UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE) KONAM,

NAGERCOIL – 629 004

RECORD NOTE BOOK

CCS356-OBJECT ORIENTED SOFTWARE ENGINEERING

REGISTERNO :

NAME :

YEAR/SEMESTER:

DEPARTMENT :

UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE) KONAM,

NAGERCOIL - 629 004

Register No:

Certified that, this is the bonafide record of work done by ***Mr./Ms****……..of VI Semester Computer Science and Engineering of this college, in the* ***CCS356–OBJECT ORIENTED SOFTWARE ENGINEERING*** *during academic year 2023-2024 in partial fulfillment of the requirements of the B.E Degree course of the Anna University Chennai.*

Staff-in-charge **Head of the Department**

This record is submitted for the University Practical Examination held on……

Internal Examiner **External Examiner**

INDEX

EXNO: DATE:	**IDENTIFY A SOFTWARE SYSTEM THAT NEEDS TO BE DEVELOPED –RECRUITMENT SYSTEM**

Aim:

To identify a software system that needs to be developed – Recruitment System.

Introduction:

In the dynamic realm of modern businesses, effective recruitment practices are paramount for organizational success. A Recruitment Management System stands as the cornerstone in the quest for talent acquisition, offering a strategic approach to streamline processes, elevate candidate experience, and fortify employer branding. This introduction underscores the pivotal role of such a system in today's competitive talent landscape and delineates its fundamental components.

Features of Recruitment Management System:

1. **Candidate Relationship Management:**

- Allow administrators to manage client profiles, contracts, and requirements.
- Implement authentication and authorization mechanisms for secure client access.
- Define client categories and access levels (e.g., premium clients, standard clients).

2. **Job Posting and Talent Acquisition:**

- Facilitate seamless job posting across multiple platforms, including company careers page, job boards, and social media channels.
- Provide tools for recruiters to source and attract top talent, including resume parsing, candidate matching algorithms, and talent pools.

3. **Task Assignment and Progress Tracking:**

- Enable real-time monitoring of recruitment tasks, such as candidate sourcing, screening, and interview scheduling.
- Streamline task assignment processes by matching candidate skills and qualifications with job requirements.

4. **Communication and Collaboration:**

- Incorporate Offer communication tools, such as messaging, email, and video conferencing, for seamless interaction between recruiters, hiring managers, and candidates.
- Implement notifications and alerts to keep stakeholders informed about task updates, interview invitations, and application statuses.

5. **Quality Assurance and Compliance:**

- Incorporate quality control mechanisms to ensure compliance with hiring standards and regulations.
- Enable recruiters to track recruitment metrics, such as time-to-hire, applicant satisfaction, and diversity metrics, to continuously improve recruitment processes.
- Provide feedback mechanisms for candidates and hiring managers to evaluate the recruitment experience and suggest improvements

6. **Reporting and Analytics**:

- Generate comprehensive reports and analytics on recruitment metrics, such as applicant demographics, source effectiveness, and hiring trends.
- Utilize data insights to make informed decisions, optimize recruitment strategies, and align hiring processes with organizational goals.

7. **Data Security and Privacy:**

- Ensure compliance with data protection regulations, such as GDPR or EEOC, by securely managing candidate data and consent forms.
- Implement data encryption, access controls, and audit trails to protect sensitive information and mitigate data breaches.

Result:

Thus the software system that is need to be developed for Recruitment management system was executed successfully.

Ex No: Date:	**DOCUMENT THE SOFTWARE REQUIREMENTS SPECIFICATION(SRS) FOR THE RECRUITMENT SYSTEM**

AIM:

To implement a software for Recruitment system.

PROBLEM STATEMENT:

- With the dynamic evolution of the job market and the increasing diversity in recruitment needs across industries, there is a growing demand for streamlined and efficient recruitment systems.
- In this competitive landscape, organizations, particularly those with extensive recruitment requirements, encounter a significant obstacle: the effective management of candidate applications, ensuring thorough evaluation, and smooth progression through the recruitment pipeline.
- The primary challenge lies in establishing a cohesive system capable of receiving candidate applications from various sources and in different formats. Once received, the system must efficiently assign these applications to the appropriate recruiters or hiring teams for comprehensive assessment and timely decision-making.
- Furthermore, the system must integrate features for candidate evaluation, ensuring compliance adherence, scheduling interviews, and providing feedback to applicants. This integration is crucial for maintaining alignment with regulations and fostering a positive candidate experience, which is essential for organizational reputation.
- Addressing these challenges is critical for organizations to maintain recruitment efficiency, eliminate processing bottlenecks, and uphold their competitiveness in talent acquisition.

SOFTWARE REQUIREMENT SPECIFICATION:

INTRODUCTION:

- Recruitment remains a crucial function within organizations, serving as a cornerstone for sourcing, evaluating, and integrating talent to meet business objectives. Traditionally, companies managed recruitment internally. However, evolving business practices have given rise to specialized services such as Recruitment Process Outsourcing (RPO). Much like broader Business Process Outsourcing (BPO), RPO involves delegating recruitment functions to external service providers.

PURPOSE:

- Substantial reduction in manual paperwork and associated expenses.
- Implementation of a seamless, fully integrated recruitment process to elevate servicequality for candidates and hiring stakeholders alike.
- Cost-effectiveness achieved through automating candidate data entry processesand employing advanced algorithms for streamlined application routing to recruiters.
- Instantaneous access to candidate profiles and recruitment progress for recruitment management and stakeholders, irrespective of their location Enhanced data security through digital storage of candidate information.
- Improved user experience with the capability to search and manage candidate profiles online, fostering efficiency and convenience throughout the recruitment journey.

INTENDED AUDIENCE AND READING SUGGESTIONS:

This Recruitment System document is primarily intended for the project development team,encompassing key stakeholders involved in the creation and maintenance of the system. This includes:

Recruitment Project Manager:

The Recruitment Project Manager plays a crucial role in overseeing the development and implementation of the RMS. This document provides essential information for cost estimation, project planning, and ensuring alignment with customer requirements.

Recruitment Developer:

For the Recruitment Developer, this SRS document serves as a comprehensive guide to understand the functional requirements and design specifications of the RMS. It ensures that the developed product meets the needs and expectations of the customer.

Documentation Writer:

The Documentation Writer utilizes the SRS document to gain a clear understanding of the system's functionality and requirements. This knowledge forms the basis for creating user manuals and instructional guides to assist users in navigating the RMS effectively.

Recruitment Tester:

The Recruitment Tester relies on the SRS document to comprehend the functional requirements and expected behavior of the RMS. This enables effective test case design and validation to ensure the software's reliability and functionality.

Maintenance Engineers:

For Maintenance Engineers tasked with maintaining and updating the RMS, the SRS document serves as a valuable reference tool. It provides insights into the system's functionality, enabling efficient troubleshooting, enhancement, and modification as needed.

DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS:

a. **Client**- Refers to organizations seeking recruitment services.
b. **Recruitment System (RS)**- Refers to the system designed for managing recruitment processes.
c. **Recruitment Agency (RA)-** Refers to the organization providing recruitment services.
d. **HTML**- Hypertext Markup Language, utilized for structuring web documents.
e. **J2EE** – Java 2 Enterprise Edition, a Java platform for developing and running distributed enterprise applications.
f. **HTTP**- Hyper Text Transfer Protocol , facilitating data communication on the World Wide Web.
g. **TCP/IP** –Transmission Control Protocol/Internet Protocol, the standard communication protocol for internet connectivity.

