

## Contents

User requirements .....	2
Use Case Diagram.....	3
Use Case scenario: «Select suitable applicants» .....	4
Class Diagram – analytical.....	4
Class Diagram – design.....	5
Activity Diagram.....	6
State diagram.....	6
The GUI Design .....	7
The discussion of design decisions .....	9

# Job Board System

## User requirements

The system should provide Job Board experience. System should have different user Roles such as User, Employee, Recruiter, CompanyOwner. Users, Employees and Recruiters should be able apply to different jobs. Recruiters, Recruiters Owners then should be able to look through the list of applicants to the given job posting. They should be able leave some remarks on a certain candidate, accept or reject candidate. To be more precise in their choice, they can view details of the applicant, and even download their cv.

The Company Owner and Recruiter Owner may buy promotions for their company or to pay fee for job postings. System should have two different types of payments – by credit card, and by transfer.

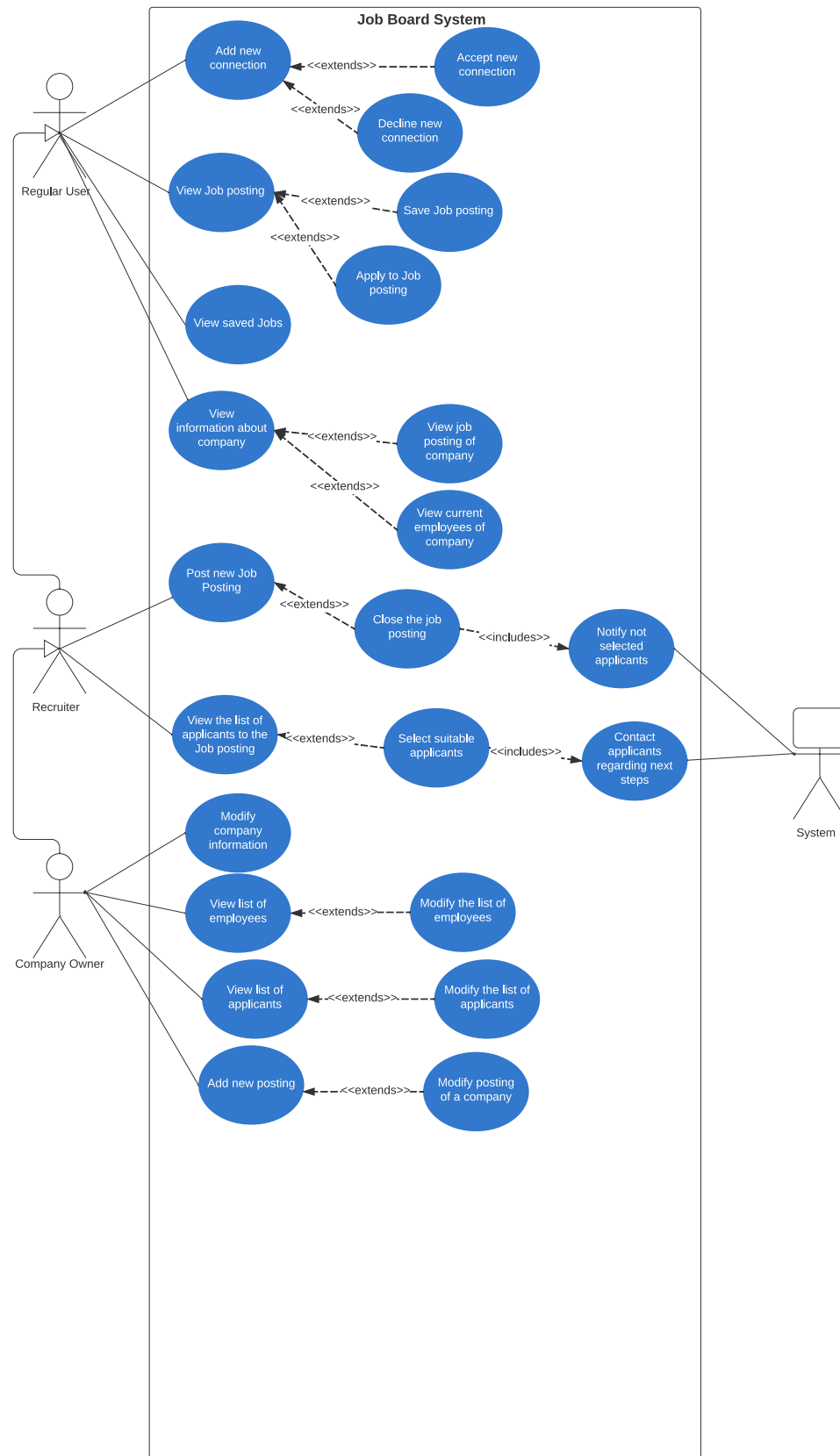
Users should be able to add new connections/reject the requests to join their network.

