**APPLICANT TRACKING SYSTEM**

***by***

**Pravin Bhandari**

**Category**

***Part 1: Planning*** ............................................................................................................................. 3

1.1 Introduction about the problem domain..................................................................................... 3

1.2 System Request.......................................................................................................................... 4

1.3 Feasibility analysis.................................................................................................................... 5

1.4 Requirements Definition............................................................................................................ 7

***Part 2: Analysis*** .............................................................................................................................. 8

2.1 Activity Diagram....................................................................................................................... 8

2.2 Use Case Diagram...................................................................................................................... 9

2.3 Use Case Description............................................................................................................... 10

2.4 Initial Class Diagram............................................................................................................... 16

2.5 Sequence Diagram................................................................................................................... 17

**Part 3: Design** .............................................................................................................................. 20

3.1 Final Class Diagram................................................................................................................. 20

3.2 Package Diagram..................................................................................................................... 21

3.3 Database Design...................................................................................................................... 22

3.4 Data Access and Manipulation Design .................................................................................... 23

### ***PART 1 – PLANNING***

### ***Part 1.1: Introduction about the problem domain***

A company’s hiring process plays a major role in its future success. Traditional hiring process still seems to be time consuming, messy, and frustrating for both the company and the applicants. HR team have been struggling to keep up with job posts, applicants, resumes, interviews, and boarding process.

The traditional hiring process requires Human Resource groups or recruiters to manually type and process the hiring data. The Applicant Tracking System (ATS) helps to parse the resumes of applicants and breakdown the data, integrate job postings with different job boards, with better user interface and visualization of the hiring process.

### ***Part 1.2: System Request***

***Project Sponsor:*** Company Sponsored

***Business Need:*** The system is designed for the ease of the HR Department of the company by saving time and complexity during Hiring Process.

***1.2.1 Business Requirement:***

There will primarily be three users – Recruiters, Account Manager, and Administrators

* The system should be able to provide User Interface that is interactive and helpful to understand the overall hiring process.
* System should be integrated with different job boards and allow the user to post job position at once to all the job boards
* System should be able to gather applicant’s profile by parsing the submitted resume and show the best candidate to recruiters by matching the keywords from job description and the resume.
* The HR department should have access to all the applicants for each job positions. There will be multiple users (recruiters, account managers) and one administrator. Each user should have their own profile and login credentials. Users should be allowed to make changes of the hiring process (accepting applicants, interview rounds, and boarding).

***1.2.2 Business Value***

1. ***Efficiency Value:***  
   Building the applicant tracking system helps the company to make the hiring process easier. The system automates the process of hiring applicants and tracking their status making the process time and cost efficient.
2. ***Commercial value:***

The company or department will save time and money in long run by implementing the applicant tracking system. We can cut operation costs of manual hiring of applications.

1. ***Market value:***

The company or department in which the applicant tracking application is developed will add more features to HR hiring process. The ATS will help to make more placements and play a key role in improving competitiveness.

***Part 1.3: Feasibility Analysis***

***1.3.1 Technical Feasibility:***

The Applicant Tracking System is feasible technically, although there is some risk.

1. ***Project Risk Delivery:*** Medium

* The basic process of the applicant tracking system is clear and concise. Most are familiar with the general front end of an application tracking system.
* The unclear part is the database/backend that controls and supports all the data that applicants enter or system parses, but we can get information from HR to tackle this risk.

1. ***Project Risk Programming:*** Medium

* There are 4 people in the project, 2 people have programming and web application development experience with front end development.
* One has decent experience in database and backend development.
* One has primarily HR experience and have used other applicant tracking systems and will showcase what other systems don’t do so well, that way we can make substantial benefits.

1. ***Project Size:*** Low

* The team will include 4 people.
* The project will last 2 months and will start rolling out partially in 6 weeks, roughly a month and a half.

1. ***Project Compatibility:*** Great

* The ATS will use latest web base code that can work on any browser.
* The ATS will need to integrate to current database and few web-based applications.
* The ATS will connect seamlessly to a database storing the applicant’s information.

***1.3.2 Economic Feasibility***

This project will help to reduce the cost of hiring process by cancelling various recruiting platform subscriptions. The company will be able to make more placements with the modern hiring process resulting in gains.

