

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ програмної інженерії \_\_\_\_\_  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_

Програмна система для автоматизації онлайн-рекрутингу.  
Front-end  
(тема)

Виконав:  
студент 4 курсу, групи ПЗПІ-20-4

Харченко П.Д.  
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного  
забезпечення  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія  
(повна назва освітньої програми)

Керівник доц. кафедри ПІ Побіженко І. О.  
(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри

\_\_\_\_\_  
(підпис)

З.В.Дудар  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерних наук  
 Кафедра \_\_\_\_\_ програмної інженерії  
 Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський)  
 Спеціальність \_\_\_\_\_ 121 – Інженерія програмного забезпечення  
 Тип програми \_\_\_\_\_ Освітньо-професійна  
 Освітня програма \_\_\_\_\_ Програмна Інженерія  
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Харченко Поліні Дмитрівні  
 (прізвище, ім'я, по батькові)

1. Тема роботи Програмна система для автоматизації онлайн-рекрутингу. Front-end  
 Затверджена наказом по університету від 20.05.2024 № 471 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 16.06.2024
3. Вихідні дані до роботи Розробити і створити клієнтську частину додатку для автоматизації онлайн-рекрутингу за допомогою мови програмування TypeScript і фреймворку Angular.
4. Перелік питань, що потрібно опрацювати в роботі  
Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	08.04.2024	<i>виконано</i>
2	Створення специфікації ПЗ	15.04.2024	<i>виконано</i>
3	Проектування ПЗ	24.04.2024	<i>виконано</i>
4	Розробка ПЗ	05.05.2024	<i>виконано</i>
5	Тестування ПЗ	30.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	05.06.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	06.06.2024	<i>виконано</i>
8	Попередній захист	16.06.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	16.06.2024	<i>виконано</i>
10	Здача роботи у електронний архів	16.06.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	16.06.2024	<i>виконано</i>

Дата видачі завдання 08 квітня 2024р.

Студент (ка) \_\_\_\_\_  
(підпис)

Харченко П. Д.

Керівник роботи \_\_\_\_\_  
(підпис)

доц. кафедри ПІ Побіженко І. О.  
(посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра: 90 стор., 33 рис., 7 джерел, 5 додатки.

АВТОМАТИЗАЦІЯ, ВЕБ-ДОДАТОК, ДОШКА, КАЛЕНДАР, НАЙМ, РЕКРУТИНГ, СТАТИСТИКА, ANGULAR, ANGULAR MATERIALS, REST API, TYPE SCRIPT.

Об'єкт розробки – клієнтська частина програмної системи автоматизації рекрутингу.

Мета розробки полягає у створенні веб-додатку, який надає користувачам можливість ефективного пошуку вакансій та забезпечує роботодавцям можливість публікації вакансій та здійснення управління ними. Веб-додаток повинен забезпечувати зручний інтерфейс для взаємодії користувачів із системою, а також надавати широкий спектр функціональності для зручності використання та підвищення задоволеності користувачів.

Метод рішення – середовище розробки Microsoft Visual Studio Code, фреймворк Angular, мова програмування TypeScript, REST API, хмарне рішення Azure DevOps.

У результаті розробки було створено веб-додаток, який взаємодіє з серверною частиною за допомогою REST API і вирішує проблему автоматизації рекрутингу.

ANGULAR, ANGULAR MATERIALS, AUTOMATION, BOARD, CALENDAR, HIRING, RECRUITMENT, REST API, STATISTICS, TYPE SCRIPT, WEB APPLICATION.

The object of development is the client part of the recruiting automation software system.

The purpose of the development is to create a web application that provides users with the ability to effectively search for vacancies and provides employers with the ability to publish and manage vacancies. The web application should provide a user-friendly interface for user interaction with the system, as well as provide a wide range of functionality for ease of use and increase user satisfaction.

The solution method is Microsoft Visual Studio Code development environment, Angular framework, TypeScript programming language, REST API, Azure DevOps cloud solution.

As a result of the development, a web application was created that interacts with the server side using the REST API and solves the problem of recruitment automation.

Я, Харченко Поліна Дмитрівна , студентка гр. ПЗП-20-4, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програмна система для автоматизації онлайн-рекрутингу. Front-end», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомена з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ .....	9
1.1 Аналіз предметної галузі.....	9
1.2 Аналіз існуючих рішень .....	10
1.3 Виявлення проблем та актуалізація рішень .....	14
1.4 Аналіз способів подальшої монетизації .....	15
1.5 Постановка задачі.....	16
2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ .....	18
2.1 Функціональні вимоги .....	18
2.2 Нефункціональні вимоги.....	19
3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМУВАННЯ СИСТЕМИ .....	21
3.1 UML Проектування ПЗ.....	21
3.2 Проектування ПЗ.....	28
4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ.....	33
4.1 Загальні відомості .....	33
4.2 Розробка програмної системи .....	33
4.3 Опис програмної системи.....	37
5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	49
ВИСНОВКИ.....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	53
ДОДАТОК А .....	54
ДОДАТОК Б .....	55
ДОДАТОК В .....	73
ДОДАТОК Г .....	89
ДОДАТОК Д.....	90

## **ПЕРЕЛІК СКОРОЧЕНЬ**

HTTP – Hypertext Transfer Protocol

IDE – Integrated Development Environment

JSON – JavaScript Object Notation

JWT – JSON Web Token

REST API – Representational State Transfer Application Programming Interface

SPA – Single Page Application

UML – Unified Modeling Language

## ВСТУП

У сучасному світі, де конкуренція на ринку праці постійно зростає, важливість ефективного рекрутингу стає визначальною для успішного функціонування підприємств. При цьому виникає потреба в розробці та впровадженні програмних систем, спрямованих на автоматизацію процесів підбору та управління персоналом. Дослідження та розробка таких систем стають актуальним завданням в контексті вдосконалення управління людськими ресурсами та підвищення конкурентоспроможності підприємств.

Метою даної кваліфікаційної роботи є розробка програмної системи для автоматизації рекрутингу, спрямованої на оптимізацію процесів відбору кандидатів, зменшення часу, необхідного для здійснення цих процесів, та підвищення ефективності вибору кваліфікованого персоналу. Для досягнення цієї мети передбачено розв'язання наступних завдань:

- аналіз сучасних підходів до рекрутингу та існуючих програмних засобів;
- визначення вимог на основі аналізу потреб потенційних користувачів;
- проектування архітектури програмної системи;
- розробка програмної системи;
- проведення тестування розробленої програмної системи з метою перевірки її працездатності та відповідності вимогам.

Галузь застосування результатів роботи охоплює широкий спектр підприємств та організацій, які потребують ефективного управління людськими ресурсами. Впровадження розробленої системи дозволить здійснювати рекрутинг як великих, так і малих підприємств, незалежно від їх специфіки та галузі діяльності.

Отже, дана кваліфікаційна робота спрямована на вирішення актуальних завдань у сфері рекрутингу та управління персоналом, зокрема розробку програмної системи, що має значний потенціал для застосування в різних галузях бізнесу.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Аналіз предметної галузі

Сучасний рекрутинг є ключовим елементом успішного управління людськими ресурсами в організаціях. Він відіграє важливу роль у формуванні ефективного та конкурентоспроможного колективу, що є критичним для досягнення стратегічних цілей підприємства. Поняття рекрутингу охоплює в собі комплекс процесів, починаючи від оголошення вакансій та закінчуючи прийняттям на роботу кандидатів, які відповідають вимогам вакантних посад.

У сучасному світі рекрутинг піддався значним змінам через розвиток технологій та зміну підходів до роботи. Однією з головних тенденцій є широке використання цифрових інструментів у даних процесах. Це охоплює розміщення вакансій на онлайн-платформах, використання аналітики даних для оцінки ефективності рекрутингових стратегій та автоматизацію певних аспектів відбору.

Програмні засоби грають важливу роль у сфері рекрутингу, надаючи компаніям інструменти для ефективного керування відносинами з кандидатами, оцінки та відбору найбільш підходящих кандидатів на вакантні посади. Існують також системи для проведення онлайн співбесід та інші інструменти, спрямовані на полегшення та прискорення процесів рекрутингу.

Проте, незважаючи на наявність програмних засобів, у даній сфері виникають різноманітні виклики. До них відносяться висока конкуренція на ринку праці, скорочення термінів пошуку та відбору кандидатів, а також нестабільність вимог до кандидатів у зв'язку з швидкими змінами в технологіях та ринкових умовах.

У висновку, аналіз предметної галузі рекрутингу підкреслює важливість розуміння сучасних тенденцій та викликів, а також необхідність використання програмних засобів для оптимізації та покращення рекрутингових процесів в організаціях.

## 1.2 Аналіз існуючих рішень

На сучасному ринку існує значна кількість програмних рішень та аналогів, спрямованих на автоматизацію рекрутингових процесів. Проте, деякі з них можуть не повністю відповідати потребам та вимогам потенційного користувача.

Перше рішення, яке ми розглянемо, це ресурс Glassdoor [1] – американський веб-сайт, де компаніям пропонується представити свій бренд (див. рис. 1.1), розповісти свою історію, продемонструвати нагороди, рейтинги, огляди та ключові оновлення компанії, щоб проілюструвати, чому шукачі роботи повинні хотіти працювати з ними.

Рисунок 1.1 – Сторінка бренду, опублікована в Glassdoor (за даними [1])

Особливість ресурсу полягає в тому, що він надає доступ до конфіденційної інформації про робочі умови в різних компаніях. Сервіс дозволяє користувачам залишати відгуки про свій досвід роботи (див. рис. 1.2), оцінювати керівництво, отримувати інформацію про заробітну плату, ділитися досвідом інтерв'ювання та порівнювати компанії за різними критеріями.

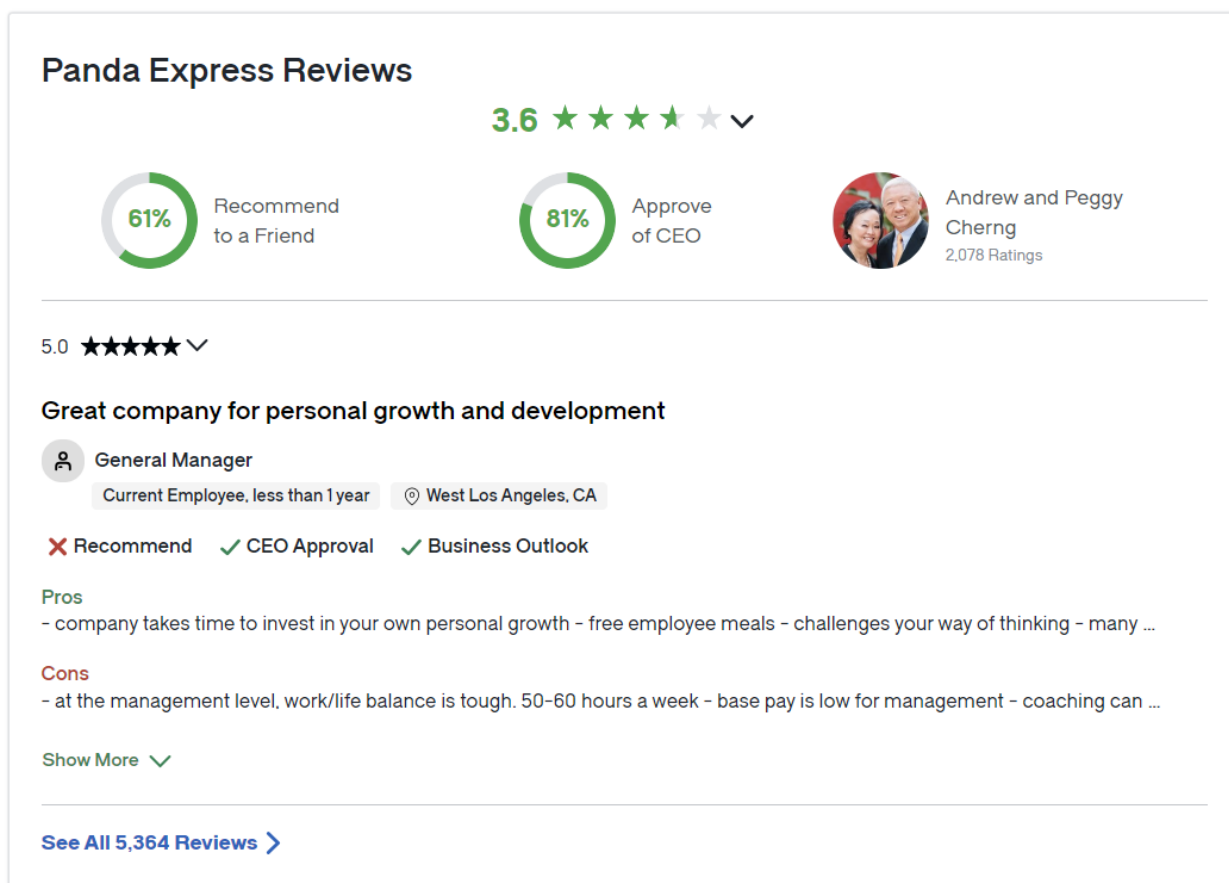


Рисунок 1.2 – Система відгуків, запропонована Glassdoor (за даними [1])

Розмістити оголошення на сайті Glassdoor можливо тільки для компанії, ресурс не націлений на приватні оголошення, а можливості користувача сильно обмежені тим, як багато інформації він залишив про себе і свій досвід роботи.

Work.ua – це провідна онлайн-платформа в Україні для пошуку роботи та розміщення вакансій [2]. Її функціонал забезпечує можливість як шукати вакансії в різних галузях та регіонах, так і роботодавцям розміщати оголошення про вакансії.

Платформа надає користувачам широкий спектр можливостей для фільтрації вакансій, що дозволяє їм точно визначити свої умови пошуку. Зокрема, можна обирати вакансії за рівнем зарплати, галуззю, містом, а також за наявністю обмежень, таких як інвалідність, неповний робочий день чи відсутність вищої освіти (див. рис. 1.3).

The screenshot displays the work.ua job search interface. It features three job listings on the left and a filter sidebar on the right.

**Job Listing 1:** "Менеджер по роботі з добровольцями (волонтерами)". Salary: 15 000 – 30 000 грн. Employer: Здоров'я українського народу, міжнародний благодійний фонд. Location: Київ. Description: Повна зайнятість. Також готові взяти людину з інвалідністю, ветерана. Досвід роботи від 2 років. Вища освіта. Міжнародний Благодійний Фонд «ЗДОРОВ'Я УКРАЇНСЬКОГО НАРОДУ» ОГОЛОШУЄ КОНКУРС на заміщення вакантної посади... Date: вчора. Action: Зберегти.

**Job Listing 2:** "Заступник керівника управління з питань ризику ліквідності". Employer: ПриватБанк. Location: Київ. Description: Повна зайнятість. Також готові взяти людину з інвалідністю. Досвід роботи від 2 років. Вища освіта. ПриватБанк — є найбільшим банком України та одним з найбільш інноваційних банків світу. Займає лідируючі позиції за всіма... Date: вчора. Action: Зберегти.

**Job Listing 3:** "Виконавець робіт (з бригадою)". Employer: Будівельна компанія-ВР. ТОВ. Location: Київ, 5.3 км від центрів. Description: Відгук без резюме. Logo: ВР БУДІВЕЛЬНА КОМПАНІЯ.

**Filter Sidebar:**

- Аміністрація, керівництво середньої ...
- Людам з інвалідністю
- Вид зайнятості**
  - Повна 195
  - Неповна 19
- Зарплата**
  - від: будь-яка
  - до: будь-яка
  - Не вказана 70
- Знання мов**
  - Англійська 28
  - Українська 28
- Підходить**
  - Кандидатам без досвіду 45
  - Кандидатам без резюме 37
  - Студентам 74
  - Людам з інвалідністю 198
  - Ветеранам 131
  - Пенсіонерам 36

Рисунок 1.3 – Пошук вакансії запропонований work.ua (за даними [2])

Однак платформа не має вбудованих інструментів зручного керування процесом найму, відсутня можливість мережевого спілкування, а також система відгуків і статистики.

Системи оцінки кандидатів, такі як HackerRank [3] та Codility [4], надають компаніям можливість створювати та відправляти кандидатам завдання для вирішення, які перевіряють їхні знання та навички у програмуванні (див. рис. 1.4). Часто ці завдання можуть включати рішення алгоритмічних задач, написання коду або розв'язання конкретних програмних завдань. Вони корисні для технічних позицій у сфері інформаційних технологій. Однак, ці системи можуть бути менш ефективними для оцінки кандидатів на позиціях, що вимагають інші навички.

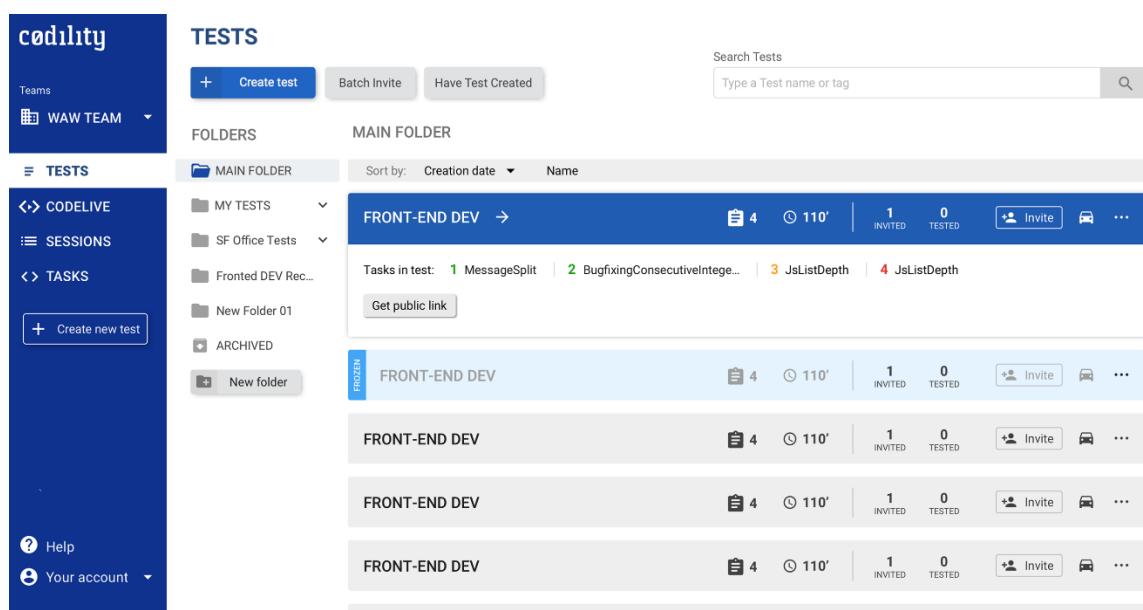


Рисунок 1.4 – Інтерфейс додатку Codility (за даними [4])

Jira – це інструмент для керування проектами, який дозволяє створювати різні типи дошок для організації завдань та проектів. У контексті рекрутингу, дошки Jira [5] можуть бути використані для керування процесом найму, відслідковування кандидатів на різних етапах відбору та планування роботи команди (див. рис. 1.5).

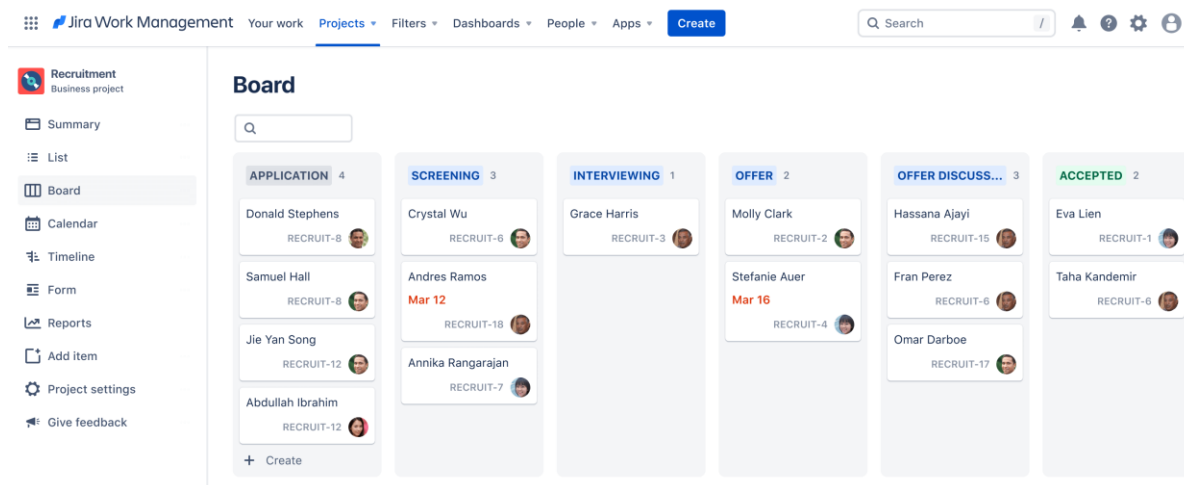


Рисунок 1.5 – Шаблон дошки Jira для найму персоналу (за даними [5])

Проте важливо зауважити, що Jira-дошки, хоч і є корисним інструментом для організації роботи команди та керування проектами, є лише частиною функціоналу, необхідного для повноцінної системи рекрутингу. Вони не мають

вбудованих інструментів для аналітики, зберігання резюме, комунікації з кандидатами тощо.

У сфері найму також використовуються табличні інструменти, такі як Excel та подібні, які дозволяють зберігати та аналізувати дані шляхом побудови діаграм. Однак ці інструменти, хоч і мають корисні функції, не є спеціалізованими рекрутинговими системами і не забезпечують всіх потреб, які виникають у процесі найму персоналу. Вони можуть бути обмеженими в розмірах обробки та аналізу великого обсягу даних, а також не мають вбудованих функцій співпраці та комунікації між рекрутерами та кандидатами.

### 1.3 Виявлення проблем та актуалізація рішень

У результаті аналізу різноманітних програмних рішень та аналогів, які використовуються в галузі рекрутингу, виявлено ключові функції, які додаток повинен включати для оптимізації та централізації всіх процесів.

Зокрема, він має надавати можливість використовувати платформу для створення та презентації бренду, спрощену систему зберігання інформації про кандидатів, а також інструменти оцінки їхніх навичок та професійної підготовки. Додаток також повинен забезпечувати зручне використання дошок для управління процесом найму та відслідковуванням кандидатів, а також надавати доступ до інструментів аналізу і статистики для вдосконалення рекрутингових стратегій і процесів.

Під час розробки нового технічного продукту можуть виникати різні труднощі, які варто врахувати на ранніх етапах. Наприклад, непередбачувані технічні проблеми, такі як конфлікти між компонентами програмного забезпечення чи архітектурні обмеження, можуть призвести до затримок у розробці та збільшення витрат на їх вирішення.

Недоліки у дизайні інтерфейсу, наприклад, неякісна навігація або неінтуїтивне розташування елементів, можуть суттєво вплинути на користувацький досвід та відображення продукту на ринку.

Під час розробки важливо забезпечити безпеку і конфіденційність інформації

користувачів. Використання шифрування є ефективним методом захисту інформації в базі даних, а також під час передачі даних через мережу, щоб захистити їх від несанкціонованого доступу.

Також важливо врахувати можливість невідповідності технічних характеристик з іншими системами або платформами. Для уникнення таких проблем необхідно вести детальний аналіз вимог, тестувати продукт на різних етапах розробки, враховувати кращі практики безпеки та розробляти продукт з урахуванням можливих сумісностей та інтеграцій.

#### 1.4 Аналіз способів подальшої монетизації

У сучасному світі монетизація є важливим аспектом для кожного додатку. Вона забезпечує стабільний дохід, необхідний для подальшого розвитку та підтримки продукту. Успішна стратегія монетизації дозволяє не лише покрити витрати на розробку, а й залучати кошти для впровадження нових функцій та поліпшень. Без ефективної монетизації додаток може стати нерентабельним і призвести до його зникнення з ринку.

Почнемо з розгляду моделі підписки як одного зі способів монетизації. Можливість підписки на різні рівні функціоналу може включати доступ до різних наборів функцій та сервісів. Наприклад, базовий план може включати основні можливості, такі як розміщення вакансій та базова аналітика. У той час як преміум план може додатково надавати розширені функції, такі як інтеграція з іншими системами або розширена звітність. Така модель може забезпечити користувачам гнучкість у виборі тарифного плану в залежності від їхніх потреб та можливостей.

Надання консультаційних послуг може включати індивідуальні зустрічі з компаніями з питань оптимізації рекрутингових процесів та використання додатку. Такі послуги можуть охоплювати різні аспекти, включаючи ознайомлення з додатком, розробку стратегій рекрутингу, підготовку користувачів та аналіз ефективності.

Варто врахувати можливість співпраці з іншими компаніями в сфері рекрутингу. Це може включати партнерські програми, з можливістю отримати

комісійні винагороди за успішні рефералізації або спільні маркетингові заходи.