REFERENCES:

IEEE Recruitment System Requirement Specification format.

TECHNOLOGIES TO BE USED:

- HTML (Hypertext Markup Language) - for creating structured web pages.
- JSP (JavaServer Pages) - for dynamic content generation server-side.
- Javascript - for client-side scripting to enhance user interactions.

TOOLS TO BE USED:

- Star UML

OVERVIEW:

The SRS comprises two sections: an Overall Description and Specific Requirements.

The Overall Description will outline the primary functions of system components and their interaction.

OVERALL DESCRIPTION:

PRODUCT PERSPECTIVE:

The Recruitment System acts as a bridge between the 'recruitment agency' and the 'client'. This system aims to streamline the interface while ensuring the security of stored data. The goal is to minimize the time taken for the client to receive candidate information and documents.

SOFTWARE INTERFACE:

- **Front End Client**- The exporter online interface is built using JSP and HTML.
- **Web Server** – Apache Tomcat Server (Oracle Corporation)
- **Back End**- Oracle 11g database

HARDWARE INTERFACE:

The Recruitment System's server is directly connected to client systems via FTP (File Transfer Protocol). Client systems have access to the database on the server.

SYSTEM FUNCTIONS:

- The Recruitment System has been designed with a base product tailored to meet the requirements of the recruitment vertical.
- It is a comprehensive web-based solution that enables organizations in the recruitment domain.
- Organizations can store candidate documents automatically on the server.
- Candidates can upload documents from various locations, and an automatic uploader module, which can be scheduled, facilitates the automatic upload of documents.
- Uploaded documents are automatically routed to specific users based on rules defined by the admin user.
- Recruitment staff, including Data Entry and Quality Check users, can access the documents via a web-based interface to perform their respectiveoperations on the uploaded documents

USER CHARACTERISTICS:

- **Recruitment Organization:** They are the entities interested in utilizing the recruitment system to efficiently manage their hiring processes. They aim to receive job applications from candidates and store relevant information in the database.
- **Client:** The client has specific privileges to outsource their recruitment needs and approve the processing of candidate applications. They may have a team to assist in verifying candidate information and providing recommendations on whether to proceed with the hiring process.

CONSTRAINTS:

- The Recruitment System requires users to have access to a computer with FTP capabilities to submit candidate information.
- While security measures are prioritized, the online environment is susceptible to potential intrusions, necessitating ongoing monitoring.
- Users must exercise caution when submitting candidate information, as accuracy and attention to detail are essential.

ASSUMPTIONS AND DEPENDENCIES:

It is assumed that both the organization and clients have a fundamental understanding of computers and proficiency in the English language.

The system depends on the implementation of comprehensive privacy and security measures to protect documents and client information.

SYSTEM FEATURES:

System features refer to the specific functionalities or capabilities that the Recruitment System offers to its users. Here are some potential system features for a Recruitment Management System:

- **Candidate Application Intake and Management:** Ability to receive and store candidate applications uploaded by applicants. Organization and categorization of applications for efficient processing
- **Task Assignment and Workflow Management:** Automated routing of applications to designated recruiters or hiring teams based on predefined criteria. Task tracking and monitoring to ensure timely processing and evaluation.

- **Candidate Evaluation and Screening:** User-friendly interfaces for recruiters to assess and screen candidate applications. Real-time evaluation tools to ensure candidate suitability and alignment with job requirements**.**

- **Interview Scheduling and Coordination:** Integration of features for scheduling interviews with candidates. Automated reminders and notifications to candidates and interviewers**.**
- **Compliance Adherence:** Built-in mechanisms to ensure compliance with regulations and organizational policies throughout the recruitment process.

Reporting and Analytics:

Generation of reports on recruitment process metrics, such as Application ,volume, hire,camdidate conversion rate. Analytical tool to deliver insights and optimize recruitment strategies.

SYSTEM REQUIREMENTS:

System requirements outline the capabilities and constraints that the Recruitment System must meet to fulfill the needs of its users and stakeholders. These requirements are typically categorized into functional and non-functional requirements.

Functional Requirements:

1. Candidate Application Management:

- Ability to receive, store, and organize candidate applications.
- Facilitate easy retrieval and access to candidate information for recruiters.

2. Task Assignment and Workflow Management:

- Automated routing of applications to appropriate recruiters or hiring teams based on predefined criteria.
- Task tracking and monitoring to ensure timely processing and evaluation of applications.

3. Candidate Evaluation and Screening:

- User-friendly interfaces for recruiters to assess and screen candidate applications.
- Real-time evaluation tools to ensure candidate suitability and alignment with job requirements.

4. Interview Scheduling and Coordination:

- Integration of features for scheduling interviews with candidates.
- Automated reminders and notifications to candidates and interviewers.

5. Compliance Adherence:

- Implementation of mechanisms to ensure compliance with regulations and organizational policies throughout the recruitment process.

Non-Functional Requirements:

1. Security:

- Robust security measures to protect candidate data and maintain confidentiality.
- Access control mechanisms to restrict unauthorized access to sensitive information.

2. Performance:

- Fast and responsive system performance to handle large volumes of candidate applications efficiently.
- Minimal downtime and system interruptions to ensure continuous availability.

3. Usability:

- Intuitive user interfaces for easy navigation and usage by recruiters and hiring teams.
- Accessibility features to accommodate users with diverse needs.

4. Reliability:

- Dependable system reliability to ensure consistent and accurate processing of candidate applications.
- Backup and recovery mechanisms to prevent data loss in case of system failures.

5. Integration:

- Seamless integration with existing recruitment tools and systems used by the organization.

- .

Result :

Thus the documentation of the Software Requirement System(SRS) for Recruitment System was developed successfully.

EX NO: DATE:	**TO IDENTIFY THE USE CASES AND DEVELOP THE USE CASE MODEL**

AIM :

To identify the use cases and develop use case model.

Introduction of Use Case Diagram:

A use case diagram is a type of behavioral diagram in the Unified Modeling Language (UML) that represents the interactions between a system and its actors (users or external systems) in terms of the system's behavior. It provides a graphical overview of the functionalities provided by a system and the actors involved in those functionalities. Use case diagrams are widely used in software engineering to capture the requirements of a system and to visualize the high-level system structure.

Purpose of Use Case Diagram:

The purpose of a use case diagram is to visually represent the functional requirements of a system from the perspective of its users or external systems. It helps in understanding the system's behavior by illustrating how users interact with the system to accomplish specific tasks or goals. Use case diagrams facilitate communication between stakeholders, including developers, designers, and clients, by providing a clear and concise representation of the system's functionality.

Components of Use Case Diagram:

Actors: In the recruitment system, the actors represent the various entities interacting with the system. These typically include the hiring organization, job applicants, and recruitment agents or managers.

Use Cases: Use cases in the recruitment system describe the functionalities or services provided by the system to facilitate the recruitment process. Examples of use cases may include "Post job openings," "Receive and review applications," "Schedule interviews," "Conduct assessments,""Makejoboffers," and"Onboard newhires."

Relationships: Relationships in the recruitment system illustrate the connections between actors and use cases. Associations depict the interactions between actors and use cases, while dependencies signify the reliance of certain use cases on others to complete the recruitment process effectively.

Benefits of Use Case Diagram for Recruitment System:

Requirements Elicitation:

Use case diagrams assist in eliciting and defining the functional requirements of a recruitment system by identifying the key interactions between stakeholders such as hiring organizations, job applicants, and recruitment agents.

Visualization:

Use case diagrams offer a visual representation of the recruitment system's functionalities, enhancing stakeholders' comprehension of how the system operates and how different entities interact within it.

