



COMSATS University
Islamabad
Abbottabad Campus

FROM:
TAYYAB ALAM
FA22-BSE-043
PROJECT:
FOR
SOFTWARE DESIGN &
ARCHITECTURE
SECTION:
BSE: 5A
TO:
SIR MUKHTIAR ZAMIN

TOPIC:

JOB PORTAL

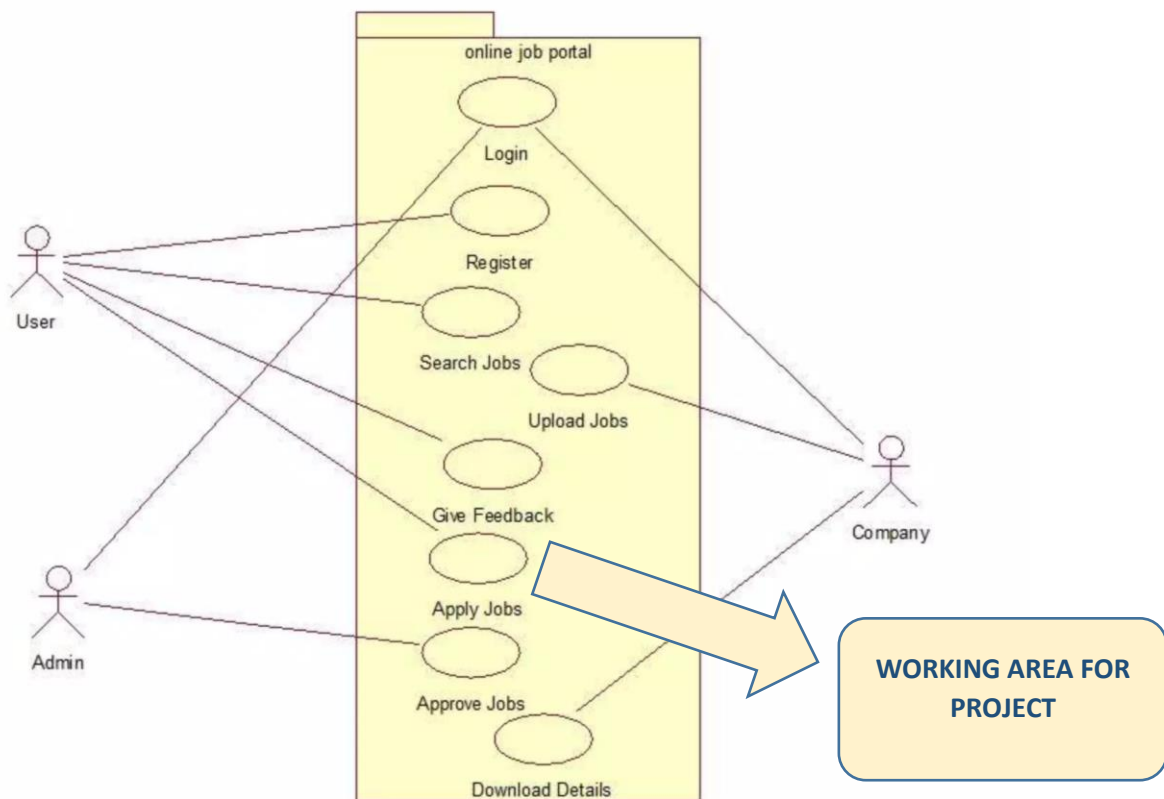
Introduction

- This web based portal basically concerned with different job services provided by different companies.
- It is also concerned with details of jobseekers.
- Jobseeker can view the list of different jobs and can apply for jobs.
- Then the company going to select the required job seekers for their qualification and update the database.
- The portal going to follow different companies policy.
- Their be an admin to manage all things.

