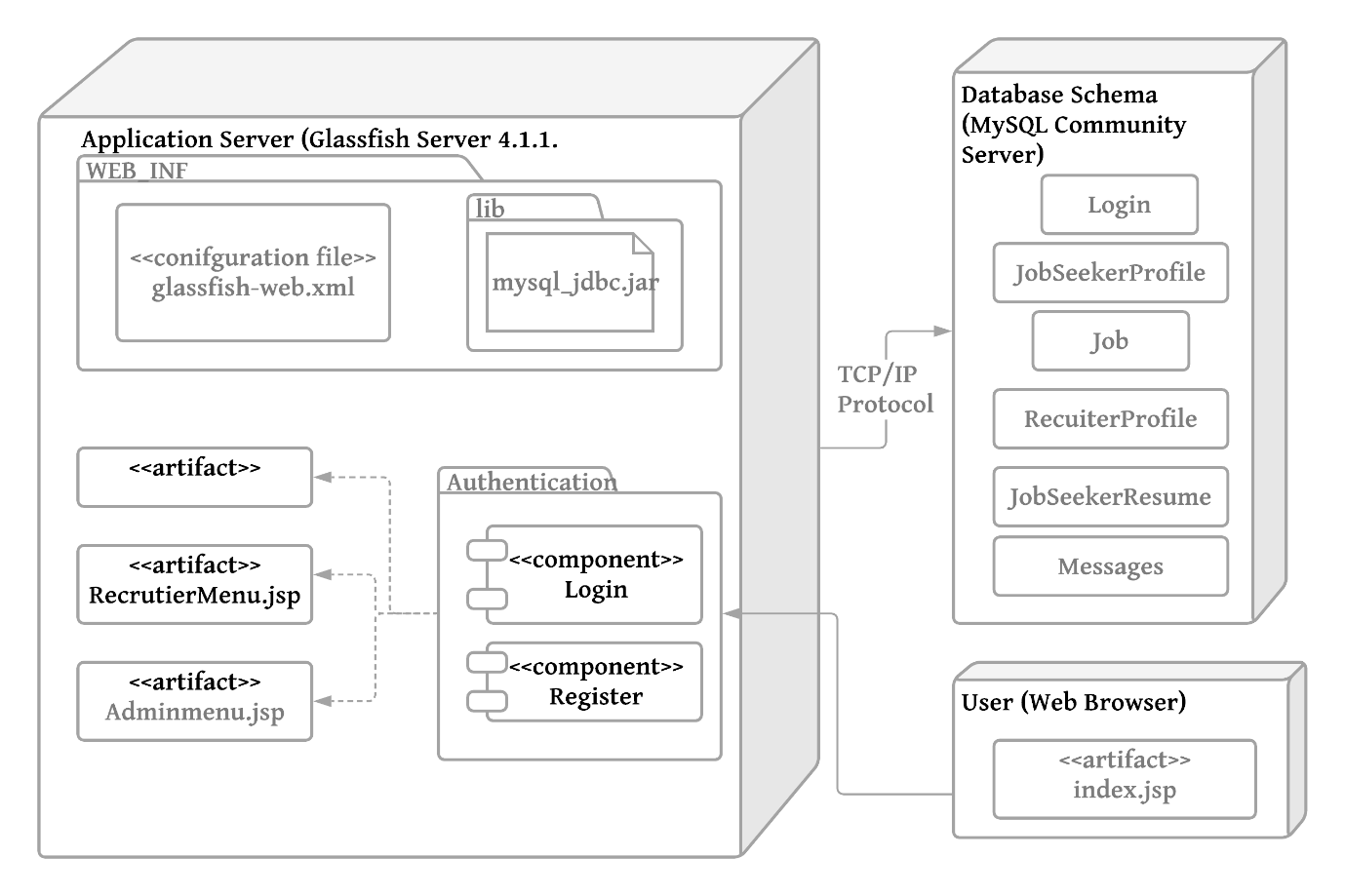
***Activity Diagrams***

****

***Component Diagram***

******

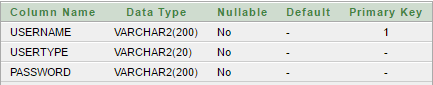
***Deployment Diagram***

******

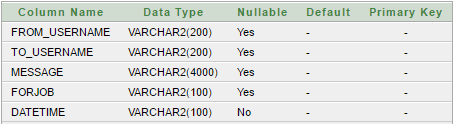
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Machine | **Software Configuration** | **Hardware Configuration** | **Operating System** | **Compiler** | **Any other software** | **Software Modules and path** |
| Client | Google Chrome  /Internet Explorer/  any browser with JSP Support |  | Windows 7 and after | Java | NetBeans, | GUI |
| Application Server | Sun GlassFish 4.1.1. Server | * Sun GlassFish Enterprise Server: 35 MB minimum * SDK: 250 MB minimum * 1GB Memory | Windows 7 and after, Red Hat, MacOS | HTML, Java, JSP,  SQL | Internet Explorer, Chrome, Mozilla, Safari |  |
| Database Server | MySQL Connector/J Driver 5.1 |  | Windwos 7 & after, Unbuntu |  |  | tables |