Можливості розміщення реклами або спонсорських матеріалів в додатку можуть привернути інші компанії або сервіси в сфері рекрутингу. Це може бути вигідним для обох сторін і створити додаткові джерела доходу для вашого додатку.

### 1.5 Постановка задачі

Під час аналізу ми дослідили різні аспекти, щоб визначити які саме задачі необхідно вирішити у процесі розробки. Проаналізувавши різноманітні аналоги, було виявлено їхні сильні та слабкі сторони, враховано можливі технічні проблеми, з якими можна стикнутись під час розробки продукту, а також ми приділили увагу питанню монетизації.

На основі цього аналізу ми прийшли до висновку, що важливо забезпечити ефективне управління процесом найму та зменшити залежність від сторонніх інструментів. Ми прагнемо зібрати всі необхідні функції в одному зручному середовищі, що дозволить оптимізувати роботу кадрового відділу та підвищити продуктивність. Враховуючи це, ми визначили ключові функції, які необхідно реалізувати у нашому продукті, а саме:

MF-1 Створення та взаємодія з особистим акаунтом.

MF-2 Створення та взаємодія з вакансіями.

MF-3 Створення та взаємодія з резюме, підтримка завантаження файлів.

MF-4 Комунікація між рекрутером та найманцем за допомогою чату.

MF-5 Система тестування для вакансій.

MF-6 Kanban-дошка [6] для контролю над процесом найму.

MF-7 Інтеграція з календарем для планування зустрічей.

MF-8 Відправлення електронних листів та використання шаблонів.

MF-9 Аналітичні інструменти для статистичного аналізу даних.

MF-10 Збір відгуків про вакансії та кандидатів.

MF-11 Можливість зміни мови інтерфейсу.

Задача для розробки клієнтської частини полягає в створенні інтуїтивного та ефективного інтерфейсу для користувачів з використанням сучасних технологій.

Важливо забезпечити зручну навігацію та швидкий доступ до всіх функцій. Реалізувати взаємодію клієнтської частини з сервером через API, щоб користувачі могли працювати з даними, збереженими на сервері, через інтерфейс користувацького додатку.

## 2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Функціональні вимоги

Програмна система для автоматизації рекрутингу повинна надавати ряд ключових функцій, щоб забезпечити ефективне управління процесом найму та підвищити продуктивність кадрового відділу. Перш за все, користувачам має бути надана можливість створювати та керувати своїми особистими обліковими записами. Система повинна мати функціонал взаємодії з такими об'єктами: вакансії, резюме, календар і дошка для планування та управління процесами, шаблони електронних листів і документів, система тестування і статистичними даними. На основі перелічених об'єктів, можна визначити ключові аспекти взаємодії:

- а) особистий акаунт:
  - 1) реєстрація нових користувачів;
  - 2) керування обліковими записами;
- б) вакансії:
  - 1) створення, перегляд, редагування і видалення вакансій;
  - 2) пошук вакансій по фільтрам;
- в) резюме:
  - 1) створення, перегляд, редагування і видалення;
  - 2) відправлення резюме на вакансію;
  - 3) порівняння резюме декількох кандидатів;
- г) комунікації:
  - 1) взаємодія між рекрутером та кандидатом через вбудований чат;
  - 2) відправлення електронних листів та використання шаблонів;
- д) система тестування:
  - 1) створення і проходження тестів з прив'язкою до вакансії;
- е) інтеграція з календарем для планування зустрічей;
- ж) kanban-дошка для керування процесом найму;
- з) аналітичні інструменти:

- 1) панель візуалізації ключових статистичних даних;
- 2) інструмент розрахунку заробітної плати;
- 3) система відгуків.

Ці аспекти є важливими для забезпечення ефективного та продуктивного процесу найму персоналу.

## 2.2 Нефункціональні вимоги

Клієнтська частина системи має забезпечувати високий рівень безпеки, зокрема, за допомогою захищеного з'єднання для передачі даних між клієнтом та сервером через API. Також важливо забезпечити захист від небажаних вторгнень, використовуючи механізми аутентифікації та авторизації, надані сервером.

Для досягнення максимальної продуктивності, клієнтська частина повинна мінімізувати затримки у відгуку інтерфейсу користувача на дії, виконані на клієнтській стороні, та ефективно використовувати ресурси пристроїв користувачів, зокрема, оптимізуючи завантаження та використання оперативної пам'яті.

Необхідно розробити інтуїтивно зрозумілий та естетичний інтерфейс, який буде зручним для користувачів різного рівня технічної підготовки. Також важливо забезпечити сумісність інтерфейсу з різними браузерами.

Важливою вимогою є надійність роботи клієнтської частини навіть у випадку непередбачуваних ситуацій, таких як втрата зв'язку з сервером чи помилки відправки запитів. Крім того, необхідно забезпечити обробку помилок та надання користувачу зрозумілих повідомлень про помилки або рекомендацій щодо подальших дій.

Клієнтська частина має бути готовою до підтримки росту обсягів даних та користувачів без втрати продуктивності. Це вимагає гнучкості і готовності до майбутніх змін, що дозволяє легко впроваджувати нові функції та змінювати інтерфейс відповідно до потреб бізнесу і змін в API серверної частини.

Система буде побудована відповідно до архітектури клієнт-сервер, що передбачає розділення функціоналу між клієнтською та серверною частинами.

Клієнтська частина буде відповідальна за інтерфейс взаємодії з користувачем та відображення даних, тоді як серверна частина буде виконувати бізнес-логіку та взаємодію з базою даних. Така архітектура дозволить забезпечити ефективну роботу додатку та оптимізувати роботу його компонентами. Взаємодія між сервером та клієнтом буде здійснюватися через API з використанням HTTP протоколу.

Клієнтська частина веб-додатку буде реалізована з використанням TypeScript та фреймворка Angular [7], а також Angular Material для створення користувацького інтерфейсу. TypeScript полегшує розробку та підвищує безпеку коду. Angular надає широкий набір інструментів для реалізації функціональності додатку, включаючи маршрутизацію та управління станом, а також має готові до використання компоненти з сучасним дизайном, Angular Materials, що допоможуть швидко створити стильний інтерфейс.

## 3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМУВАННЯ СИСТЕМИ

### 3.1 UML Проектування ПЗ

UML проектування використовується для моделювання та аналізу програмного забезпечення, що допомагає розробникам уявити структуру системи, її функціональність та взаємозв'язки між компонентами.

Для клієнтської частини проекту було обрано розробити Use Case діаграму (див. рис. 3.1) для ідентифікації основних функцій та можливостей користувача.

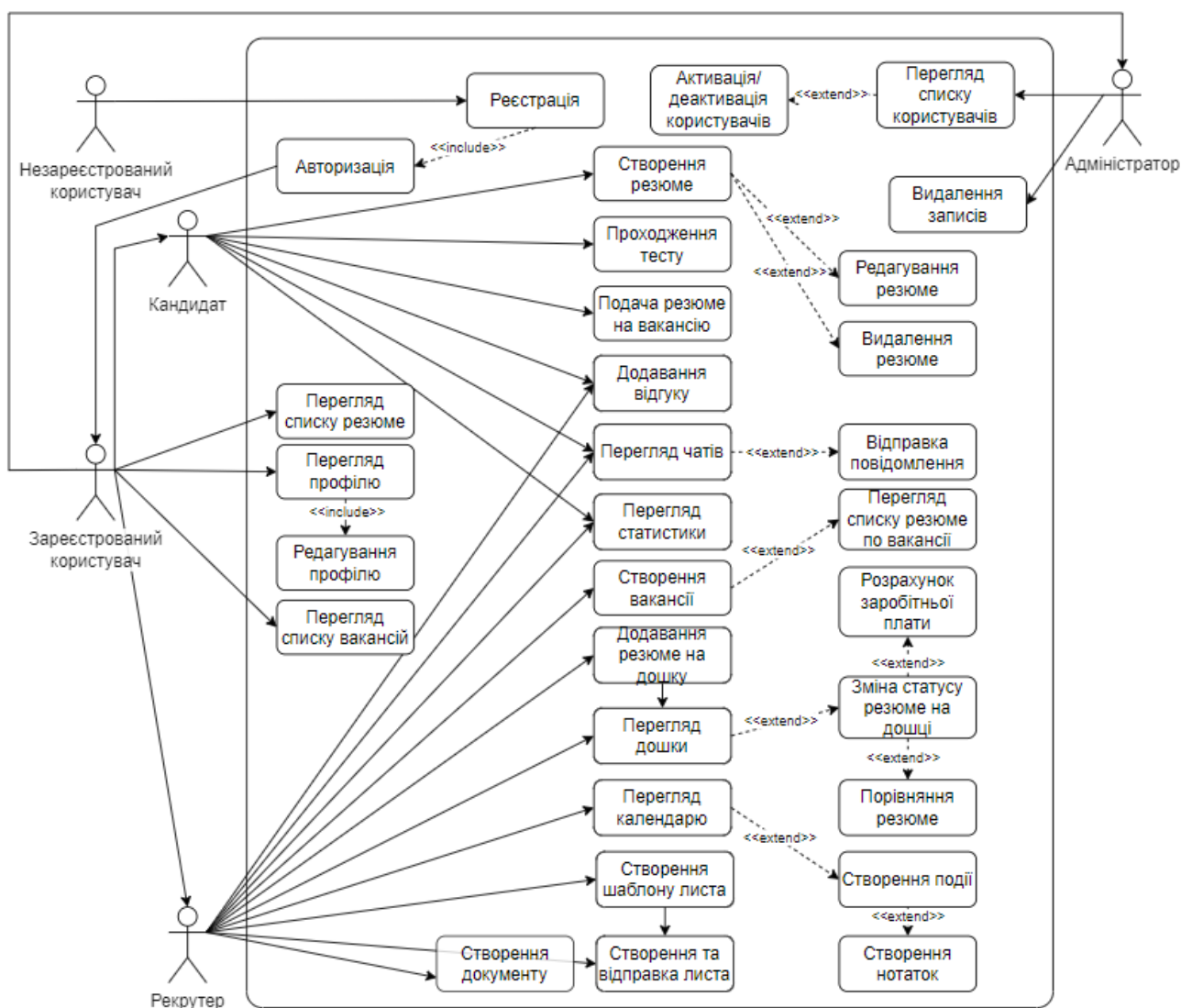


Рисунок 3.1 – Діаграма Use Case (рисунок виконано самостійно)

Проаналізувавши Use Case діаграму, видно, що кожен тип користувача має свій набір дій, які відображають їхні можливості в системі. Незареєстрований

користувач може виконати дії реєстрації та авторизації, щоб мати доступ до додаткових функцій. Зареєстрований користувач, поділяється на адміністраторів, рекрутерів та кандидатів, може переглядати резюме та вакансії, редагувати свій профіль та здійснювати інші дії, пов'язані з навігацією.

Адміністратор системи має можливість керувати користувачами, включаючи їхню активацію та деактивацію, а також виконувати інші адміністративні функції, наприклад, перегляд і видалення записів. Рекрутери мають широкий спектр можливостей, включаючи створення вакансій, взаємодію з кандидатами через чат, аналіз статистики та інші дії, які спрямовані на ефективне управління рекрутинговим процесом. Кандидати можуть створювати та редагувати свої резюме, подавати їх на вакансії, взаємодіяти з рекрутерами та здійснювати дії, що допомагають їм знайти відповідну роботу.

Діаграма станів є корисним інструментом для моделювання поведінки системи, візуалізуючи стани та переходи між ними (див. рис. 3.2).

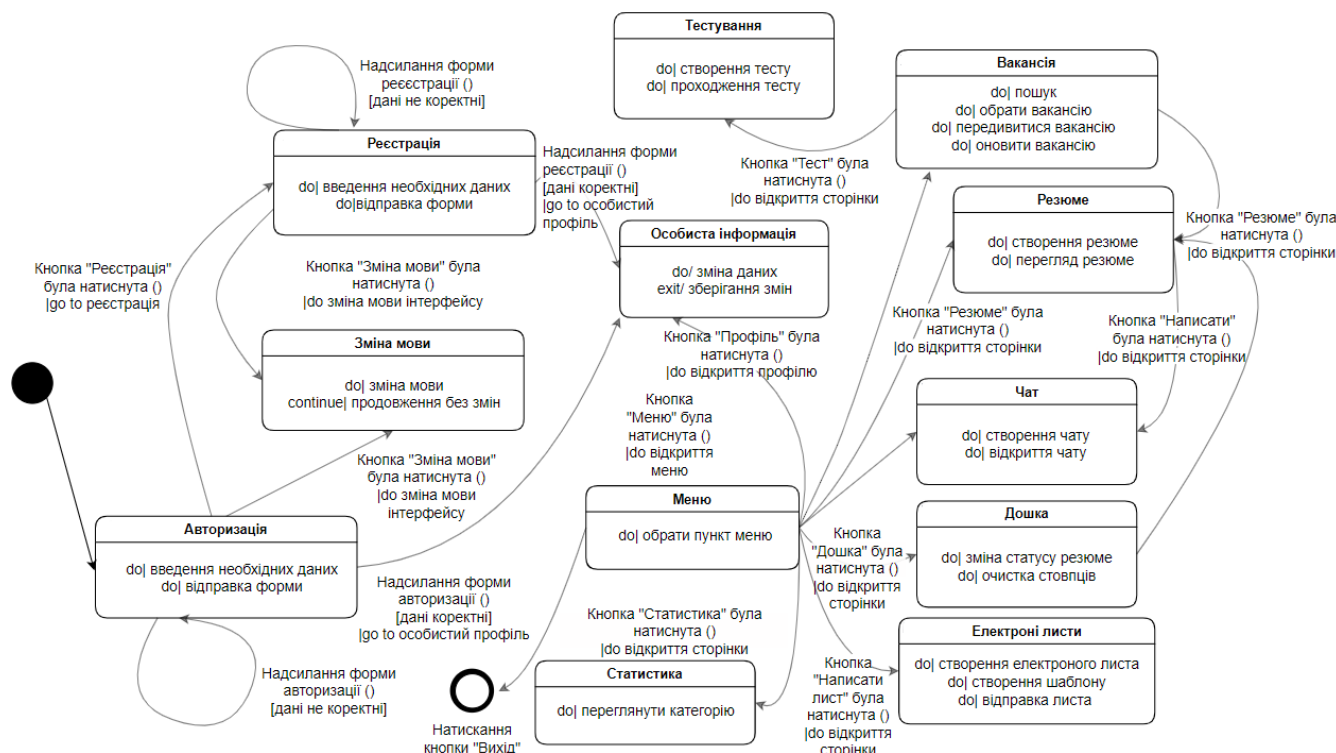


Рисунок 3.2 – Діаграма станів (рисунок виконано самостійно)

За діаграмою станів можна зрозуміти, як система повинна поводитись у різних умовах і виявити можливі недоліки ще до початку розробки коду. Такий

підхід сприяє високій якості програмного забезпечення та зменшує час розробки і тестування.

Система для автоматизації рекрутингу складається з 12 станів, кожен з яких представляє собою певну функціональність та взаємодіє з користувачем унікальним способом.

Користувач починає станом «Авторизація», де має ввести свої дані для входу в систему, що будуть перевірені системою, дозволяє користувачу перейти до наступного стану лише у випадку коректного введення. У разі некоректних даних, система залишається в поточному стані або відправляється до стану «Реєстрація», якщо у нього немає акаунту. Стан «Реєстрація» вимагає від користувача введення обов'язкових даних та дозволяє перейти до наступного стану при їхній валідності. Це є важливим кроком для створення особистого облікового запису в системі. Стани «Авторизація» та «Реєстрація» схожі за процесом і можуть посилатися один на одного.

«Зміна мови» – стан, який доступний з моменту запуску додатку, користувач може змінити мову інтерфейсу за своїм вибором. Це створює зручні умови для використання додатку в різних мовних середовищах.

«Меню» є центральним елементом навігації і розширює варіанти сценаріїв після успішної авторизації користувача. Він надає доступ до всіх інших станів та функцій, що дозволяє користувачам зручно переходити між різними розділами системи.

Перехід до стану «Особиста інформація» можливий з різних точок входу в систему: авторизації, реєстрації або з меню. Цей стан дозволяє користувачам переглядати та редагувати особисті дані, забезпечуючи повну інтеграцію з системою.

Стан «Вакансія» володіє рядом функцій, включаючи можливість пошуку, створення, перегляду і оновлення вакансій. Крім того, з цього стану можна перейти до стану «Резюме», звідки доступний стан «Чат» між кандидатами та рекрутерами. Також можливий перехід до стану «Тестування», який включає як створення, так і проходження тестів. Стани «Вакансія», «Резюме» і «Тестування» тісно пов'язані

між собою, що створює інтегрований і ефективний процес найму.

Наступні стани складаються з інструментів, спрямованих на оптимізацію процесу найма з боку рекрутера. «Дошка» дозволяє керувати статусами резюме та управляти процесом найму. Перехід до стану «Резюме» і «Електронні листи» можливий з меню або зі стану резюме. Ці стани включають у себе процес створення шаблонів листів, самі листи і їх відправку.

Крім того, існує стан «Статистика», де дані змінюються в залежності від вибраної категорії в дашборді, що допомагає рекрутерам аналізувати та вдосконалювати свою роботу.

Таким чином, кожен стан в системі для автоматизації рекрутингу має свою унікальну роль і взаємозв'язок з іншими станами, що дозволяє забезпечити ефективну та зручну роботу користувачів в системі.

Діаграма діяльності – це графічний інструмент, що використовується для моделювання послідовності дій або активностей у системі та є важливим для розуміння того, як взаємодіє користувач з додатком і як саме відбувається процес виконання різних завдань.

Діаграма діяльності дозволяє уточнити послідовність дій, що виконує користувач або система. Це допомагає зрозуміти, які саме кроки потрібно виконувати, щоб завершити певне завдання або процес. Також, вона допомагає виявити можливі проблеми або помилки, що можуть виникнути під час взаємодії з додатком, та дозволяє заздалегідь передбачити ці ситуації.

Окрім того, аналізуючи діаграму діяльності, можна знайти можливості для оптимізації робочих процесів та забезпечити більш ефективну взаємодію користувача з додатком (див. рис.3.3).

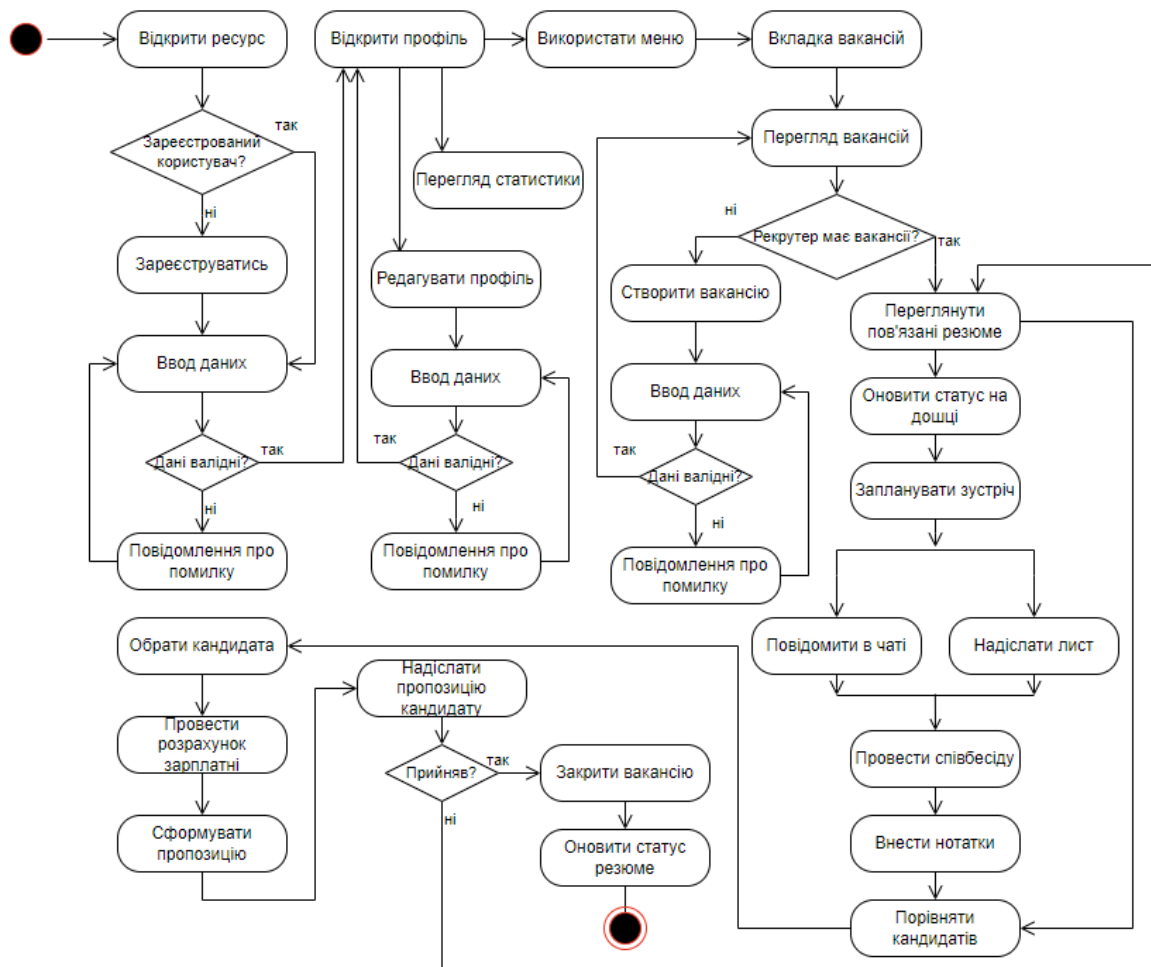


Рисунок 3.3 – Діаграма діяльності рекрутера (рисунок виконано самостійно)

Отже, діаграма діяльності грає важливу роль у розробці програмного забезпечення, надаючи учасникам проекту чітке уявлення про послідовність дій та взаємодію між різними елементами системи.

Наступною була розроблена діаграма класів, яка є важливим інструментом для проектування програмного забезпечення, оскільки вона дозволяє візуалізувати структуру системи та взаємозв'язки між її компонентами. Її загальна мета полягає в тому, щоб візуалізувати, як система повинна бути організована і які класи, атрибути та методи вона має включати.

У контексті проектування клієнтської частини, діаграма класів стає важливим інструментом для моделювання структури програми. Вона створення для розуміння, які компоненти існують у клієнтській частині, як вони взаємодіють між собою та як можуть бути організовані для досягнення певного функціоналу, а

також які дані можна очікувати та які операції можна виконати. Діаграма класів є важливим інструментом для забезпечення ефективної комунікації між розробниками клієнтської та серверної частини проекту, дозволяючи розробникам зосередитися на реалізації необхідного функціоналу та підтримувати консистентність у роботі обох частин програми.

Створена діаграма класів налічує 12 сутностей і 4 переліки (див. рис. 3.4).