Communication:

Use case diagrams act as a communication tool among various stakeholders involved in the recruitment process, including HR personnel, hiring managers, recruiters, and job applicants. They facilitate a shared understanding of the system's requirements and processes.

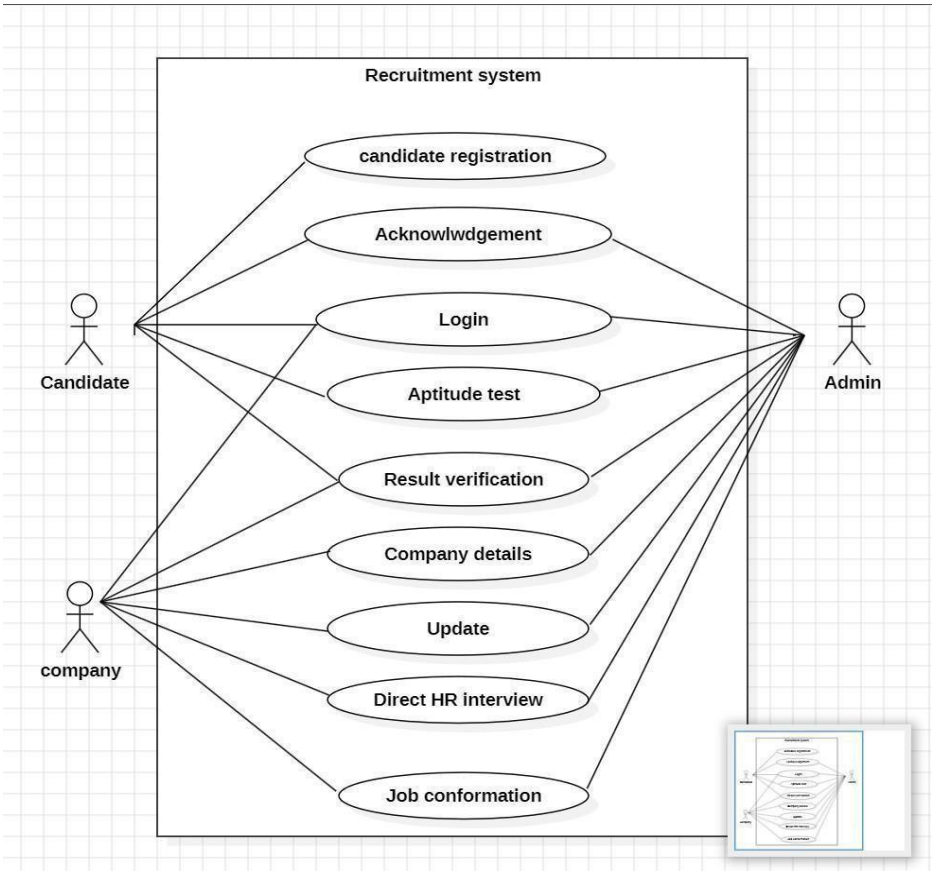
Requirements Verification:

Use case diagrams enable stakeholders to verify whether all essential recruitment functionalities and interactions have been adequately addressed in the system design, ensuring that the system aligns with the organization's recruitment needs and objectives.

Search for Job Candidates:

- Involves the process of the hiring organization identifying potential candidates for open for positions.
- Recruitment staff may utilize various channels such as online job boards, social media platforms, referrals, or recruitment agencies to source candidates.
- Factors considered during the search may include candidate qualifications, skills, experience, cultural fit, salary expectations, and availability.

Use Case diagram for Recruitment system:



Negotiate the Project:

- Once a suitable candidate is identified, negotiation with the candidate ensues to establish terms of employment.
- Negotiation may involve discussions on salary, benefits, job responsibilities, work schedule, and other employment terms.
- Hiring managers or HR representatives may engage in communication with candidates through interviews, phone calls, or emails to reach mutual agreements.
- The negotiation phase aims to ensure alignment between candidate expectations and the hiring organization's requirements.

Receive Applicant Documents:

- After the job offer is accepted by the candidate, the candidate uploads necessary documents or information to the hiring organization's system.
- Applicant documents may include resumes, cover letters, portfolios, certifications, or any other relevant information required for the recruitment process.
- The recruitment system mayprovide secure upload interfaces or email communication for candidates to submit their documents.
- Candidates may also utilize integration APIs or online application portals to streamline the document submission process, ensuring efficiency and accuracy in the recruitment process.

Evaluate Applicant Qualifications:

- Evaluation tasks may vary depending on the job position and could include reviewing resumes, assessing qualifications, analyzing experience, and verifying credentials.
- Recruitment staff may utilize applicant tracking systems (ATS), screening tools, or manual review processes to evaluate applicant qualifications accurately and efficiently.

Candidate Screening:

- After the evaluation process, the recruitment system conducts candidate screening to ensure the suitability of applicants for the job position.
- Screening may involve automated algorithms, manual review by recruiters, or a combination of both.
- Criteria for screening may include qualifications, skills, experience, cultural fit, and alignment with job requirements.
- Screening measures aim to identify and select the most qualified candidates for further consideration, ensuring a high-quality pool of applicants for the hiring organization.

Job Offer Delivery:

- After the candidate screening process, the finalized job offer is prepared for delivery to the selected candidate.
- Delivery methods may vary depending on the organization's practices and candidate preferences, including email, postal mail, or electronic signature platforms.
- The hiring organization ensures secure transmission of the job offer to protect confidentiality and privacy.

RESULT:

Thus the usecase diagram for recruitment management system is implemented and executed successfully.

EX NO: DATE:	**IDENTIFY THE CONCEPTUAL CLASSES AND DEVELOP A DOMAIN MODEL AND DERIVE CLASS DIAGRAM**

Aim:

To identify the conceptual classes and develop a domain model and derive a class diagram for a Recruitment System.

Conceptual Classes:

1. **Employer:** Represents the companies or organizations seeking to hire employees. Attributes may include employer ID, name, contact information, etc.
2. **Job Position:** Represents the specific roles or positions offered by employers. Attributes may include position ID, title, description, requirements, etc.
3. **Candidate:** Represents individuals seeking employment opportunities. Attributes may include candidate ID, name, contact information, skills, etc.
4. **Recruiter:** Represents the individuals or entities responsible for managing the recruitment process. Attributes may include recruiter ID, name, contact information, etc.
5. **Application:** Represents the submissions made by candidates for specific job positions. Attributes may include application ID, candidate ID, position ID, status, etc.
6. **Interview:** Represents the scheduled meetings or assessments between recruiters and candidates. Attributes may include interview ID, candidate ID, recruiter ID, date, feedback, etc.
7. **Department:** Represents the organizational units within the employer's company. Attributes may include department ID, name, description, etc.
8. **Offer:** Represents the formal job offers extended to candidates by employers. Attributes may include offer ID, candidate ID, position ID, date, status, etc.