Scope of Project

- The online job portal that is developed provide jobseekers with different jobs information like:
 - online applying for jobs
 - search for jobs
- Supported by well-designed database and web-based design.
- A friendly user interface is provided to facilitate different services.
- The basic scope contains:
 - Jobseeker's Area
 - Company's Area
 - Administrator's Area

USECASE:



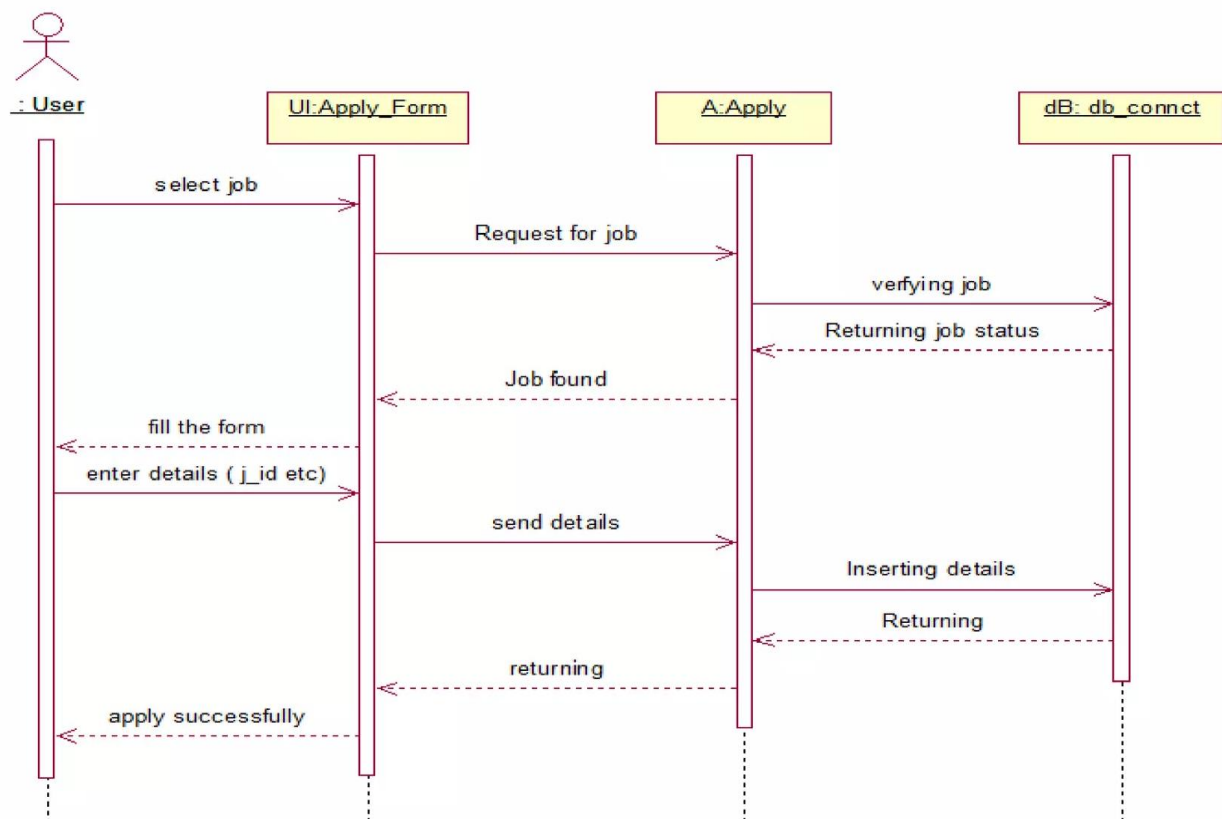
TEXTUAL USECASE

Apply Jobs

Use Case Name:	Apply jobs
Type:	Primary
Actors:	User
Purpose:	To apply for a particular job.
Pre-Condition:	A user going to login to account search for a job and fill the application form and submit the button.
Flow of Events	
Actor Action	System Resposnse
1) The user login to his account.	2) The system checks the account validation and returns with the result of login.
3) The user then clicks the search job option and search for particular job to apply for.	4) The system tells and evaluate search criteria enter by user and return the result with appropriate search "If found".
	5) The system displays the search result.
6) The user then go to application form tab and fill out details and click "submit" button.	7) The system saves the record and displays the message "Application submitted successfully".
8) Then user waits for company to response for his/her application.	
Post-Condition:	The user submit the form and receive a message "Application submitted successfully".