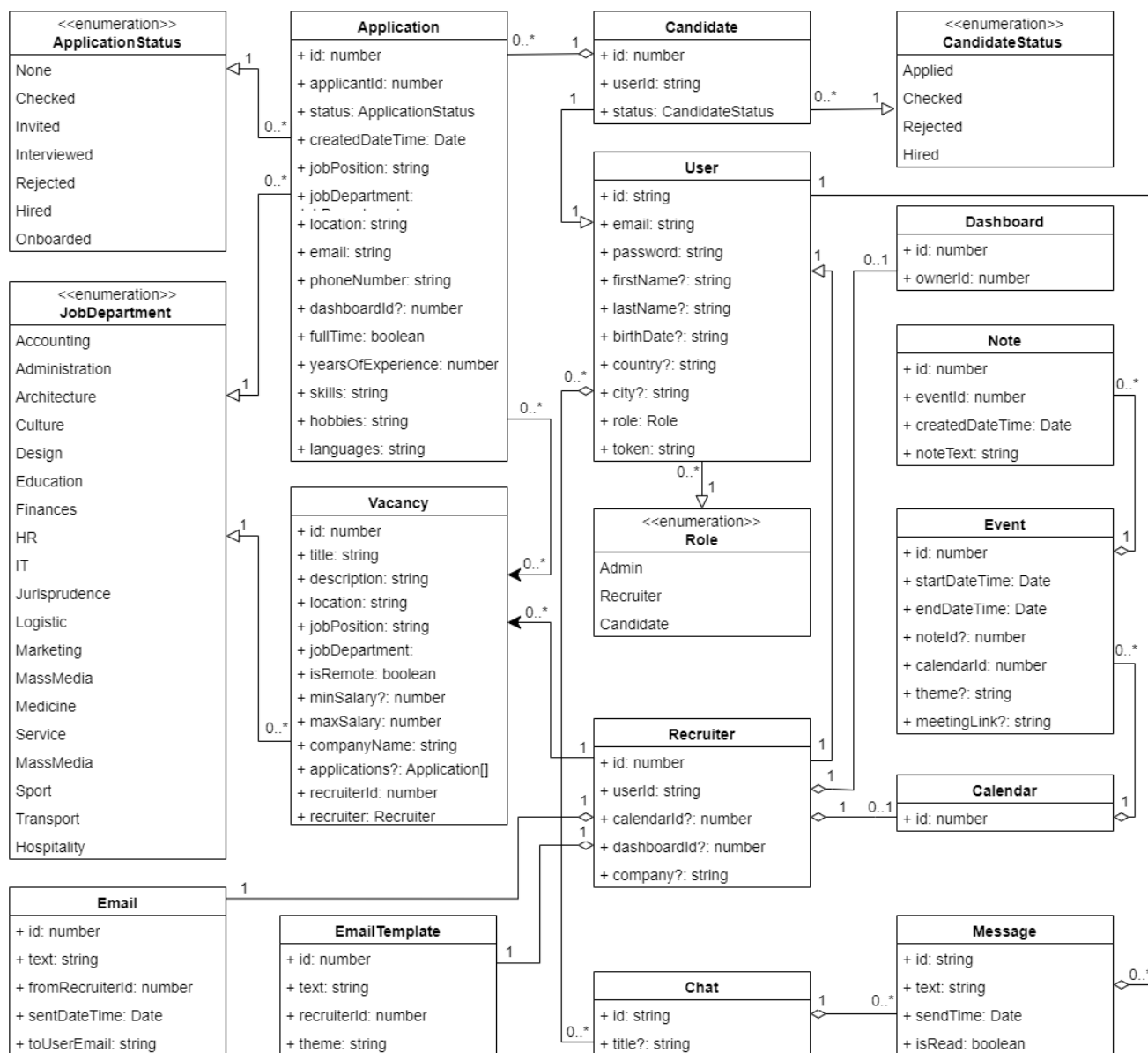


Рисунок 3.4 – Діаграма класів (рисунок виконано самостійно)

Розглянемо структуру діаграми класів:

– клас «Application» має атрибути, такі як унікальний ідентифікатор, ідентифікатор заявника, ідентифікатор дошки, стан, текст, дата створення,

- посада, відділ, розташування, електронна пошта, номер телефону, повний робочий день, роки досвіду, навички, хобі, мови;
- клас «Candidate» має атрибути, такі як унікальний ідентифікатор, ідентифікатор користувача, стан;
  - клас «Recruiter» має атрибути, такі як унікальний ідентифікатор, ідентифікатор користувача, ідентифікатор календаря, ідентифікатор дошки;
  - клас «User» має атрибути, такі як унікальний ідентифікатор, електронна пошта, пароль, ім'я, прізвище, ім'я користувача, дата народження, країна, місто, роль та токен;
  - клас «Calendar» має атрибут унікальний ідентифікатор;
  - клас «Dashboard» має атрибути, такі як унікальний ідентифікатор, ідентифікатор власника;
  - клас «EmailTemplate» має атрибути, такі як унікальний ідентифікатор, ідентифікатор рекрутера, текст, тема;
  - клас «Email» має атрибути, такі як унікальний ідентифікатор, ідентифікатор рекрутера, текст, електронна пошта отримувача та дата відправлення;
  - клас «Event» має атрибути, такі як унікальний ідентифікатор, ідентифікатор нотатки, ідентифікатор календаря, дата початку, дата закінчення, тема, посилання на зустріч;
  - клас «Note» має атрибути, такі як унікальний ідентифікатор, ідентифікатор події, дата створення, текст запису;
  - клас «Vacancy» має атрибути, такі як унікальний ідентифікатор, назва, опис, розташування, посада, відділ, віддалена робота, мінімальна зарплата, максимальна зарплата, назва компанії;
  - клас «Chat» має атрибути, такі як унікальний ідентифікатор, назва;
  - клас «Message» має атрибути, такі як унікальний ідентифікатор, текст, автор, час відправлення та ознака прочитаності;

- перелік «Role» має властивості, такі як Адміністратор, Рекрутер, Кандидат;
- перелік «ApplicationStatus» має властивості, такі як Нічого, Перевірено, Запрошено, Пройшов співбесіду, Відхилено, Найнято, В роботі;
- перелік «CandidateStatus» має властивості, такі як Подано заявку, Перевірено, Найнято, Відхилено;
- перелік «JobDepartment» має властивості, такі як Облік, Адміністрація, Архітектура, Культура, Дизайн, Освіта, Фінанси, HR, IT, Юриспруденція, Логістика, Маркетинг, ЗМІ, Медицина, Сервіс, Спорт, Транспорт, Готельно-ресторанна справа.

Створення діаграм, таких як Use Case, State, Activity і Class стала важливим кроком у процесі розробки програмного забезпечення для системи автоматизації рекрутингу. Ці діаграми надали нам чітке уявлення про структуру програми, взаємозв'язки між її компонентами та основні функції системи. Вони сприяли кращому розумінню внутрішньої логіки додатку, спростили комунікацію в команді розробників та допомогли ідентифікувати можливі проблеми ще на ранніх етапах розробки. Крім того, ці діаграми стали основою для подальшої розробки, надаючи важливу інформацію для написання коду, реалізації функціоналу та тестування програмного забезпечення. Таким чином, вони допомогли забезпечити високу якість та ефективність проекту в цілому.

### 3.2 Проектування ПЗ

Процес проектування програмного забезпечення включає в себе аналіз вимог, визначення архітектури системи та проектування ключових компонентів. Діаграма компонентів грає важливу роль у цьому процесі, оскільки вона надає візуальне відображення структури програми та зв'язків між її складовими частинами.

Angular, як сучасний фреймворк для розробки веб-додатків, надає структуру та шаблони для побудови додатків. Використання його основних елементів, таких як модулі, компоненти, сервіси та директиви, допомагає створити добре організований, масштабований та підтримуваний проект. Діаграма компонентів на

основі архітектури Angular допоможе нам уявити, як кожен компонент взаємодіє з іншими, як вони організовані та виконують свої функції (див. рис. 3.5).

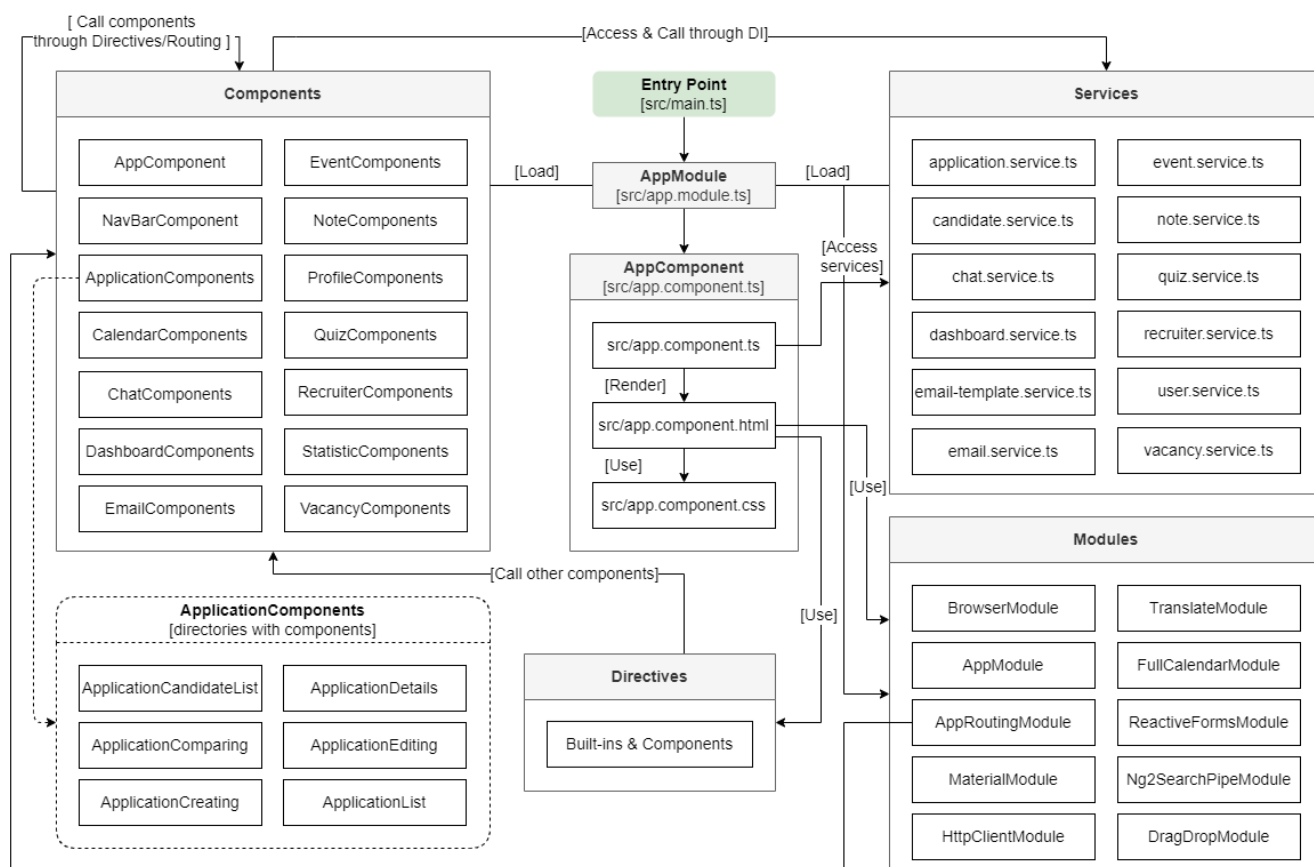


Рисунок 3.5 – Діаграма компонентів (рисунок виконано самостійно)

Архітектура фреймворку Angular ґрунтується на чотирьох основних концепціях: компонентах, шаблонах з прив'язкою даних та директивах, модулях та сервісах з впровадженням залежностей. Кожен з цих компонентів має важливе значення для створення структури додатку та його функціональності.

Компонент є основним блоком будь-якого Angular-дodatku. Він складається з класу TypeScript, HTML-шаблону та асоційованого імені. Кожен компонент може включати в себе інші компоненти за допомогою їх селекторів. Компонент також може мати опційні CSS-стилі.

Шаблон це розширений HTML, який включає функції прив'язки даних та директив. Прив'язка даних дозволяє зв'язувати дані компонента з його шаблоном, тоді як директиви додають логіку та створюють складні DOM-елементи.

Колекція пов'язаних функціональностей називається модуль. Angular

дозволяє групувати компоненти та сервіси у модулі, що полегшує організацію та розширення додатку. Один модуль може використовувати функціональність іншого модуля, в той час як сервіси – це прості класи, які надають конкретну функціональність та можуть використовуватись у компонентах для виконання різних завдань. Використання сервісів дозволяє багаторазового використовувати логіку додатку.

Діаграма компонентів дозволяє нам побачити загальну структуру додатку та взаємозв'язки між його елементами. Вона надає візуальне уявлення про те, як компоненти взаємодіють між собою та як вони вбудовані в загальну архітектуру додатку. Крім того, діаграма компонентів дозволяє нам зосередитися на одному з компонентів, такому як `AppComponent`.

Розглянувши `AppComponent` детальніше, ми можемо побачити всі елементи, які потрібні для забезпечення його основної функціональності, а саме:

- компонент кандидатів, які подали заяви;
- компонент порівняння заявок;
- компонент створення заявки;
- компонент редагування заявки;
- компонент детального перегляду заявки;
- компонент для відображення списку заявок.

Кожен з цих елементів містить свій HTML-шаблон, TypeScript-файл та файл стилів, які разом створюють компонент. Це допомагає нам зрозуміти, як кожен компонент працює, які дані він обробляє та як він взаємодіє з іншими частинами додатку. Такий підхід дозволяє нам легше розуміти та розробляти кожен компонент окремо, а також виявляти та виправляти можливі проблеми швидше та ефективніше.

Визначення основного функціоналу і архітектури проекту важливий крок у процесі розробки програмного забезпечення, який дозволяє зрозуміти, які саме можливості потрібно реалізувати у додатку та як вони будуть взаємодіяти між собою. Тепер можна перейти до ознайомлення з модулями Angular, необхідними для реалізації запланованих задач.

Angular – це потужний фреймворк, який надає різноманітні модулі та компоненти для розробки веб-додатків, розглянемо перелік тих, що можуть бути корисними для реалізації запланованих функціональних можливостей:

- RouterModule відповідає за маршрутизацію у додатку. Він дозволяє визначити шляхи, за якими можуть переходити користувачі, і пов'язані з ними компоненти;
- HttpClientModule призначений для роботи з HTTP-запитами. Він надає сервіс HttpClient для виконання HTTP-запитів до сервера, що дозволяє взаємодіяти зі зовнішніми джерелами даних;
- FormsModule/ReactiveFormsModule відповідають за роботу з формами у додатку. Вони надають різні можливості для збору та валідації даних, введених користувачем;
- NgModule основний модуль, який дозволяє організувати додаток у модульні структури, визначаючи компоненти, сервіси та інші ресурси, які потрібні для функціонування додатку;
- Ng2SearchPipeModule надає можливості для пошуку та фільтрації даних у веб-додатку. Він дозволяє швидко і ефективно виконувати пошук серед набору даних, включаючи масиви об'єктів;
- JwtModule є ключовим компонентом для забезпечення безпеки веб-додатку шляхом реалізації механізму аутентифікації та авторизації. Він дозволяє створювати та перевіряти JSON Web Tokens, що дозволяє ідентифікувати користувачів і керувати їх доступом до різних ресурсів в системі. За допомогою цього модуля відбувається обмін цифровими підписами між сервером і клієнтом, що гарантує безпеку передачі даних та забезпечує валідність користувацьких сесій;
- FullCalendarModule надає можливість використовувати готовий календар у веб-додатку для планування подій, зустрічей, важливих дат тощо. Цей модуль дозволяє інтегрувати календарну функціональність, що полегшує користувачам відстеження їхнього розкладу та організацію подій;
- DragDropModule є важливим компонентом для створення інтерактивних

інтерфейсів веб-додатків. З його допомогою рекрутер може зручно перетягувати заявки кандидатів на kanban-дошці в залежності від їхнього поточного статусу;

- TranslateModule дозволяє легко реалізувати функціонал перекладу веб-додатку. За допомогою цього модуля можна забезпечити підтримку мультимовності, дозволяючи користувачам вибирати мову інтерфейсу, яка найбільше їм підходить. Це робить додаток більш доступним і зручним для користувачів із різних країн і культур;
- Material Design бібліотека компонентів, яка може значно спростити розробку інтерфейсу користувача, надаючи готові компоненти для використання.

Ознайомлення з цими модулями та бібліотеками дозволить нам ефективно використовувати можливості Angular для реалізації запланованого функціоналу та забезпечення користувачам зручного та ефективного досвіду використання додатку.

## 4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

### 4.1 Загальні відомості

Клієнтська частина системи для автоматизації рекрутингу реалізована як SPA (Single Page Application) з використанням сучасних технологій. Основу додатку складає Angular, фреймворк, який обрано завдяки його потужності, модульній побудові, підтримці TypeScript та активній спільноті розробників. Використання TypeScript дозволяє підвищити якість коду завдяки статичній типізації, що допомагає зменшити кількість помилок на етапі розробки. Крім того, він забезпечує кращу підтримку сучасних засобів розробки (IDE), що спрощує процес написання та тестування коду.

Для побудови користувацького інтерфейсу використовується Angular Material, набір компонентів, який реалізує принципи Material Design від Google. Це дозволяє швидко створювати привабливі та функціональні інтерфейси, дотримуючись єдиних стандартів дизайну. Взаємодія з серверною частиною здійснюється за допомогою REST API, що дозволяє клієнтському додатку отримувати та надсилати дані через HTTP запити, забезпечуючи динамічну взаємодію з сервером та зберіганням даних.

Для аутентифікації та авторизації в системі використовується механізм JWT (JSON Web Tokens). Застосовується бібліотека `auth0/angular-jwt`, яка дозволяє безпечно передавати та зберігати токени доступу, забезпечуючи захищений доступ до ресурсів системи і зменшуючи ризик несанкціонованого доступу. Ці технології та інструменти забезпечують стабільну, масштабовану та безпечну роботу клієнтської частини системи для автоматизації рекрутингу, що сприяє ефективному управлінню процесом підбору персоналу.

### 4.2 Розробка програмної системи

Система реалізована у вигляді веб-додатку з використанням Angular, що дозволяє створювати компоненти, які можна повторно використовувати. Це забезпечує високу модульність та гнучкість у розробці інтерфейсу користувача,

спрощуючи підтримку та розширення функціональності додатку.

У системі створено моделі, необхідні для зручної роботи додатку і його взаємодії з API. Ці моделі розташовані в папці `models` (див. рис. 4.1), що забезпечує централізоване зберігання та легкий доступ до визначень типів даних, які використовуються в додатку. Це полегшує керування даними та забезпечує їх відповідність між клієнтською та серверною частинами системи.

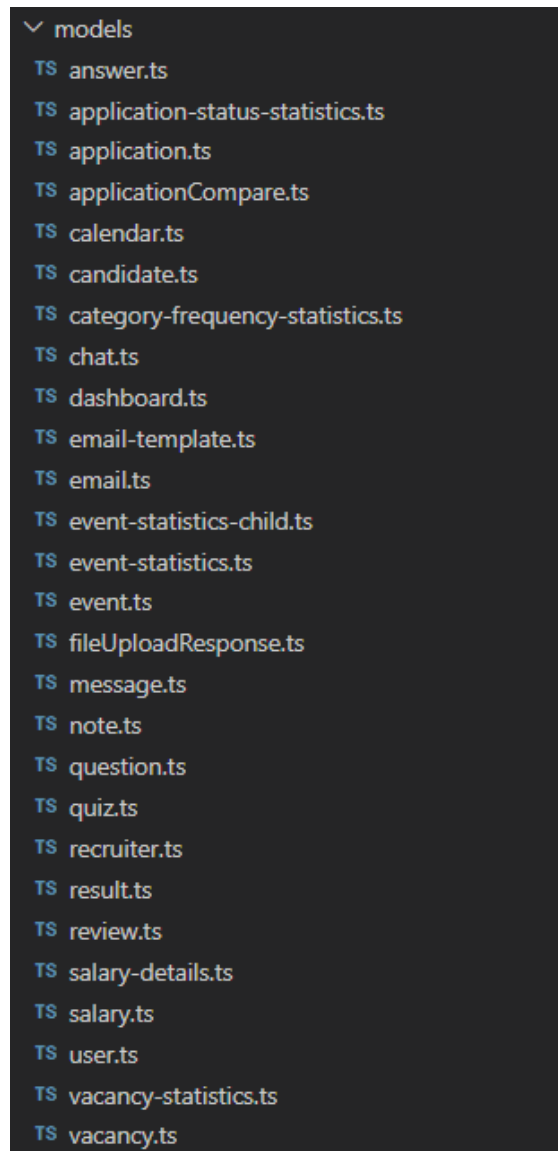


Рисунок 4.1 – Папка `models` (рисунок виконано самостійно)

Взаємодія з API відбувається через HTTPS запити. Кожна група ендпоінтів зібрана в окремому сервісі, які знаходяться в папці `services` (див. рис. 4.2). Така структура дозволяє організувати код більш зрозумілим та легким для підтримки. Кожен сервіс відповідає за певний набір функціональностей, що спрощує модульне

тестування та розширення системи.

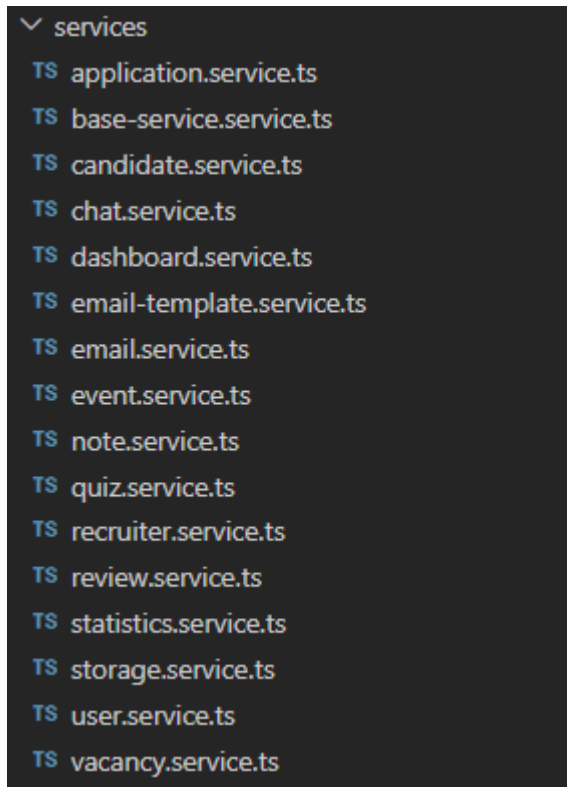


Рисунок 4.2 – Папка services (рисунок виконано самостійно)

Для забезпечення уніфікованого підходу до взаємодії з API, в системі реалізовано клас BaseService, який є базовим для всіх інших сервісів. Код сервісу наведено нижче:

```
export class BaseService {
  public getErrors(error: HttpResponse): string {
    let result: string = error.error.title;

    for (const ex of Object.values(error.error.errors)) {
      (ex as any[]).forEach((text) => (result += `\n${text}`));
    }

    return result;
  }
}
```

Кожен сервіс наслідує цей клас, що дозволяє уникнути дублювання коду та спрощує реалізацію загальних функцій, таких як обробка помилок. Це сприяє підвищенню якості коду та його повторного використання.

Робота з кожним з ендпоінтів реалізована за принципом REST API і

виконується за допомогою стандартних HTTP запитів GET, POST, PUT, DELETE, які забезпечують ефективну взаємодію між клієнтом і сервером. Приклад звернення до сервера:

```

    getRecruiterCategoryStatistic(id:number):Observable<CategoryFrequency
Statistics[]>{
    return
this.http.get<CategoryFrequencyStatistics[]>(`${this.baseUrl}/Recruiter/Cat
egory/${id}`).pipe(
    catchError((error) => {
        this.snackbar.open(`${error.error.errors == undefined
? error.message
: this.getErrors(error)}`, 'Close', {
            duration: 5000,
            horizontalPosition: 'right',
            verticalPosition: 'top',
        });
        return throwError(error);
    })
);
}

```

Дані з серверу приходять у вигляді JSON файла і конвертуються у масив моделі `CategoryFrequencyStatistics`, що дозволяє зручно обробляти та використовувати отримані дані у додатку. Приклад коду моделі:

```

export class CategoryFrequencyStatistics {
    department: JobDepartment;
    count: number;
}

```

Для організації навігації використовується App Routing модуль, який містить список доступних маршрутів, пов'язаних з компонентами, їх параметри та важливу інформацію про механізм аутентифікації. Продемонструємо групу маршрутів на прикладі чату:

```

{
  path: 'chat',
  canActivate: [AuthenticationGuard],
  data: { roles: [Role[Role.Recruiter], Role[Role.Candidate]] },
  children: [
    { path: '', component: ChatListComponent },
    { path: ':id', component: ChatComponent },
    { path: 'details/:id', component: ChatDetailsComponent }
  ]
}

```

```
  ]  
}
```

За аутентифікацію і авторизацію відповідає клас `AuthenticationGuard`, який реалізує інтерфейс `CanActivate`. Код класу аутентифікації наведено в додатку Г. За допомогою `JwtHelperService` з бібліотеки `@auth0/angular-jwt` ми перевіряємо актуальність токена. У випадку, якщо токен актуальний, перевіряємо, чи співпадає роль користувача, отримана з токена, з доступними для цього маршрута. Для роботи з ролями використано перелічення `Role`, код якого можна подивитись нижче:

```
export enum Role {  
  Admin,  
  Recruiter,  
  Candidate  
}
```

Кожен компонент має файл `TypeScript`, який містить логіку компонента, включаючи обробку подій, взаємодію з сервісами та обробку даних. `HTML`-файл містить шаблон компонента, що визначає його структуру та вигляд вмісту на сторінці. Файл `CSS` відповідає за стилізацію компонента, включаючи кольори, розміри, розташування елементів та інші властивості відображення. Така структура дозволяє зберігати логіку, вигляд та стилі компонента окремо, що спрощує розробку, тестування та підтримку.

### 4.3 Опис програмної системи

Знайомство користувача з додатком починається зі сторінки логіна, де йому пропонується ввести дані або створити аккаунт. На цій сторінці користувач може ввести свої облікові дані, такі як ім'я користувача та пароль, для входу в систему (див. рис. 4.3).

