

Лабораторна №3-4

Тема: Веб-портал для пошуку роботи і підбору персоналу

1.Визначення основних класів і їх відповідальностей

User (логічний базовий клас) - базова сутність для користувачів системи (логічно від якої походять: Applicant, Employer, Admin). Відповідальність: управління аутентифікацією/профілем, загальні атрибути (пароль, роль).

Атрибути	Тип	Опис
user_id	int	унікальний ідентифікатор.
username	string	логін (опціонально).
passwordHash	string	хешований пароль.
role	string	роль (applicant, employer, admin).

Методи	Повертає	Опис
RegisterUser(login, password, role)	void	Реєстрація користувача та створення відповідного профілю (Applicant, Employer, Admin)
LogoutUser()	void	Вихід користувача (видалення сесії)
LoginUser(user)	void	Логін користувача (запис у сесію)

Applicant (кандидат) - Відповідальність: профіль кандидата, резюме, відправка заявок на вакансії, перегляд/оновлення резюме.

Атрибути	Тип	Опис
applicant_id	int	унікальний ідентифікатор.
firstName	string	ім`я користувача
lastName	string	прізвище користувача
phone	string	номер телефону користувача
user_id	int	ПК від user

Методи	Повертає	Опис
Register(user_id)	void	Створення профілю Applicant
EditInfo(name, surname, email, phone, applicant_id)	void	Редагування профілю

Employer (роботодавець) - Відповідальність: створення вакансій, управління вакансіями, перегляд заявок від кандидатів, керування профілем компанії.

Атрибути	Тип	Опис
employer_id	int	унікальний ідентифікатор.
companyName	string	назва компанії
companyDescription	string	інформація про компанію
phone	string	номер телефону компанії
user_id	int	РК від user

Методи	Повертає	Опис
Register(user_id)	void	Створення профілю Employer
EditInfo(name_company, email, phone, about_company, employer_id)	void	Редагування профілю

Admin (Адміністратор\Модератор) - Відповідальність: модерація контенту (вакансій, новин, користувачів), управління акаунтами, формування звітів.

Атрибути	Тип	Опис
admin_id	int	унікальний ідентифікатор.
username	string	нікнейм адміністратора
email	string	пошта адміністратора
user_id	int	РК від user

Методи	Повертає	Опис
Register(user_id)	void	Створення профілю Admin
EditInfo(username, email, admin_id)	void	Редагування профілю Admin

Vacancy (вакансія) Відповідальність: представлення вакансії - опис, вимоги, зарплата, статус публікації, зв'язок із роботодавцем.

Атрибути	Тип	Опис
vacancy_id	int	ID Вакансії
title	string	Назва вакансії
description	string	Опис вакансії
employment	string	Вид зайнятості
salary	float	Заробітна плата
location	string	Місто
employer_id	int	РК від employer

Методи	Повертає	Опис
AddVacancy(title, description, employment, salary, location, employer_id)	void	Створення вакансії
DeleteVacancy(vacancy_id)	void	Видалення вакансії
EditVacancy(vacancy_id, title, description, employment, salary, location, employer_id)	void	Редагування вакансії

Resume (резюме) Відповідальність: зберігання резюме кандидата (досвід, освіта, навички), версії резюме.

Атрибути	Тип	Опис
\$resume_id	int	унікальний ідентифікатор.
\$education	string	освіта
\$work_experience	string	досвід роботи
\$skills	string	навички
\$personal_qualities	string	особисті якості
\$about_me	string	про мене
\$applicant_id	int	РК від applicant

Методи	Повертає	Опис
AddResume(education,work_experience,skills,personal_qualities,about_me,applicant_id)	void	Додавання резюме
EditResume(education,work_experience,skills,personal_qualities,about_me,applicant_id)	void	Редагування резюме
DeleteResume(education,work_experience,skills,personal_qualities,about_me,applicant_id)	void	Видалення резюме

Applications (подане резюме на вакансію) Відповідальність: заявка кандидата на конкретну вакансію — статус (submitted, reviewed, rejected, accepted), супровідні матеріали.