Domain Model for Recruitment System: Agency:

Attributes:

- Agency_name: String
- Address: String
- Contact_no: Integer
- Emailid: String

Operations:

- addAgency()
- updateAgency()

Job Vacancy:

Attributes:

- Job_title: String
- Vacancy_id: Integer
- Description: String
- Requirements: String
- Location: String
- Salary_range: String
- Status: String

Operations:

- addVacancy()
- deleteVacancy()
- updateVacancy()

Employer:

Attributes:

- Employer_name: String
- Employer_id: Integer
- Company_name: String
- Address: String
- Contact_no: Integer
- Emailid: String

Operations:

- addEmployer()
- deleteEmployer()
- updateEmployer()

Candidate:

Attributes:

- Candidate_name: String

- Candidate_id: Integer
- Address: String
- Phone_no: Integer
- Emailid: String
- Skills: String
- Experience: Integer
- Education: String

Operations:

- addCandidate()
- deleteCandidate()
- updateCandidate()

Recruitment Process:

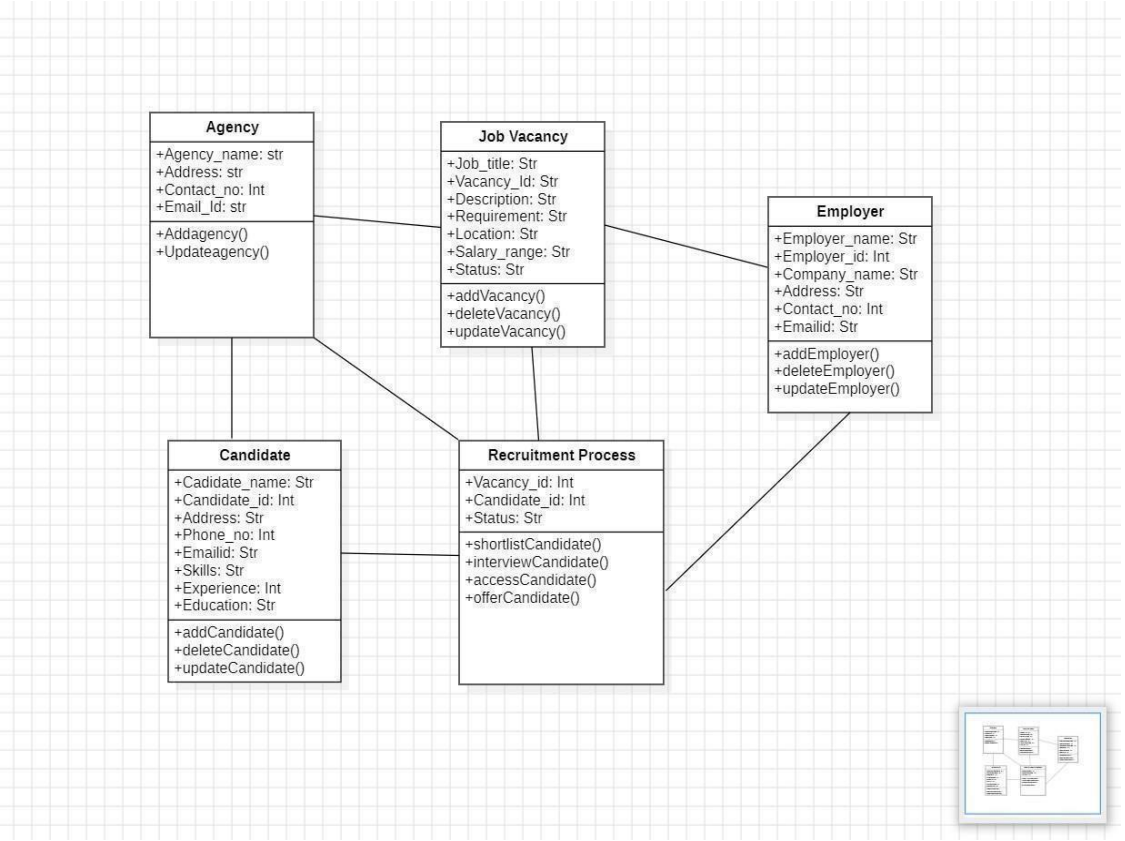
Attributes:

- Vacancy_id: Integer
- Candidate_id: Integer
- Status: String

Operations:

- shortlistCandidate()
- interviewCandidate()
- assessCandidate()
- offerCandidate()

Class diagram for Recruitment system:



RESULT:

Thus the class diagram for the recruitment system is implemented and executed successfully.

EXNO: DATE:	**USING THE IDENTIFIED SCENARIOS, FIND THE IN BETWEEN OBJECTS AND REPRESENTS THEM USING UML SEQUENCE AND COLLABORATION DIAGRAM**

AIM:

To find the interaction between objects and represent them using UML Sequence and collaboration diagram.

Introduction of Interaction Diagrams:

Overview of Interaction Diagrams:

1. Interaction diagrams are graphical representations in UML (Unified Modeling Language) used to visualize the dynamic behavior of a system by showing how objects interact over time.
2. These diagrams capture the flow of messages or actions between objects during the execution of a system or process.

Categorization into Two Types:

Interaction diagrams are categorized into two main types: Activity Diagrams and Sequence Diagrams.

Activity Diagrams: These diagrams focus on modeling the flow of activities or tasks within a system, showing these quence of actions and decisions.

Sequence Diagrams: These diagrams focus on modeling the chronological order of messages exchanged between objects in a system, illustrating how objects interact over time.

Sequence Diagram:

- A sequence diagram is a type of interaction diagram in UML that represents theinteractions between objects over time.
- It illustrates the sequence of messages exchanged between objects within asystem to accomplish a specific task or scenario.

Components of a Sequence Diagram:

1. **Lifelines:**

Lifelines represent the lifespan of objects participating in the interaction within the system.

- Each object involved in the interaction is represented by a lifeline, which is depicted as a vertical dashed line.
- The length of the lifeline represents the duration of the object's existence or involvement in the interaction.