Here’s why:

* For this project our full-time tech experts will develop the system, labor covered.
* The front-end will be built into our current CRM system and published to our current website; those costs are already covered.
* The back-end database will be bridged to the front end with the same CRM system that we already cover.
* The company will see a significant increase in the quality of interviewees saving time and money, it will speed up the tracking process, it will increase revenue compared to the old system that was very non user friendly to both types of users.

***1.3.3 Organizational Feasibility***

* Overall Risk from an organizational perspective, the Applicant Tracking System (ATS) has a low risk.
* The recruiters of the company are well-familiar with the ATS.
* The HR will appreciate the new system in the company because of the ease of use, accessibility of candidate’s data including the applicant’s status and hiring progress.

### ***Part 1.4: Requirements Definition***

***1.4.1 Non-Functional Requirements***

### ***Operational Requirements:***

### The system will provide option to add and edit job positions, scan applicant’s profile

* The system will search and show vacancies and candidates
* System will be Web based Application
* System will back up automatically

### ***Performance Requirements:***

* System will store all applicant’s profile and resume to the database
* System will retrieve the most suitable candidate for a job position from database
* System will be able to retrieve job interviews and notify candidate and recruiters

### ***Security Requirements:***

* Log in / User Roles: Provide login platform and manage User Roles for the system with specific access matching the user’s role. For example: Recruiter won’t have a delete access, but the Account Manager would.

### ***Cultural and Political Requirements:***

### None

***1.4.2 Functional Requirements***

* ***Search:*** Search option using keywords, Name, or job titles. For Example, Recruiter might search using candidate Name, or specific title like – Java Developer, or keywords like JavaScript, Azure.
* **Changes:** System will provide option to add, update, and delete job postings, candidate’s profile and availability, or interview schedules.
* ***Integration:*** Integration with email and database for direct profile upload or email reminders to candidate using calendar.
* ***Profile match:*** Determine and pull the most suitable applicant for the specific job position from Database using keywords used in job description and resume or profile matching.
* ***Reports:*** Generate the report on specific job position as well as recruiter’s performance.

***2.1 Activity Diagram***

The system has 3 users: HR (Human Resource), Interviewer, and Candidate. The process for each user is different. The diagram below illustrates the process flow in detail.

**A screenshot of a cell phone

Description automatically generated**

***2.2 Use Case Diagram***

There are six primary Use cases for the system which contains the majority of it’s functions.

* Position Management
* Candidates Management
* Job Application Management
* Interview Management
* Position Overview
* Status Overview