Атрибути	Тип	Опис
applications_id	int	унікальний ідентифікатор.
date_create	DateTime	Дата створення відгуку
date_editing	DateTime	Дата зміни статусу
type_application	string	тип заяви
resume_id	int	РК від резюме
vacancy_id	int	РК від вакансії

applicant_id	int	РК від кандидата
--------------	-----	------------------

Методи	Повертає	Опис
AddApplications(applicant_id, resume_id, vacancy_id)	void	Додавання заявки
ApplicationsForApplicant(applicant_id)	array	Перегляд заявок Applicant
ApplicationsForVacancy(vacancy_id)	array	Перегляд заявок на вакансію

News (новини) Відповідальність: публікація новин/оголошень portalу.

Атрибути	Тип	Опис
news_id	int	унікальний ідентифікатор.
title	string	Заголовок новини
news	string	Текст новини
date_create	Date	Дата публікації новини
photourl	string	Посилання на фотографію
admin_id	int	РК від адміна

Методи	Повертає	Опис
AddNews(title, newsText, photourl, admin_id)	void	Додавання новини
DeleteNews(news_id)	void	Видалення новини
EditNews(title, newsText, photourl, admin_id)	void	Редагування новини

2. Зв'язки між класами

Зв'язки показують, як класи взаємодіють. Вони включають наслідування (стрілка з порожнім трикутником) та асоціації (лінія з мультиплікативністю, наприклад, "1" -- "0..*").

Наслідування (Generalization)

- User <|-- Applicant : Applicant наслідує від User (має всі атрибути/методи User + свої).
- User <|-- Employer : Employer наслідує від User (має всі атрибути/методи User + свої).
- User <|-- Admin : Admin наслідує від User (має всі атрибути/методи User + свої).

Асоціації (Association):

- Applicant "1" -- "0..*" Resume : has (один кандидат може мати багато резюме).
- Applicant "1" -- "0..*" Application : submits (один кандидат може подавати багато заявок).
- Employer "1" -- "0..*" Vacancy : posts (одна компанія - багато вакансій).
- Vacancy "1" -- "0..*" Application : receives (одна вакансія - багато заявок).
- Application "1" -- "1" Resume : uses (заявка може посилатись на одне резюме). (опціонально)
- News "0..*" -- "1" Admin : authoredBy (адмін може створювати новини).
- User (або Applicants/Employers) - взаємодіють із сервісами пошуку/фільтрації (послугами системи - поза моделлю класів).

Композиція (Composition):

- Видалення Employer призводить до логічного або каскадного видалення Vacancy (композиція залежності).
- Видалення Applicant - видалення Resume та Application (за політикою збереження даних).

Описані класи та зв'язки між ними продемонстровані на діаграмі:

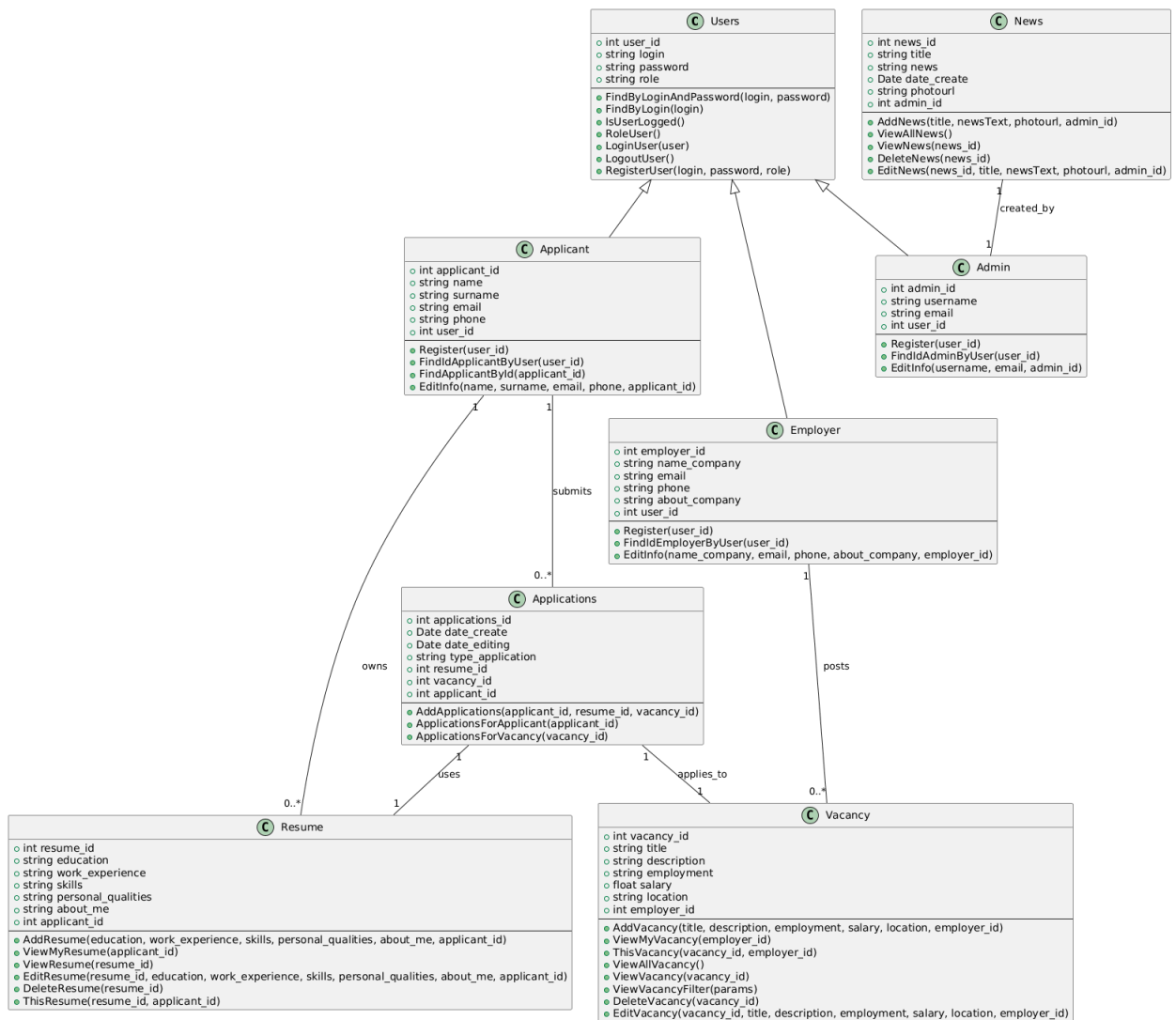


Рис. 3.1. Діаграма класів системи

Діаграма класів системи із використанням патернів GOF:

UsersManager "1" -- "*" Users : manages (один менеджер керує всіма користувачами системи; патерн Singleton)