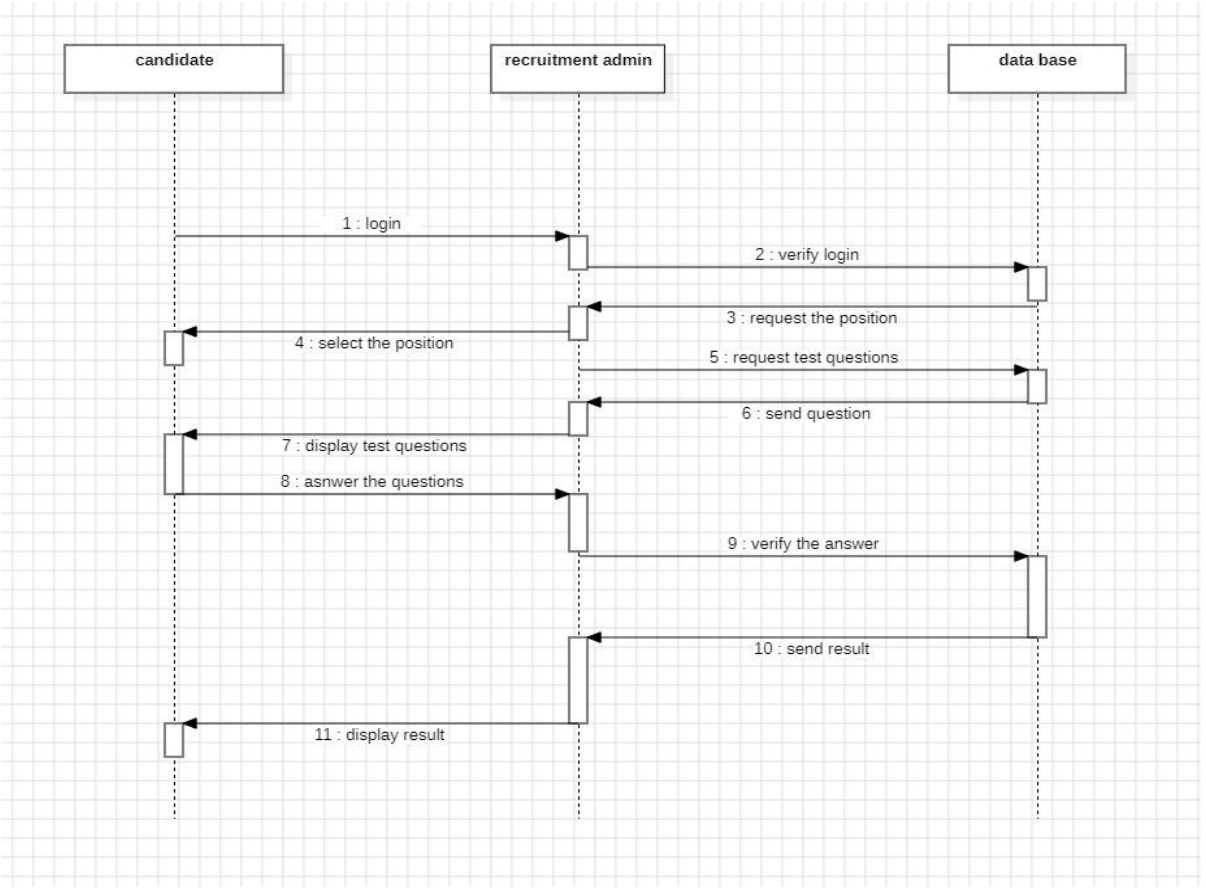
2. **Messages:**

- Messages represent communication or interactions between objects in the system.
- They depict the flow of control or data between objects during the execution of a scenario.
- Messages are represented by arrows that connect lifelines, indicating the direction of communication.

There are different types of messages:

1. **Synchronous Message:** Represents a direct call or invocation between objects, denoted by a solid arrow.
2. **Asynchronous Message:** Represents a non-blocking call or invocation between objects, denoted by a dashed arrow.
3. **Return Message:** Indicates the response or return value from a method call, represented by a dashed arrow with a labeled return value.
4. **Self-Message:** Represents a message sent from an object to itself, depicted by a looped arrow that returns to the same lifeline.

Sequence Diagram:



Result :

Thus the interaction between objects and represent them using UML Sequence and Collaboration diagram was executed successfully.

EX NO:

DATE

Draw relavant state chart and activity diagram for the same system

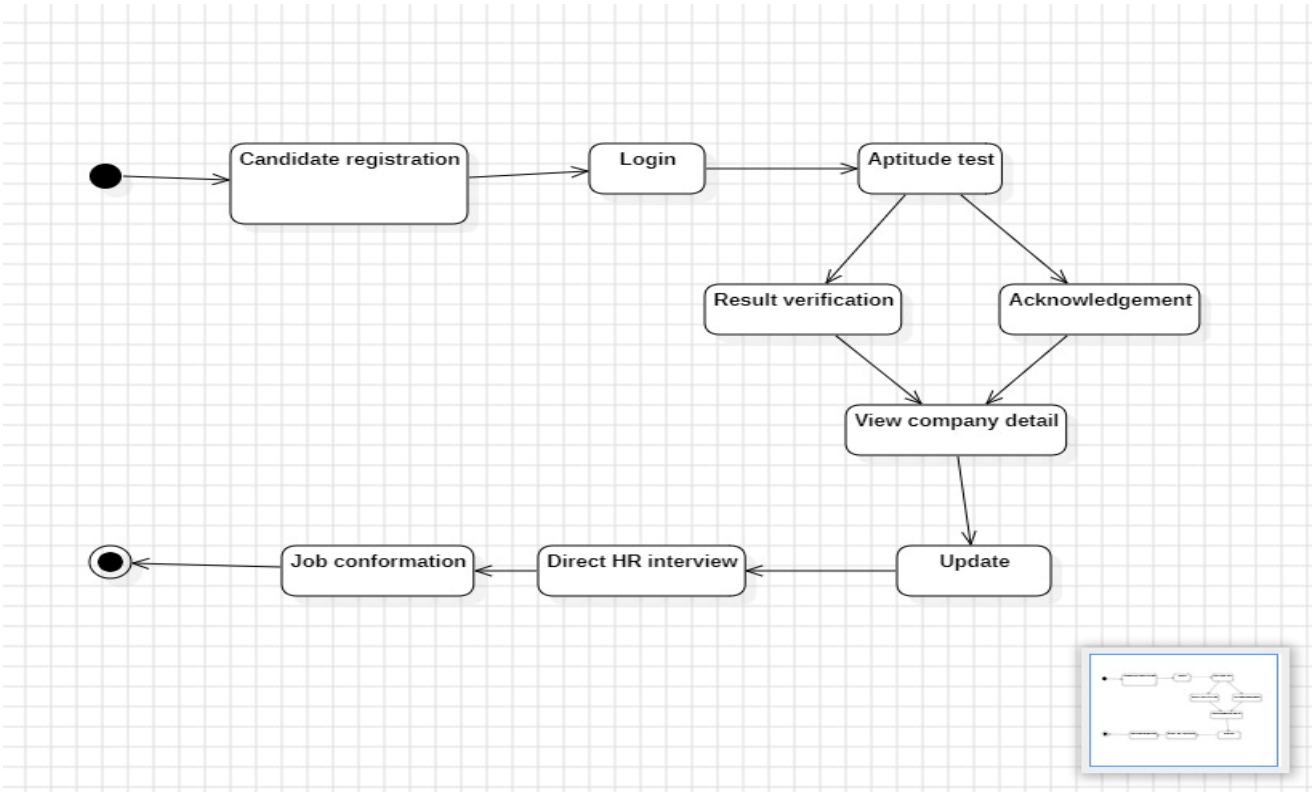
AIM:

To draw relevant state chart and activity diagram for same system.

Start chart Diagram:

A state chart diagram, also known as a state machine diagram, is a type of diagram used in UML (Unified Modeling Language) to describe the behavior of a system by showing its states and the transitions between those states. For a campus recruitment system, the state chart diagram would represent the various states an application goes through during the recruitment process and the events that cause transitions between these states.

Activity Diagram:



Initial Node:

Represents the starting point of the activity diagram. It is denoted by a solid circle with an arrow pointing outward. The initial node indicates where the process flow begins within thediagram. It is the first step executed in the sequence of activities.

Final Node:

Represents the ending point of the activity diagram. It is denoted by a solid circle with a border. The final node indicates the completion of the process flow within the diagram. It isthe last step executed in the sequence of activities.

Action Node (Activity):

Represents a specific task or operation to be performed within the system. It is denoted by a rectangle with rounded corners. Action nodes represent individual activities or tasks that need to be completed as part of the overall process flow. They encapsulate the actions performed by the system or its components.

Decision Node (Branch):

Represents a branching point based on a condition or decision. It is denoted by a diamond shape. Decision nodes allow the process flow to take different paths based on the evaluation of one or more conditions. They enable conditional behavior within the diagram, where different actions are taken depending on the outcome of a decision.

Merge Node:

Represents the merging of multiple paths of control flow back into a single path. It is denoted by a diamond shape with a single incoming flow and multiple outgoing flows. Merge nodes combine multiple paths of control flow back into a single path. They are used to synchronize parallel branches of the process flow, ensuring that all parallel activities are completed before continuing with the next step.

Fork Node:

Represents the creation of parallel paths of control flow. It is denoted by a bar with a single incoming flow and multiple outgoing flows. Fork nodes split the process flow into multiple parallel paths, allowing activities to be executed concurrently. They enable parallelism within the diagram, where multiple tasks can be performed simultaneously.