Users also should be able to look at Info of the company, save their jobs and then apply to them later.

When it comes to jobs postings from side of company, Recruiter roles users should be able to delete them. If for example the job posting was on the job board for longer than 60 days it gets automatically deleted.

Every job posting may have a list of required skills.

# Use Case Diagram

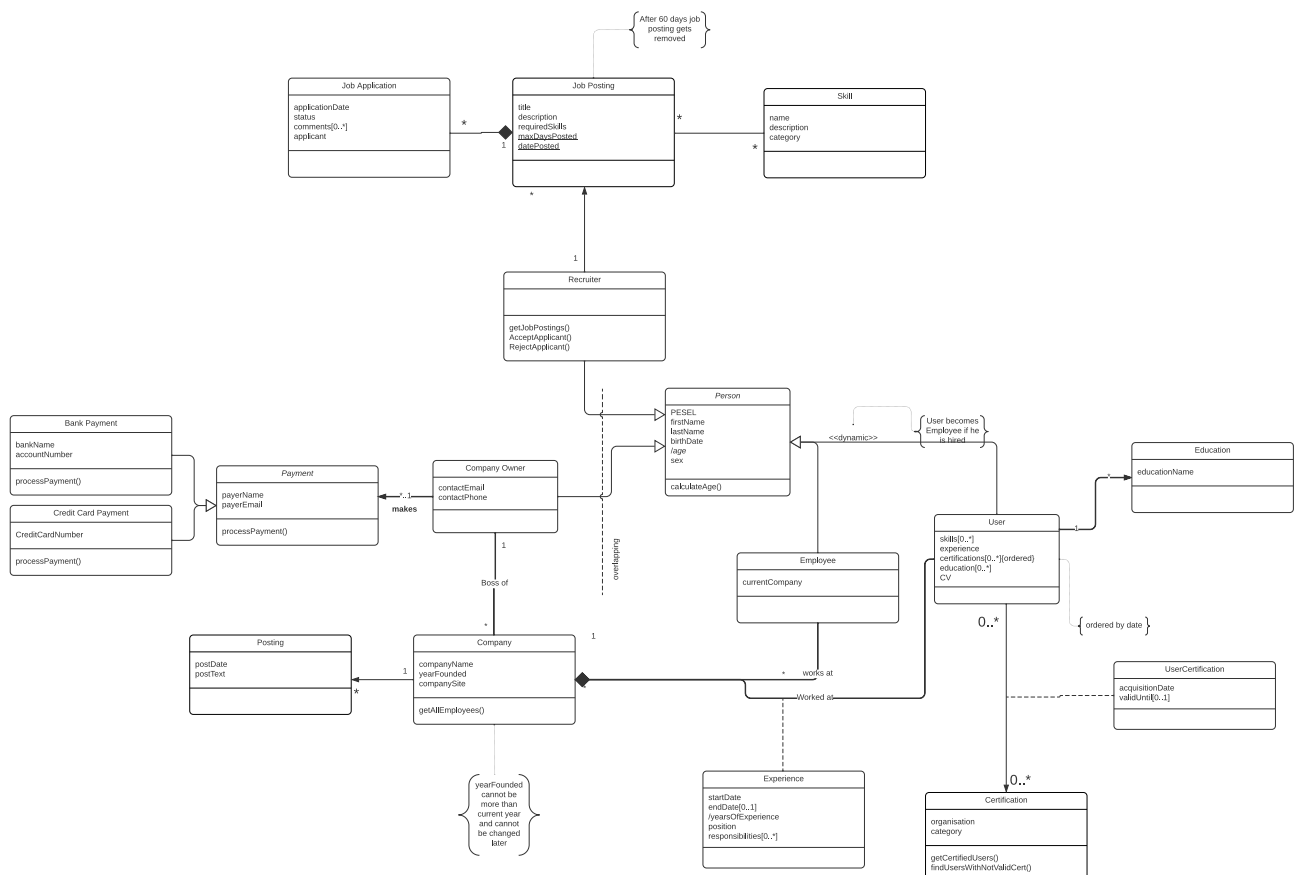


## Use Case scenario: «Select suitable applicants»

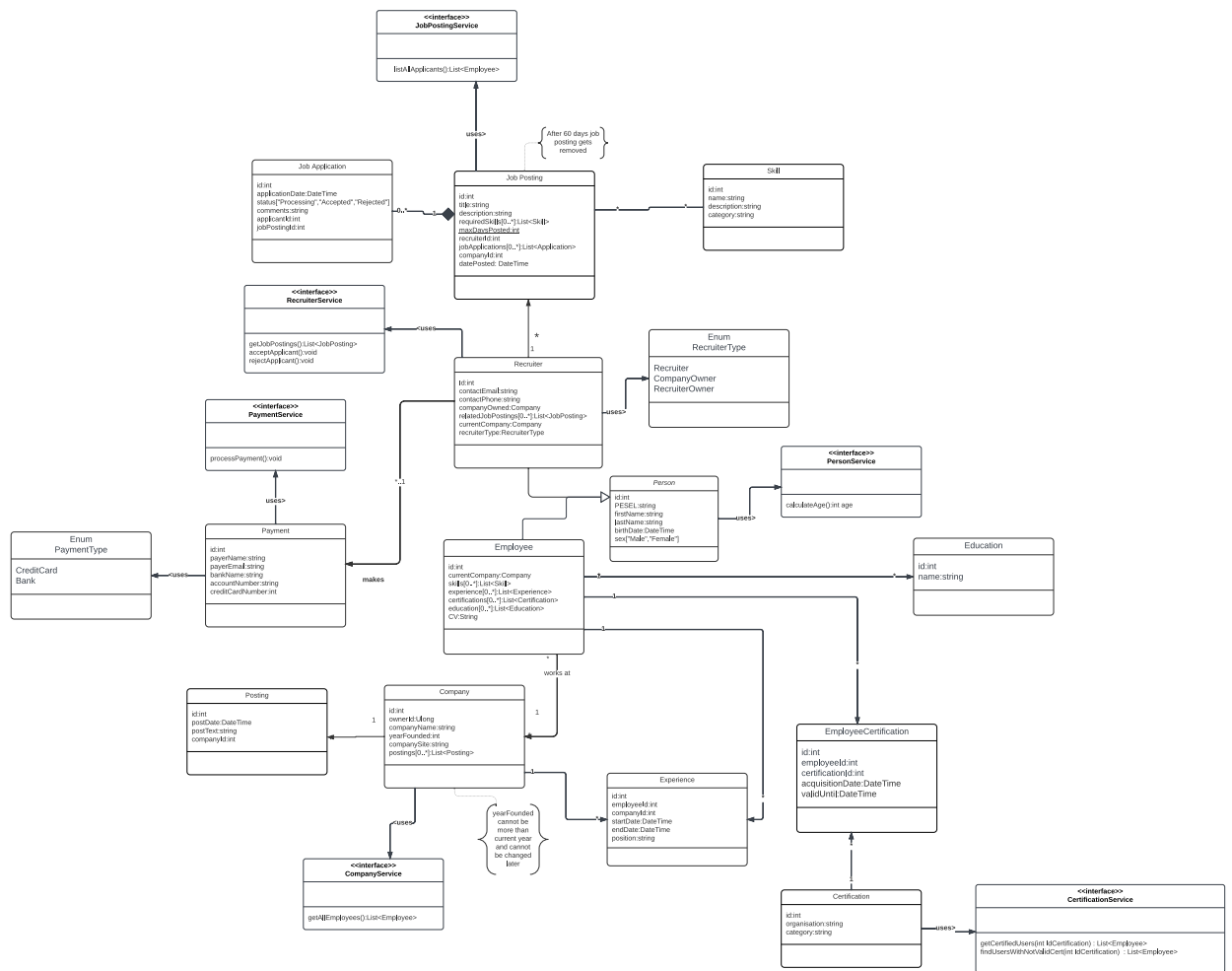
1. The process starts with a Recruiter logged into their accounts with corresponding permissions.
2. The list of all Job Postings related to the given recruiter is then listed and shown
3. After this user can pick the Job Posting of their company to check the applicants. To do so, user has to click into the job posting tile
4. User is able to see information about applicants such as first name, last name, years of experience, their CV.
5. User evaluates whether the applicant is suitable for the job posting
6. User accepts or rejects the applicant by pressing corresponding buttons
7. After this User can either get back to the page with job postings or keep picking applicants

