

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ ім. Ю. М. ПОТЕБНІ
Кафедра інформаційної економіки, підприємництва та фінансів
(повна назва кафедри)

Кваліфікаційна робота(проект)

магістра
(рівень вищої освіти)

на тему Розробка рекомендаційної системи управління підбором персоналу ІТ-підприємств

Виконав: студент 2 курсу, групи 8.0510-іє
спеціальності 051 Економіка
(код і назва спеціальності)

спеціалізації _____
(код і назва спеціалізації)

освітньої програми Інформаційна економіка
(назва освітньої програми)

М. К. Стеценко
(ініціали та прізвище)

Керівник доц., д.е.н., доц. Клопов І.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Інженерний навчально-науковий інститут ім. Ю. М. Потебні
Кафедра інформаційної економіки, підприємництва та фінансів

Рівень вищої освіти магістр
Спеціальність 051 Економіка
(код та назва)

Спеціалізація _____
(код та назва)

Освітня програма Інформаційна економіка

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

« _____ » _____ 20 _____ року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЕКТ) СТУДЕНТОВІ (СТУДЕНТЦІ)
Стеценко Максим Костянтинович
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Розробка рекомендаційної системи управління підбором персоналу ІТ-підприємств

керівник роботи Клопов І. О., д.е.н., доц.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом ЗНУ від «30» червня 2021 року № 975-с

Строк подання студентом роботи _____

2. Вихідні дані до роботи інформаційні потоки управління підбором персоналу ІТ-підприємств.

3. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Теоретичні засади рекомендаційних систем. 2. Розробка рекомендаційної системи управління підбором персоналу ІТ-підприємств. 3. Практична реалізація рекомендаційної системи.

4. Перелік графічного матеріалу: кваліфікаційна робота містить 26 зображень концептуальна модель рекомендаційної системи управління підбором персоналу іт-підприємств, схема бази даних, діаграма компонентів системи, діаграма послідовностей.

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	доцент, д.е.н. доцент кафедри інформаційної економіки, підприємництва та фінансів Клопов І.О.	17.09	19.10
2	доцент, д.е.н. доцент кафедри інформаційної економіки, підприємництва та фінансів Клопов І.О.	19.10	29.10
3	доцент, д.е.н. доцент кафедри інформаційної економіки, підприємництва та фінансів Клопов І.О.	29.10	16.11

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Призначення наукових керівників. Затвердження тем дипломних робіт	25.10.2021	
2	Напрацювання теоретичного матеріалу: дослідження сутності об'єкту та предмету дослідження, критичний аналіз існуючих методологічних засад, вибір та обґрунтування напрямку проведення дослідження	29.10.2021	
3	Апробація результатів на Міжнародних та Всеукраїнських конференціях	05.11.2021	
4	Розробка економіко-математичного забезпечення основних елементів концептуального підходу	19.11.2021	
5	Збір та систематизація статистичного та нормативного матеріалу дослідження.	30.11.2021	
6	Узагальнення отриманих результатів. Оформлення роботи	01.12.2021	
7	Надання роботи та автореферату до рецензії. Нормоконтроль	03.12.2021	
8	Прилюдний захист дипломної роботи на засіданні ЕК	16.12.2021	

Студент _____
(підпис)

М. К. Стеценко
(ініціали та прізвище)

Керівник роботи (проєкту) _____
(підпис)

І. О. Клопов
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис) _____ (ініціали та прізвище)

АНОТАЦІЯ

Стеценко М. К. Розробка рекомендаційної системи управління підбором персоналу ІТ-підприємств.

Кваліфікаційна випускна робота для здобуття ступеня вищої освіти магістра за спеціальністю 051 - Економіка, науковий керівник І. О. Клопов. Інженерний навчально-науковий інститут ім. Ю. М. Потебні ЗНУ, кафедра інформаційної економіки, підприємництва та фінансів, 2021.

Магістерська робота присвячена розробці рекомендаційної системи управління підбором персоналу ІТ-підприємств. В роботі досліджено класифікацію та принципи проектування рекомендаційних систем; проаналізовано сучасне програмне забезпечення та системи для розробки рекомендаційних систем; описана структура рекомендаційної системи; розроблено проектну реалізацію рекомендаційної системи управління підбором персоналу ІТ-підприємств.

Ключові слова: РЕКОМЕНДАЦІЙНА СИСТЕМА, ПРИНЦИПИ ПРОЕКТУВАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ, УПРАВЛІННЯ ПІДБОРОМ ПЕРСОНАЛУ, СТРУКТУРА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ, ІТ-ПІДПРИЄМСТВА.

SUMMARY

Stetsenko M. K. Development of a recommendation system for managing the selection of personnel of IT enterprises.