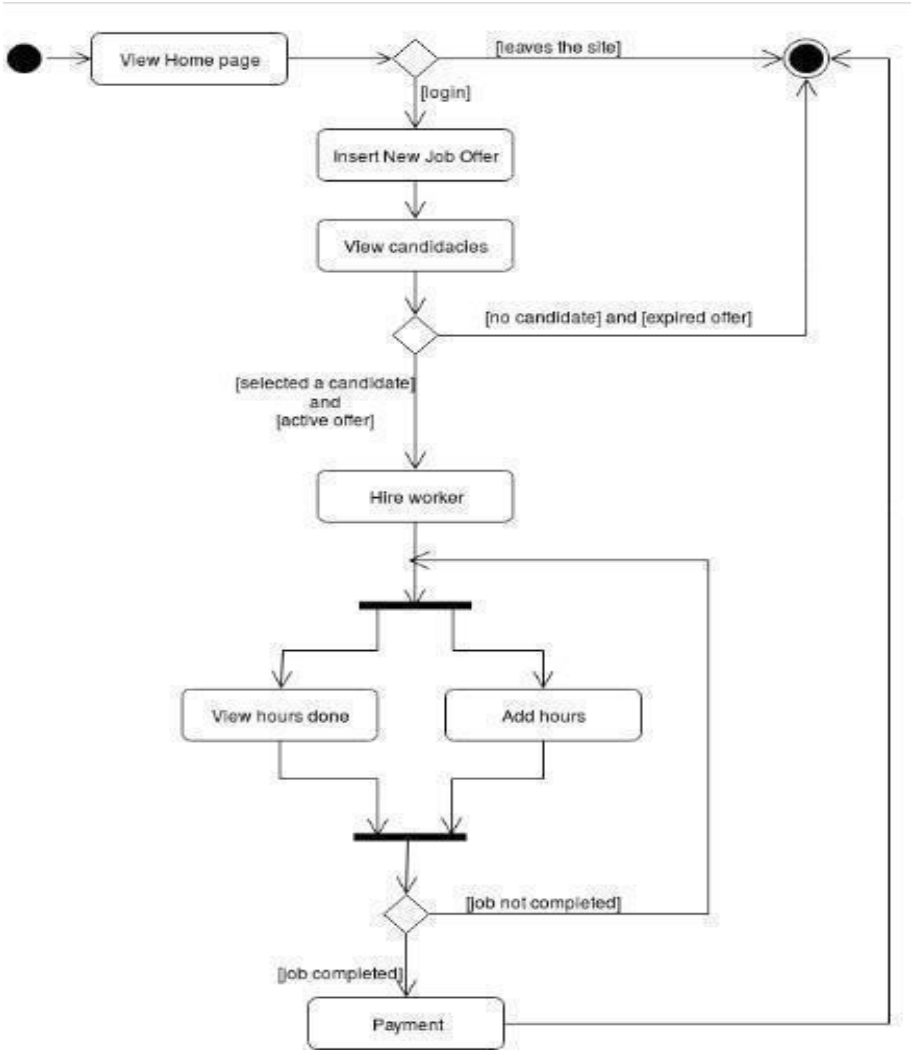
Join Node:

Represents the merging of parallel paths of control flow back into a single path. It is denoted by a bar with multiple incoming flows and a single outgoing flow. Join nodes synchronize parallel branches of the process flow, ensuring that all parallel activities are completed before continuing with the next step. They are used in conjunction with fork nodes to manage parallelism within the diagram.

Control Flow Arrows:

- Control flow arrows indicate the direction of flow between nodes in the activity diagram.
- Solid arrows represent the primary flow of control, while dashed arrows represent alternative or conditional flows.

Activity Diagram For Recruitment System:



Activities:

1. **Search for Job**: The Job Seeker searches for job opportunities.
2. **Negotiate Terms**: The Recruiter negotiates the terms and conditions of the job with the hiring company or client.
3. **Upload Documents**: The Job Seeker uploads necessary documents such as resumes, cover letters, and certificates for job applications.
4. **Perform Conversion**: The system automatically converts uploaded documents into a standardized format for easier processing.
5. **Request Additional Information**: If necessary, the Recruiter requests additional information or documents from the Job Seeker.
6. **Shortlist Candidates**: Based on the quality checks, the Recruiter shortlists candidates for further consideration.
7. **Schedule Interviews**: The Recruiter schedules interviews with shortlisted candidates.
8. **Conduct Interviews**: The Recruiter conducts interviews with the shortlisted candidates.
9. **Send Offer**: After interviews, the Recruiter sends job offers to successful candidates.
10. **Negotiate Salary**: The Recruiter negotiates salary and benefits with the selected candidate.
11. **Finalize Hiring**: Once terms are agreed upon, the hiring process is finalized.
12. **Provide Feedback**: The Recruiter provides feedback to unsuccessful candidates to help them improve their chances in future applications.
13. **Update Job Status**: The system updates the job status to filled or closed.
14. **Monitor Onboarding**: The Recruiter monitors the onboarding process for the newly hired employee.

Decisions:

1. **Satisfied / Not Satisfied:** After receiving the client's feedback, the user decides whether the client is satisfied with the job or not.
2. **Is any Error:** During the quality check process, the user decides if any errors or defects are found in the converted data.
3. **Is any Defects:** During the client's quality check, the user decides if any defects are found in the data.
4. **Yes / No (Send Feedback):** Based on the presence of defects, the user decides whether to send feedback to address them or not.

Result:

Thus the State Chart and Activity diagram for Recruitment System was designed successfully.

EXNO: DATE:	**IMPLEMENT THE SYSTEM AS PER THE DETAILED DESIGN**

AIM:

To implement the Recruitment System as per the detailed design.

IMPLEMENTATION FOR RECRUITMENT SYSTEM:

Candidate. java

```
import  java.util.*;

/**
 *
 */
public class candiate {

   /**
    *Default constructor
    */
   public candiate(){
   }

   /**
    *
    */
   public integer name;

   /**
    *
    */
   public integer id;

   /**
    *
    */
   public void select position(){
      //TO DO implement here
   }

   /**
    *
    */
   public void atten_test(){
      //TODO implement here
   }

       /**
```

```
public void get_result(){
        //TO DO implement here
    }

}
```

Class1.java

```
import java.util.*;

/**
 *
 */
public class Class1{

    /**
     *Default constructor
     */
    public Class1(){
    }

}
```

Database.java

```
import java. util.*;
/**
 *
 */
public class database{
    /**
     *Default constructor
     */
    public database(){
    }
    /**
     *
     */
    public integer questation;
    /**
     *
     */
    public integer login_id;
    /**
     *
     */
    public integer score;
    /**
```

```
     *
     */
    public void display_questation(){
        //TO DO implement here
    }
    /**
     *
     */
    public void calculate(){
        //TO DO implement here
    }
    /**
     *
     */
    public void display_result(){
        //TO DO implement here
    }
    /**
     *
     */
    public void new Operation(){
        //TO DO implement here
    }
}
```

Recruitment admin.java

```
import java.util.*;
/**
 *
 */
public class recruitment admin{
    /**
     *Default constructor
     */
    public recruitment admin(){
    }
    /**
     *
     */
    public integer position_list;
    /**
     *
     */
    public integer questation;
    /**
     *
     */
    public void display_questation(){
        //TO DO implement here
```

```
    }
    /**
     *
     */
    public void get_answer(){
        //TO DO implement here
    }
    /**
     *
     */
    public void display_result(){
        //TO DO implement here
    }
}
```

Result :

Thus the system as per the Detailed Design was implemented successfully.