Users are able to reject the application or accept it using corresponding buttons in form of Tick and Cross. In case if the application was rejected, the automatic notification by the system is sent to the applicant. Whereas, if the application is accepted, the notification about it is sent to the user with the instruction regarding next steps of his application.

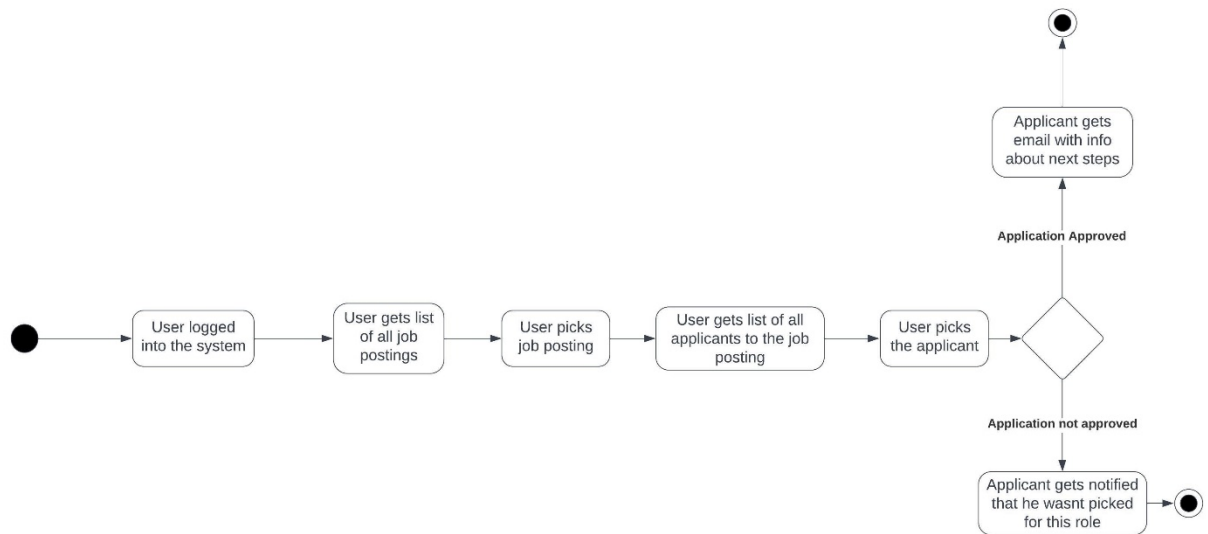
## Class Diagram – analytical



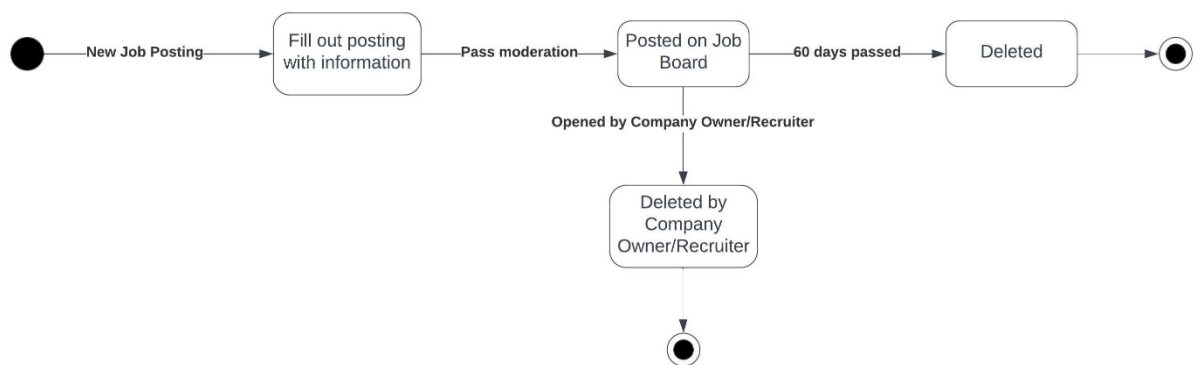
# Class Diagram – design



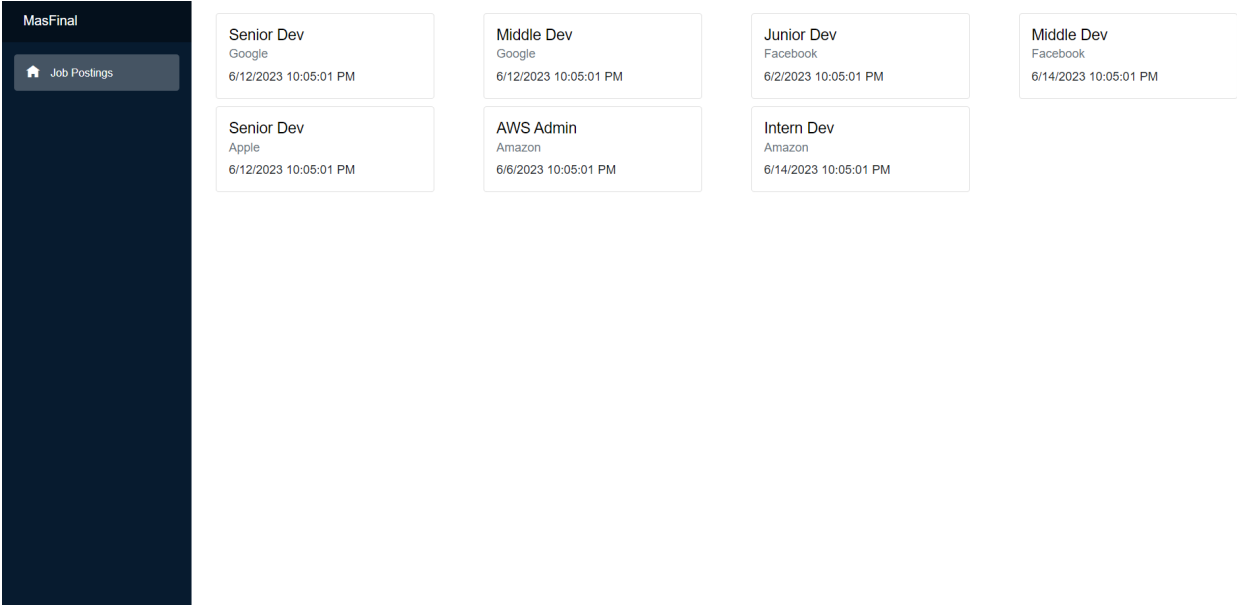
## Activity Diagram



## State diagram



## The GUI Design



Logged in user gets the list of the related Job Postings. He sees the title of the job posting, company that posted, and date posted. When User hovers over the tile with the job postings, the text gets underlined



After user picks job posting list of applicants shows up

First name	Last name	CV	Details	Status		
Steve	Jobs	<a href="#">View CV</a>	<a href="#">View Details</a>	Processing	<a href="#">Accept</a>	<a href="#">Reject</a>
Guy	Ritchie	<a href="#">View CV</a>	<a href="#">View Details</a>	Processing	<a href="#">Accept</a>	<a href="#">Reject</a>
Mad	Dimon	<a href="#">View CV</a>	<a href="#">View Details</a>	Processing	<a href="#">Accept</a>	<a href="#">Reject</a>
Hugh	Jackman	<a href="#">View CV</a>	<a href="#">View Details</a>	Processing	<a href="#">Accept</a>	<a href="#">Reject</a>

Here User can view the full name of applicants, can accept or reject the applicant, can download applicant's cv and view details