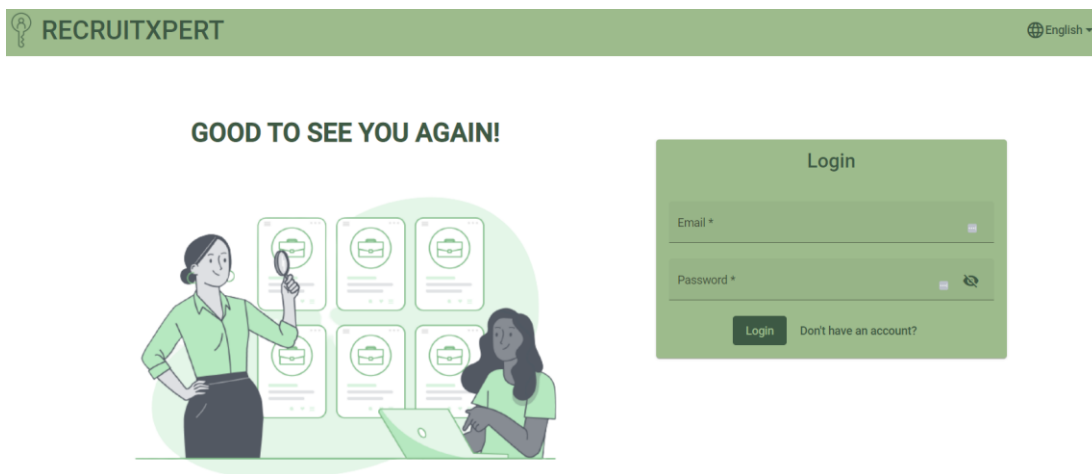


Рисунок 4.3 – Сторінка авторизації (рисунок виконано самостійно)

Якщо користувач ще не має облікового запису, він може скористатися функцією створення нового аккаунту (див. рис. 4.4), де йому буде запропоновано ввести необхідні дані для реєстрації в системі.

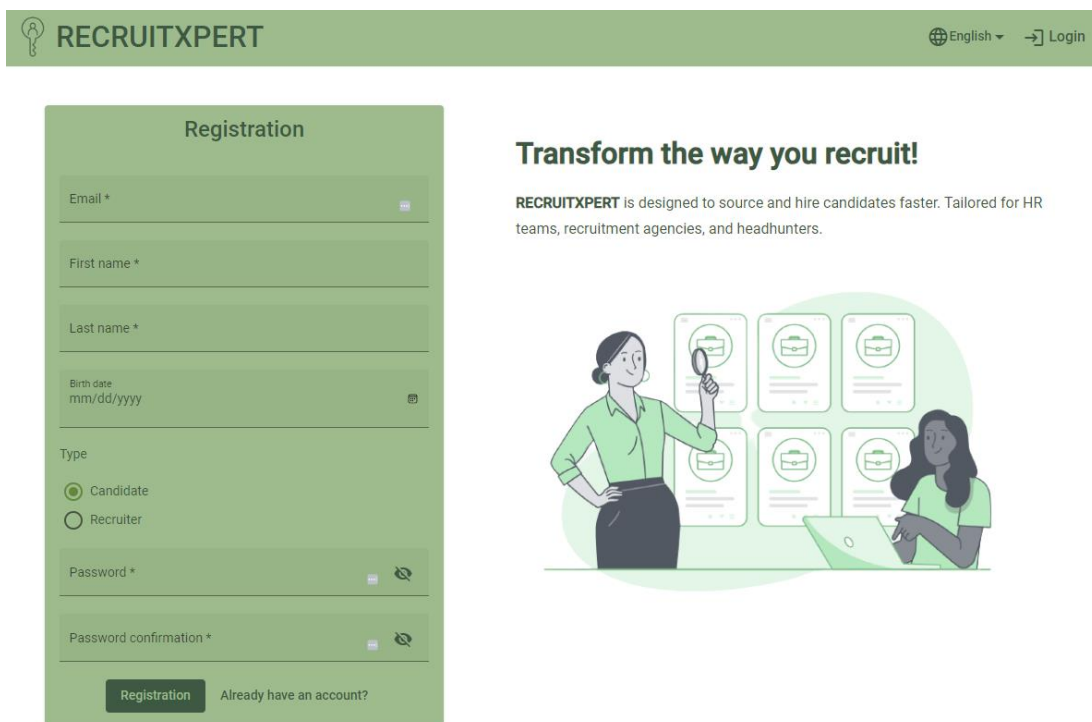


Рисунок 4.4 – Сторінка реєстрації (рисунок виконано самостійно)

Для створення аккаунта необхідні такі поля: електронна пошта, ім'я, прізвище, дата народження, роль, пароль, підтвердження пароля.

При успішній авторизації рекрутеру доступна сторінка списку резюме, де він

може скористатися фільтрами та взаємодіяти з резюме та їх кандидатами (див. рис. 4.5). На цій сторінці рекрутер може перейти до детального перегляду резюме, відправити електронний лист кандидатові, відкрити чат для подальшої комунікації та додати резюме на kanban-дошку для подальшого відстеження процесу найму.

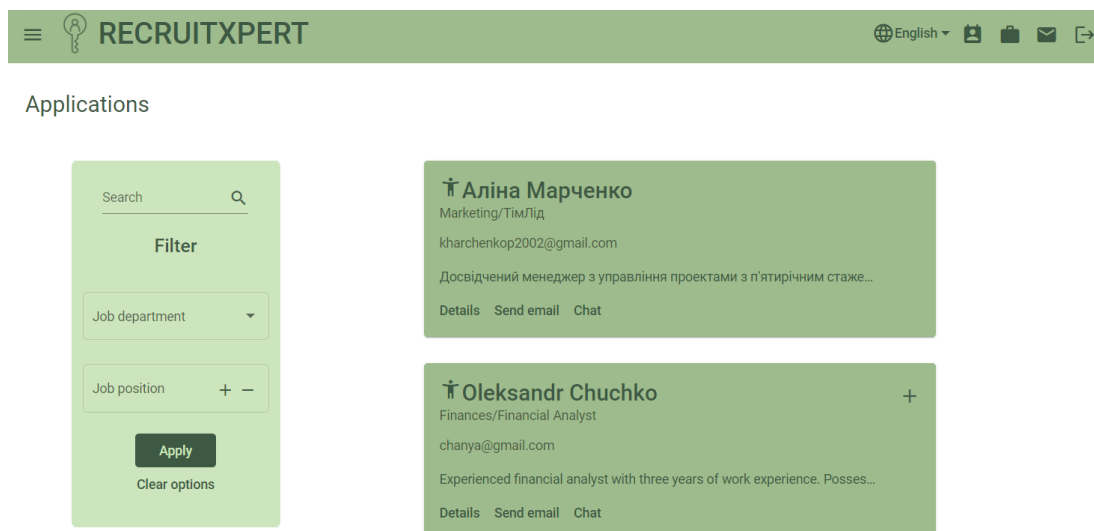


Рисунок 4.5 – Сторінка резюме (рисунок виконано самостійно)

Відкривши деталі резюме, рекрутер може побачити всю наявну інформацію про кандидата, включаючи особисті дані, досвід роботи, навички та інше (див. рис. 4.6).

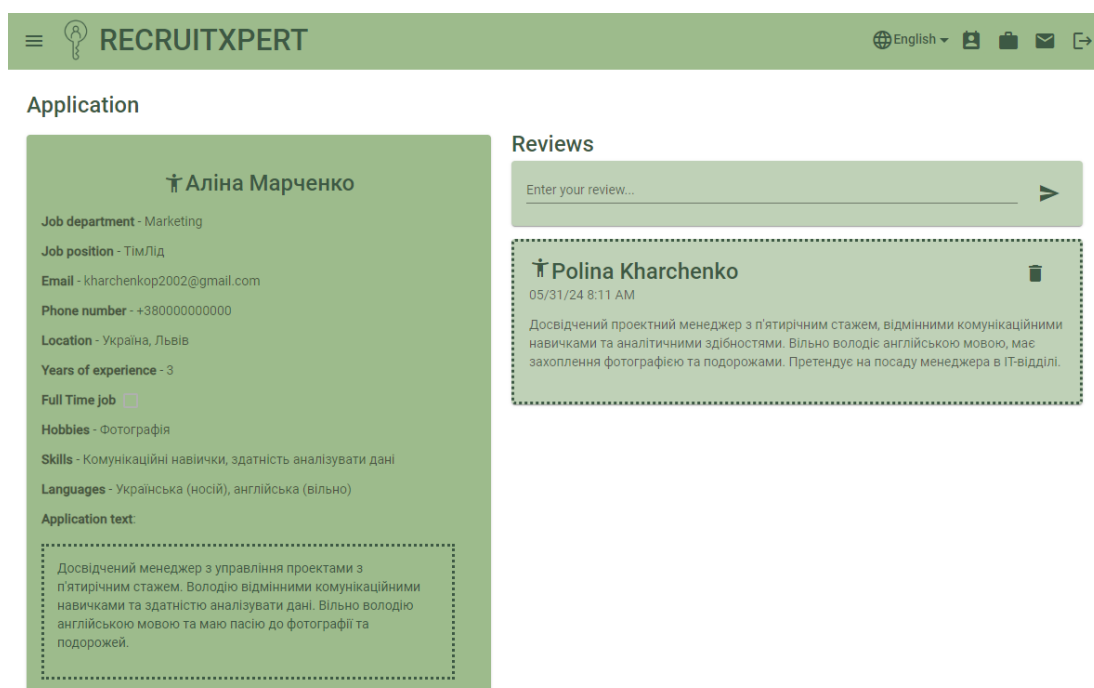


Рисунок 4.6 – Деталі резюме (рисунок виконано самостійно)

Також на цій сторінці рекрутер може залишити свій власний відгук про кандидата або переглянути ті, які залишили інші рекрутери. Це дозволяє створити комплексне уявлення про кандидата та його придатність для певної посади, а також забезпечує можливість обміну враженнями та співпраці між рекрутерами.

Усі користувачі мають доступ до списку вакансій з можливістю фільтрації та пошуку (див. рис. 4.7). Рекрутер може керувати своїми вакансіями, включаючи створення, редагування та видалення, тоді як інші користувачі можуть лише переглядати деталі вакансій.

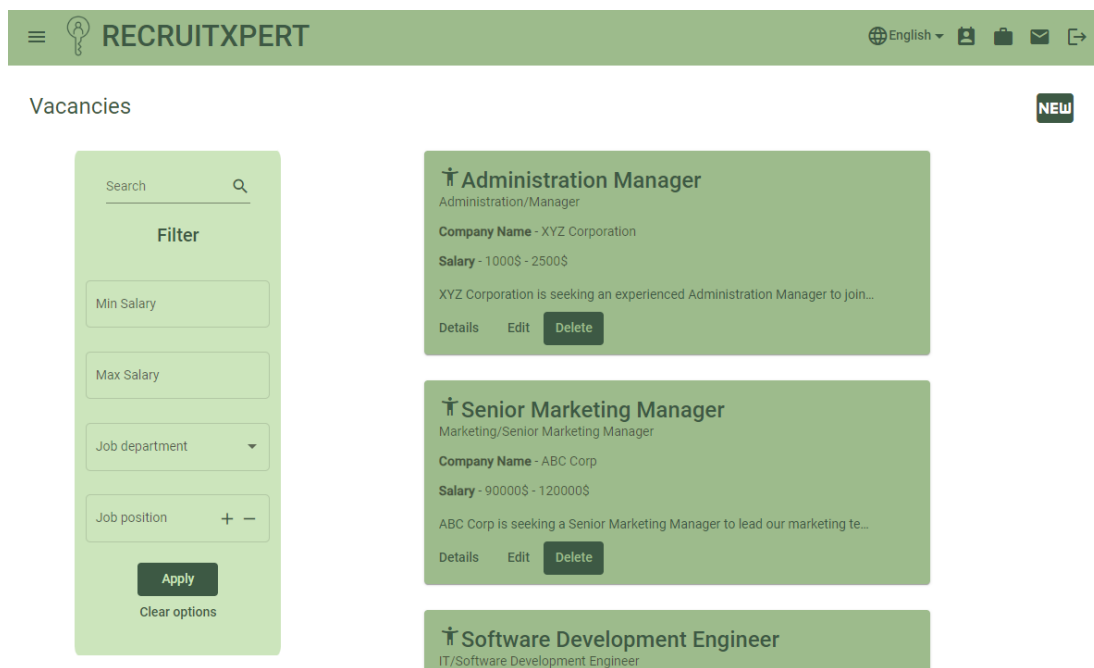


Рисунок 4.7 – Список вакансій (рисунок виконано самостійно)

Також рекрутеру доступна сторінка, на якій він може побачити лише вакансії, які створив сам (див. рис. 4.8). Це забезпечує зручність та ефективність у керуванні власними вакансіями без зайвого завантаження інших даних.



Рисунок 4.8 – Список вакансій, які належать рекрутеру (рисунок виконано самостійно)

Детальний перегляд вакансії відрізняється в залежності від ролі користувача. Рекрутер, якщо він є власником вакансії, бачить всі деталі вакансії та має доступ до елементів керування нею, таких як редагування, видалення, створення або перегляд тесту (див. рис. 4.9). Він також бачить список резюме, які подали на цю вакансію.

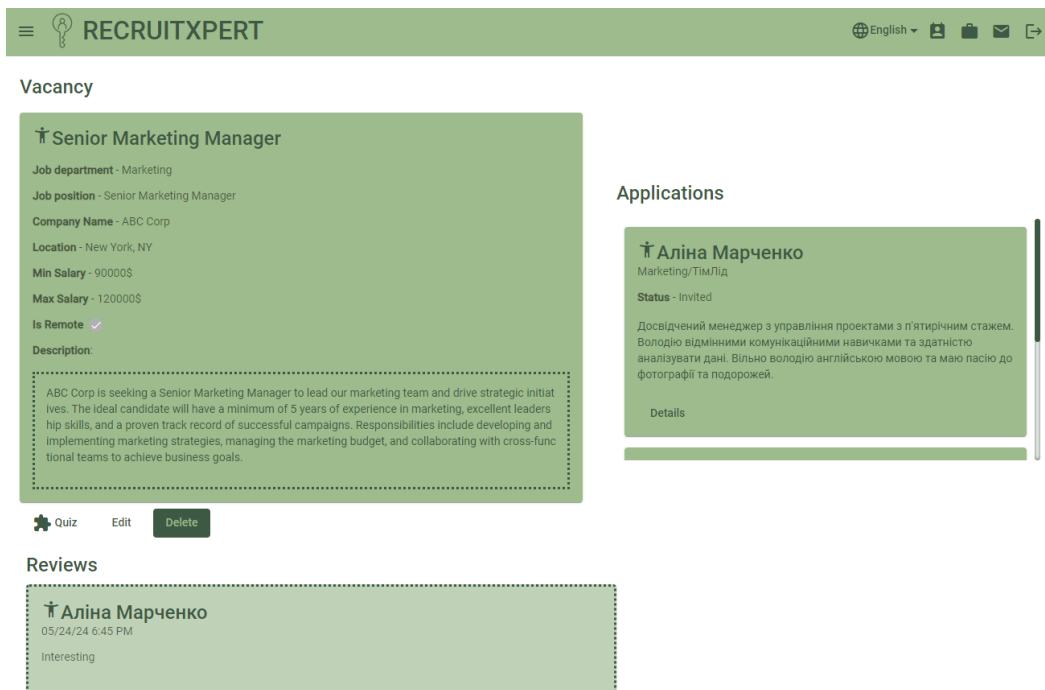


Рисунок 4.9 – Деталі вакансії в ролі рекрутера  
(рисунок виконано самостійно)

Усі користувачі можуть переглядати відгуки кандидатів про вакансію. Кандидати мають можливість опублікувати своє резюме до цієї вакансії, пройти тест або переглянути його результати (див. рис. 4.10).

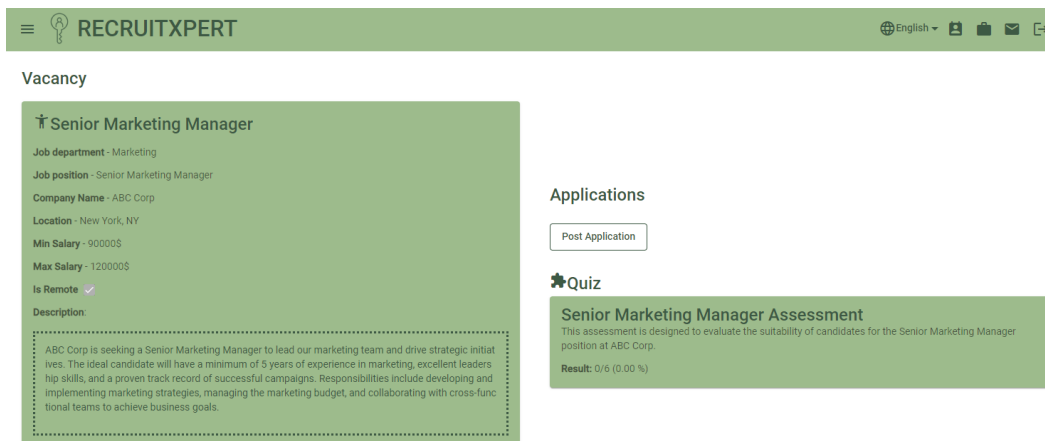


Рисунок 4.10 – Деталі вакансії в ролі кандидата  
(рисунок виконано самостійно)

Кандидати можуть подати своє резюме на вакансію, обравши з вже існуючих резюме або створивши нове (див. рис. 4.11).

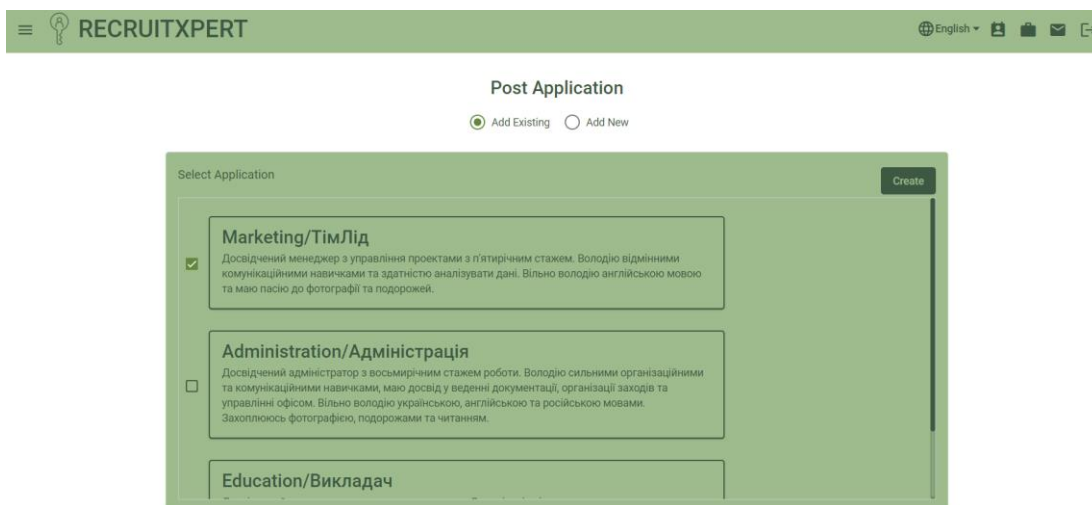


Рисунок 4.11 – Публікація резюме (рисунок виконано самостійно)

Кандидати можуть переглядати свої резюме, взаємодіяти з ними, а також бачити статистику, яка показує загальну кількість резюме та скільки з них знаходяться в кожному зі статусів (див. рис. 4.12).

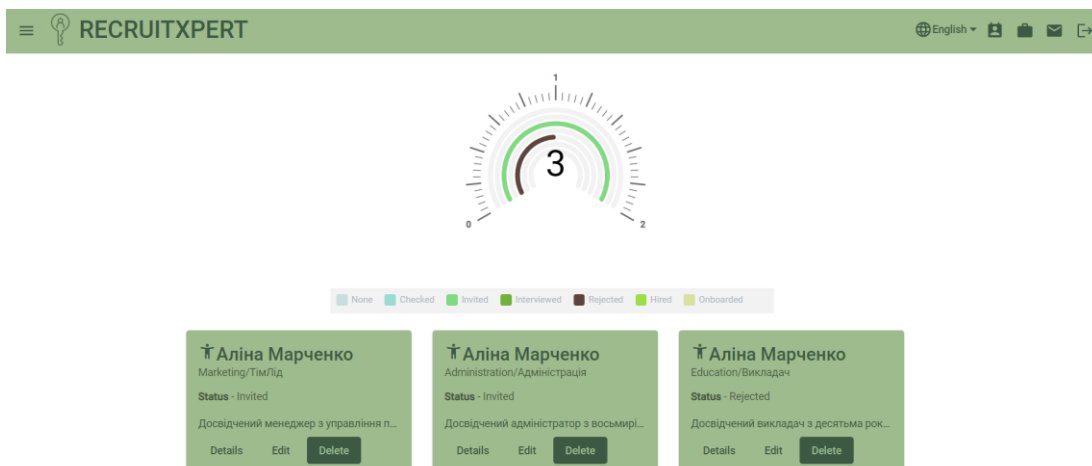
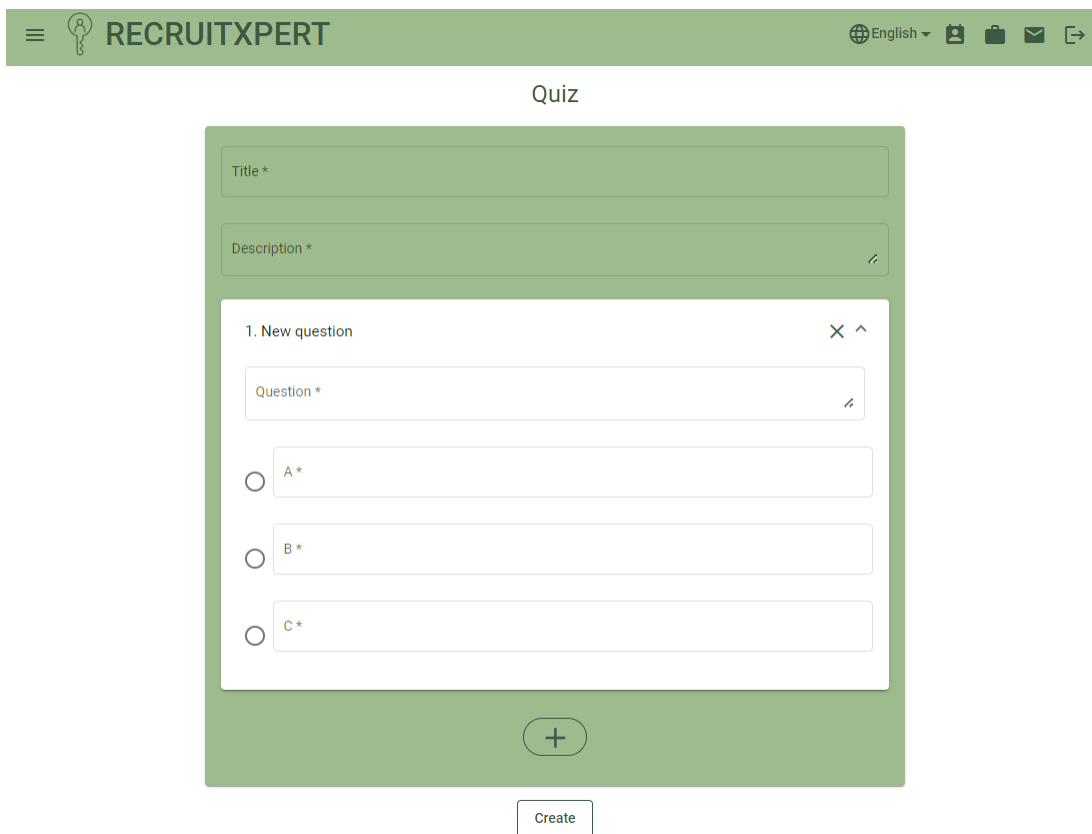


Рисунок 4.12 – Список резюме в ролі рекрутера (рисунок виконано самостійно)

Розглянемо функціональність чатів. Рекрутер може створювати та переглядати тести (див. рис. 4.13), тоді як кандидатам доступно проходити тести та переглядати результати (див. рис. 4.14).



The screenshot shows the RECRUITXPERT interface for creating a quiz. At the top, there is a navigation bar with the RECRUITXPERT logo and a language dropdown set to 'English'. Below the navigation bar, the word 'Quiz' is centered. The main content area is a form with the following elements:

- A text input field labeled 'Title \*'.
- A text input field labeled 'Description \*' with a small icon on the right.
- A section titled '1. New question' with a close button (X) and an expand/collapse arrow (^).
- Inside this section, a text input field labeled 'Question \*' with a small icon on the right.
- Below the question field, three radio button options: 'A \*', 'B \*', and 'C \*', each with its own text input field.
- A circular button with a '+' sign is located below the options.
- A 'Create' button is positioned at the bottom center of the form.