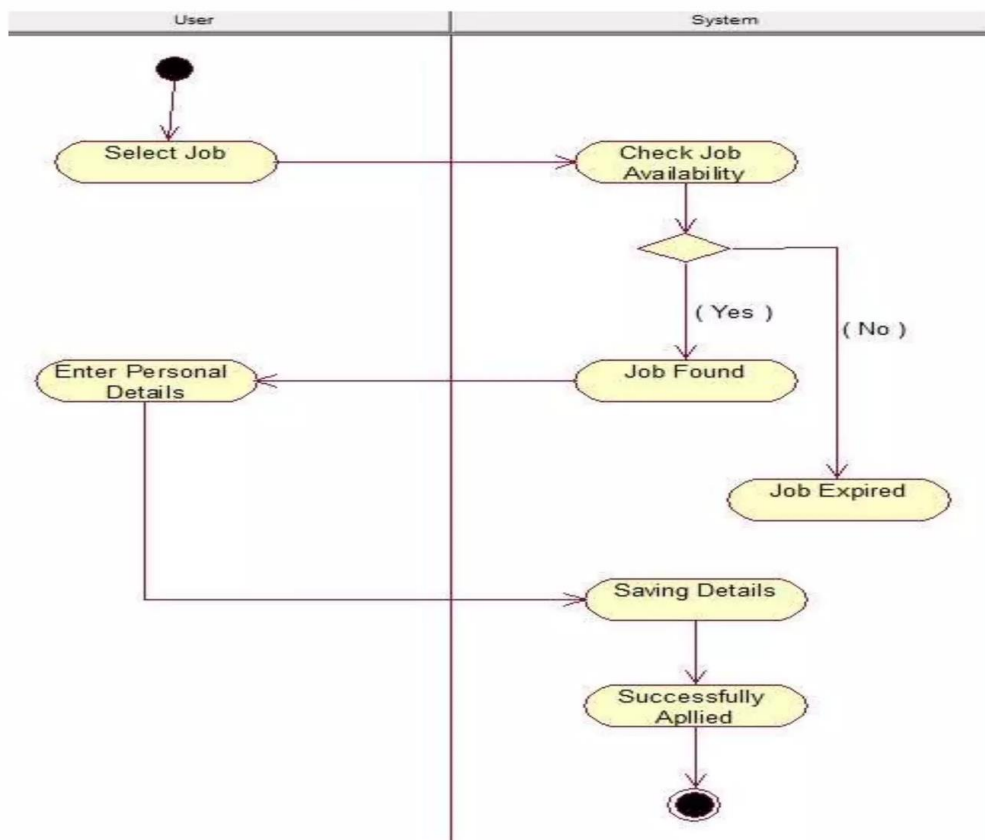
SEQUENCE DIAGRAM

Apply Jobs

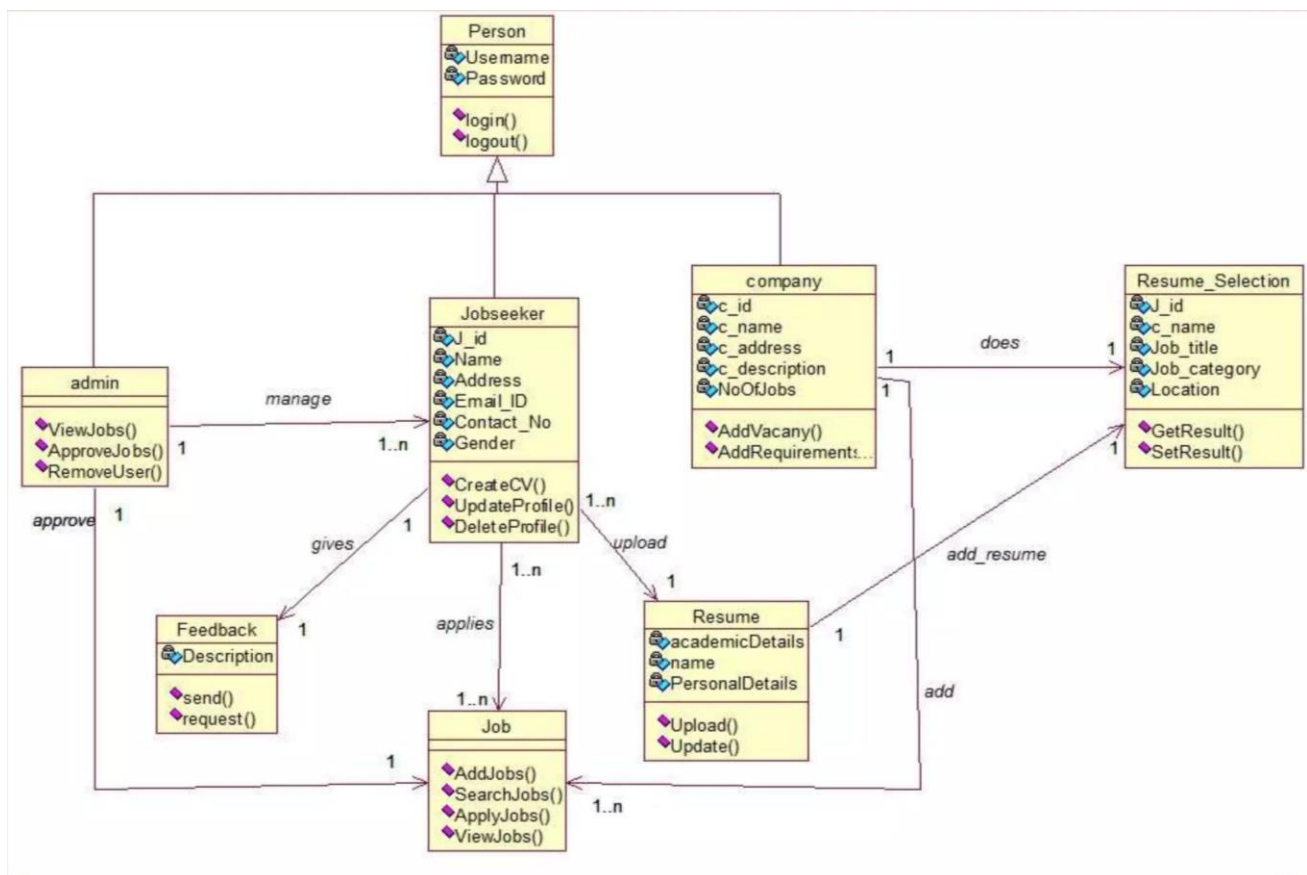


ACTIVITY DIAGRAM

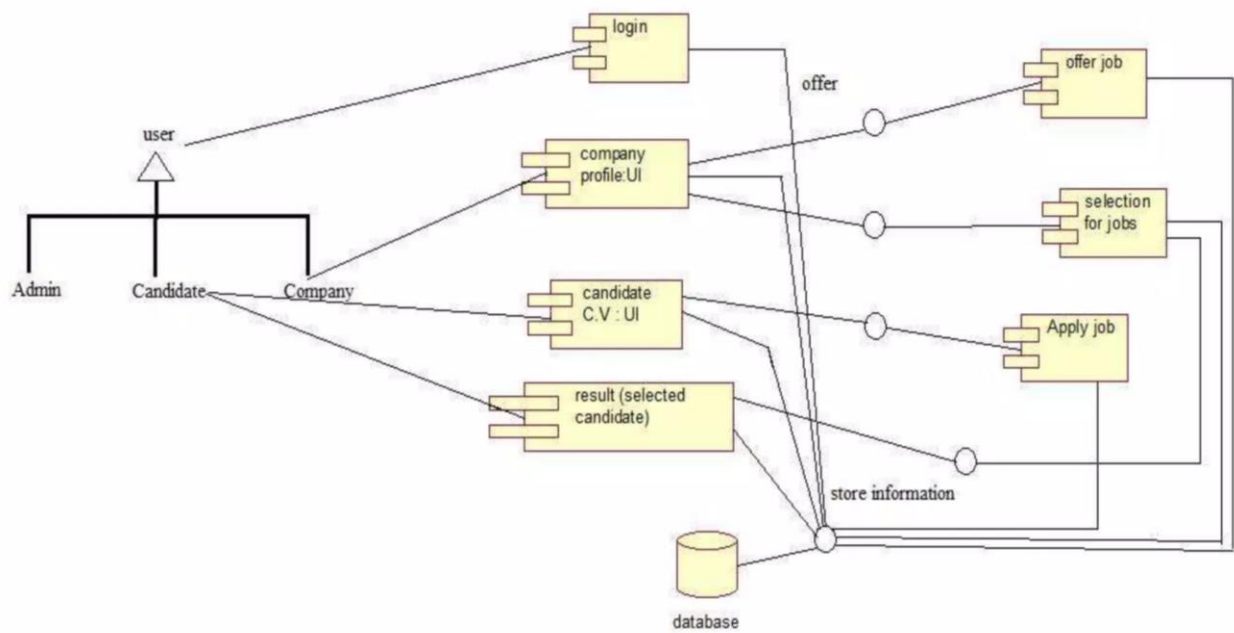
Apply Jobs



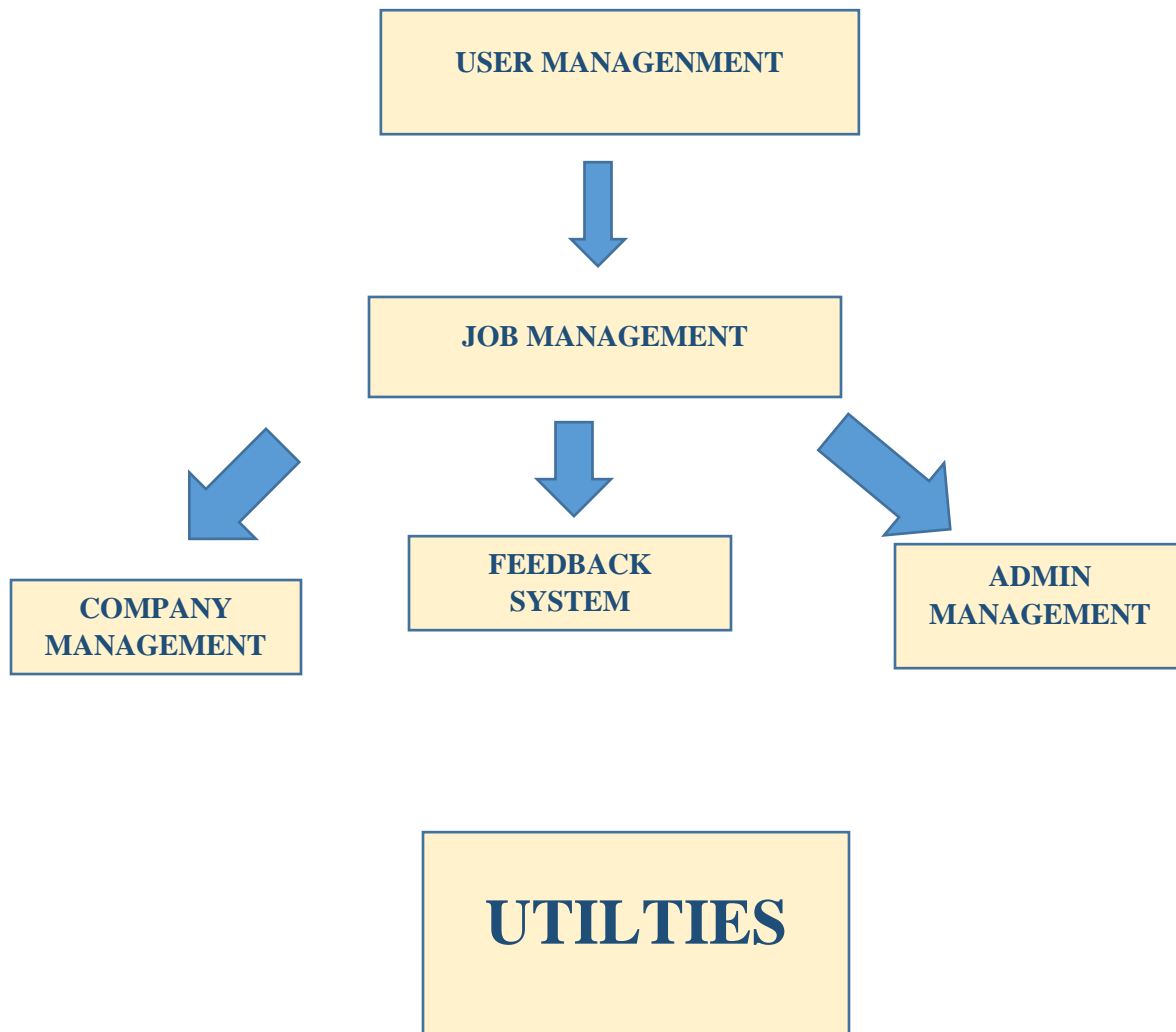
CLASS DIAGRAM



COMPONENT DIAGRAM



PACKAGE DIAGRAM



GRASP Principles

1. Information Expert:

- The Job, JobApplication, and JobPortalAppSimplified/JobPortalFrame classes follow the Information Expert principle. Each class is responsible for handling data it is most knowledgeable about:
 - Job manages job details such as jobId, jobTitle, and companyName.
 - JobApplication holds the application details, including userName, job, and resumeText.