Data Structure

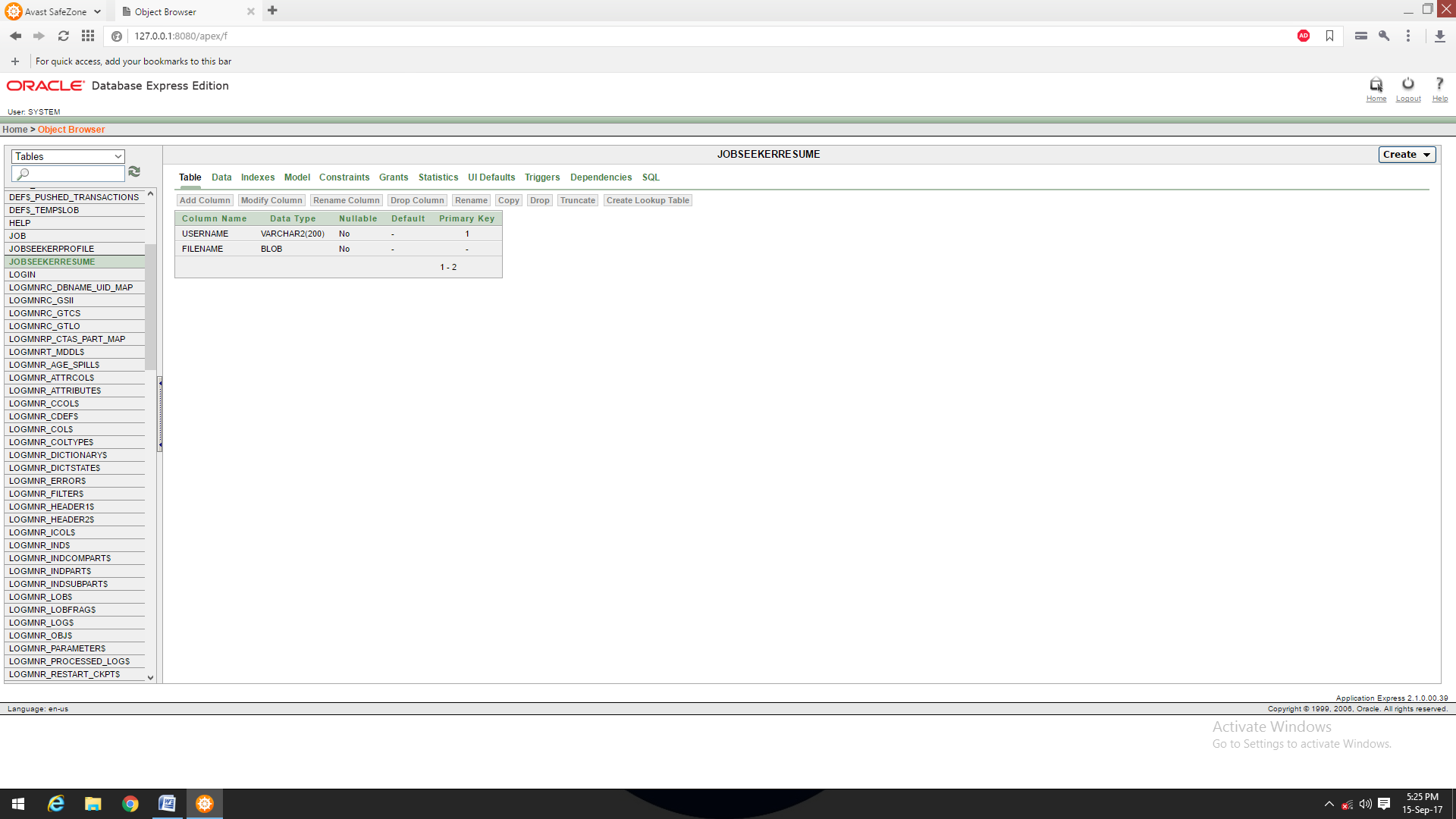
LOGIN



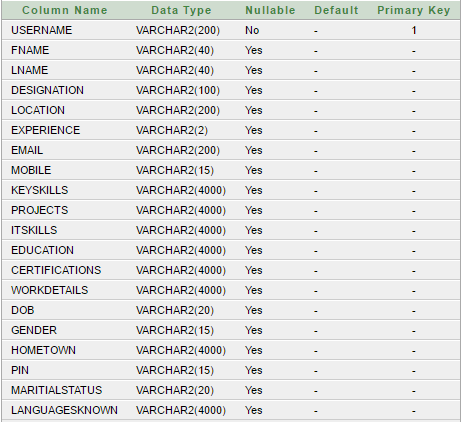
MESSAGE



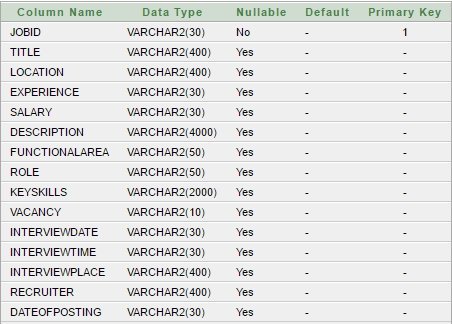
JOBSEEKERRESUME



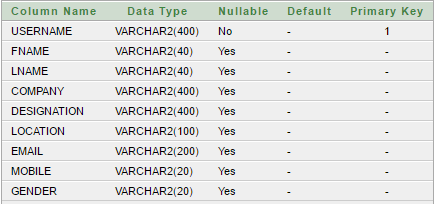
JOBSEEKERPROFILE



JOB



RECRUITERPROFILE



ALGORITHM DESIGN

## Use Case: Registration

|  |  |
| --- | --- |
| Name | Register |
| Summary | A new user (a user which doesn’t have an account) should first register into the system to use it. This feature will allow the user to enroll into the system if the user is a new user and doesn’t already have an existing account. |
| Rationale | The user can access the system |
| Actor | Job Seeker and Recruiter |
| Pre-Condition | None |
| Basic course of Event | 1. User opens the desired module of the system. 2. The system displays GUI the registration. 3. User enters the valid details required to join the system 4. Submits the form |
| Post- Condition | User is registered |
| Alternate Flow | User goes back to the Login Page |
| Exception | Username already taken |

**Algorithm:**

**Input:** username, password, usertype;

**Output:**  user added to the system

**Pseudo Code:**

* + 1. Start
    2. Go to Registration Page
    3. Enter the username and usertype
    4. Enter password and confirm password
    5. If username!=NULL && usertype!=NULL &&password!=NULL && confirm\_password!=NULL then

If password==confirm\_password

If username already exists in table then enter another username

1. Else: Enter the entry into login table of database

Else

Alert: Password and confirm password do not match

Try again

1. Else:
   * + - 1. The field is required, do not leave it empty

## Use Case: Login

|  |  |
| --- | --- |
| Name | Login |
| Summary | Job seeker can change information into their profiles which include profile edit, delete, and update. |
| Rationale | After registration of the user, the valid user can login into the system |
| Actor | All Users |
| Pre-Condition | The login name and password should match with the login name and password provided while registering.  If the username and or password do not match, the user cannot login successfully into the system. |
| Basic course of Event | 1. User opens the desired module of the system. 2. The system displays GUI the login form 3. The user writes its username, password and type 4. If valid, user can login into the system. If the username and or password do not match, the user cannot login successfully into the system. |
| Post- Condition | User goes to his/her dashboard |
| Alternate Flow | 1. The username and password is blank 2. The type of user is not specified 3. The username and password do not match |

**Algorithm:**

**Input:** username, password, usertype;

**Output:**  allow/denied access to home page

**Pseudo Code:**

1. Start
2. Login Page open
3. Enter the username, password, type input
4. If password==NULL | username==NULL|| type==NULL
   1. The field is required
5. Retrieve the entry from database for the given username
   1. If username=username(database) && password=password(database)&& type=usertype
   2. Go to Dashboard
6. Else
   1. Alert: Invalid entry, please try again

End

## Use Case: Manage Profile

|  |  |
| --- | --- |
| Name | Manage Profile |
| Summary | User can change information into their profiles which include profile edit, delete, and update. |
| Rationale | If there is unnecessary date and fake profiles that are running on the site then Admin will have rights to remove or delete the profile. |
| Actor | All Users |
| Pre-Condition | Profile and record must exist |
| Basic course of Event | 1. User opens the desired module of the system. 2. The system displays GUI of different pages. 3. User will input the necessary data into the fields. 4. System will update the record as per requirements. |
| Post- Condition | User receives the notification that profile has been approved or profile has been updated. |
| Alternate Condition | The details enter to not validate, e.g. age is <18, date of bitrth is out of range. |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Go to Profile section
3. Click on edit
4. Update the sections you want
5. Click Update
6. End

## Use Case: Upload CV

|  |  |
| --- | --- |
| Name | Upload CV |
| Summary | Job seeker can post his/her CV on job portal |
| Actor | Job seeker |
| Pre-Condition | New Jobs and other features should be displayed. |
| Basic course of Event | 1. Job seeker search for a specific job category. 2. The system displays GUI for the new jobs and features. 3. Job seeker clicks on the new jobs option and upload his/her CV there. 4. System will show all the CV on the admin and job seeker end. |
| Post- Condition | CV successfully uploaded. |
| Alternate Flow | * + - 1. The type of file uploaded does not match the required file type       2. The size of file is more than the maximum available size |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Go to Upload Resume section
3. Click on choose file
4. Choose the file from your computer
5. Click on Upload
6. End

## Use Case: Apply/Search for Job

|  |  |
| --- | --- |
| Name | Search/Apply Jobs |
| Summary | Job seeker can search for jobs related to his interest using filters, if interested he can apply for the job |
| Rationale | Job Seeker is searching for a job |
| Actor | Job Seeker |
| Pre-Condition | Job Seeker must have an account |
| Basic course of Event | 1. Jobseeker access the required page to search the job. 2. System will show the appropriate GUI for searching and filtering the jobs 3. Job Seeker will fill all the required fields and describe the category and type of a job. 4. If, he/she is interested he will Apply for the job |
| Post- Condition | Apply for the job and wait for Interview notification. |
| Alternate Flow | * + - 1. The search criteria entered has no jobs listed |

**Algorithm:**

**Input:** keyword, location, experience, expected salary

**Output:** List of Mathing Jobs

**Pseudo Code:**

* + - 1. Prompt the user for search query
      2. Check whether entry is valid or not
      3. If entry is invalid
  1. Then Information to the user & prompt again for entry
     1. Else
  2. check with database
  3. Show list of jobs
  4. Select the job of your choice
  5. The details of the same will be displayed
  6. Apply for the job
  7. If vacancy>0
     1. Successfully Applied
  8. Else
     1. Alert: “No vacancy, search other jobs”

5. End

## Use Case: Job Application Status

|  |  |
| --- | --- |
| Name | Job application status |
| Summary | Job seeker can check its status either their application is accepted. |
| Rationale | If their application is accepted and rejected, then they must know about the status, so that they can further proceed. |
| Actor | Job seeker |
| Pre-Condition | Job should have been applied for |
| Basic course of Event | * + - 1. The job seeker login to system and check status of his application       2. The system displays GUI for the job application page.       3. Job seeker clicks on the page and check its status.       4. System will show all the results regarding their post on the jobs. |
| Post- Condition | Result successfully shown to the job seeker |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Go to messages tab
3. Check for the application status
4. End

## Use Case: Post jobs

|  |  |
| --- | --- |
| Name | Post Jobs |
| Summary | Recruiter can post the job |
| Rationale | Recruiter wants to find the right employees for specific job. |
| Actor | Recruiter |
| Pre-Condition | Recruiter must have a task and reasonable price for that task |
| Basic course of Event | 1. Recruiter access the required page to post the job. 2. System will show the appropriate GUI for posting the project. 3. Recruiter will fill all the required fields and describe the category and type of a project. 4. System will post the job. |
| Post- Condition | Anyone can see the posted job and apply for it after registration. |
| Alternate Flow | * + - 1. The job details are not sufficient enough to be posted |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Go to Post a Job Tab
3. Initialize: JobID
4. Initialize: Title
5. Initialize: Location
6. Initialize: Experience
7. Initialize: Salary
8. Initialize: Description
9. Initialize: Role
10. Initialize: KeySkills
11. Initialize: Vacancy
12. Initialize: Recruiter
13. Initialize: DateofPosting
14. Click on Post Job
15. End