Рисунок 4.13 – Створення теста (рисунок виконано самостійно)



The screenshot shows the RECRUITXPERT interface displaying the results of a quiz. At the top, there is a navigation bar with the RECRUITXPERT logo and a language dropdown set to 'English'. Below the navigation bar, the word 'Quiz' is centered. The main content area is a green box with the following elements:

- Senior Marketing Manager Assessment**
- This assessment is designed to evaluate the suitability of candidates for the Senior Marketing Manager position at ABC Corp.
- 1. What is the minimum required experience for the Senior Marketing Manager position? \***
- 3 years
- 5 years
- 7 years
- 10 years
- 2. Which skills are important for the Senior Marketing Manager role? \***
- Leadership skills
- Basic accounting
- Strategic planning
- Graphic design
- 3. What is the minimum salary for this position? \***
- \$70,000

Рисунок 4.14 – Проходження теста (рисунок виконано самостійно)

Для зручної візуалізації роботи рекрутера було створено канбан-дошку з резюме, які знаходяться в роботі і мають різні статуси (див. рис. 4.15).

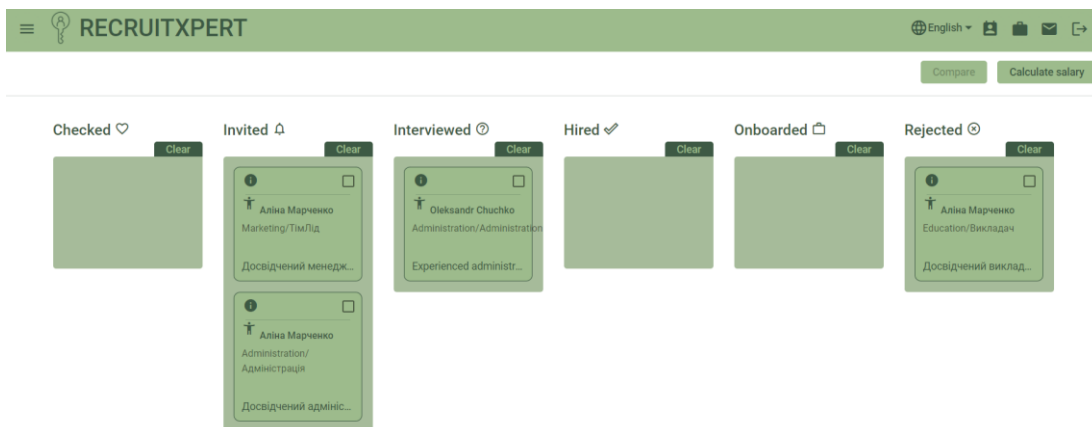


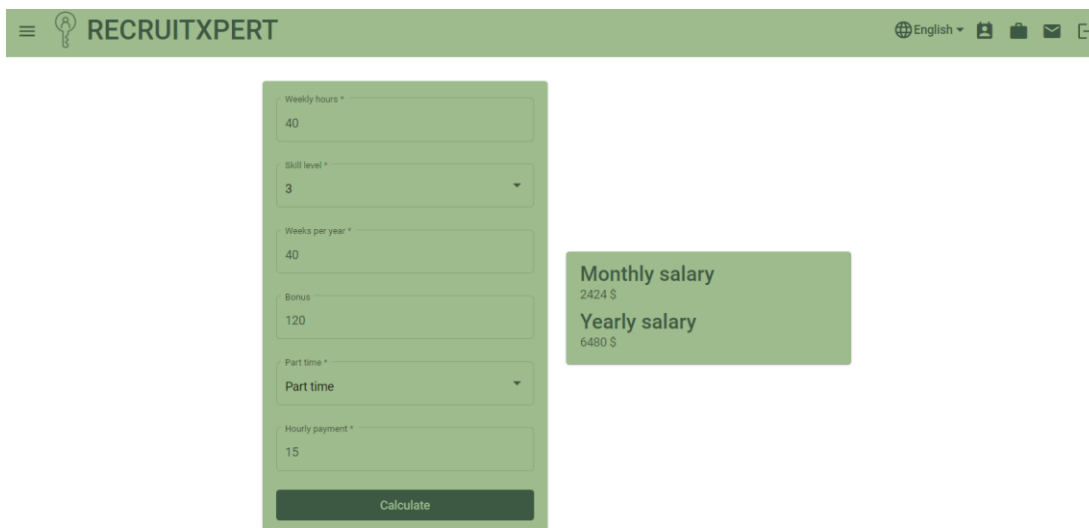
Рисунок 4.15 – Kanban-дошка (рисунок виконано самостійно)

Оскільки основна робота рекрутера відбувається на сторінці з kanban-дошкою, для зручності було додано можливість порівняння резюме (див. рис. 4.16) та швидкий розрахунок зарплатні (див. рис. 4.17).



Рисунок 4.16 – Порівняння резюме (рисунок виконано самостійно)

Ці елементи дозволяють рекрутеру мати орієнтири під час інтерв'ю з кандидатом і робить процес відбору більш ефективним.



RECRUITXPERT

English

Weekly hours \*  
40

Skill level \*  
3

Weeks per year \*  
40

Bonus  
120

Part time \*  
Part time

Hourly payment \*  
15

Calculate

Monthly salary  
2424 \$

Yearly salary  
6480 \$

Рисунок 4.17 – Розрахунок зарплатні (рисунок виконано самостійно)

Для комунікації користувачі мають такі елементи, як чат (див. рис. 4.18) і можливість відправити електронний лист зсередини додатку. Рекрутери можуть створювати необхідні шаблони електронних листів для зручної відправки інформації (див. рис. 4.19).



Рисунок 4.18 – Вбудований чат (рисунок виконано самостійно)



Рисунок 4.19 – Інтеграція додатка з електронною поштою  
(рисунок виконано самостійно)

Для рекрутера було розроблено календар, який дозволяє планувати зустрічі та назначати посилання на платформу для онлайн зустрічей, такі як відеоконференції (див. рис. 4.20). Цей календар пропонує три різних режими перегляду, щоб рекрутер міг вибрати оптимальний для себе: режим місяця, де відображаються всі заплановані події протягом місяця, режим тижня, який показує розклад на тиждень, та режим дня, який детально відображає розклад на конкретний день.

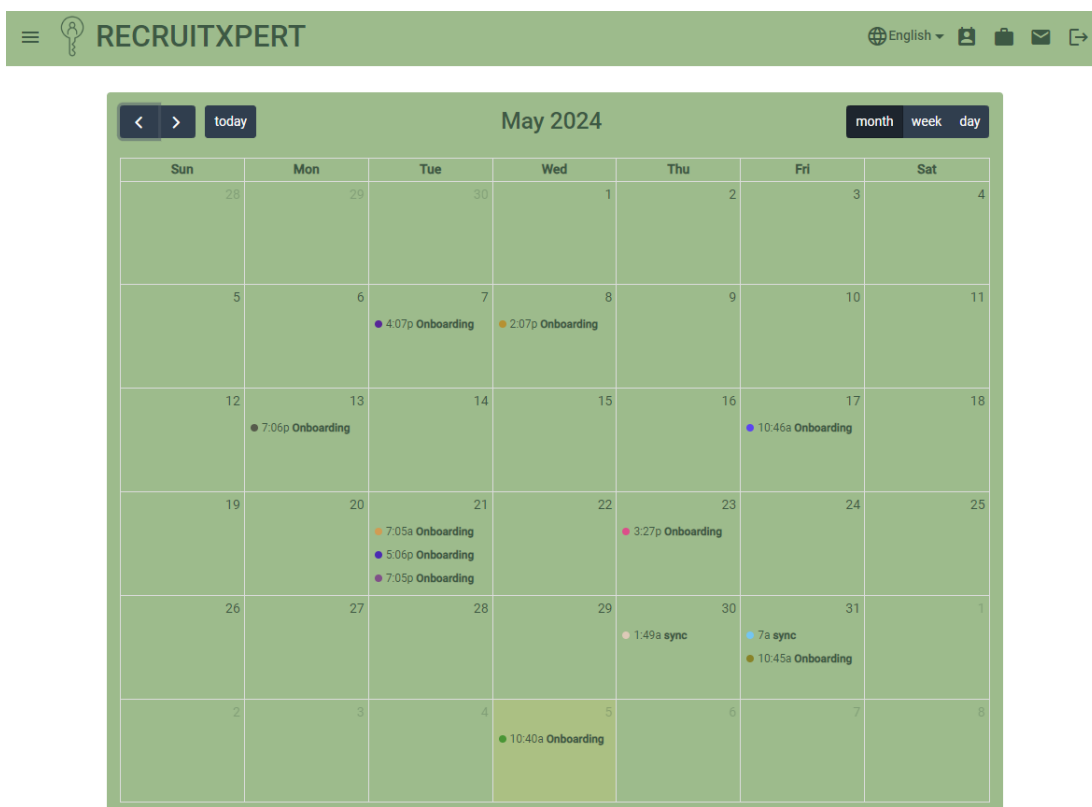


Рисунок 4.20 – Інтеграція календаря (рисунок виконано самостійно)

Сторінка статистики розділена на три секції з діаграмами для аналізу різних аспектів роботи (див. рис. 4.21).

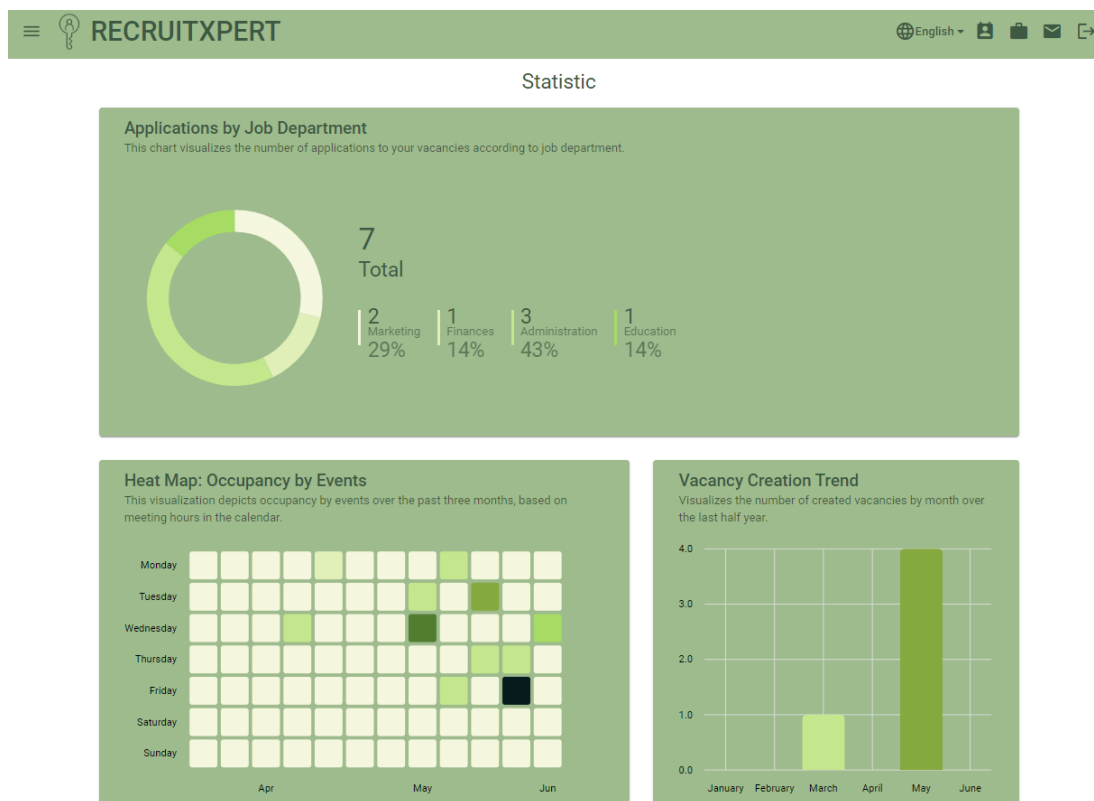


Рисунок 4.21 – Статистика доступна рекрутеру  
(рисунок виконано самостійно)

Перша діаграма «Резюме за відділами» візуалізує кількість резюме на ваші вакансії за відділами. Друга – теплова карта «Зайнятість за подіями» відображає зайнятість за подіями протягом останніх трьох місяців, на основі годин зустрічей у календарі. Третя діаграма, «Тенденція створення вакансій», візуалізує кількість створених вакансій за місяць протягом останніх півроку.

В будь-який момент користування додатком користувач може змінити мову з англійської на українську та навпаки. Функція зміни мови знаходиться у навігаційній панелі (див. рис. 4.22).



Рисунок 4.22 – Випадаючий список зміни мови  
(рисунок виконано самостійно)

Під час зміни мови всі шаблонні слова і речення автоматично перекладаються, але текст, введений користувачами, залишається незмінним (див. рис. 4.23). Така функціональність забезпечує користувачам зручність та стабільність у роботі з додатком, незалежно від обраної мови.

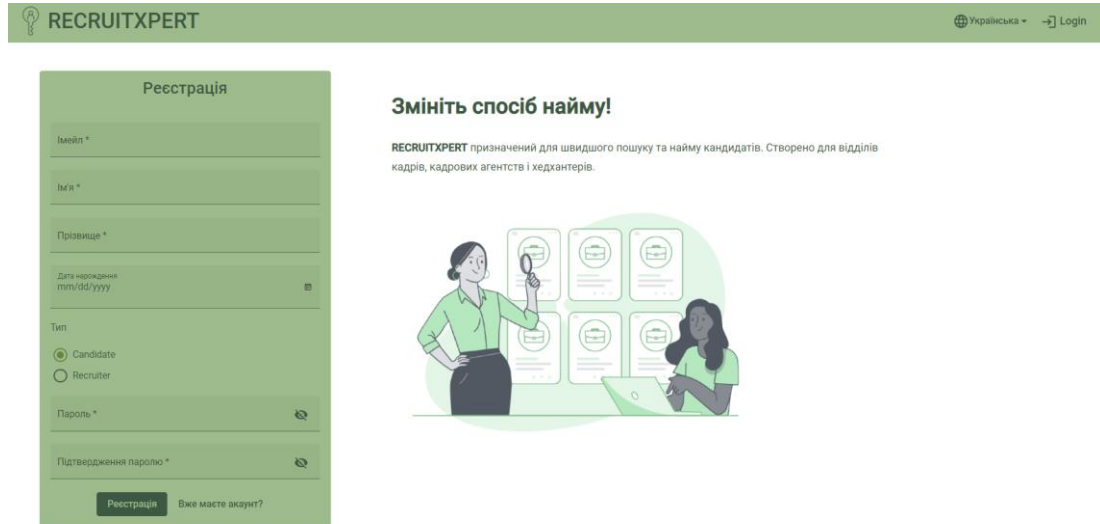


Рисунок 4.23 – Переклад українсько мовою  
(рисунок виконано самостійно)

В ході розробки було створено зручний додаток, який спрямований на автоматизацію процесів рекрутингу та полегшення взаємодії між рекрутерами та кандидатами. Додаток реалізує основні функціональні вимоги, такі як управління резюме та вакансіями, планування зустрічей, спілкування через чат та електронні листи, а також надає зручний інтерфейс для аналізу даних. Особлива увага приділяється забезпеченню надійності та коректності введених даних, тому у додатку вбудована валідація полів та обробка помилок. Крім того, для полегшення використання додатку, було додано можливість перекладу інтерфейсу з однієї мови на іншу за бажанням користувача. Не менш важливим аспектом є наявність елемента завантаження, який активується у випадку повільного відгуку програми, що забезпечує комфортну роботу користувача та покращує загальний досвід використання додатку.

## 5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування – це процес перевірки програмного забезпечення з метою виявлення та виправлення помилок, а також забезпечення його відповідності вимогам та специфікаціям. Тестування є невід'ємною частиною процесу розробки, оскільки дозволяє виявити та виправити дефекти на ранніх етапах, що сприяє підвищенню якості кінцевого продукту.

Для клієнтської частини додатка тестування особливо важливе, оскільки воно гарантує, що користувацький інтерфейс функціонує правильно та забезпечує позитивний досвід для користувачів. Це включає перевірку всіх елементів інтерфейсу, взаємодій, відображення даних, а також коректної роботи різних функцій та логіки.

Ми будемо здійснювати тестування за допомогою тест-кейсів, набору умов або змінних, за якими тестувальник визначає, чи працює додаток правильно та відповідає вимогам. Тест-кейс включає опис передумов, кроків виконання, очікуваних результатів і фактичних результатів, які наведено в таблиці 5.1.

Таблиця 5.1 – Тест-кейс №1 (таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:	Тест-кейс №1		
Опис функції:	Публікація резюме		
Власник тесту:	Харченко Поліна Дмитрівна		
Дата створення:	30.05.2024		
Мета тесту:	Перевірити коректність реалізації публікації резюме до вакансії (створення нового і додавання вже наявного)		
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити веб-застосунок	Користувач має доступ до сайту, який відкритий	Пройдено
2	Користувач має обліковий запис та хоче авторизуватися	Перенаправлено на сторінку профілю	Пройдено
3	В системі створено вакансію	Вакансія доступна	Пройдено
Створення і публікація нового резюме			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити сторінку вакансії	Перенаправлено на сторінку вакансії	Пройдено

Продовження таблиці 5.1

Створення і публікація нового резюме			
№	Опис випадку	Очікуваний результат	Висновок
2	Натиснути кнопку «Додати своє резюме»	Перенаправлено на сторінку створення резюме, вкладку «Додати нове»	Пройдено
3	Заповнити обов'язкове поле «Посада»	Обов'язкове поле було заповнено	Пройдено
4	Обрати «Відділ»	Було обрано відділ	Пройдено
5	Заповнити необов'язкове поле «Місцезнаходження»	Необов'язкове поле було заповнено	Пройдено
6	Заповнити обов'язкове поле «Імейл»	Обов'язкове поле було заповнено, пройдено валідацію електронної пошти	Пройдено
7	Заповнити обов'язкове поле «Номер телефону»	Обов'язкове поле було заповнено, пройдено валідацію номера телефону	Пройдено
8	Заповнити обов'язкове поле «Досвід (кількість років)»	Обов'язкове поле було заповнено	Пройдено
9	Обрати «Повний робочий день»	Прапорець поставлено	Пройдено
10	Заповнити обов'язкове поле «Навички»	Обов'язкове поле було заповнено	Пройдено
11	Заповнити обов'язкове поле «Знання мов»	Обов'язкове поле було заповнено	Пройдено
12	Заповнити обов'язкове поле «Хобі»	Обов'язкове поле було заповнено	Пройдено
13	Заповнити обов'язкове поле «Мінімальна заробітна плата»	Обов'язкове поле було заповнено	Пройдено
14	Заповнити обов'язкове поле «Максимальна заробітна плата», меншим значенням ніж «Мінімальна заробітна плата»	Відображається повідомлення про некоректне заповнення поля	Пройдено
15	Заповнити обов'язкове поле «Максимальна заробітна плата» правильними даними	Обов'язкове поле було заповнено, повідомлення про помилку зникло	Пройдено
16	Заповнити обов'язкове поле «Текст резюме»	Обов'язкове поле було заповнено	Пройдено
17	Натиснути кнопку «Створити»	Резюме успішно створено	Пройдено
18	Користувача перенаправлено на сторінку деталей резюме	Резюме має вказані дані і посилається на правильну вакансію	Пройдено

Кінець таблиці 5.1

Публікація вже наявного резюме на вакансію			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити сторінку вакансії	Перенаправлено на сторінку вакансії	Пройдено
2	Натиснути кнопку «Додати своє резюме»	Перенаправлено на сторінку створення резюме, вкладку «Додати нове»	Пройдено
3	Обрати вкладку «Додати існуюче»	Відкриття списку наявних резюме	Пройдено
4	Обрати резюме	Обране резюме відмічено прапорцем	Пройдено
5	Обрати інше резюме	Прапорець перенесено на обране резюме	Пройдено
6	Натиснути кнопку «Створити»	Резюме успішно опубліковано	Пройдено
7	Користувача перенаправлено на детальний перегляд обраного резюме	Список вакансій оновлено	Пройдено
Результати тестування			
Тестувальник: Харченко П. Д.		Дата прогону тесту: 01.06.2024	Результат тесту (P/F/V): ПРОЙДЕНО (P)

Ми провели тест-кейс тестування з метою перевірки коректності реалізації публікації резюме до вакансії, як для створення нового резюме, так і для додавання вже наявного. Успішно виконавши всі етапи тестування, можна зробити висновок, що функціональність публікації резюме до вакансії працює коректно. Додаток забезпечує правильне створення нових резюме та дозволяє без помилок додавати вже існуючі резюме до відповідних вакансій. Це свідчить про те, що розроблена функція відповідає вимогам та специфікаціям, забезпечуючи користувачам надійний інструмент для управління резюме.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було створено веб-додаток, який інтегрується з серверною частиною за допомогою REST API. Клієнтська сторона програми була реалізована з використанням найсучасніших технологій, таких як Angular, HTML/CSS, а також використанням Angular Material і Bootstrap для забезпечення гарного інтерфейсу користувача. Для керування версіями, автоматичної поставки, розгортання і багатьох інших аспектів розробки було використано Azure DevOps інструменти.

Розроблена програмна система є надійним інструментом для автоматизації процесу найму, що відкриває перед користувачами широкі можливості для ефективного керування вакансіями та кандидатами. Вона об'єднує в собі різноманітні функції, такі як kanban-дошка, календар, система тестування, вбудований чат та інтеграцію з електронною поштою, що спрощує процес планування, взаємодії та спілкування між рекрутерами та кандидатами. Завдяки цим інструментам, рекрутери можуть ефективно вести переговори з кандидатами, проводити тести для оцінки їхніх навичок, а також управляти всіма аспектами найму, що робить цю систему незамінною для рекрутингових агентств та великих компаній.

У результаті було створено простий у використанні та ефективний інструмент, який є конкурентоспроможним на ринку завдяки своїй широкій функціональності та зручному інтерфейсу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Glassdoor: Working at Panda Express [Електронний ресурс] – URL: [https://www.glassdoor.com/Overview/Working-at-Panda-Express-EI\\_IE5659.11,24.htm](https://www.glassdoor.com/Overview/Working-at-Panda-Express-EI_IE5659.11,24.htm) (дата звернення: 12.04.2024).
2. Work.ua – сайт пошуку роботи [Електронний ресурс] – URL: <https://www.work.ua/> (дата звернення: 12.04.2024).
3. HackerRank [Електронний ресурс] – URL: <https://www.hackerrank.com/> (дата звернення: 13.04.2024).
4. Codility: Online Coding Tests & Technical Interviews [Електронний ресурс] – URL: <https://www.codility.com/> (дата звернення: 13.04.2024).
5. Jira Templates: Simple Recruitment Template [Електронний ресурс] – URL: <https://www.atlassian.com/software/jira/templates/recruitment> (дата звернення: 13.04.2024).
6. Сазерленд Дж. Scrum. Навчись робити вдвічі більше за менший час / пер. з англ. Я. Лебедецько. КСД, 2022. 280 с.
7. Seshadri S. Angular: up and running: learning angular, step by step. O'Reilly Media, 2018. 312 с.

## ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:  
Олійник Олена Володимирівна каф. ПІ

ID перевірки:  
1016334181

Дата перевірки:  
08.06.2024 06:22:09 EEST

Тип перевірки:  
Doc vs Library

Дата звіту:  
08.06.2024 06:25:59 EEST

ID користувача:  
100012353

Назва документа: 2024\_Б\_ПІ\_ПЗПІ-20-4\_Харченко\_П\_Д

Кількість сторінок: 46 Кількість слів: 6827 Кількість символів: 53087 Розмір файлу: 2.33 MB ID файлу: 1016134591

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

2.72%

**Схожість**

Найбільша схожість: 0.97% з джерелом з Бібліотеки (ID файлу: 1015174616)

Пошук збігів з Інтернетом не проводився

2.72% Джерела з Бібліотеки

126

Сторінка 48

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%

**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

12  
сторінок

Рисунок А.1 – Результат перевірки кваліфікаційної роботи бакалавра на плагіат

ДОДАТОК Б  
Слайди презентації



# ПРОГРАМНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ОНЛАЙН-РЕКРУТИНГУ. FRONT-END

Харківський національний університет радіоелектроніки  
Кваліфікаційна робота бакалавра

Виконала  
Науковий керівник

ст. гр. ПЗПІ-20-4 Харченко П.Д.  
доц. каф. ПІ Побіженко І.О.