 - JobPortalAppSimplified or JobPortalFrame interacts with both classes to control the application flow, as it has overall knowledge of available jobs and applications.

2. Controller:

- The JobPortalAppSimplified class in the console version or JobPortalFrame in the GUI version acts as the controller. It coordinates the application flow by displaying jobs, collecting user inputs, and managing job applications. It responds to user interactions by invoking relevant methods, like applyForJob and displayJobs.

3. Creator:

- The JobPortalAppSimplified or JobPortalFrame class creates instances of JobApplication when a user applies for a job, making it the "creator" of JobApplication objects. This follows the Creator principle, where a class that aggregates or closely uses another class is responsible for creating instances of that class.

4. Low Coupling:

- By using separate classes for Job, JobApplication, and JobPortalFrame, you maintain low coupling. Each class focuses on a specific responsibility, minimizing dependencies. For example, JobPortalFrame doesn't need to know internal details of Job or JobApplication; it just interacts with them through defined methods.

5. High Cohesion:

- Each class has a clear, cohesive purpose: Job encapsulates job details, JobApplication manages application details, and JobPortalAppSimplified/JobPortalFrame handles the main job application logic. This separation of concerns promotes high cohesion.

Layered Architecture

1. **Presentation Layer:**

- In JobPortalFrame, the GUI elements and methods for displaying information (e.g., updateJobList, labels, and input fields) make up the presentation layer. This layer is responsible for interacting with the user.

2. **Application Logic Layer:**

- The methods in JobPortalAppSimplified and JobPortalFrame (like applyForJob and submit Application) handle the main application logic, including processing applications and validating user input. This layer coordinates between the presentation and data layers.

3. **Data Layer:**

- The Job and JobApplication classes represent the data layer, as they encapsulate the data for each job and job application, respectively. These classes only manage the data and provide methods for retrieving or displaying that data.

THE END