A close up of a map

Description automatically generated

***2.3 Use Case Description***

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Position Management | | ID: 1 | Importance Level: High |
| Primary Actor: Human Resource | Use Case Type: Detail, Essential | | |
| Stake Holders and Interests: Human Resource: wants to add, update, and modify current job openings of the company | | | |
| Brief Description: This use case allows the HR Department users (Recruiters) to add and update the current openings in the company. | | | |
| Trigger: User can – “add’, “modify”, “delete”  Type: External | | | |
| Relationships:  Association: Human Resource Include: Position Overview Extend: Generalization: | | | |
| Normal Flow of Events:   1. The user logins to the system’s account 2. User clicks “Manage Positions” 3. User clicks “Add”, “Modify”, “Delete” options according to the requirements. 4. User access the job positions and makes edits as per the requirements. | | | |
| Sub flows:   1. User can choose to go to candidate management or Job Applications Management | | | |
| Alternate/Exceptional Flows:   1. HR inputs wrong Position ID 2. System shows error message 3. The user is asked to input the ID again. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Candidates Management | ID: 2 | | Importance Level: High |
| Primary Actor: Human Resource, Interviewer | | Use Case Type: Detail, Essential | |
| Stake Holders and Interests:   1. Human Resource: Wants to view Candidates that fit the job requirements 2. Interviewer: Wants to know details about the candidate for possible interview | | | |
| Brief Description: This Use Case will allow the HR department and interviewer to check candidates available for the position and their detailed profile | | | |
| Trigger: User can “View”, “Edit”, and “Delete” candidates.  Type: External | | | |
| Relationships:  Association: Human Resource, Interviewer Include: Extend: Generalization: | | | |
| Normal Flow of Events:   1. The user will click on “Candidates” button to see available candidate 2. User can “View”, “Edit”, and “Delete” profiles 3. User can “filter” the profiles based on Keywords | | | |
| Sub flows: | | | |
| Alternate/Exceptional Flows:   1. User inputs incorrect details 2. System shows an error message 3. The user is asked to input the details again | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Job Application Management | | ID: 3 | Importance Level: High |
| Primary Actor: Candidate, Human Resource | Use Case Type: Essential | | |
| Stake Holders and Interests:   1. Candidate: wants to apply the certain job position or to make changes in the applications 2. Human Resource: wants to make change in application and status | | | |
| Brief Description: The use case will allow the users to make changes in job application and allow Human resource to make changes in application status | | | |
| Trigger: View/Edit Application  Type: External | | | |
| Relationships:  Association: Human Resource, Candidate Include: Status Extend: Generalization: | | | |
| Normal Flow of Events:   1. Candidate Edits the job application using the job application Number and Pin code 2. Human resource checks the application of the candidate 3. Human resource changes the application status to “Under Review” or “Declined” | | | |
| Sub flows: | | | |
| Alternate/Exceptional Flows:   1. User enters wrong PIN or Application Number / Fields are left black 2. System shows and error 3. User is asked for input again | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Interview Management | | ID: 4 | Importance Level: High |
| Primary Actor: Interviewer | Use Case Type: Essential | | |
| Stake Holders and Interests:   1. Interviewer wants to schedule an interview the candidate. 2. Candidate wants to know Interview schedule and status | | | |
| Brief Description: The use case allows the Interviewer to setup interview and update the status of the scheduled interview for candidate. | | | |
| Trigger: Manage Interviews  Type: External | | | |
| Relationships:  Association: Candidate Include: Status overview Extend: Generalization: | | | |
| Normal Flow of Events:   1. Interviewer selects the candidate to interviewed. 2. Interviewer requests the interview time and make changes to the status to “Interview Pending” with time options for interview 3. Interviewer makes changes in “Status Overview” as “Accepted” or “Declined” after interview. | | | |
| Sub flows: | | | |
| Alternate/Exceptional Flows:   1. User inputs wrong data / Candidates deletes the job application / Interview Schedule conflicts 2. System shows an error. 3. Fields are cleared 4. Ask for details again | | | |

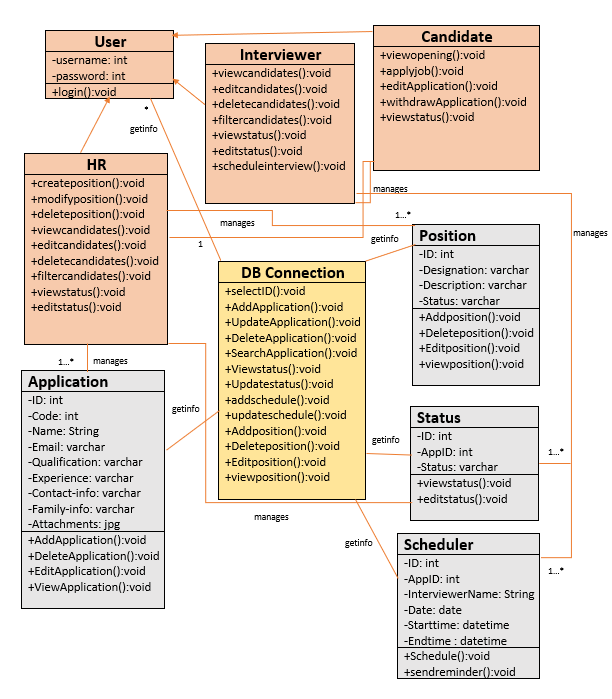
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Position Overview | | ID: 5 | Importance Level: Middle |
| Primary Actor: Candidate | Use Case Type: Essential | | |
| Stake Holders and Interests:   1. Candidate wants to check any open positions to apply for 2. Human Resource wants to open or close any positions | | | |
| Brief Description: The Use case will allow candidate to keep up with any updates on position applied. | | | |
| Trigger: “View Openings” button  Type: External | | | |
| Relationships:  Association: Human Resource, Candidate Include:  Extend:  Generalization: | | | |
| Normal Flow of Events:   1. Candidate clicks the “View Openings” button 2. Candidate will check the position is filled or not 3. Candidates can “Apply” for the position if not applied already 4. Candidate can “Withdraw” the application if necessary. | | | |
| Sub flows:   1. Candidate can go to “Status Overview” | | | |
| Alternate/Exceptional Flows:   1. Candidate inputs invalid Job ID 2. Ask user for valid ID number | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Status Overview | | ID: 6 | Importance Level: High |
| Primary Actor: Candidate | Use Case Type: Essential | | |
| Stake Holders and Interests:   1. Candidate wants to check the status of job application 2. Human Resource wants to check the status of a candidates’ job application | | | |
| Brief Description: The Use case allows the Candidate the status of the applied Job Application after Human Resource or Interviewer changes the status. | | | |
| Trigger: “View Application Status”  Type: External | | | |
| Relationships:  Association: Candidate, Human Resource, Interviewer Include:  Extend: Generalization: | | | |
| Normal Flow of Events:   1. Candidate Clicks “View Application Status” button 2. Candidate can check the status as “Under Review”, “Interview Pending”, “Accepted”, or “Declined” | | | |
| Sub flows: | | | |
| Alternate/Exceptional Flows:   1. Candidate withdraws the application 2. Show error message notifying Application was withdrawn 3. Redirect to main page. | | | |