Рисунок Б.1 – Слайд 1



## МЕТА РОБОТИ

Аналіз предметної галузі	01
Проектування	02
Розробка клієнтської частини	03
Тестування	04

Рисунок Б.2 – Слайд 2



Рисунок Б.3 – Слайд 3

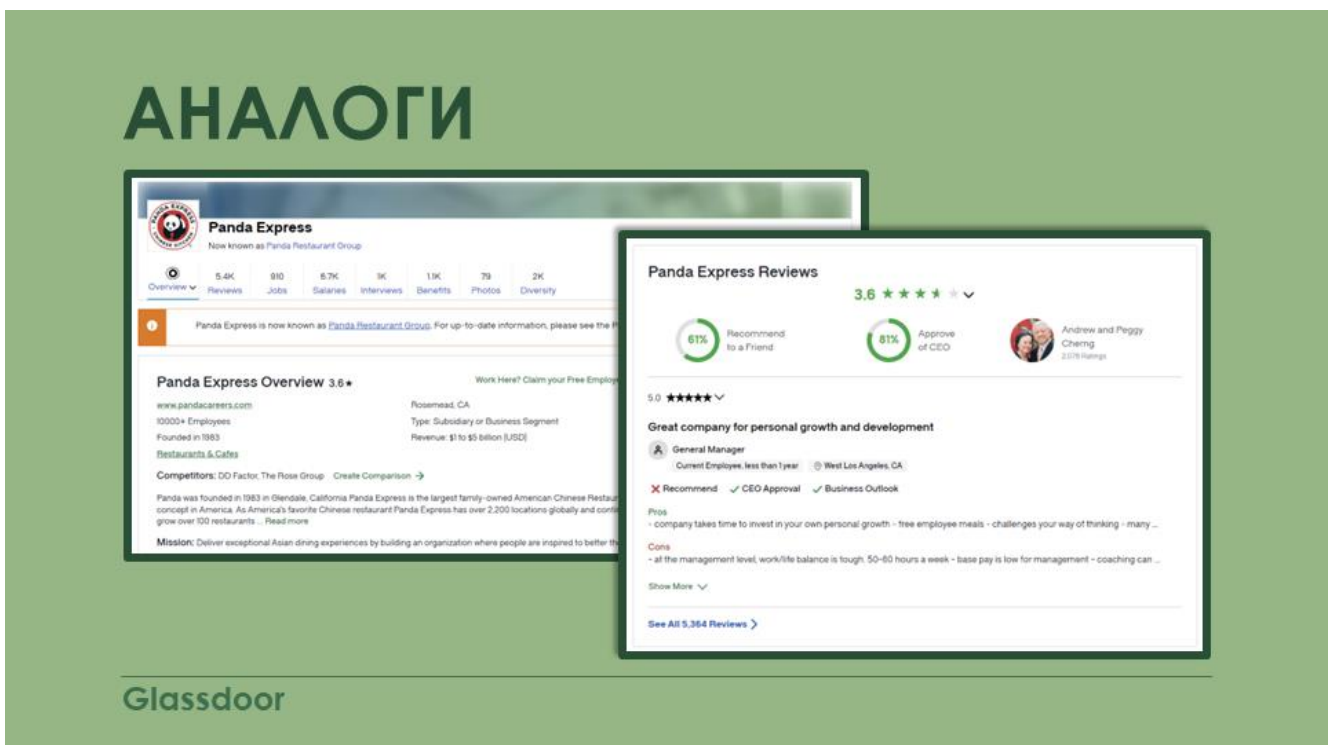
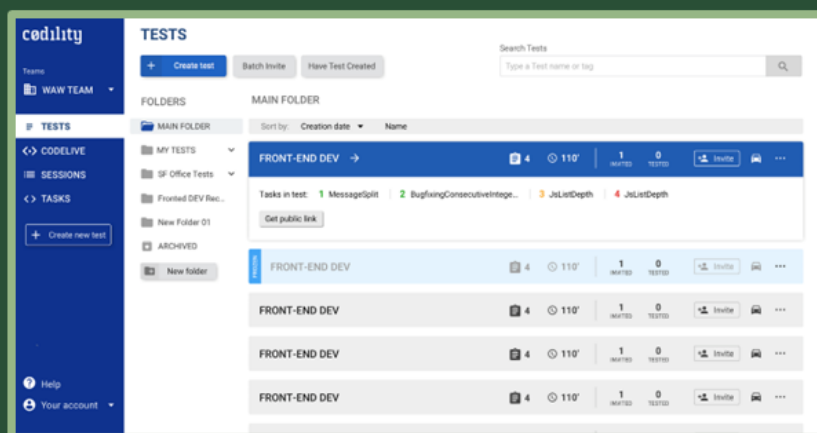