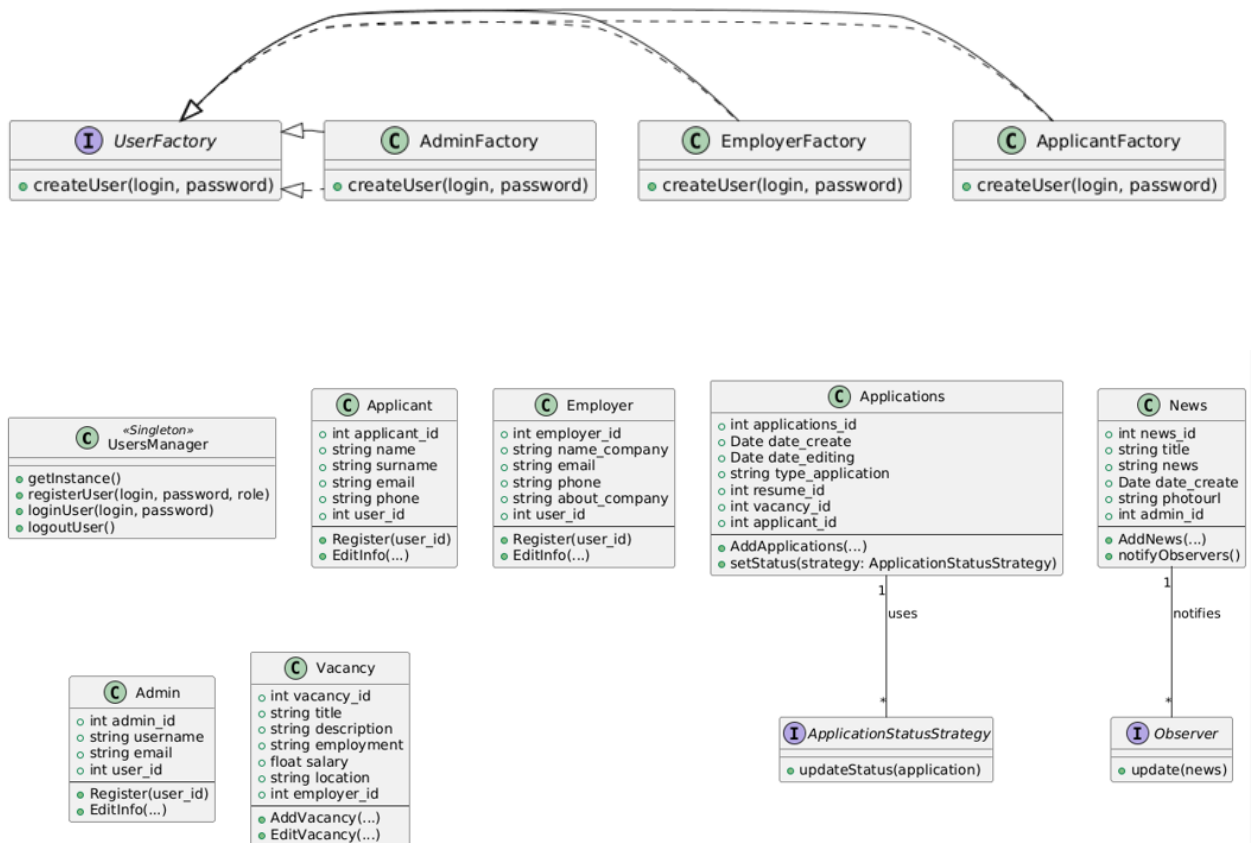
ApplicantFactory "1" --> "*" Applicant : creates (фабрика створює кандидатів; патерн Factory)

EmployerFactory "1" --> "*" Employer : creates (фабрика створює роботодавців; патерн Factory)

AdminFactory "1" --> "*" Admin : creates (фабрика створює адміністраторів; патерн Factory)

Applications "1" -- "*" ApplicationStatusStrategy : uses (заявка динамічно змінює свій статус через стратегію; патерн Strategy)

News "1" -- "*" Observer : notifies (новини повідомляють підписників; патерн Observer)



Singleton (UsersManager) - Гарантує єдину точку доступу до менеджера користувачів. Всі операції логіну, реєстрації і керування сесіями проходять через один екземпляр UsersManager

Factory (UserFactory / ApplicantFactory / EmployerFactory / AdminFactory) - Інкапсулює створення користувачів різних типів. Замість switch у RegisterUser використовується фабрика, що повертає об'єкт потрібного типу.

Strategy (ApplicationStatusStrategy) - Динамічна зміна статусу заявки без модифікації класу Applications. Клас Applications приймає стратегію AcceptedStrategy, RejectedStrategy або UnderReviewStrategy для оновлення статусу.

Observer (NewsObserver / UserObserver) - Повідомлення користувачів про новини. Клас News повідомляє всіх підписаних користувачів через метод notifyObservers()