***2.4 Initial Class Diagram***

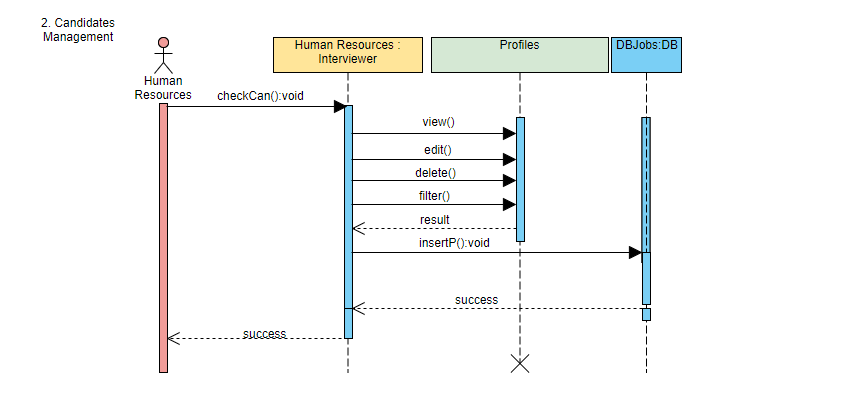
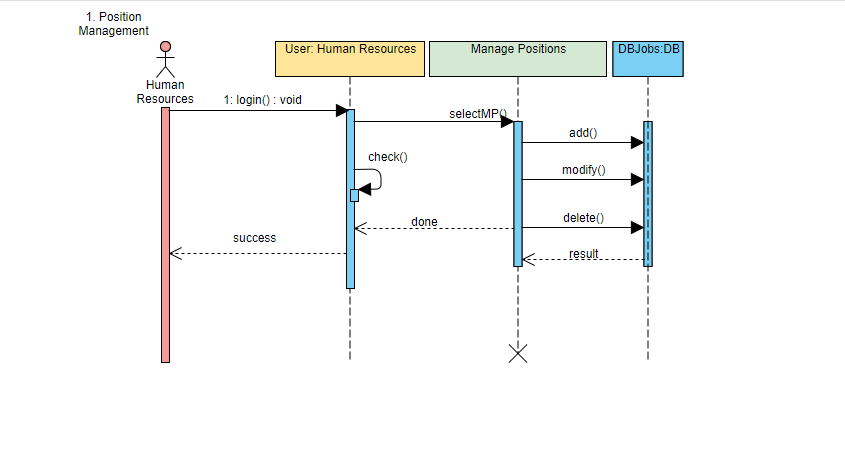
This is our initial Class Diagram. We have produced nine classes for our project: User, Interviewer, Candidate, HR, DB connection, Position, Application, Status and Scheduler. There is a generalization relationship between HR, Candidate, and Interviewer. The user class is a generalization of HR, Interviewer, and Candidate classes.