## Use Case: View Job seeker profile

|  |  |
| --- | --- |
| Name | View Job seeker profile |
| Summary | Recruiter can search the job seeker by name and can view the whole portfolio of job seeker. |
| Rationale | If a Recruiter has a job then he can search the job seeker and see entire profile of all employees or jobseekers. |
| Actor | Recruiter |
| Pre-Condition | Recruiter must be existing in a list and Job Seeker has applied for the Job |
| Basic course of Event | 1. The user indicates that the system is to perform a search function. 2. System responds by the requesting the search term and shows the results. 3. System will show the entire job seeker in a relevant field in which Recruiter wants. |
| Post- Condition | Recruiter can see whole information about the job seeker. |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Go to Search Jobseeker tab
3. Enter the name of jobseeker
4. If no entry
   1. Display: No such job seeker
5. Else  
   b. From the list of searches, chose the one you want to view profile of
6. End

## Use Case: Hire Jobseeker

|  |  |
| --- | --- |
| Name | Hire Job seeker |
| Summary | Recruiter can hire the employee for a specific job required |
| Rationale | Recruiter has a job and wants to search the employee for the project if found, Recruiter will hire schedule the interview. |
| Actor | Recruiter |
| Pre-Condition | Recruiter should have posted a job. |
| Basic course of Event | 1. The user indicates that the system is to perform a search function. 2. System responds by the requesting the search term and shows the results 3. System will show the entire list of job seeker. 4. System will send the interview notification to the job seeker. |
| Post- Condition | Job seeker will receive the notification that he or she has been asked for interview. |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Go to Messages tab
3. Check the list of applications
4. Accept the one, you want
5. End

## Use Case: Handle Users

|  |  |
| --- | --- |
| Name | Handle Users |
| Summary | Admin can view all the user’s proceedings. |
| Rationale | If admin wants to job seeker and recruiter to be added to the system or denied. |
| Actor | Admin |
| Pre-Condition | Job Seeker or Recruiter has registered through the website |
| Basic course of Event | 1. Job Seeker or Recruiter has registered through the website 2. Admin verifies the details and credentials of the user 3. If valid, he admits them into the system, else removes them |
| Post- Condition | Job Seeker and Recruiter’s account will be created |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Login as Admin
3. Go to Jobseeker Report
   1. View or Remove Job Seeker
4. Go to Recruiter Report
   1. View or Remove Job Seeker
5. End