EX NO: DATE:	**TEST THE SOFTWARE SYSTEM FOR ALL SCENARIOS IDENTIFIED AS PER THE USE CASE**

Aim:

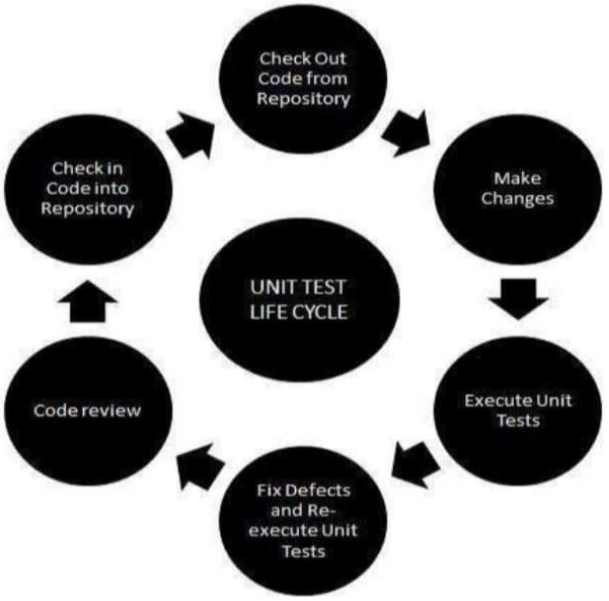
To test the software system for all scenarios identified in the use case diagram.

Recruitment Management System:

In the domain of recruitment, efficient management of processes is essential for operational effectiveness and client contentment. A Recruitment Management System (RMS) simplifies the handling of client requirements, allocation of tasks to recruiters, maintaining quality standards, and enhancing client communication.

Unit Testing:

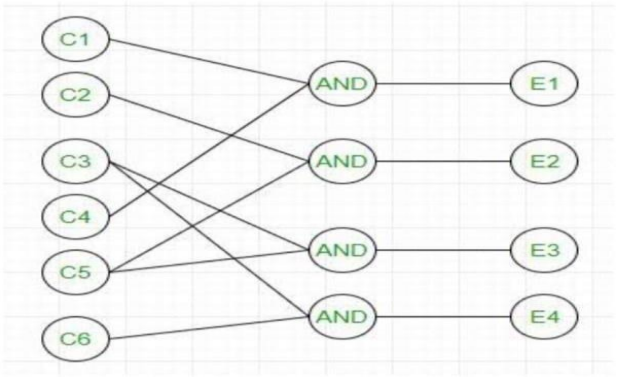
- Unit testing involves testing individual components or modules of the software to ensure they function correctly in isolation.
- Each module, such as client management, project management, employee management, etc., would undergo unit testing.
- Example: Testing the "add Client()" function to ensure that a new client is successfully added to the system.



Black Box Testing:

- Black box testing is a technique where the internal workings of the system are not known to the tester. The tester only tests the system's functionality based on its specifications.
- Testers would input various sets of data into the system and verify that the expected output is produced.

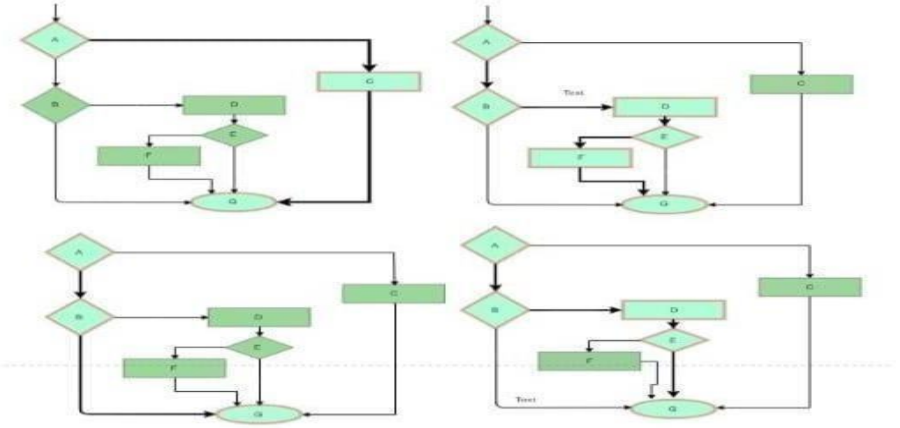
Example: Testing the "search For Job()" functionality to ensure that it returns the expected results based on different search criteria.



		1	2	3	4
CAUSES	C1	1	0	0	0
	C2	0	1	0	0
	C3	0	0	1	1
	C4	1	0	0	0
	C5	0	1	1	0
	C6	0	0	0	1
EFFECTS	E1	X	-	-	-
	E2	-	X	-	-
	E3	-	-	X	-
	E4	-	-	-	X

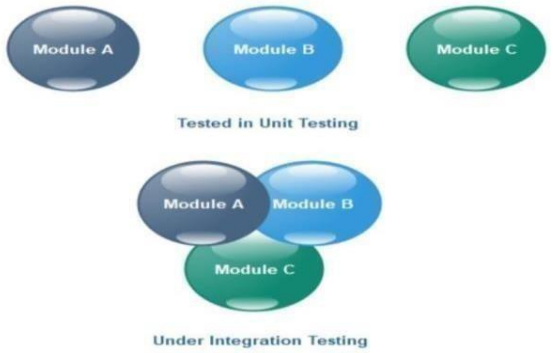
White Box Testing:

- White box testing involves testing the internal logic and structure of the software code.
- Testers would examine the code of individual modules to ensure that all code paths are tested.
- Example: Testing the "perform QC()" function to ensure that it adequately checks th equality of the processed data.



Integration Testing:

- Integration testing verifies that different modules of the software work together as expected.
- Testers would test the interaction between different modules, such as client management, project management, and employee management.



System Testing:

- System testing involves testing the entire system as a whole to verify that it meets the specified requirements.
- Testers would perform end-to-end testing of the entire system, including all modules andtheir interactions a client successfully.

Result:

Thus the software system for all scenarios identified in the use case diagram

was tested successfully.

EXNO: DATE:	**IMPROVE THE REUSABILITY AND MAINTAINABILITY OFTHE RECRUITMENT SYSTEMBY THE IMPLEMENTING SUITABLE DESIGN PATTERNS**

Aim:

To demonstrate the reusability and maintainability of the recruitment software system by implementing suitable design patterns.

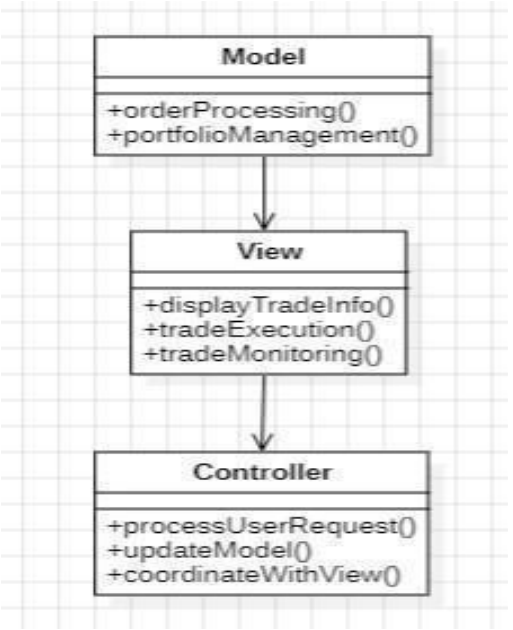
Recruitment Management System:

A Recruitment Management System is a software application tailored to streamline and optimize the recruitment processes within an organization or recruitment agency. It facilitates the organization, monitoring, and management of candidate sourcing, screening, interviewing, and hiring. The system typically includes functionalities for job posting, applicant tracking, candidate evaluation, communication with clients, and reporting.