First name	Last name	CV	Details	Status
Steve	Jobs	<a href="#">View CV</a>	<a href="#">View Details</a>	Accepted
Guy	Ritchie	<a href="#">View CV</a>	<a href="#">View Details</a>	Rejected
Mad	Dimon	<a href="#">View CV</a>	<a href="#">View Details</a>	Accepted
Hugh	Jackman	<a href="#">View CV</a>	<a href="#">View Details</a>	Processing

Accept Reject

When clicked on View details text, the details of the applicant such as skills, education show up

First name	Last name	Status
Steve	Jobs	Accepted
Guy	Ritchie	Rejected
Mad	Dimon	Accepted
Hugh	Jackman	Processing

### Application details

**Applicant name:** Steve Jobs

**ASP.NET**

Skill description: Web Framework

Skill category: Programming

**C#**

Skill description: Programming Language

Skill category: Programming

**JavaScript**

Skill description: Same as Java

Skill category: Programming

**Education**

Polsko-Japonska Akademia Technik Komputerowych

Close

When clicking View CV, the CV of the applicant gets downloaded

First name	Last name	CV
Steve	Jobs	<a href="#">View CV</a>
Guy	Ritchie	<a href="#">View CV</a>
Mad	Dimon	<a href="#">View CV</a>
Hugh	Jackman	<a href="#">View CV</a>

resume (3).pdf - Adobe Acrobat Pro DC

File Edit View Window Help

Home Tools resume (3).pdf x

1 / 1 28.4% Share

**Bohdan Andoniiev**

Student in the last year of Bachelor's degree in Computer Science with the first experience in Software Development looking for opportunities in the .NET (C#) development. I am a highly motivated, fast learner, and collaborative team player who thrives on effective communication. I continuously enhance my technologies and continuously strive to enhance my knowledge and skills.

**Education**

**Polsko-Japonska Akademia Technik Komputerowych**

**B.S. Computer Science (B.S.)**

**Skills**

**Software Development**

- Developed functional modules for gathering and processing data for clients in the given sector for "GSM"
- Used the creation of a desktop module that utilizes loading, HTML, and JavaScript to collect and organize data into a structured report
- Assisted in organizing safe systems for clients
- Worked as a first and second line of support for clients
- Delivered training and knowledge transfer sessions to a team of client developers

**Software Development Intern**

- Worked under the guidance of my mentor
- Developed my latest programming skills
- Contributed relevant regarding the page gap in the most expensive Polish companies

**Tools**

- VS, .NET, Entity Framework, PostgreSQL, AWS, JavaScript, C#, SQL Server

**Certifications**

- AWS Certified Solutions Architect - Associate
- AWS Certified Cloud Practitioner
- SCS English Exam - Result (A - C)

**Other Skills**

- Python - C#
- PHP - JS
- JavaScript - HTML
- Russian - Native



## The discussion of design decisions

I decided to create a web application to implement the solution. I used Blazor Server SPA framework with server-side rendering.

The Database is generated using Code-First approach as it's more flexible and convenient way to work with database using code, rather than building the schema of database and then import it to code.

The interaction with database is realized using Entity Framework 7.0.

To have a better control over the Models FluentApi was used, as using it we get more freedom in terms of the behavior of the data.

We have abstract class and polymorphic method invocation in class Payment. Company Owner may purchase ad, pay fee for job postings etc. The polymorphic method is ProcessPayment(), as depending on the type of the Payment depends the process itself. We can have CreditCardPayment and BankTransferPayment, depending on the type we check the different data of our payer.

In program we have abstract class Person, from which can Inherit Employee, Recruiter, CompanyOwner. We have overlapping inheritance between classes Recruiter and CompanyOwner. It is implemented using flattening method. We pass the list of enum values, which defines what values will instance have.

We also have dynamic inheritance between classes User and Employee, as if the Person doesn't have current company e.i. doesn't work it becomes User, once the job is found, User becomes Employee. Our employee holds information about his education, his skills which we can later retrieve. Also we can download the cv of the employee if the employee or user applies to a job.

For derived attribute I have age in class Person. To implement it in my app I have PersonService which has method CalculateAge() which return the age of a given person. To interact with data from database Services were used such as PersonService, JobPostingService which implement base interfaces.