## Use Case: Feedback

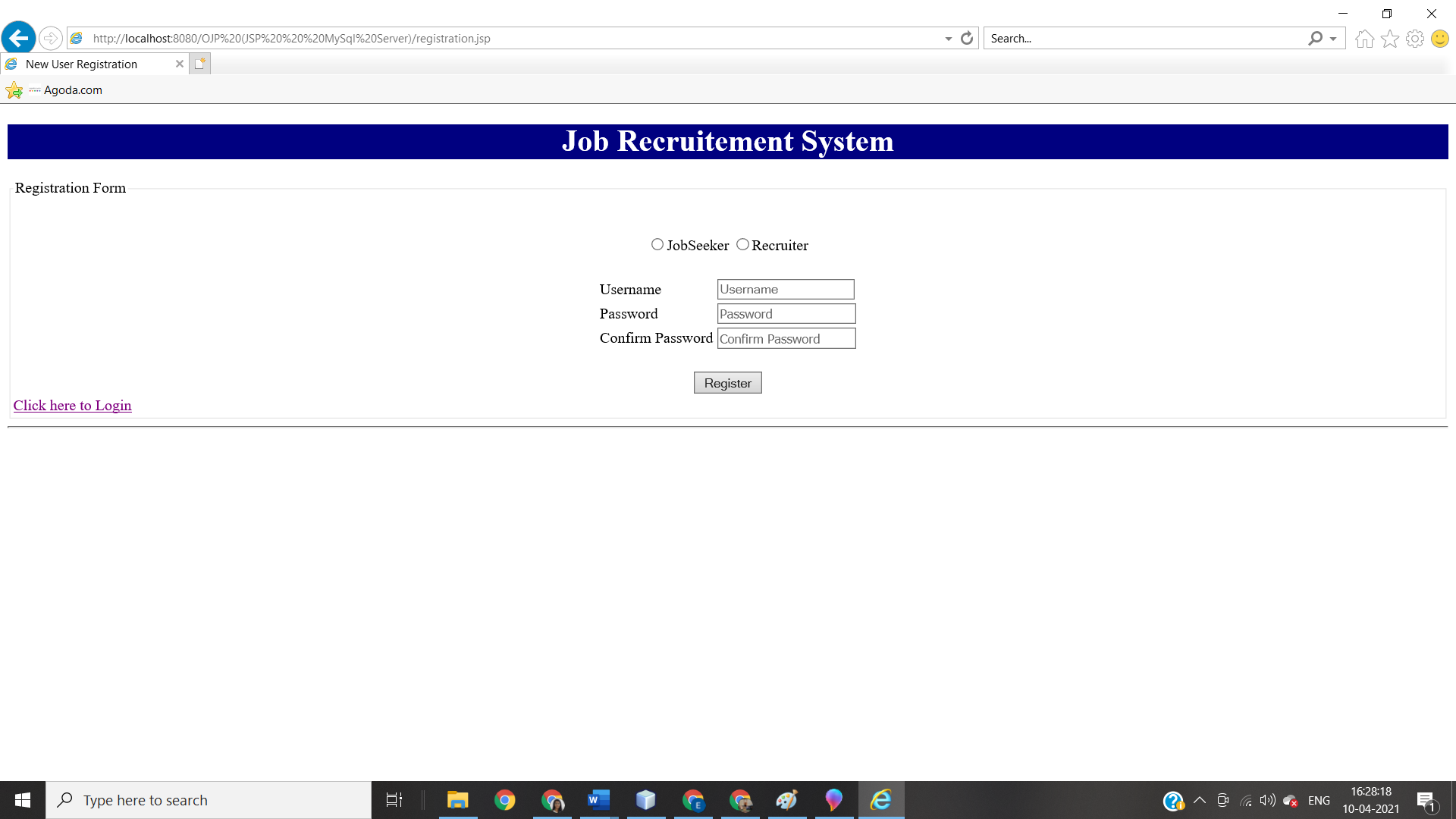
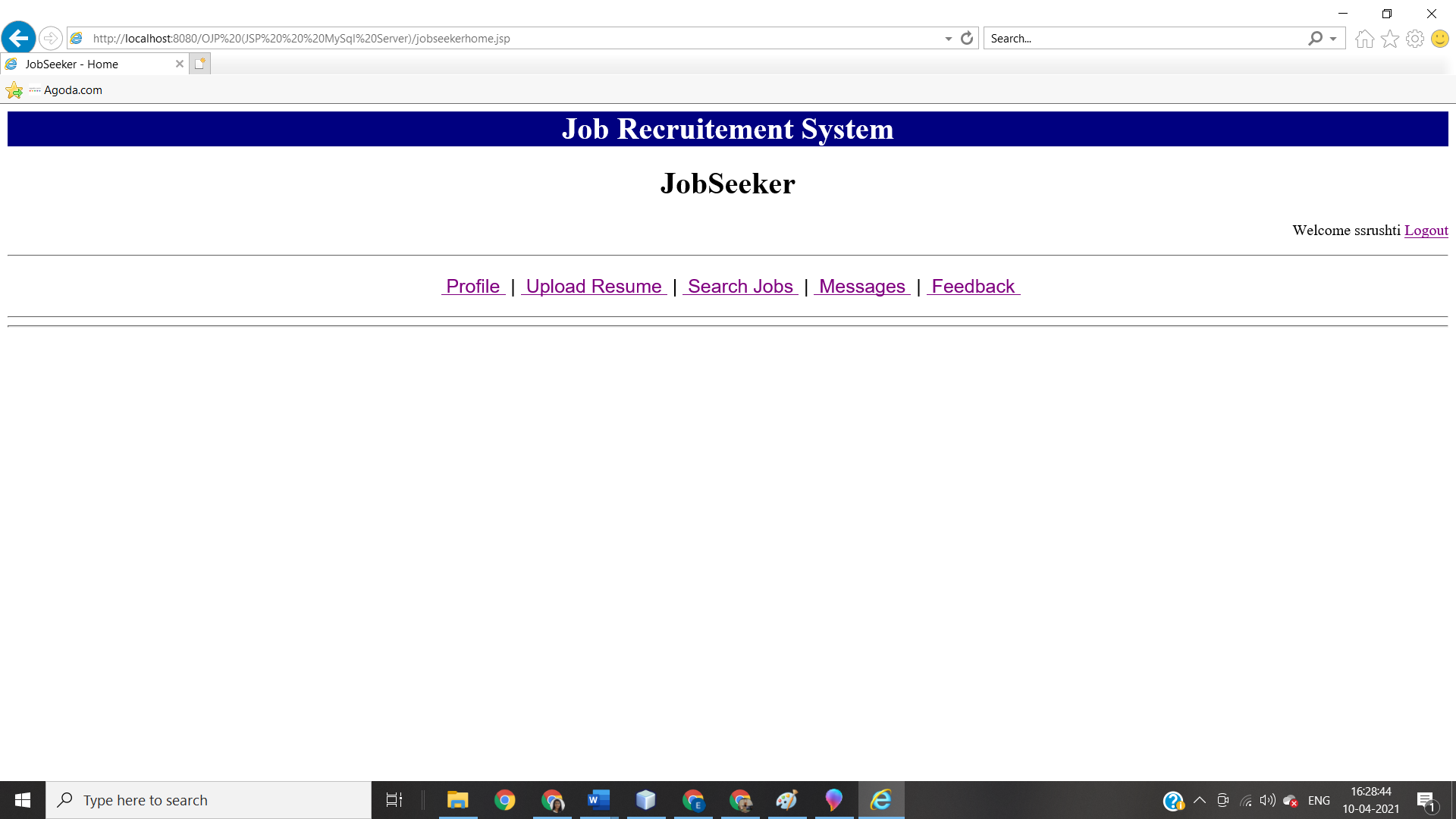
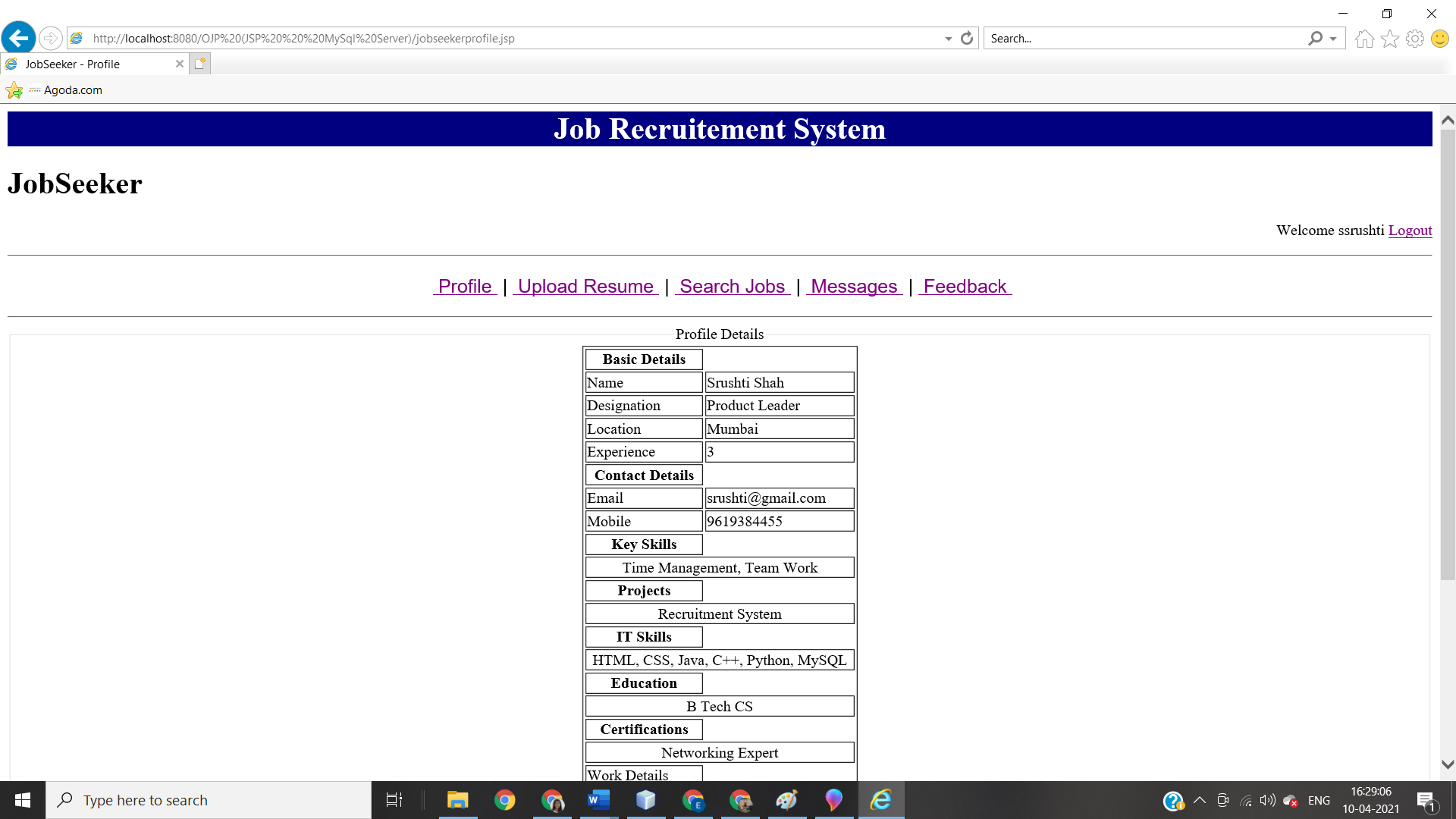
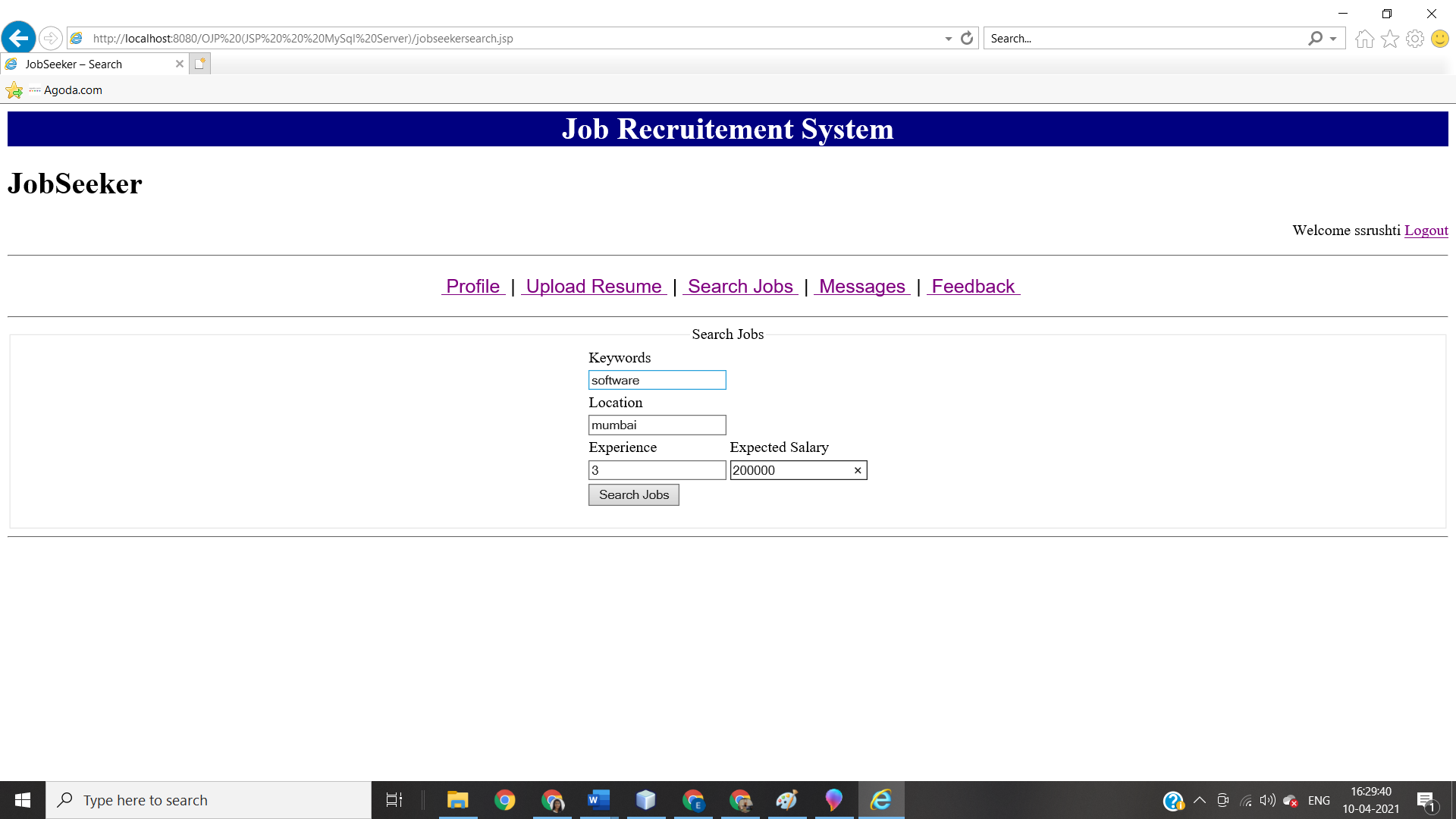
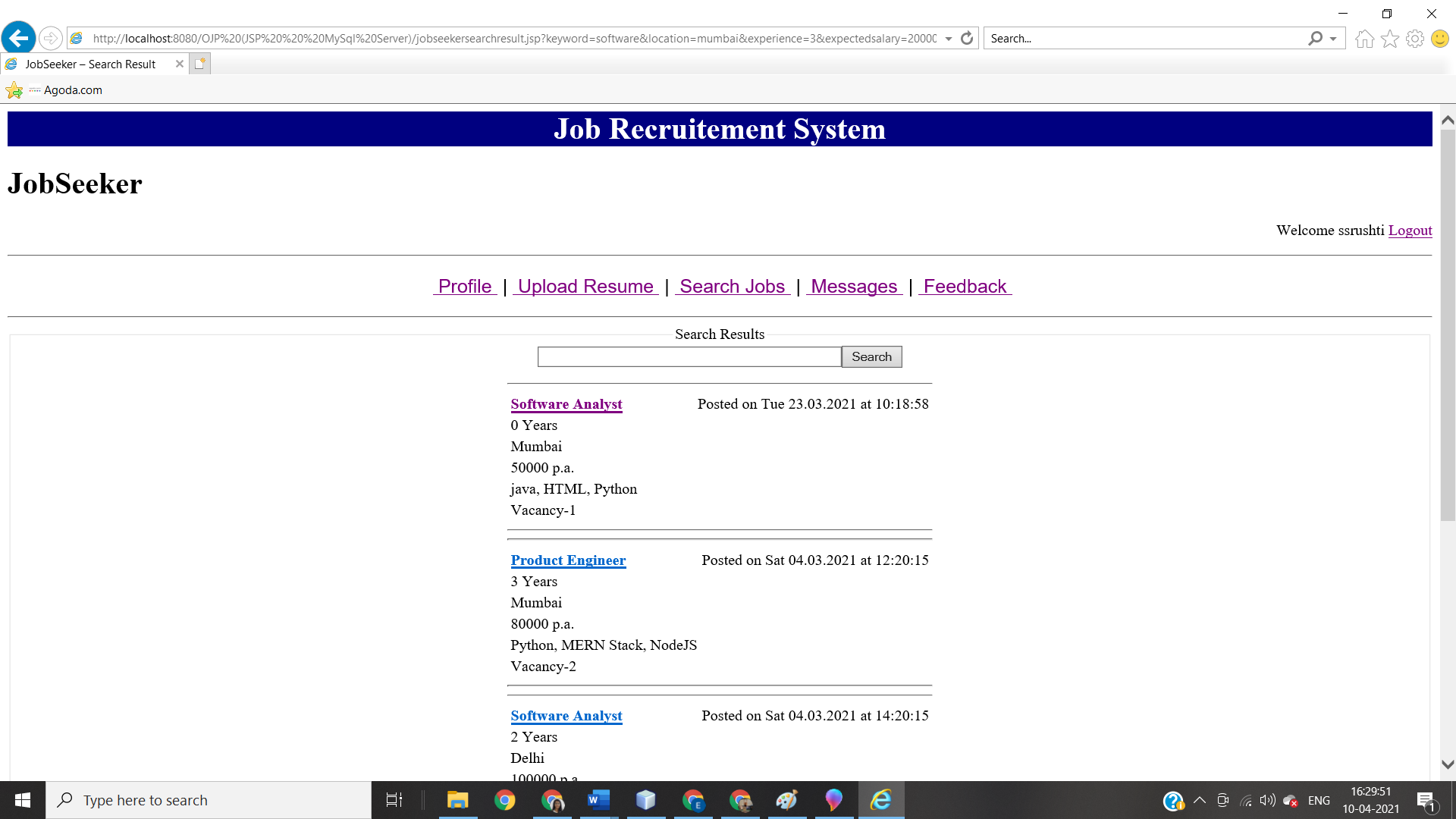
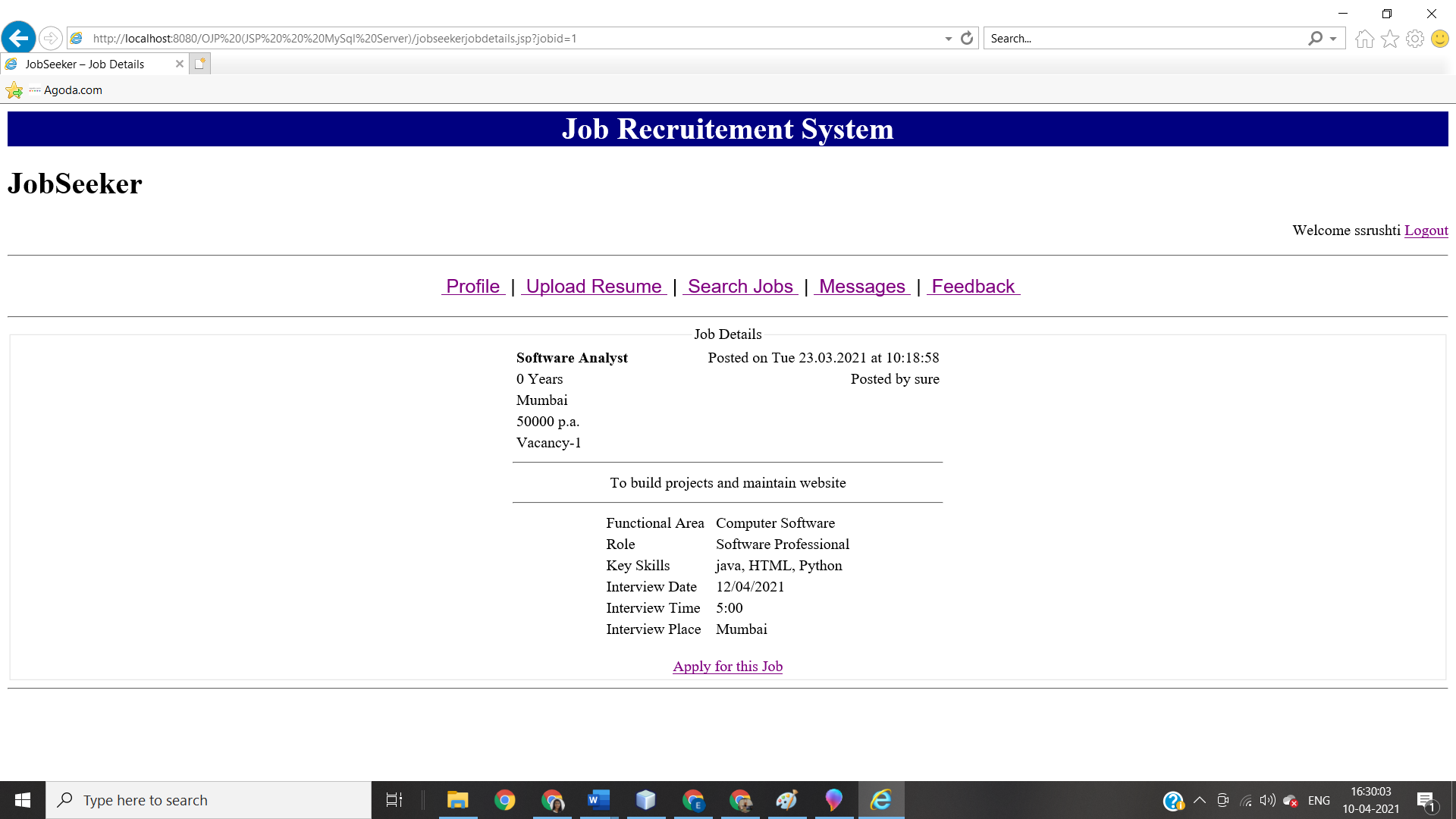
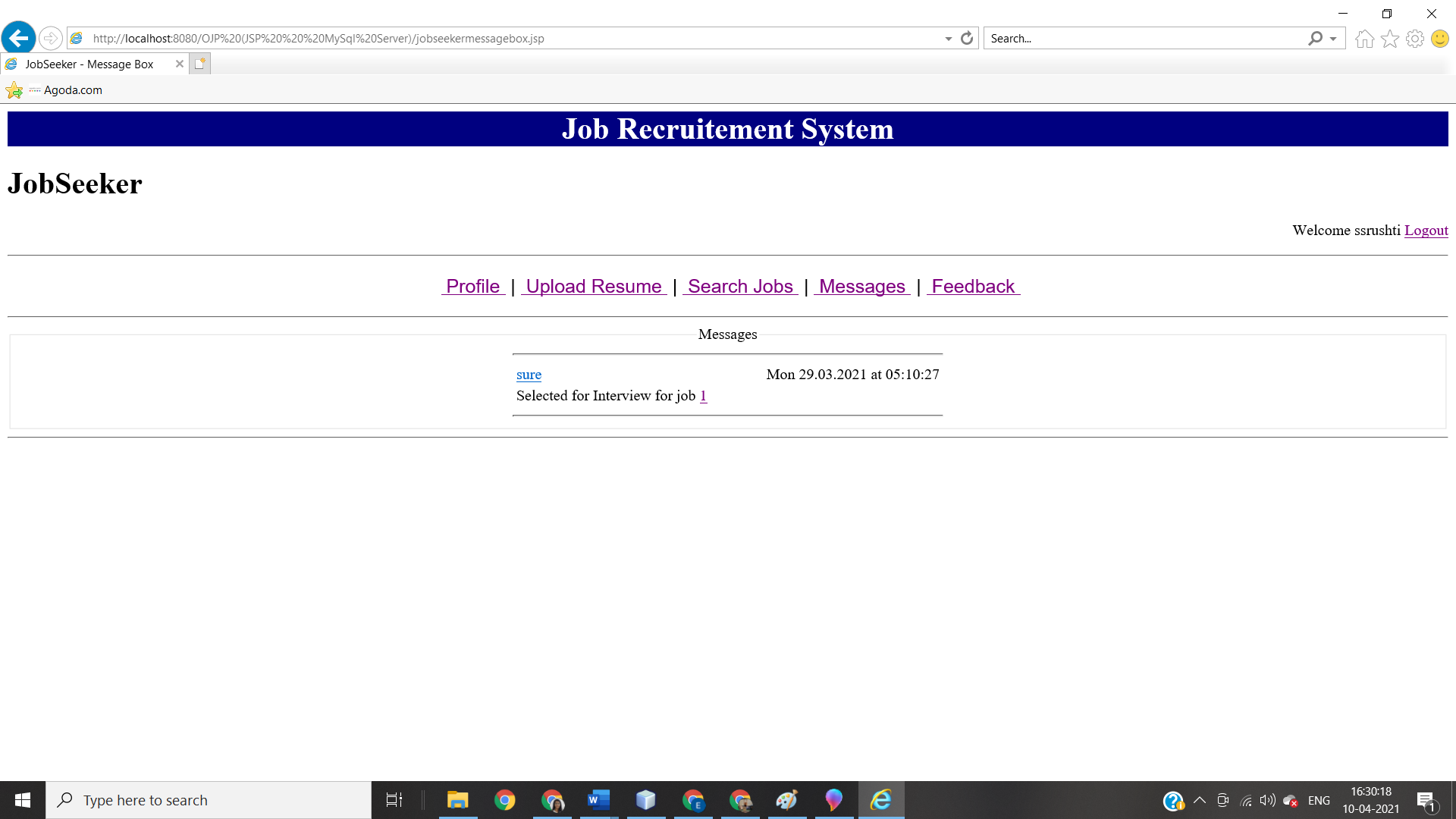