Рисунок Б.4 – Слайд 4

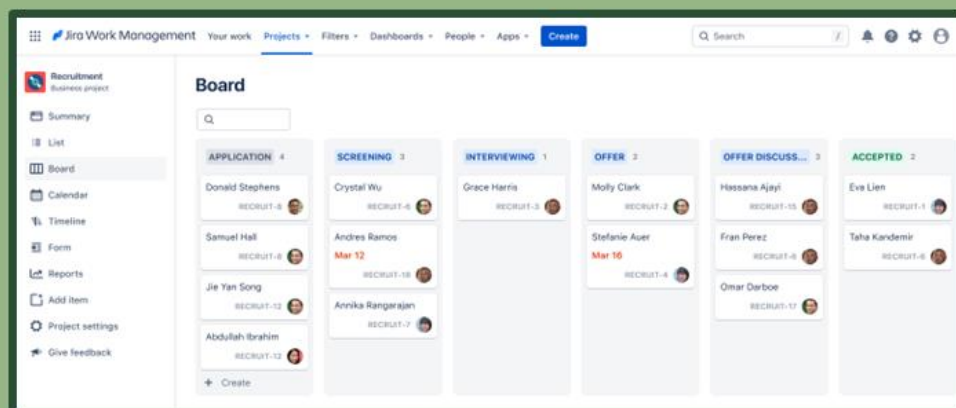
# АНАЛОГИ



Codility

Рисунок Б.5 – Слайд 5

# АНАЛОГИ



JIRA

Рисунок Б.6 – Слайд 6

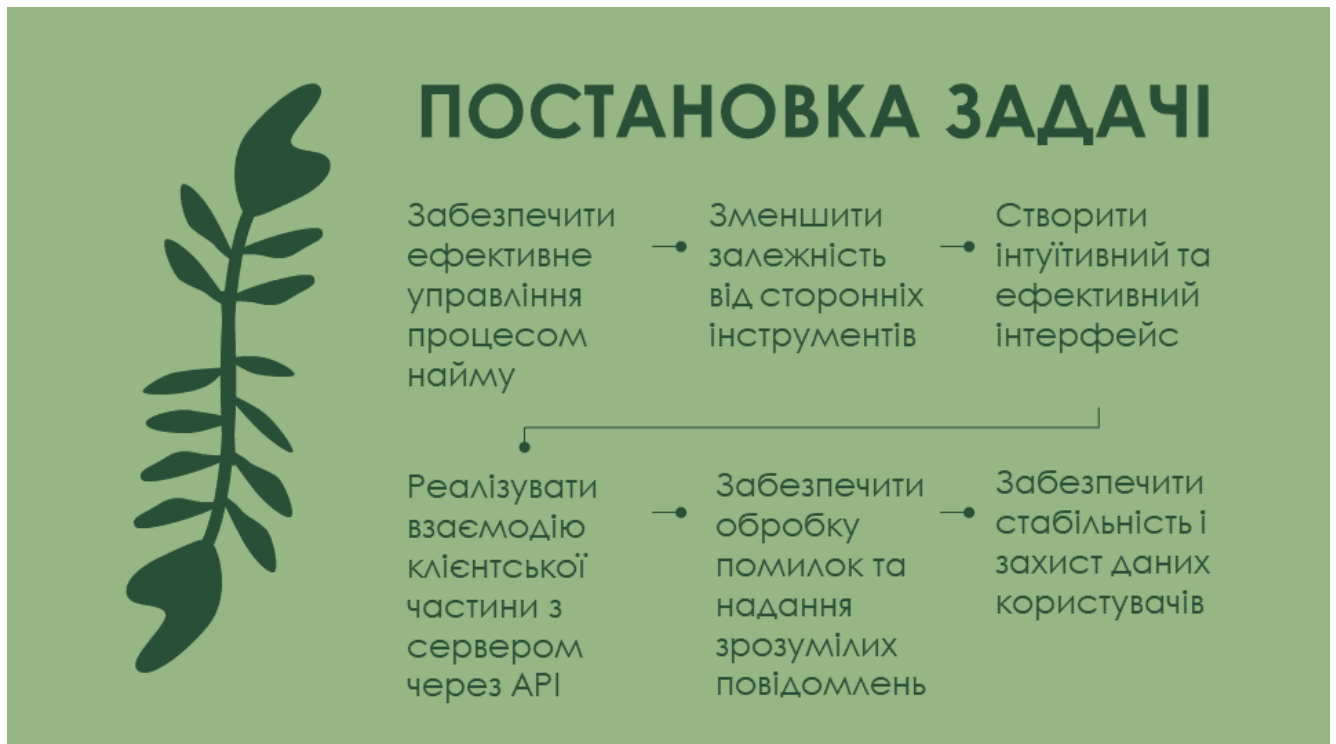


Рисунок Б.7 – Слайд 7

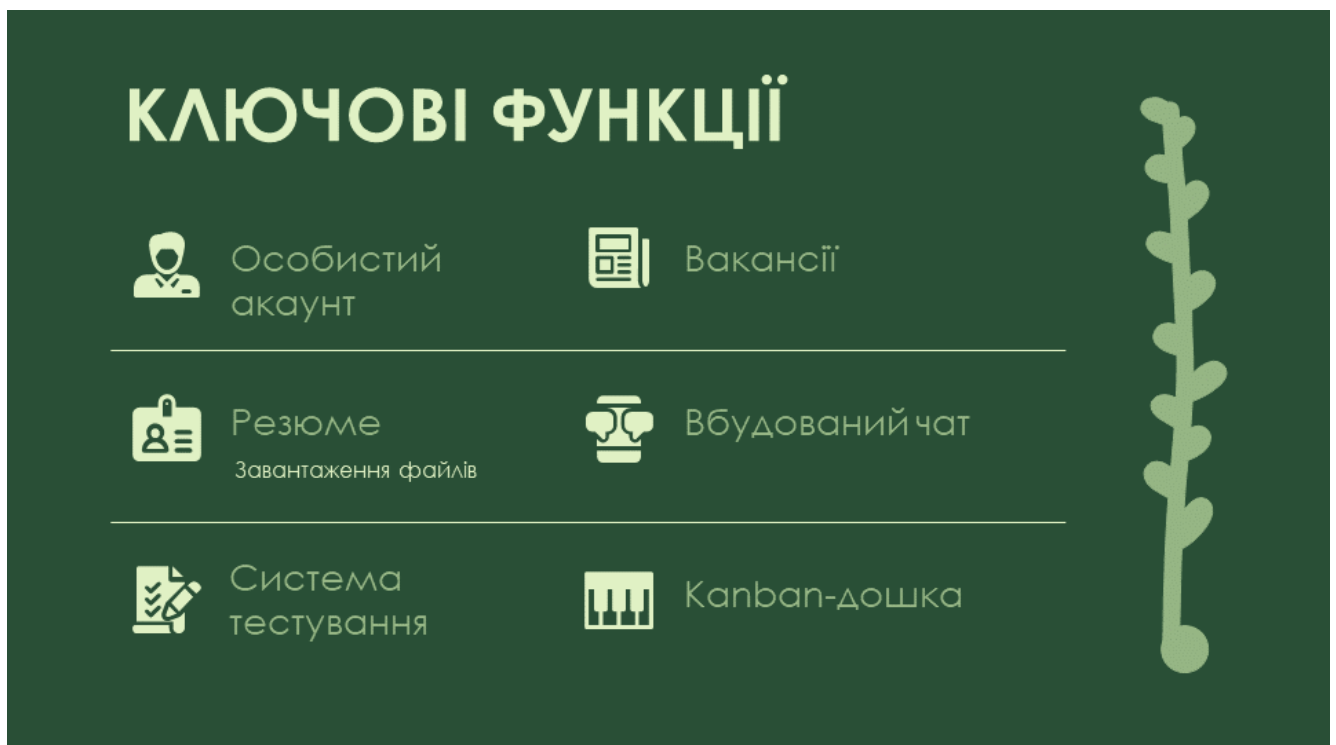


Рисунок Б.8 – Слайд 8

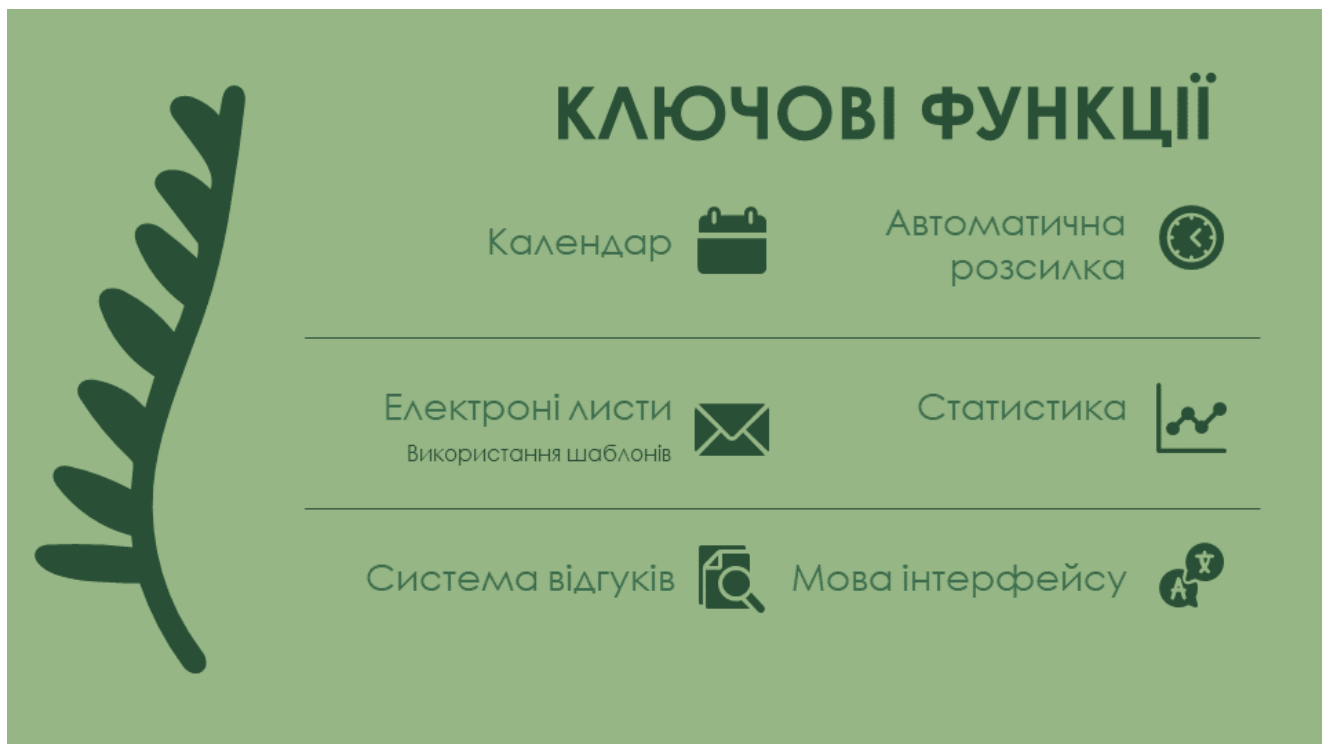


Рисунок Б.9 – Слайд 9



Рисунок Б.10 – Слайд 10



Рисунок Б.11 – Слайд 11

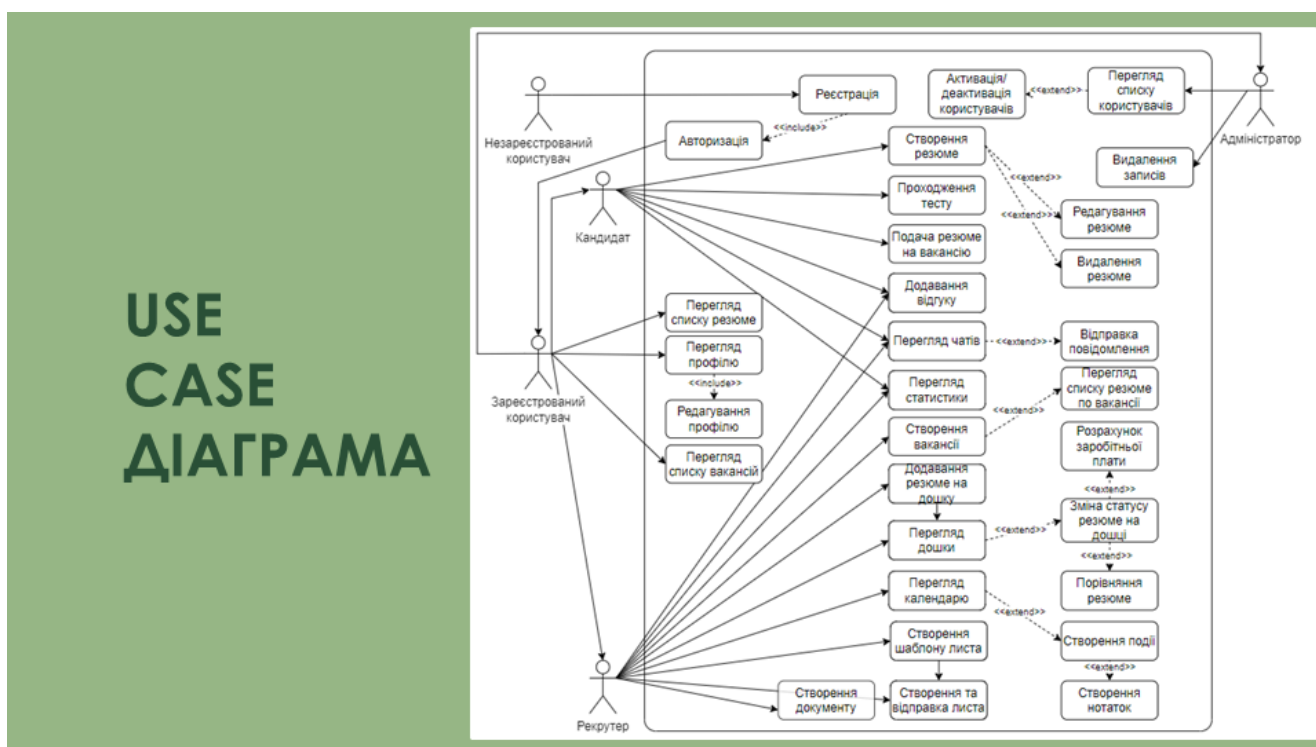


Рисунок Б.12 – Слайд 12

# ДІАГРАМА СТАНІВ

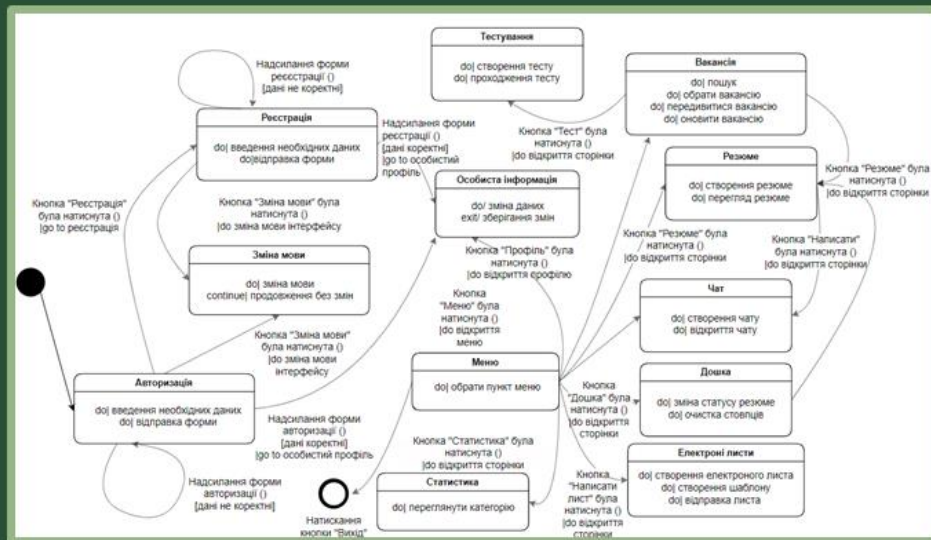


Рисунок Б.13 – Слайд 13

# ДІАГРАМА ДІЯЛЬНОСТІ РЕКРУТЕРА

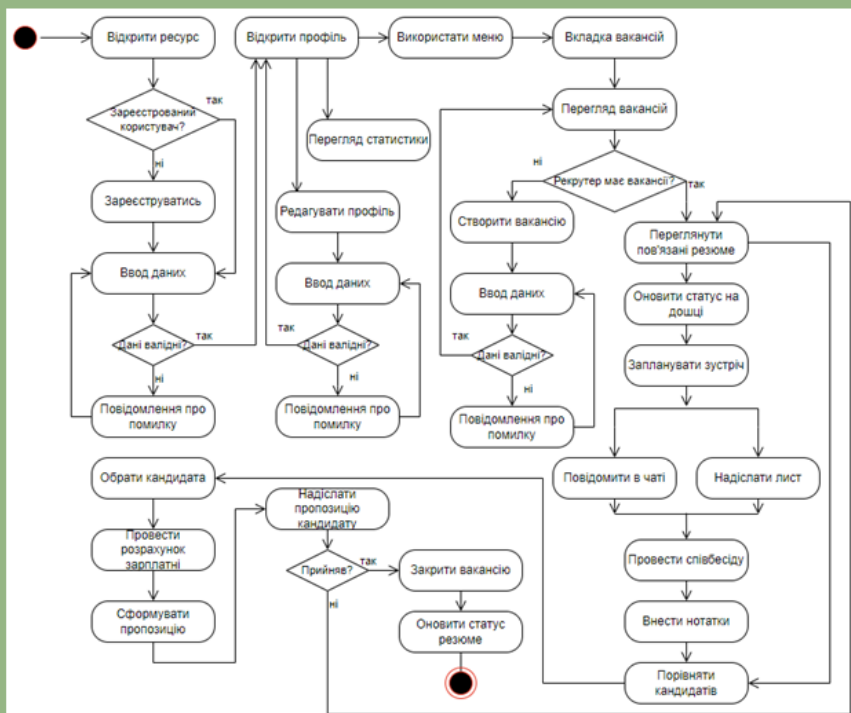


Рисунок Б.14 – Слайд 14

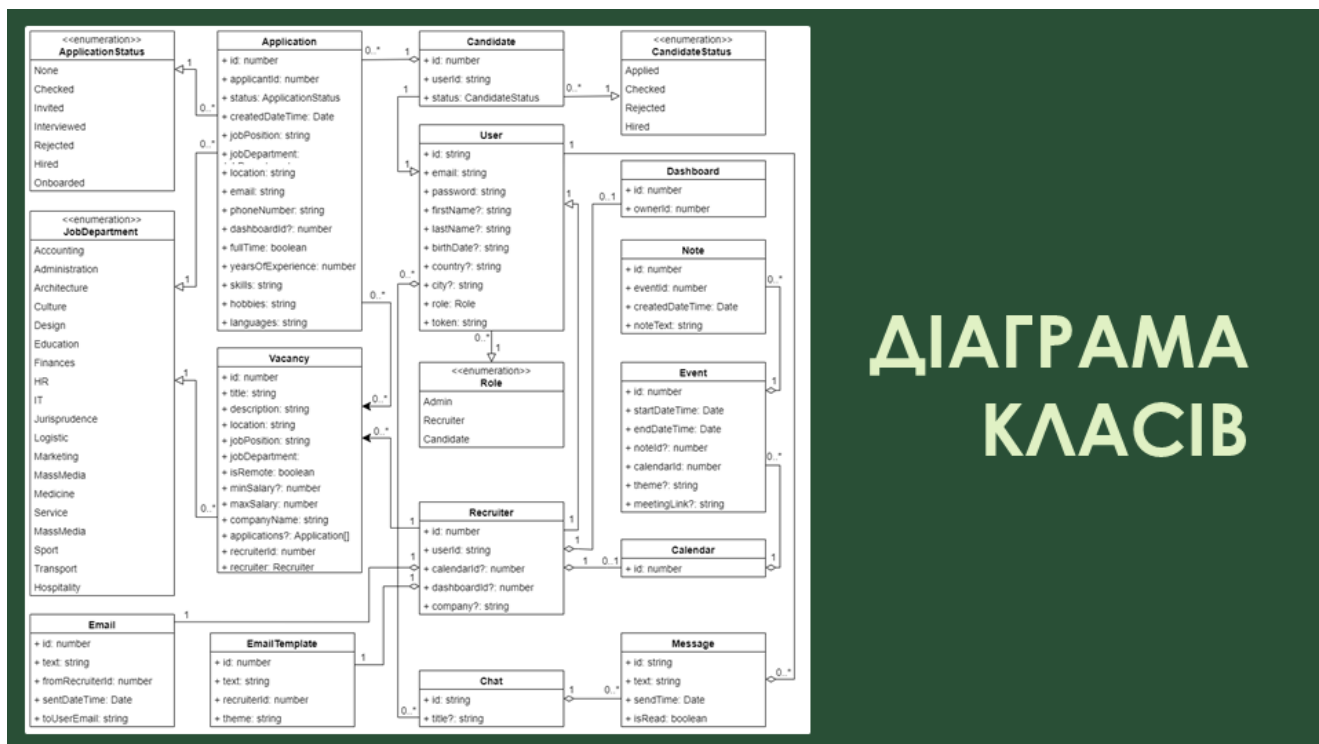


Рисунок Б.15 – Слайд 15

## ДІАГРАМА КОМПОНЕНТІВ

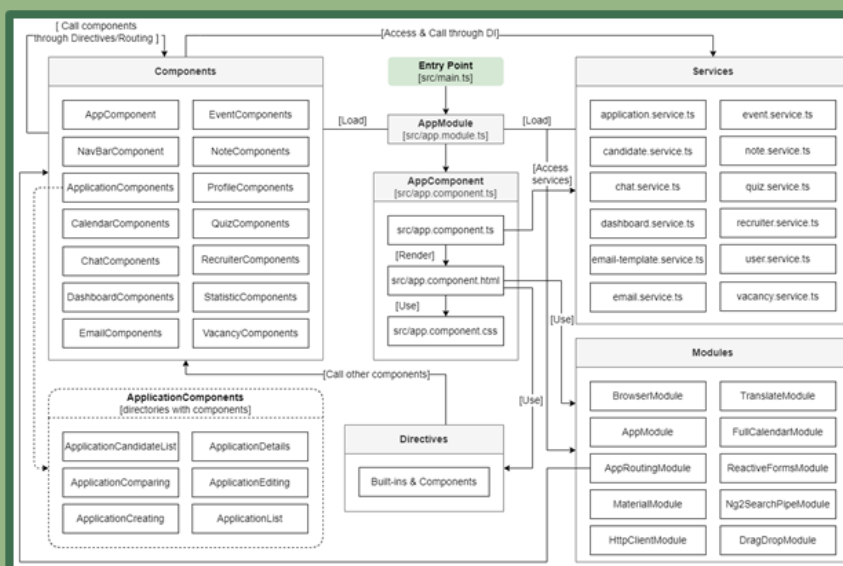


Рисунок Б.16 – Слайд 16



Рисунок Б.17 – Слайд 17

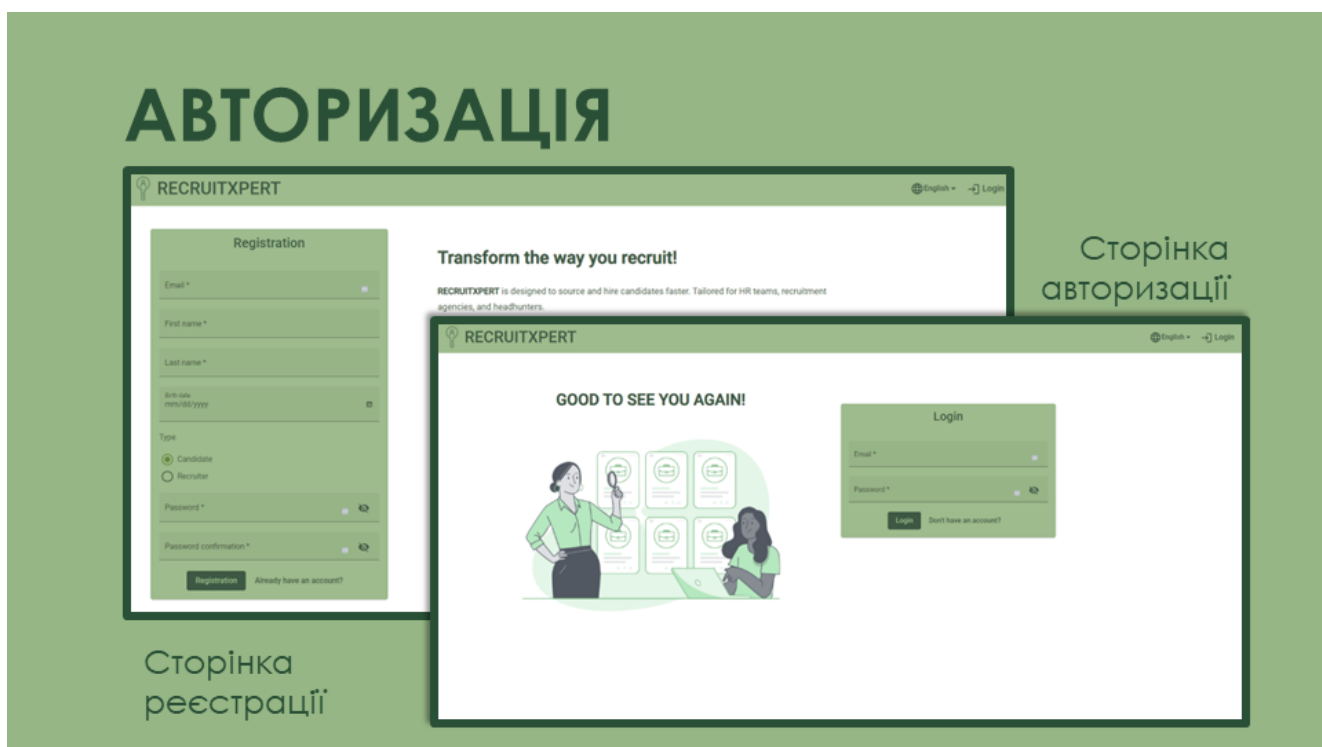


Рисунок Б.18 – Слайд 18

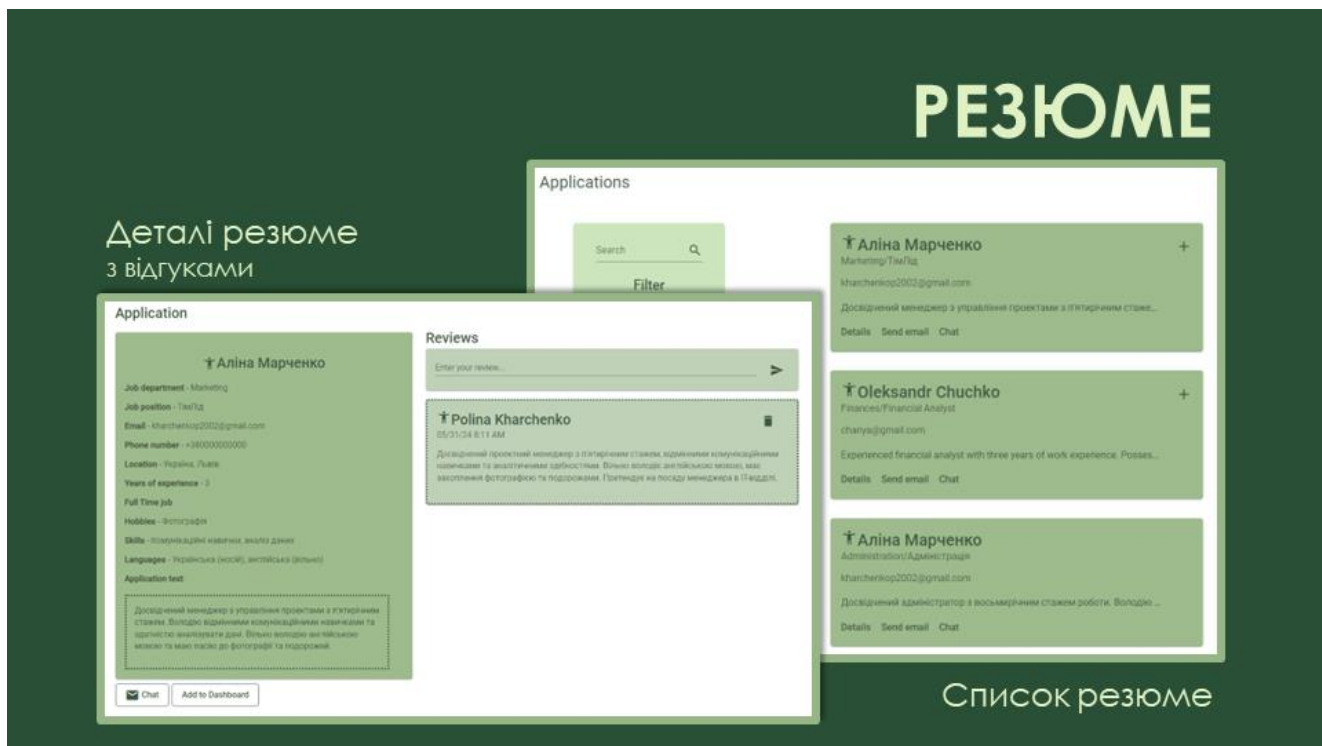


Рисунок Б.19 – Слайд 19

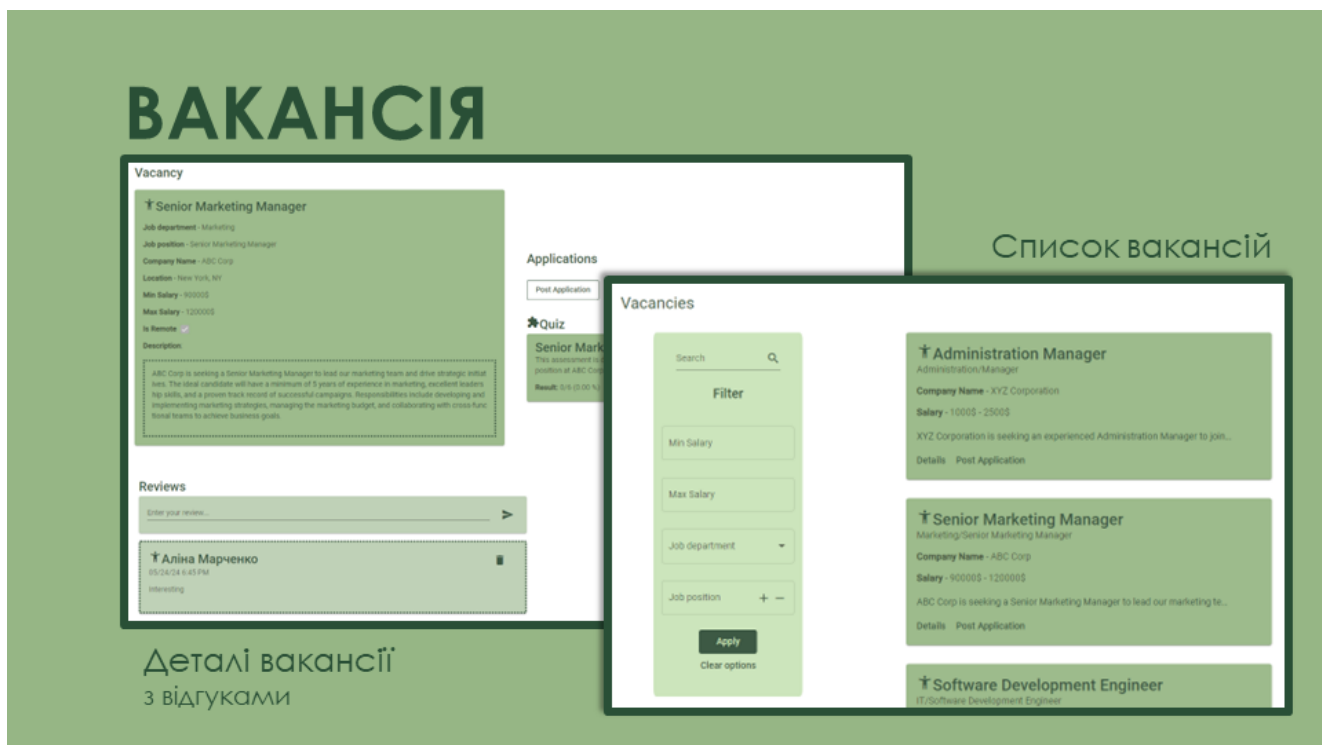


Рисунок Б.20 – Слайд 20

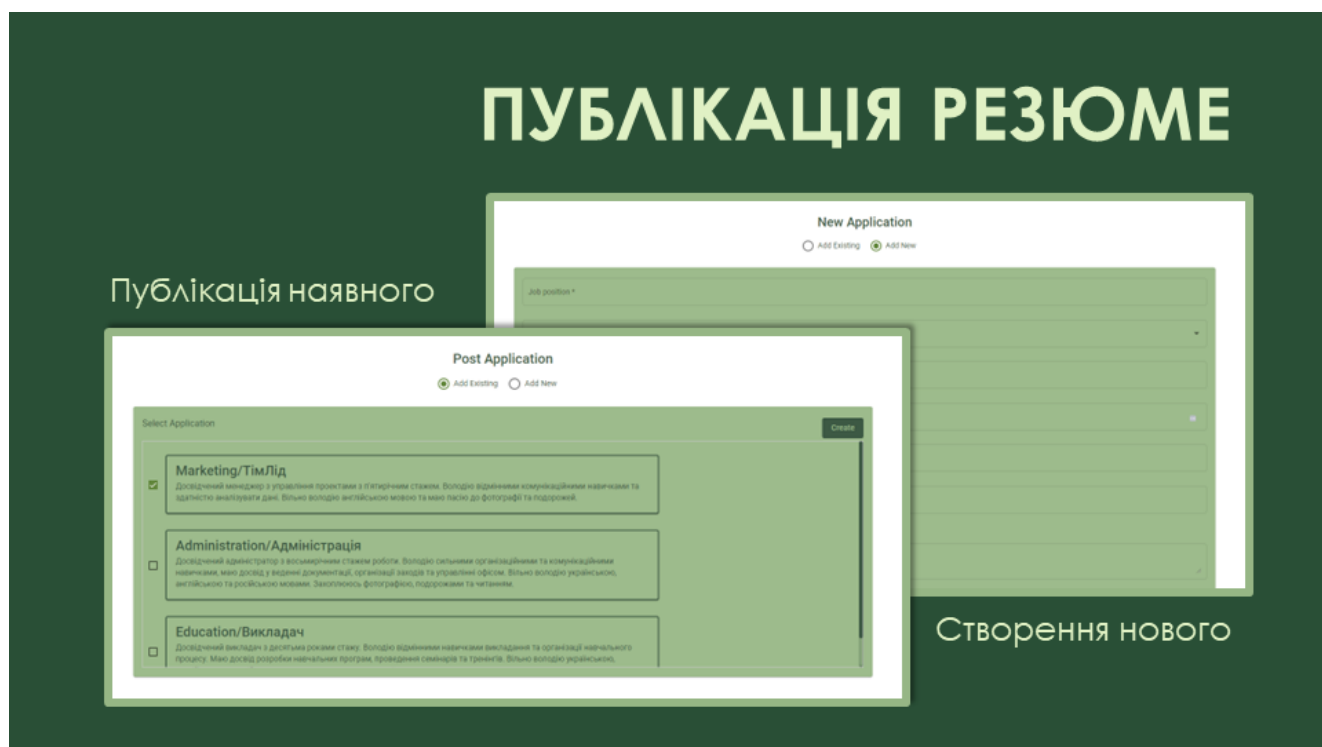


Рисунок Б.21 – Слайд 21

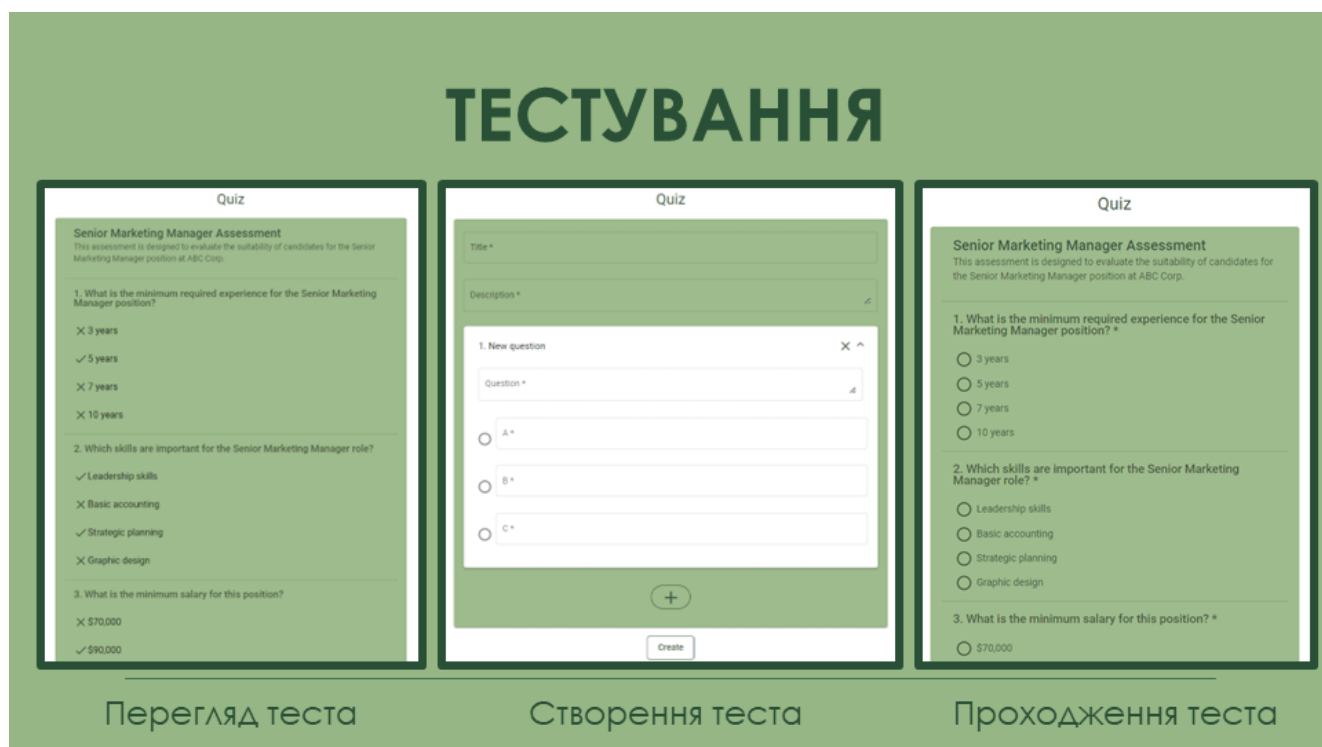


Рисунок Б.22 – Слайд 22

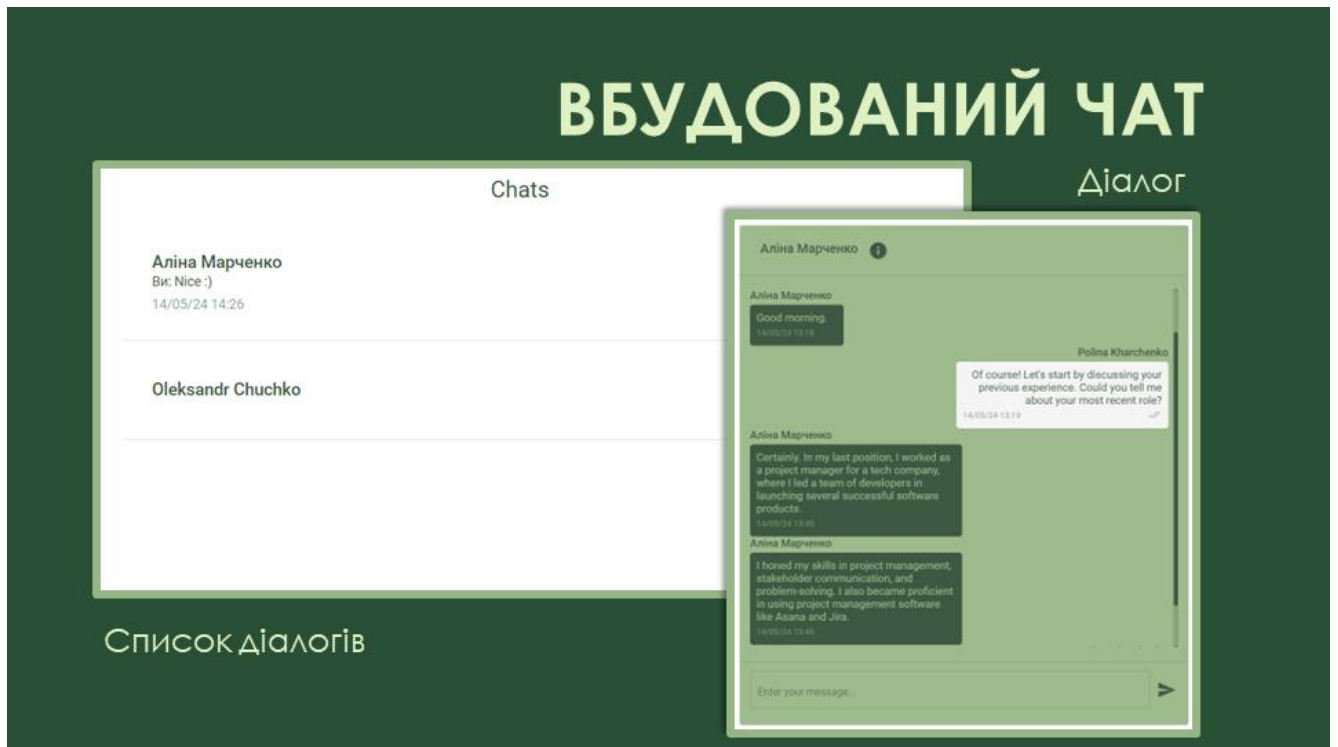


Рисунок Б.23 – Слайд 23



Рисунок Б.24 – Слайд 24

# КАЛЕНДАР



Календар рекрутера

Рисунок Б.25 – Слайд 25

# ЗАРОБІТНЯ ПЛАТА

The image shows a salary calculation form. On the left side, there are several input fields and dropdown menus:
 

- Weekly hours \*: 40
- Skill level \*: 3 (dropdown menu)
- Weeks per year \*: 20
- Bonus: 500
- Part time \*: Part time (dropdown menu)
- Hourly payment \*: 20

 At the bottom of the form is a 'Calculate' button. On the right side, there is a box displaying the results:
 

- Monthly salary: 3572 \$
- Yearly salary: 6840 \$

Розрахунок заробітної плати

Рисунок Б.26 – Слайд 26



Рисунок Б.27 – Слайд 27



Рисунок Б.28 – Слайд 28

# ТЕСТУВАННЯ

Мета тесту			
Перевірити коректність реалізації публікації резюме до вакансії, створення нового і додавання вже наявного			
Передумова			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити веб-застосунок	Користувач має доступ до сайту, який відкритий	Пройдено
2	Користувач має обліковий запис та хоче авторизуватися	Перенаправлено на сторінку профілю	Пройдено
3	В системі створено вакансію	Вакансія доступна	Пройдено

Рисунок Б.29 – Слайд 29

# ТЕСТУВАННЯ

Створення і публікація нового резюме			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити сторінку вакансії	Перенаправлено на сторінку вакансії	Пройдено
2	Натиснути кнопку «Додати своє резюме»	Перенаправлено на сторінку створення резюме, вкладку «Додати нове»	Пройдено
3	Заповнити обов'язкове поле «Посада»	Обов'язкове поле було заповнено	Пройдено
4	Обрати «Відділ»	Було обрано відділ	Пройдено
5	Заповнити необов'язкове поле «Місцезнаходження»	Необов'язкове поле було заповнено	Пройдено
6	Заповнити обов'язкове поле «Імейл»	Обов'язкове поле було заповнено, пройдено валідацію електронної пошти	Пройдено
7	Заповнити обов'язкове поле «Номер телефону»	Обов'язкове поле було заповнено, пройдено валідацію номера телефону	Пройдено
8	Заповнити обов'язкове поле «Досвід (кількість років)»	Обов'язкове поле було заповнено	Пройдено
9	Обрати «Повний робочий день»	Прапорець поставлено	Пройдено
10	Заповнити обов'язкове поле «Навички»	Обов'язкове поле було заповнено	Пройдено

Рисунок Б.30 – Слайд 30

ТЕСТУВАННЯ			
Створення і публікація нового резюме			
№	Опис випадку	Очікуваний результат	Висновок
11	Заповнити обов'язкове поле «Знання мов»	Обов'язкове поле було заповнено	Пройдено
12	Заповнити обов'язкове поле «Хобі»	Обов'язкове поле було заповнено	Пройдено
13	Заповнити обов'язкове поле «Мінімальна заробітна плата»	Обов'язкове поле було заповнено	Пройдено
14	Заповнити обов'язкове поле «Максимальна заробітна плата», меншим значенням ніж «Мінімальна заробітна плата»	Відображається повідомлення про некоректне заповнення поля	Пройдено
15	Заповнити обов'язкове поле «Максимальна заробітна плата» правильними даними	Обов'язкове поле було заповнено, повідомлення про помилку зникло	Пройдено
16	Заповнити обов'язкове поле «Текст резюме»	Обов'язкове поле було заповнено	Пройдено
17	Натиснути кнопку «Створити»	Резюме успішно створено	Пройдено
18	Користувача перенаправлено на сторінку деталей резюме	Резюме має вказані дані і посилається на правильну вакансію	Пройдено

Рисунок Б.31 – Слайд 31

ТЕСТУВАННЯ			
Публікація вже наявного резюме на вакансію			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити сторінку вакансії	Перенаправлено на сторінку вакансії	Пройдено
2	Натиснути кнопку «Додати своє резюме»	Перенаправлено на сторінку створення резюме, вкладку «Додати нове»	Пройдено
3	Обрати вкладку «Додати існуюче»	Відкриття списку наявних резюме	Пройдено
4	Обрати резюме	Обране резюме відмічено прапорцем	Пройдено
5	Обрати інше резюме	Прапорець перенесено на обране резюме	Пройдено
6	Натиснути кнопку «Створити»	Резюме успішно опубліковано	Пройдено
7	Користувача перенаправлено на детальний перегляд обраного резюме	Список вакансій оновлено	Пройдено

Рисунок Б.32 – Слайд 32



Рисунок Б.33 – Слайд 33



Рисунок Б.34 – Слайд 34



Рисунок Б.35 – Слайд 35

## ДОДАТОК В

Специфікація вимог до програмного продукту

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

### СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Програмна система для автоматизації онлайн-рекрутингу

Студент гр. ПЗПІ-20-4 Стрюкова Д. В.