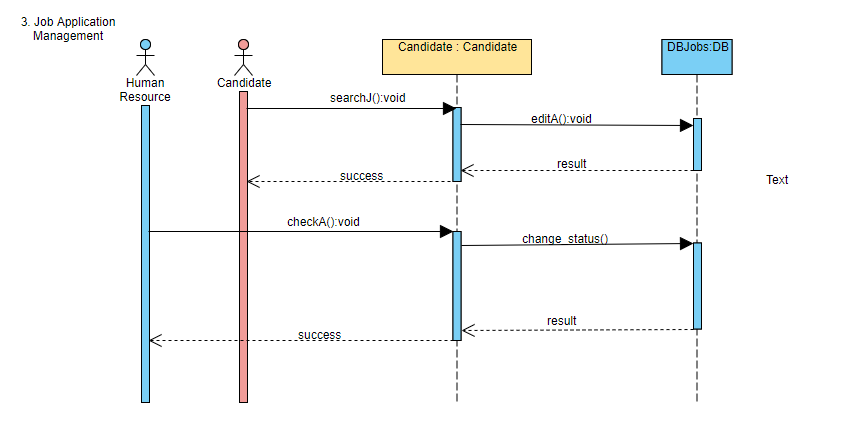
The HR class manages position class by using methods in the HR class such as create, modify and delete positions. The HR class manages Application class by keeping track of what current applications are submitted for each position. The interviewer class manages Status class through methods in the interview class. The interviewer also manages Scheduler class through creating a schedule or sending reminders to the interviewer. User, Application, Position, Status and Scheduler get into DB Connection class which glues everything in the database through methods in DB Connection class.