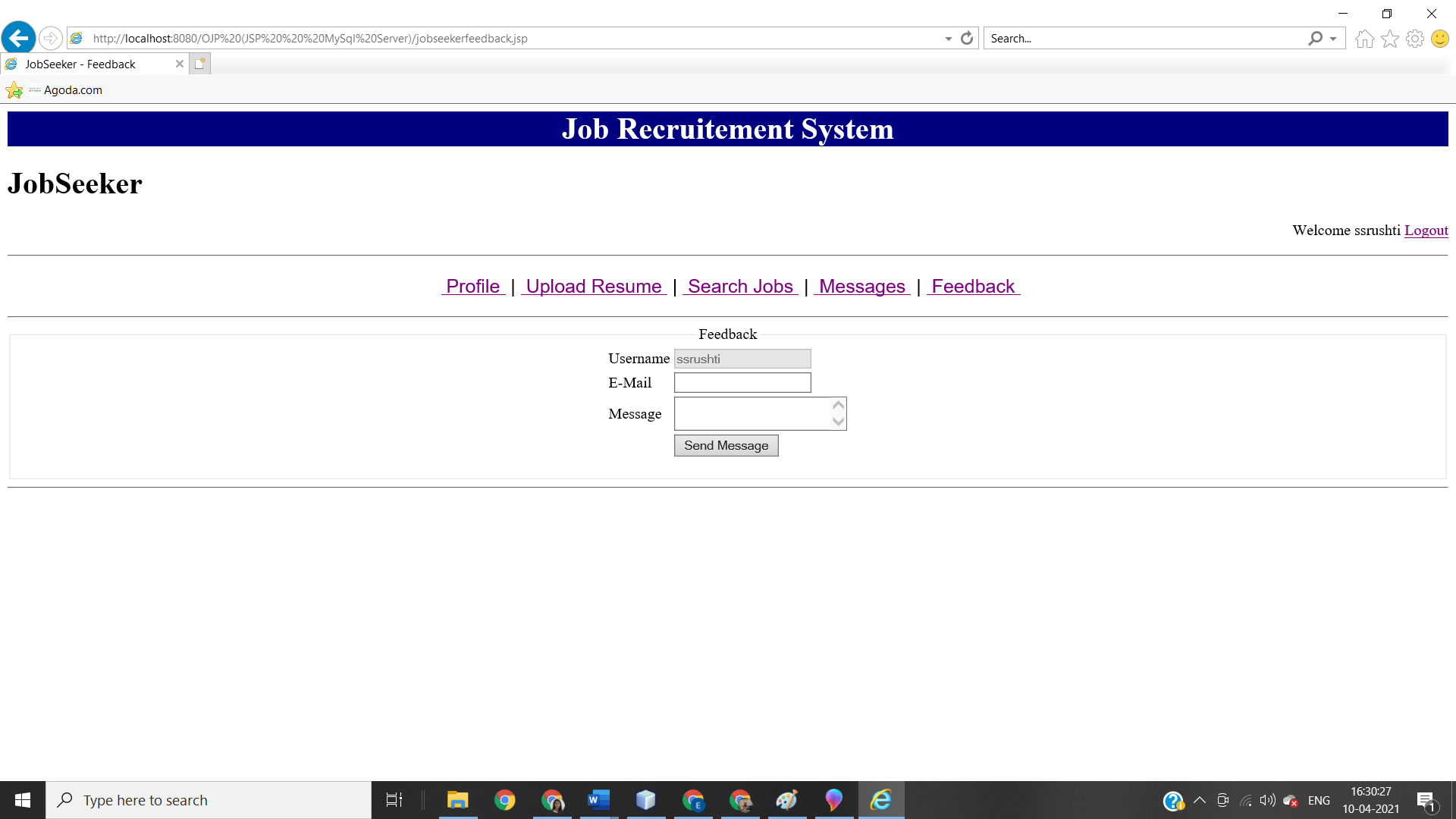
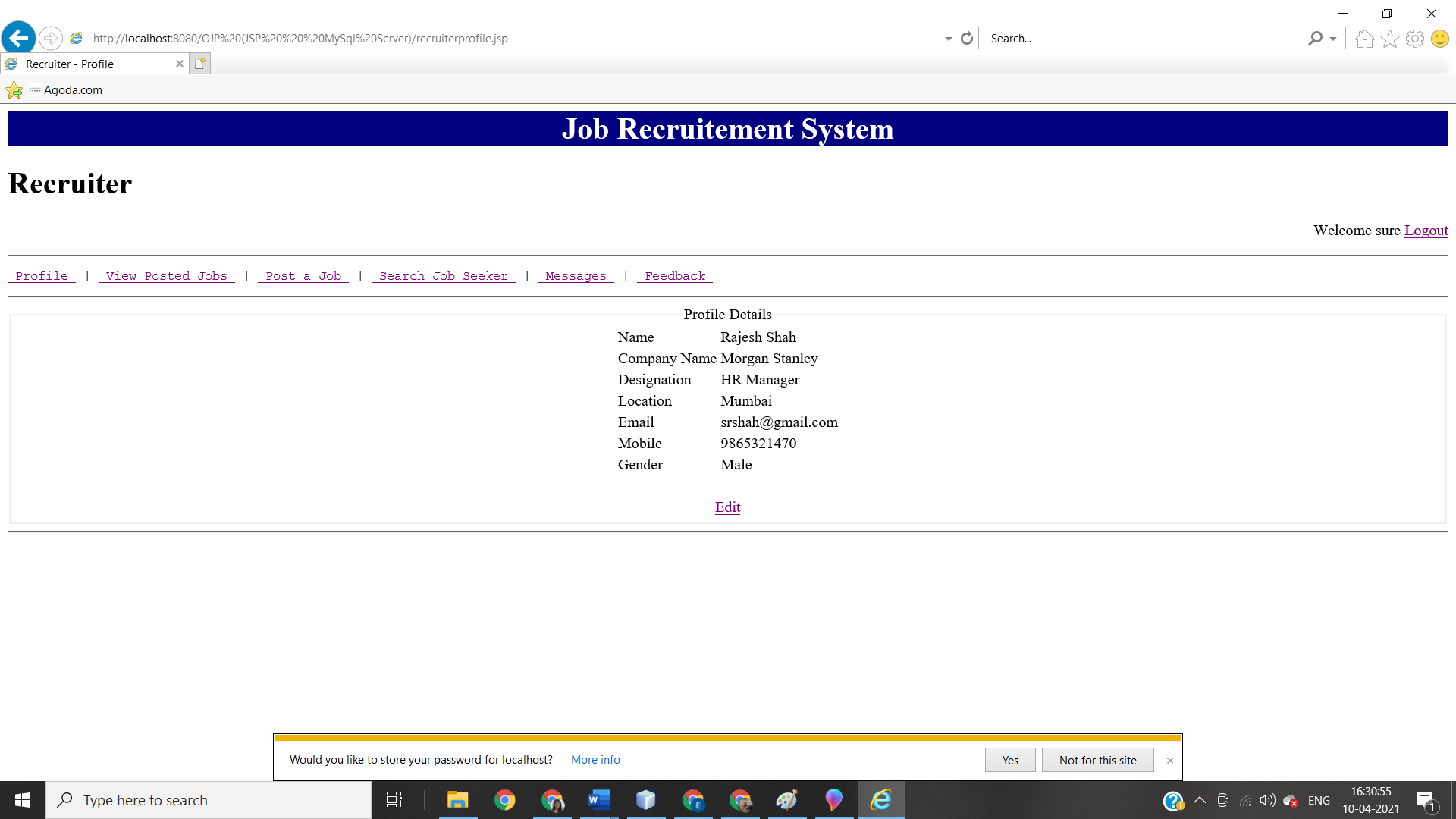
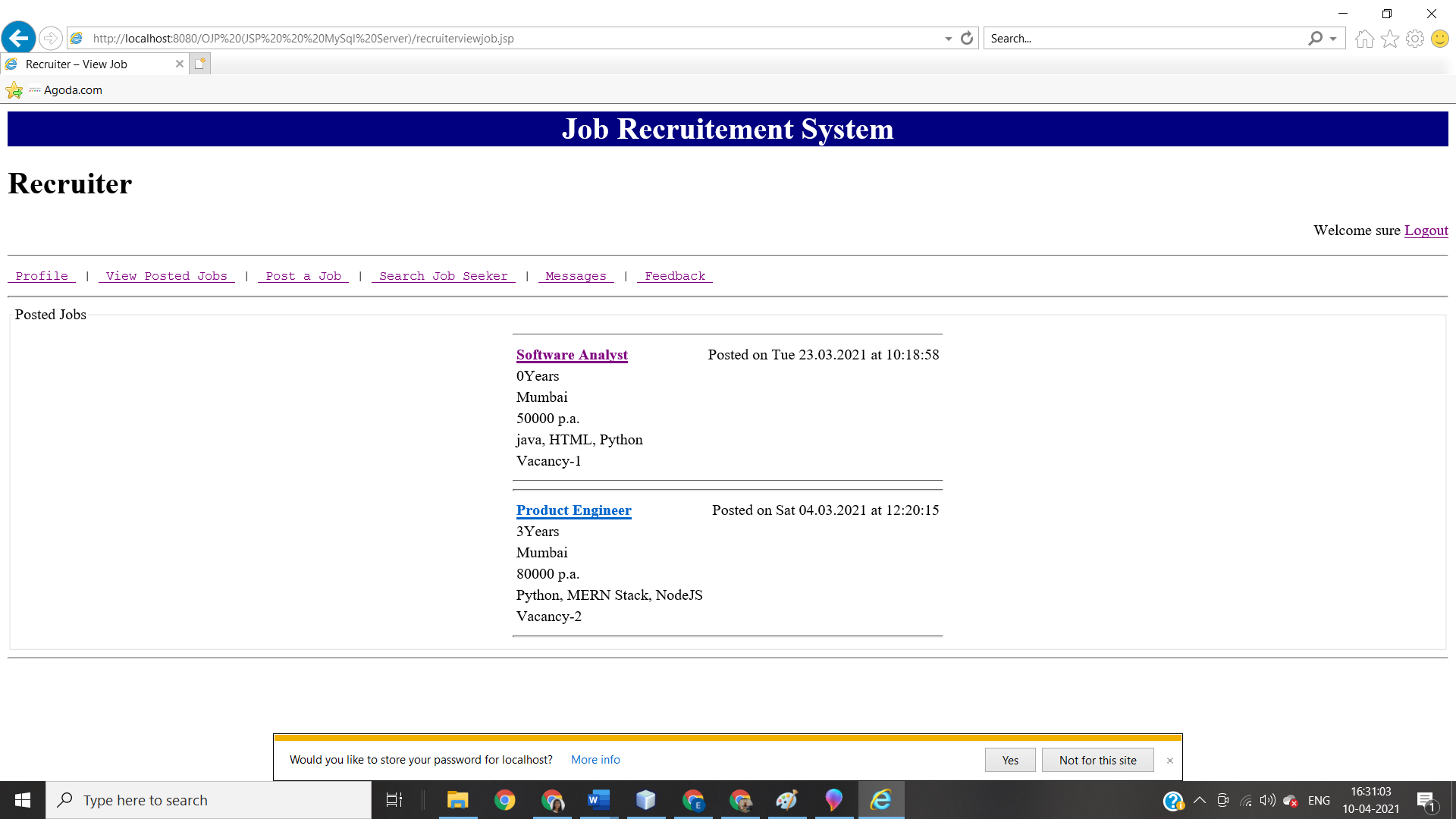
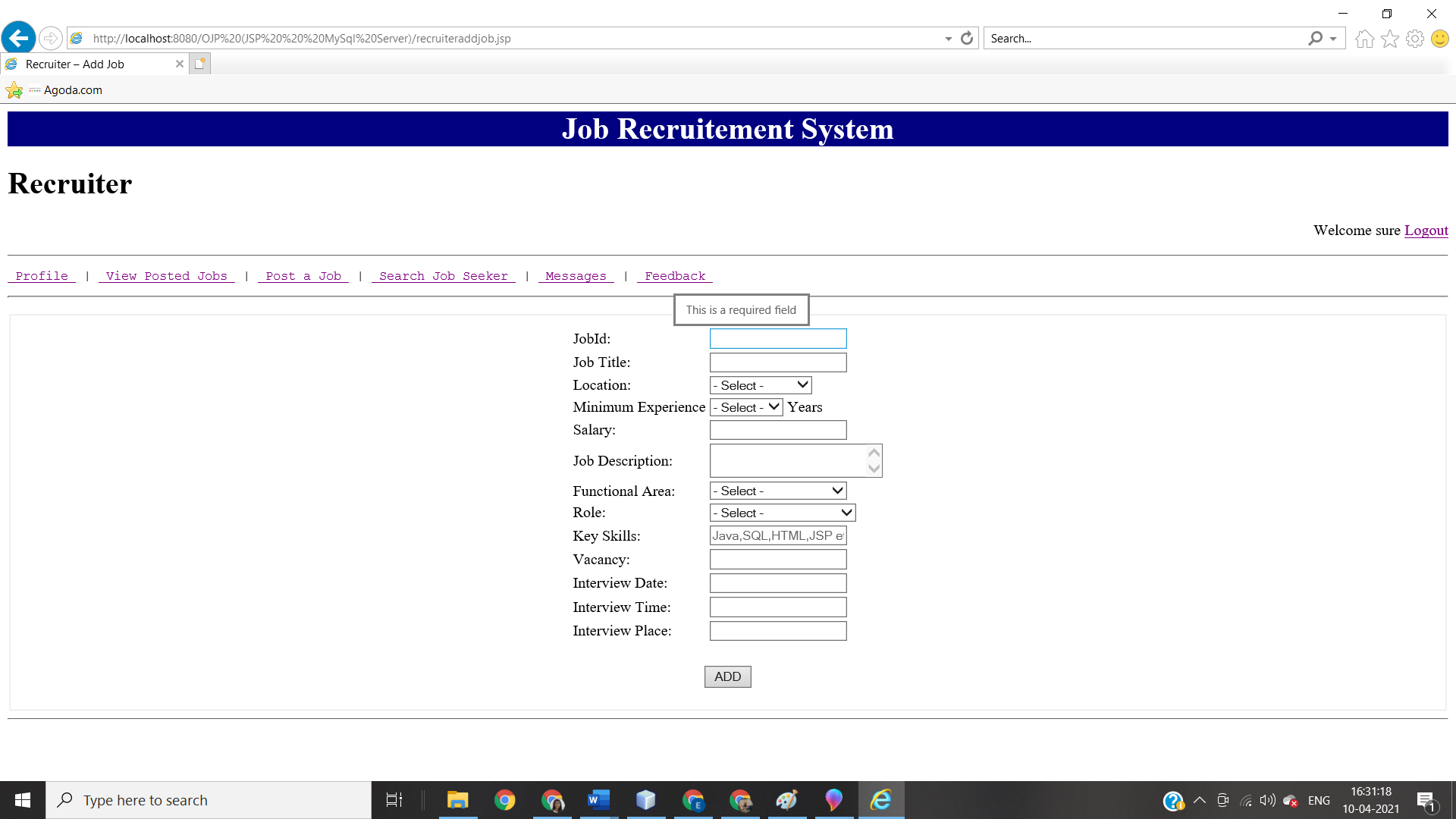
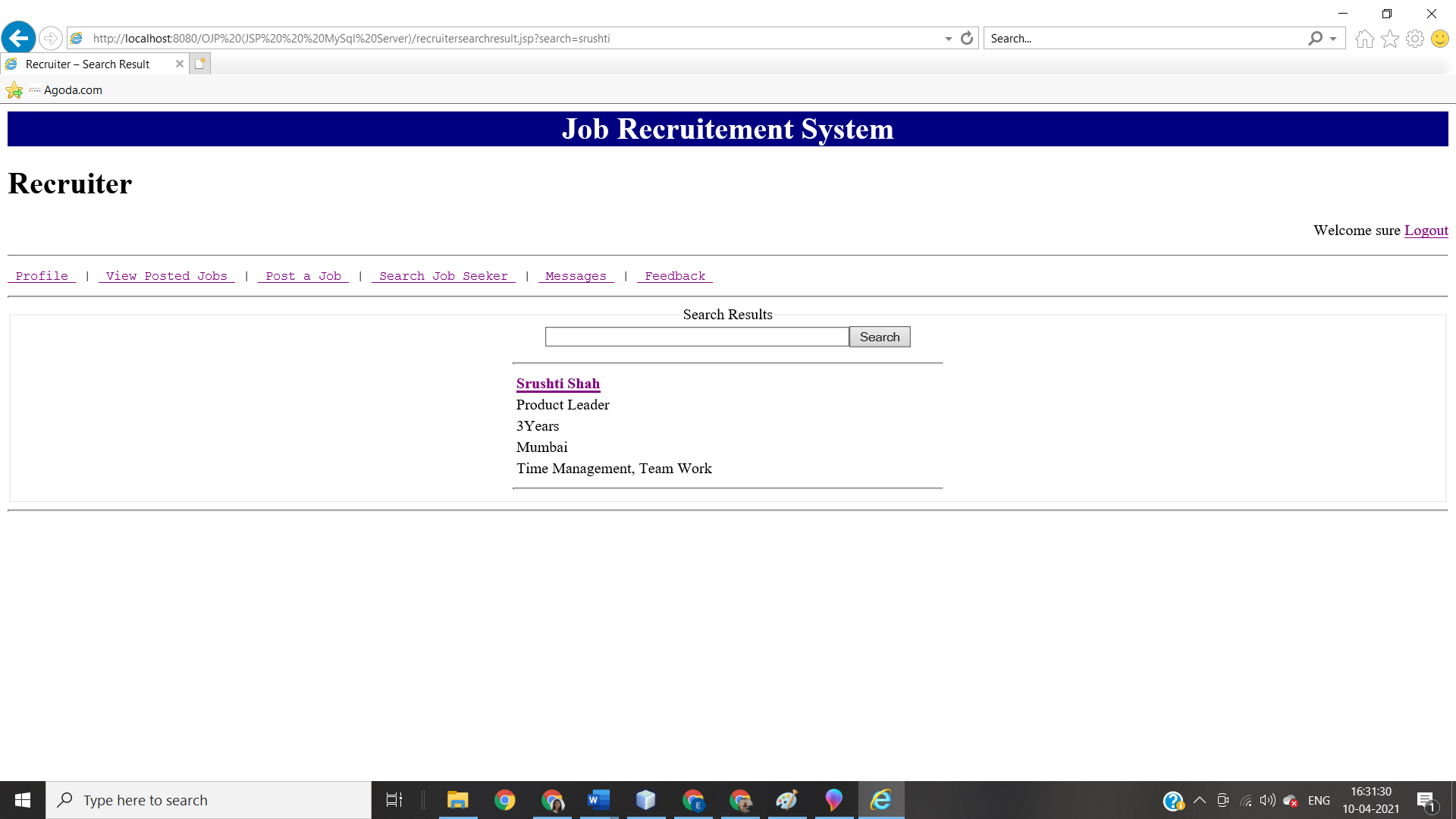
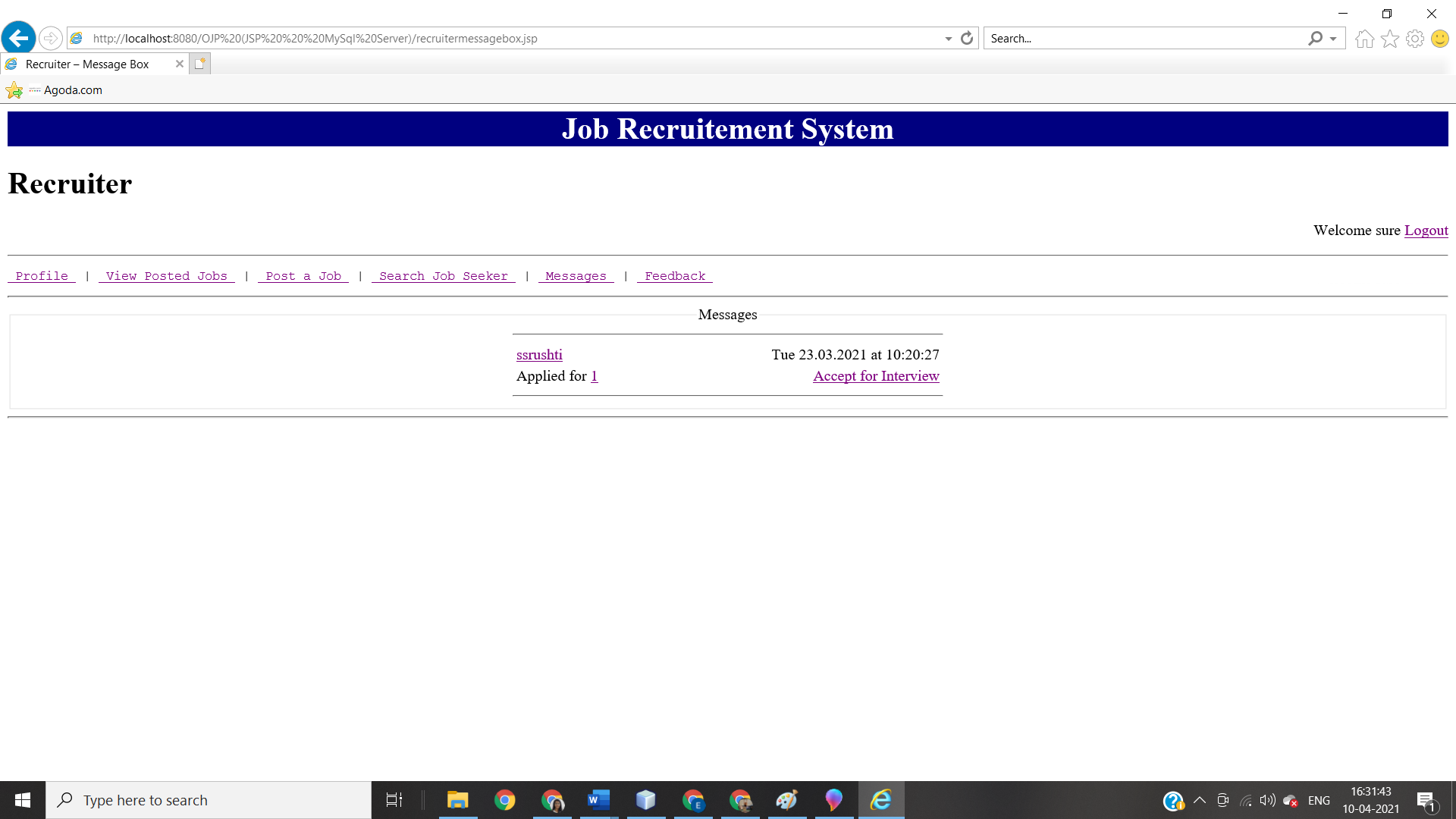
|  |  |
| --- | --- |
| Name | Feedback |
| Summary | To get a feedback from the user of the system |
| Rationale | To improve the system and understand the bugs |
| Actor | Recruiter and Job Seeker |
| Pre-Condition | Job Seeker or Recruiter has registered through the website |
| Basic course of Event | 1. Job Seeker or Recruiter has registered through the website 2. GO to the feedback tab from dashboard 3. Enter and submit the feedback |
| Post- Condition | The developer receives the feedback |

**Algorithm:**

**Pseudo Code:**

1. Start
2. Go to feedback section
3. Enter username, email, feedback
4. Click Send
5. End

***GUI DESIGN***

1. Login
2. Register
3. JobSeeker Dahsboard
4. Job seeker profile
5. Upload Resume 
6. Search job
7. Search Job result
8. Apply for job
9. Application status
10. Feedback
11. Recrutier profile
12. View posted job
13. Post a job 
14. Search Job seeker
15. Check application

***CONCLUSION:*** Thus, from this experiment I have implemented object oriented analysis and design for Job Recruitment System