Студент гр. ПЗПІ-20-4 Харченко П. Д.

Студент гр. ПЗПІ-20-5 Клименюк Е. С.

Харків

2024

## ЗМІСТ

1 ВСТУП	3
1.1 Огляд продукту	3
1.2 Мета	3
1.3 Межі	4
1.4 Означення та аббревіатури	5
2 ЗАГАЛЬНИЙ ОПИС	7
2.1 Перспективи продукту	7
2.2 Функції продукту	7
2.3 Характеристики користувачів	8
2.4 Загальні обмеження	9
2.5 Припущення й залежності	9
3 КОНКРЕТНІ ВИМОГИ	11
3.1 Вимоги до зовнішніх інтерфейсів	11
3.1.1 Інтерфейс користувача	11
3.1.2 Апаратний інтерфейс	11
3.1.3 Програмний інтерфейс	12
3.1.4 Комунікаційний протокол	13
3.1.5 Обмеження пам'яті	13
3.2 Атрибути програмного продукту	14
3.2.1 Надійність	14
3.2.2 Доступність	14
3.2.3 Безпека	15
3.2.4 Супроводжуваність	15
3.2.5 Переносність	16
3.2.6 Продуктивність	16
3.3 Вимоги до бази даних	17

## 1 ВСТУП

### 1.1 Огляд продукту

Програмна система для автоматизації онлайн-рекрутингу є потужним інструментом для компаній, які займаються наймом персоналу. Цей продукт надає комплекс функцій, які сприяють ефективному керуванню процесом рекрутингу та спрощують взаємодію з кандидатами.

Система забезпечує безпеку та контроль доступу за допомогою системи аутентифікації та авторизації. Це гарантує, що доступ до системи мають лише авторизовані користувачі, такі як рекрутери та адміністратори, забезпечуючи конфіденційність даних.

Адміністратори можуть легко керувати користувачами, створювати нові облікові записи, надавати різні рівні доступу та редагувати існуючі профілі. Це дозволяє ефективно організовувати робочий процес та розподіляти обов'язки серед персоналу.

Система дозволяє створювати та редагувати вакансії, розміщувати оголошення про вакансії та отримувати аплікації від кандидатів. Користувачі можуть переглядати деталі вакансій, такі як опис посади, вимоги до кандидатів та умови праці, а також подавати свої резюме. Це спрощує процес рекрутингу та дозволяє рекрутерам ефективно відбирати потенційних кандидатів.

Управління кандидатами є ще одним важливим аспектом продукту. Рекрутери можуть вести базу даних кандидатів, відстежувати їх статуси, проводити співбесіди та приймати рішення щодо найняття.

Ця програмна система сприяє покращенню ефективності процесу рекрутингу, зменшенню часу на пошук та відбір кандидатів, а також підвищенню якості найму.

### 1.2 Мета

Система для автоматизації онлайн-рекрутингу спрямована на створення зручного та централізованого середовища для організацій, які здійснюють найм

персоналу. Головною метою цієї системи є надання керівникам зручного і ефективного інструменту для планування та координації найму персоналу, забезпечуючи їх можливістю організовано проводити відбір кандидатів та вести облік рекрутингових процесів.

Крім того, система спрощує комунікацію між рекрутерами та кандидатами, надаючи зручний механізм обміну інформацією. Вона автоматично повідомляє про нові вакансії та статуси рекрутингових процесів, що в свою чергу дозволяє кандидату швидко реагувати на зміни та подаватися на нові вакансії.

Загалом, ця програмна система спрямована на поліпшення ефективності та організації процесу найму персоналу, надаючи зручні інструменти для планування, координації, відстеження та звітності щодо вакансій та кандидатів.

### 1.3 Межі

Програмна система спрямована на забезпечення ефективності та швидкості процесів рекрутингу, зокрема шляхом автоматизації рутинних завдань та зменшення людського втручання.

Цілі програмного продукту включають зменшення часу, потрібного для створення вакансій і подачі заявок, підвищити ефективність процесу найму.

Вона має надавати зручний API для взаємодії з клієнтською частиною додатку та має бути здатна обробляти запити від багатьох користувачів одночасно.

Однак, у системі є обмеження, такі як ресурси сервера (пам'ять та потужність процесора), які впливають на її продуктивність та масштабованість. Це може уповільнити обробку запитів та знизити доступність для користувачів. Також існують обмеження на кількість користувачів та ролей, що можуть бути створені. Додатково, система повинна забезпечувати безпеку даних, використовуючи шифрування та інші методи захисту, щоб убезпечити інформацію від несанкціонованого доступу.

## 1.4 Означення та аббревіатури

Frontend – це частина програмної системи, що відповідає за відображення інтерфейсу користувача та взаємодію з ним через браузер або інші засоби. Вона включає в себе розробку та реалізацію веб-сторінок, компонентів і елементів керування, які користувач може бачити та взаємодіяти з ними.

Backend – це складова частина програмного забезпечення, яка відповідає за обробку даних та логічні операції на серверному рівні. Вона забезпечує взаємодію з базою даних, виконує процеси автентифікації, авторизації та управління бізнес-логікою системи.

API – це скорочення від Application Programming Interface. Це набір правил і протоколів, що визначає, як програми або компоненти системи можуть взаємодіяти між собою. API визначає доступні функції та методи комунікації між різними складовими програмної системи.

ORM – це аббревіатура від Object-Relational Mapping, що означає технологію, яка дозволяє взаємодіяти з базою даних у вигляді об'єктів програми. Вона перетворює дані з реляційної бази даних у об'єкти, що можуть бути використані в програмному коді, та надає зручний спосіб управління даними.

HTTP – це протокол передачі гіпертекстових даних через мережу, що використовується для комунікації між клієнтами та серверами в Інтернеті.

REST – підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів.

CRUD – це аббревіатура, що складається зі слів Create, Read, Update, Delete (створити, прочитати, оновити, видалити). Це основні операції, які можна виконати з даними в базі даних, включаючи створення нових записів, отримання даних, їх оновлення та видалення.

Entity Framework – це технологія ORM (Object-Relational Mapping), яка дозволяє розробникам працювати з базами даних у формі об'єктів програми. Вона забезпечує зручний спосіб доступу до даних з бази даних, перетворюючи їх на об'єкти, з якими можна взаємодіяти в програмному коді.

Identity – це система аутентифікації та авторизації користувачів в програмній системі. Вона дозволяє контролювати доступ до різних частин системи, перевіряючи ідентифікаційні дані користувача та надаючи йому відповідні права доступу.

HR – це скорочення від Human Resources (кадрові ресурси). Це відділ або функціональна область в організації, що відповідає за управління персоналом. Відділ HR займається наймом та звільненням працівників, адмініструванням персональних даних, розвитком персоналу, а також забезпечує вирішення конфліктів та питань стосовно заробітної плати та соціальних пакетів.

JSON (JavaScript Object Notation) – це легкий, текстовий формат обміну даними, що базується на синтаксисі JavaScript. Він використовується для передачі структурованих даних між програмами, особливо у веб-розробці. Формат JSON складається з пар ключ-значення, де ключі є рядками, а значення можуть бути будь-якого типу даних: рядки, числа, масиви, об'єкти тощо.

CRM (Customer Relationship Management) – програмне забезпечення для організацій, призначене для автоматизації взаємодії із замовниками (клієнтами).

## 2 ЗАГАЛЬНИЙ ОПИС

### 2.1 Перспективи продукту

Останнім часом спостерігається значний інтерес до автоматизації процесу найму та підвищення ефективності управління персоналом. Наш продукт виявляє значний потенціал застосування у різних галузях, включаючи компанії будь-якого масштабу, агентства з кадрового рекрутингу та інші сфери.

Прогнозовані напрямки розвитку нашого продукту включають розширення функціоналу та впровадження нових модулів для відповіді на конкретні потреби користувачів. Наприклад, можливості планування та розподілу ресурсів, система оцінки кандидатів, інтеграція з іншими платформами або соціальними мережами, розширена аналітика та звітність.

Крім того, перспективи включають вихід на міжнародний ринок та масштабування продукту для задоволення потреб різних компаній та організацій

у різних країнах. Адаптація продукту до міжнародних стандартів та норм у сфері рекрутингу відкриє нові можливості для його впровадження на глобальному рівні.

Загалом, наш продукт має значний потенціал для розвитку та відповіді на зростаючі потреби в управлінні процесом найму персоналу.

## 2.2 Функції продукту

Основні функції цього продукту включають:

FE-1: Дошка відображення процесу найму (аналог Jira)

FE-2: Календар з відображенням онлайн зустрічей та інших подій

FE-3: Google meets, тобто створення онлайн зустрічі з кандидатом

FE-4: Імейли, відправка з системи на пошту, шаблони

FE-5: Розрахунок заробітної плати

FE-6: Резюме кандидата з можливістю прикріпити файли

FE-7: Тести для вакансії

FE-8: Чат з рекрутером

FE-9: Статистика за вакансіями, компаніями, кандидатами, загальна, стан заявки на вакансію

FE-10: Звіт по вакансії

FE-11: Порівняння кандидатів та резюме

FE-12: Генерація документів, наприклад контракти та листи пропозиції на основі даних кандидата

FE-14: Відгуки

FE-15: Підписка на вакансію та розсилка імейлів

FE-16: Релевантність вакансії вимогам

## 2.3 Характеристики користувачів

Характеристики користувачів програмної системи для керування процесом набору персоналу з back-end реалізацією можуть бути різноманітними, оскільки система взаємодіє з різними типами користувачів. Основні характеристики

користувачів включають:

- кандидати на роботу є ключовими користувачами системи. Вони можуть мати різний досвід роботи, освіту та навички, що впливають на їхню придатність для конкретної вакансії;
- HR менеджери відповідають за керування процесом набору персоналу. Вони мають розширені права доступу та можуть створювати, редагувати та видаляти вакансії, керувати кандидатами, аналізувати звітність та виконувати інші адміністративні функції;
- адміністратори системи відповідають за налаштування та підтримку програмного забезпечення. Вони забезпечують безперебійну роботу системи, вирішують технічні проблеми, які можуть виникнути.

Загалом, різноманітність користувачів враховується у розробці системи, щоб кожна категорія могла ефективно використовувати її функціонал для досягнення своїх цілей у процесі найму персоналу.

#### 2.4 Загальні обмеження

У програмній системі для управління рекрутингом з back-end реалізацією існують обмеження, які можуть вплинути на її ефективність та можливості використання.

Продуктивність та масштабованість системи обумовлені фізичними параметрами серверів, такими як кількість пам'яті, потужність обчислювального процесора, а також пропускна здатність мережі. Ці обмеження можуть суттєво вплинути на швидкість та завантаження системи під час обробки великої кількості запитів одночасно.

Забезпечення безпеки є важливою складовою для захисту конфіденційності даних користувачів та запобігання несанкціонованому доступу. Обмеження в цьому плані можуть включати криптографічні вимоги, захист від потенційних атак, встановлення правил доступу до інформації та контроль над правами користувачів.

Система повинна мати можливість масштабуватися для відповіді на

зростаючий обсяг користувачів, проектів та завдань. Обмеження масштабованості можуть впливати на продуктивність та швидкодію системи під час інтенсивного навантаження.

## 2.5 Припущення й залежності

AS-1: Доступ до Інтернету. Передбачається, що всі користувачі мають доступ до Інтернету для використання системи онлайн-рекрутингу.

AS-2: Відповідність інформації в резюме. Передбачається, що інформація, надана кандидатами у їхніх резюме, є точною і відповідає їхньому досвіду та навичкам.

AS-3: Стабільність технічних систем. Передбачається, що технічні системи, необхідні для роботи програмної платформи, будуть працювати стабільно без значних перебоїв.

DP-1: Доступ до бази даних кандидатів. Система залежить від наявності доступу до бази даних з профілями кандидатів для ефективного пошуку та відбору претендентів на вакансії.

DP-2: Інтеграція з існуючими системами рекрутингу. Може бути необхідною інтеграція з існуючими системами рекрутингу або обліку кандидатів, які вже використовуються компанією.

DP-3: Підтримка технічного персоналу. Для ефективної роботи системи може знадобитися технічна підтримка для моніторингу, усунення неполадок та оновлення програмного забезпечення.

## 3 КОНКРЕТНІ ВИМОГИ

### 3.1 Вимоги до зовнішніх інтерфейсів

#### 3.1.1 Інтерфейс користувача

Система повинна мати інтуїтивно зрозумілий та ергономічний інтерфейс, що дозволяє користувачам легко взаємодіяти з програмним забезпеченням. Інтерфейс користувача має бути зручним у використанні та забезпечувати швидкий доступ до всіх основних функцій системи.

Крім того, інтерфейс повинен бути адаптивним для різних типів пристроїв, включаючи комп'ютери, планшети та мобільні девайси.

Інтерфейс користувача представлений у вигляді веб-додатку, залежно від потреб користувачів та доступних платформ. Додаток має послідовну організацію, зрозумілу навігаційну систему та інтуїтивно зрозумілі елементи управління, що спрощують користувачам взаємодію та виконання різноманітних завдань.

### 3.1.2 Апаратний інтерфейс

Апаратне забезпечення програмної системи для управління рекрутингом з back-end реалізацією складається з необхідних компонентів та засобів, які гарантують безперебійну роботу системи. Оскільки програма зазвичай працює в онлайн-середовищі та доступна через веб-браузер, апаратне забезпечення включає:

- сервери для зберігання та обробки даних. Це можуть бути фізичні сервери, віртуальні сервери або хмарні платформи, які забезпечують необхідні обчислювальні та зберігальні ресурси для ефективної роботи системи;

- мережеві пристрої, такі як маршрутизатори, комутатори та мережеві кабелі, для забезпечення безперервного зв'язку між користувачами та серверами системи через мережу.

### 3.1.3 Програмний інтерфейс

Програмна система для автоматизації онлайн-рекрутингу впроваджує програмний інтерфейс (API) як ключовий механізм для спільної роботи з іншими додатками та системами. Цей API визначає набір правил та протоколів для взаємодії між різними компонентами програмного забезпечення, що дозволяє зовнішнім системам або розробникам використовувати функціональні можливості та отримувати доступ до даних програмної системи.

У контексті онлайн-рекрутингу, API може включати в себе різноманітні функції, такі як створення та оновлення вакансій, перегляд кандидатів, управління

процесом найму та багато іншого. Реалізація такого API у формі RESTful API є популярним підходом, оскільки він базується на використанні HTTP-протоколу, що дозволяє здійснювати просту та стандартизовану взаємодію між клієнтами та сервером.

Однією з переваг RESTful API є його простота та легкість використання. Використання стандартних HTTP-методів, таких як GET, POST, PUT та DELETE, спрощує процес комунікації з системою. Крім того, обмін даними у форматі JSON дозволяє ефективно передавати структуровану інформацію між клієнтом та сервером.

Застосування API у програмній системі для онлайн-рекрутингу відкриває безліч можливостей для інтеграції з іншими системами та сервісами.

Використання API у програмній системі для рекрутингу також дозволяє забезпечити гнучкість та розширюваність системи. Розробники можуть легко додавати нові функції та можливості, не змінюючи основну архітектуру програми, що робить систему більш адаптивною до змін потреб бізнесу та ринкових умов.

У цілому, використання програмного інтерфейсу в програмній системі для автоматизації онлайн-рекрутингу є ключовим фактором для забезпечення її ефективності, гнучкості та інтегрованості з іншими системами.

### 3.1.4 Комунікаційний протокол

Система повинна підтримувати різні інтерфейси комунікації для забезпечення обміну даними з іншими системами та сервісами. Це може включати в себе стандартні протоколи передачі даних, такі як HTTP/HTTPS для взаємодії з веб-серверами, REST для взаємодії з іншими веб-службами, а також можливості інтеграції з поштовими клієнтами. Крім того, система повинна забезпечувати захист даних та аутентифікацію при здійсненні комунікації з іншими системами, щоб забезпечити конфіденційність та цілісність інформації.

### 3.1.5 Обмеження пам'яті

Обмеження пам'яті є критичним аспектом у розробці програмного забезпечення, особливо в сучасних системах, де ефективне використання ресурсів має вирішальне значення для забезпечення продуктивності та надійності програм.

Під час розробки програмного забезпечення важливо усвідомлювати обмеження пам'яті, які встановлені для різних типів пристроїв та платформ. Наприклад, вбудовані системи можуть мати обмежену кількість доступної пам'яті, тоді як сервери або персональні комп'ютери можуть мати більші ресурси.

Одним з методів управління обмеженням пам'яті є ефективне використання алгоритмів та структур даних. Наприклад, використання компактних представлень даних, які займають менше пам'яті, може допомогти знизити використання ресурсів. Крім того, уникання зайвого копіювання даних та оптимізація роботи з пам'яттю також може покращити продуктивність програми.

Для ефективного управління пам'яттю важливо також враховувати процеси видалення та звільнення пам'яті. Наприклад, у системах, де використання пам'яті є критичним, важливо уникати витoku пам'яті шляхом правильного вивільнення непотрібних ресурсів та використанням механізмів автоматичного управління пам'яттю.

## 3.2 Атрибути програмного продукту

### 3.2.1 Надійність

Надійність програмної системи для управління онлайн-рекрутингом є ключовою складовою успішного функціонування. Система має забезпечувати безперебійну та стабільну роботу, що є критично важливим для ефективного управління процесом рекрутингу та задоволення потреб користувачів.

Надійність означає, що система працює так, як очікується, та обробляє запити користувачів, запобігаючи можливим збоям чи відмовам. Для досягнення цієї мети система має бути забезпечена вбудованим механізмом обробки помилок та винятків, що можуть виникати під час роботи. Це допомагає виявляти та

вирішувати проблеми, що впливають на роботу системи, забезпечуючи її надійну та стабільну функціональність.

Окрім цього, важливо мати механізми резервного копіювання, які забезпечують збереження та захист даних системи. Це може включати регулярне резервне копіювання даних, створення резервних копій баз даних та інших важливих ресурсів. Такий підхід дозволяє підтримувати надійність та безпеку даних у будь-який час, що є важливим аспектом для успішної роботи програмної системи з управління рекрутингом.

### 3.2.2 Доступність

Для забезпечення доступності системи можна використовувати різні технології та стратегії. Одним з ключових методів є використання дублювання та резервування. Це означає створення дублюючих екземплярів системи або компонентів, які автоматично вступають в дію у випадку відмови основного обладнання або програмного забезпечення.

Крім того, для забезпечення доступності можна використовувати системи моніторингу та управління. Такі системи постійно контролюють стан обладнання та програмного забезпечення і сповіщають адміністраторів у випадку виявлення проблем. Додатково, вони можуть автоматично виконувати заходи для відновлення роботи системи або виконувати переведення навантаження на резервні сервери.

Одним з важливих аспектів є розробка системи з урахуванням масштабованості горизонтальної або вертикальної. Горизонтальна масштабованість передбачає розширення системи за рахунок додавання нових серверів або вузлів у мережу, щоб розділити навантаження та забезпечити стабільну роботу при зростанні обсягу даних або користувачів. Вертикальна масштабованість включає у підвищення продуктивності і потужності окремих компонентів системи, наприклад, шляхом оновлення обладнання або програмного забезпечення.

### 3.2.3 Безпека

Система має забезпечувати високий рівень безпеки для захисту конфіденційності, цілісності та доступності даних. Це охоплює застосування сучасних методів шифрування для конфіденційної інформації, механізми автентифікації для запобігання несанкціонованому доступу та механізми контролю доступу для обмеження прав доступу.

Система повинна використовувати параметризовані запити і виключити можливість SQL-ін'єкцій та інших атак, для забезпечення безпеки запитів до бази даних через API. Також важливо реалізувати аутентифікацію і авторизацію на рівні API, щоб контролювати доступ до даних та забезпечити їх конфіденційність.

### 3.2.4 Супроводжуваність

Забезпечення документацією відповідає за збереження знань та інформації про систему, що дозволяє новим членам команди ефективно розібратися в її структурі та функціональності. Моніторинг та логування відіграють ключову роль у виявленні проблем та відслідковуванні причин виникнення помилок, що дозволяє оперативно реагувати на них та уникати критичних збоїв у роботі системи.

Регулярні оновлення та тестування забезпечують актуальність та надійність програмного забезпечення, а також виявлення та виправлення помилок ще до їх впливу на користувачів. Підтримка користувачів є важливою для забезпечення задоволення та впевненості користувачів у якості та ефективності системи.

Автоматизація процесів супроводження дозволяє зменшити час та зусилля, витрачені на ручні операції, а також забезпечує більшу стабільність та ефективність управління системою. Включення цих механізмів у пункт про супроводжуваність дозволяє забезпечити готовність та надійність системи для подальшого функціонування після її впровадження.

### 3.2.5 Переносність

Система повинна бути легко переносною між різними середовищами та платформами. Це включає здатність запускати систему на різних операційних системах, таких як Windows, macOS та Linux, а також на різних типах апаратного забезпечення. Переносність також означає, що система повинна бути здатною працювати в різних мережевих середовищах із різними конфігураціями мережі та доступу до Інтернету.

Крім того, важливо, щоб система могла легко інтегруватися з іншими додатками та сервісами, що використовуються у сфері рекрутингу, такими як CRM-системи, платформи соціальних мереж та рекрутингові портали, незалежно від їхніх технологій чи платформ.

### 3.2.6 Продуктивність

У програмній системі для управління онлайн-рекрутингом, важливим аспектом є продуктивність, яка включає в себе оптимізацію коду, масштабування та ефективне використання ресурсів. Написання оптимізованого коду, що забезпечує швидку обробку завдань. Масштабування системи для того, щоб ефективно розподіляти навантаження та збалансовано використовувати ресурси, що сприяє більшій продуктивності та підвищенню досвіду користувача при користуванні продуктом.

## 3.3 Вимоги до бази даних

Забезпечення вимог до бази даних є важливим етапом у процесі розробки програмного забезпечення, оскільки це визначає основні принципи організації, зберігання та доступу до даних, необхідних для ефективної роботи системи.

Перш за все, важливо визначити структуру даних, яка відповідає потребам програми. Це включає в себе створення таблиць, визначення полів та їх типів, а також встановлення зв'язків між таблицями для забезпечення консистентності даних.

Далі, важливо нормалізувати дані, щоб уникнути дублювання та забезпечити їх цілісність. Це допомагає забезпечити ефективність роботи з базою даних та уникнути виникнення проблем при маніпулюванні даними.

Однією з ключових вимог є забезпечення безпеки даних. Це включає в себе встановлення прав доступу до даних, що забезпечує конфіденційність інформації, а також застосування методів шифрування для захисту даних від несанкціонованого доступу.

Крім того, важливо забезпечити швидкодію бази даних шляхом оптимізації запитів та використання індексів. Це допомагає забезпечити ефективну обробку даних та швидку відповідь на запити користувачів.

Не менш важливою є можливість резервного копіювання та відновлення даних. Це дозволяє відновити інформацію в разі втрати або пошкодження даних, що забезпечує безпеку та надійність роботи системи.

## ДОДАТОК Г

## Код AuthenticationGuard

```
import { Injectable } from '@angular/core';
import { ActivatedRouteSnapshot, CanActivate, Router, UrlTree } from
 '@angular/router';
import { JwtHelperService } from '@auth0/angular-jwt';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root',
})
export class AuthenticationGuard implements CanActivate {
  constructor(
    private router: Router,
    private jwtService: JwtHelperService
  ) {}

  canActivate(
    route: ActivatedRouteSnapshot
  ): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean |
  UrlTree {
    if (this.jwtService.isTokenExpired()) {
      localStorage.removeItem('access_token');
      localStorage.removeItem('id');
      localStorage.removeItem('email');
      localStorage.removeItem('role');

      window.location.reload();
      this.router.navigate(['account/login']);
      return false;
    }

    const userRole = localStorage.getItem('role');

    if (route.data['roles'] && route.data['roles'].indexOf(userRole) ===
-1) {
      this.router.navigate(['vacancy/list']);
      return false;
    }

    return true;
  }
}
```

ДОДАТОК Д

Апробація



Рисунок Д.1 – Сертифікат учасника конференції «Print, Multimedia and Web» за доповідь «Аналіз трендів у сфері рекрутингу та їх відображення в функціональних можливостях додатків»