To enhance the reusability and maintainability of the Recruitment Management System, we can incorporate several design patterns. Here are some design patterns that can be applied:

Model-View-Controller (MVC):

The MVC pattern separates the representation of information from the user's interaction with it. In the Recruitment Model , it divides the system into three interconnected components: Model(data and business logic), View (user interface), and Controller (handles user input and updates the model and view accordingly).

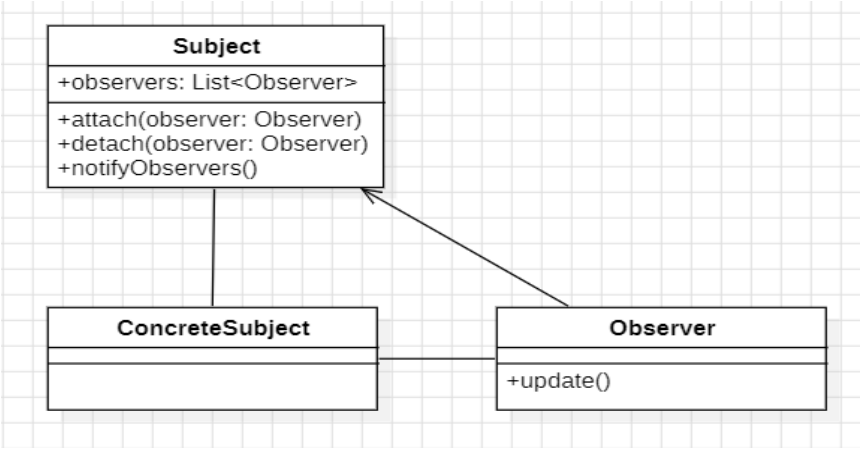


Problem: Tight coupling of user interface, business logic, and data access makes the recruitment System to maintain the scale.

Solution: Use MVC to separate concerns into Model (data and logic), View (UI), and Controller (use user's handling), improving modularity and ease of maintenance.

Observer Pattern:

The Observer pattern establishes a one-to-many dependency between objects, where multiple observers are notified of changes in a subject. In the Recruitment System, it is used to notify clients or users about the status of their projects, such as when a project is completed, when data is uploaded, when payments are processed, etc.

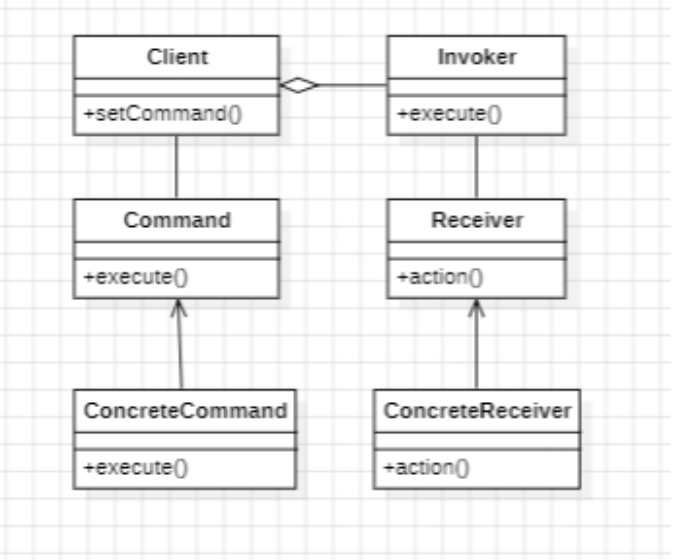


Problem: Keeping multiple stakeholders consistently informed of recruitment events is error-prone without a systematic approach.

Solution: Implement the Observer pattern to notify all registered observers (e.g., candidates, recruiters) of relevant changes automatically.

Command Pattern:

The Command pattern encapsulates requests as objects, allowing for parameterization of clients with queued or logged requests. In the Recruitment System, it is used to implement undo/redo functionality, batch processing, or asynchronous operations.

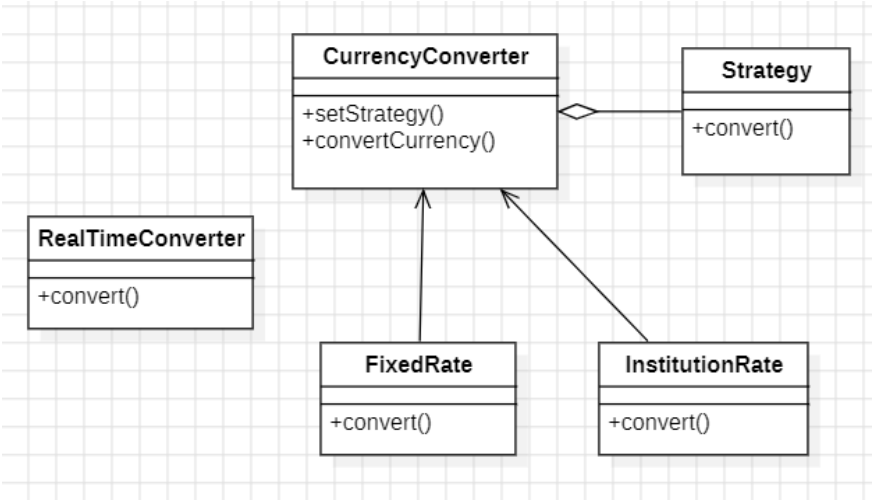


Problem: Handling user actions like undo/redo, batch processing, or asynchronous tasks is Complex with tightly coupled code.

Solution: Use the Command pattern to encapsulate requests as objects, enabling execution queuing, logging of actions.

Strategy Pattern:

The Strategy pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable. In the Recruitment System, it is used to implement different task prioritization algorithms or task allocation strategies based on workload.

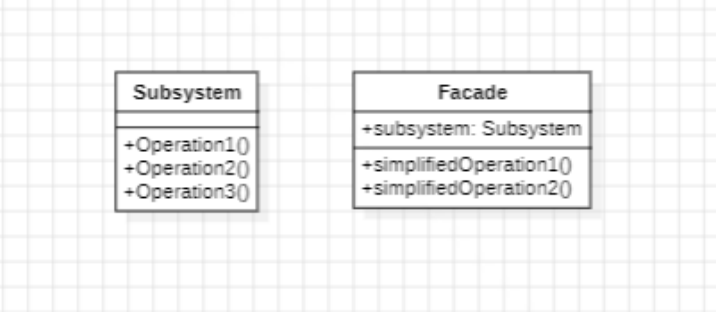


Problem: Hardcoding task prioritization and allocation algorithms makes the system rigid .

Solution: Apply the Strategy pattern to encapsulate different algorithms, allowing the system to switch strategies dynamically based on context.

Façade Pattern:

The Facade pattern provides a simplified interface to a complex subsystem, hiding its complexities from clients. In the Recruitment System, it is used to abstract away the complexities of task routing, assignment, and tracking.



Problem: Exposing clients to the complex interfaces of multiple subsystems can overwhelm them and complicate usage.

Solution: Implement a Facade to provide a simplified interface to the complex subsystems, system easier for clients to interact with reusability and maintain ability of the software .

Result:

To demonstrate the reusability and maintainability of the recruitment software system by implementing suitable design patterns.