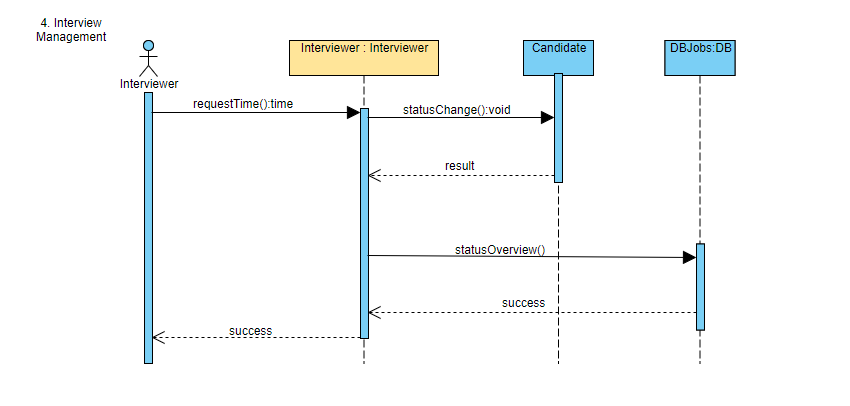
****

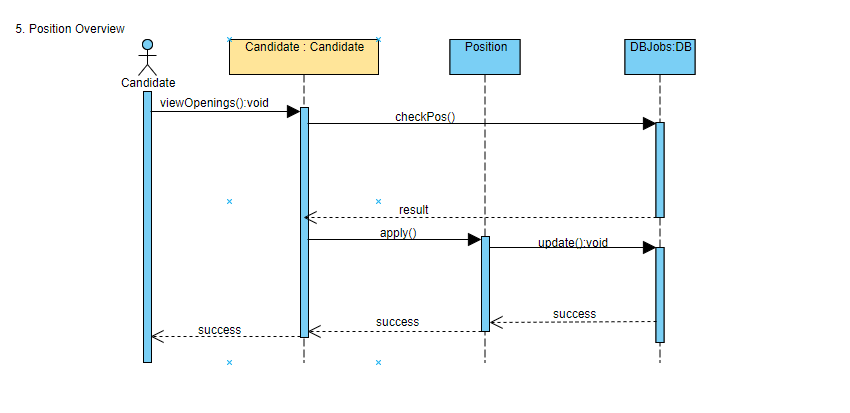
***2.5 Sequence Diagram***

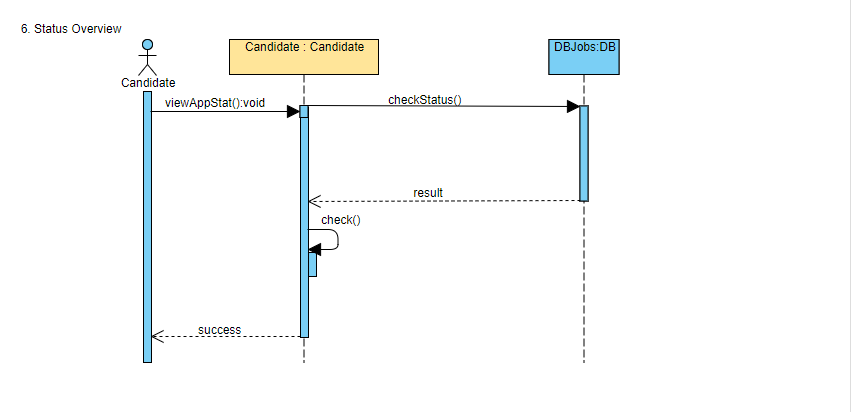
The sequence diagrams describe the use cases and the information that is communicated between them over the length of the time. Each sequence is corresponding to each use case.





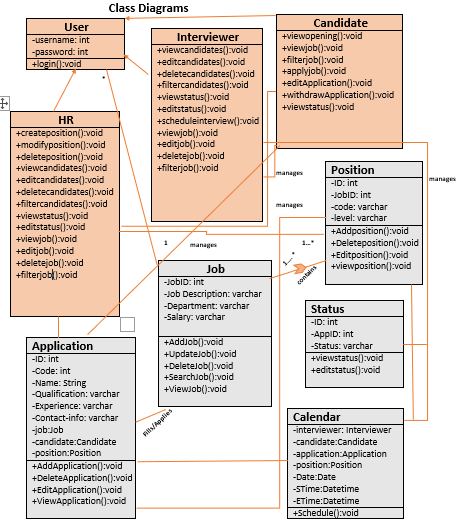






***3.1 Final Class Diagram***

The final class diagram is slightly different with the initial one. Firstly, the DBConnection class has been deleted in the diagram because we found that there is no need to put the DBConnection class in the class diagram. Secondly, two new classes are added in to the diagram: Job class and Calendar diagram. The Job class is used to contain the Addjob(), searchjob(), deletejob(), updatejob() methods and all the job details. The Calendar class is used to connect the interviewer class and Candidate class. It can be used to show which interviewee is responsible for a which candidate in the interview process.

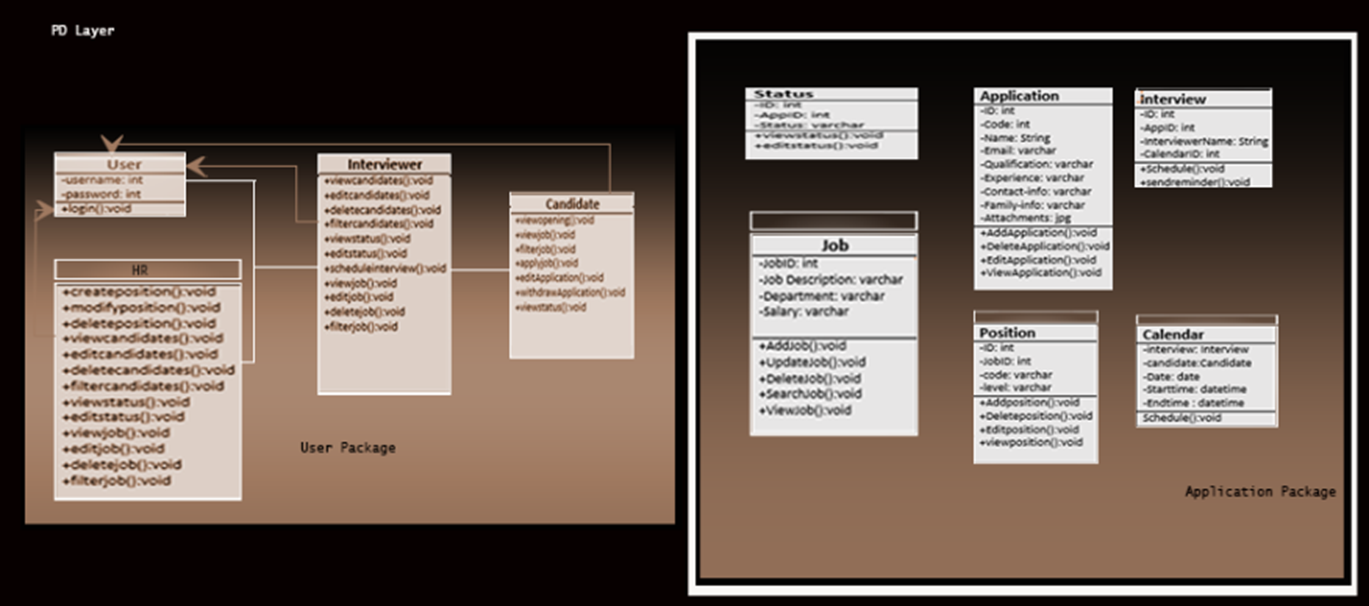


***3.2 Package Diagram***

The package diagram is show below.

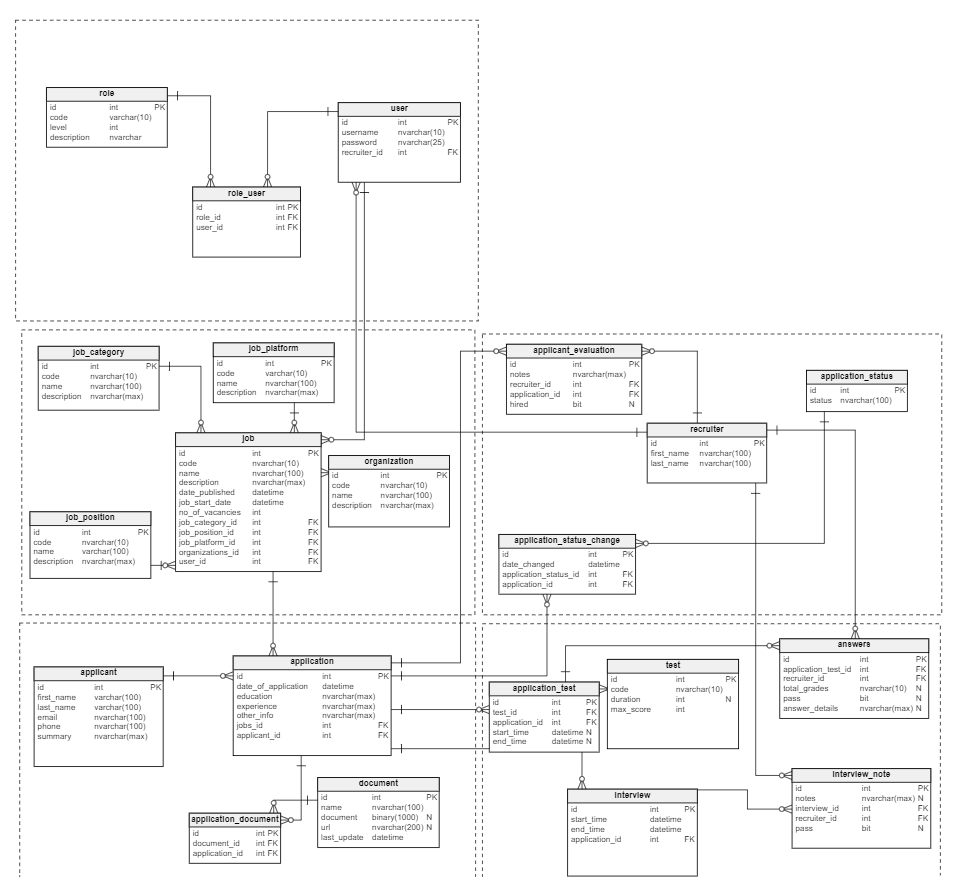
User class with HR class, Interviewer class, and Candidate is packaged together to be the Account Package.

The Application class, Job class, Calendar class, Position class, Status class, and Interview class is packaged together to be the Application Tracking Package.

****

***3.3 Database Design***

The database design for the system is presented below. There are multiple tables in the database. The user, role, and role\_user table contains each user's credentials and clearance. The tables – job, job\_category, job\_platform,job\_positions, and organization contain details about job position in detail. Tables: applicant, application, application\_document, and document contain applicant and application details. The other remaining part of database tables are for initial screening, interviews, notes, and application status.



***3.4 Data access and manipulation design***

The data access and manipulation design show the relationship between classes and databases.

A close up of text on a white surface

Description automatically generatedClasses use DAM class to connect to the database. The line pointed from database to class shows the corresponding relationship between classes and databases.