Qualification final work for obtaining a master's degree in specialty 051 - Economics, supervisor IO Klopov. Engineering Educational and Scientific Institute named after Yu. M. Potebny ZNU, Department of Information Economics, Entrepreneurship and Finance, 2021.

The master's thesis is devoted to the development of a recommendation management system for the selection of personnel of IT companies. The classification and principles of designing recommendation systems are investigated in the work; modern software and systems for development of recommendation systems are analyzed; the structure of the recommendation system is described; the project

implementation of the recommendation management system for the selection of personnel of IT enterprises has been developed.

Keywords: RECOMMENDATION SYSTEM, PRINCIPLES OF DESIGN SYSTEM DESIGN, PERSONNEL SELECTION MANAGEMENT, STRUCTURE OF RECOMMENDATIONS, RECOMMENDATIONS AND RECOMMENDATIONS

Аннотация

Стеценко М. К. Разработка рекомендательной системы управления подбором персонала ИТ-предприятий.

Квалификационная выпускная работа по получению степени высшего образования магистра по специальности 051 - Экономика, научный руководитель И. А. Клопов. Инженерный учебно-научный институт им. Ю. М. Потебные ЗНУ, кафедра информационной экономики, предпринимательства и финансов, 2021.

Магистерская работа посвящена разработке рекомендательной системы управления подбором персонала ИТ-предприятий. В работе исследованы классификация и принципы проектирования рекомендательных систем; проанализированы современное программное обеспечение и системы для разработки рекомендательных систем; описана структура рекомендательной системы; разработана проектная реализация рекомендательной системы управления подбором персонала ИТ-предприятий.

Ключевые слова: РЕКОМЕНДАЦИОННАЯ СИСТЕМА, ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ РЕКОМЕНДАЦИОННЫХ СИСТЕМ, УПРАВЛЕНИЕ ПОДБОРОМ ПЕРСОНАЛА, СТРУКТУРА РЕКОМЕНДАЦИОННОЙ СИСТЕМЫ, ИТ-ПРЕДПРИЯТИЕ.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ЗАСАДИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ.....	9
1.1. Теоретичні основи рекомендаційних систем.....	9
1.2. Класифікація рекомендаційних систем.....	12
1.3. Показники ефективності рекомендаційних систем.....	20
Висновки до розділу 1.....	23
РОЗДІЛ 2 РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПІДБОРОМ ПЕРСОНАЛУ ІТ-ПІДПРИЄМСТВ.....	25
2.1. Концепція рекомендаційної системи управління підбором персоналу ІТ-підприємств.....	25
2.2. Обґрунтування необхідності використання програмного забезпечення для підбору кадрів.....	27
2.3. Аналіз типових програмних рішень управління підбором персоналу ІТ-підприємств.....	30
Висновки до розділу 2.....	39
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ.....	41
3.1. Вибір програмних компонентів реалізації.....	41
3.2. Проектування рекомендаційної системи управління підбором персоналу ІТ-підприємств.....	54
3.3. Реалізація рекомендаційної системи управління підбором персоналу ІТ-підприємств.....	62
Висновки до розділу 3.....	69
ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72

ВСТУП

Рекомендаційні системи сьогодні є одними з найпопулярніших систем фільтрації інформації. Вони використовуються для прогнозування «рейтингу» або «переваги», які користувач поставив елементу. Практично кожна велика технологічна компанія застосувала рекомендаційну систему в тій чи іншій формі. Amazon використовує її, щоб пропонувати продукти клієнтам, YouTube, щоб вирішити, яке відео відтворювати наступним при автоматичному програванні, а Facebook застосував, щоб рекомендувати сторінки, які потрібно лайкнути, і людей, за якими слід стежити.

Рекомендаційні системи приносять очевидну вигоду власникам онлайн-магазинів, різних сервісів і додатків. Вони показують користувачеві саме те, що йому цікаво, і генерують прибуток. Але є і мінуси. Людина звикає до підказок рекомендаційних мереж при виборі і у неї зникає стимул вивчати нові незвичні жанри - адже все, що пропонує мережа, їй в принципі подобається. І це може призвести до сумних наслідків або неадекватних вчинків.

Об'єкт дослідження - процеси підбору персоналу ІТ-підприємств.

Предмет дослідження - методи та моделі розробки рекомендаційних систем.

Мета дослідження - розробка рекомендаційної системи управління підбором персоналу.

Для досягнення мети поставлено і вирішено наступні завдання:

1. Проаналізовано підходи до розробки рекомендаційних систем.
2. Розроблено концептуальну модель рекомендаційної системи підбору персоналу.
3. Удосконалено етапи розробки рекомендаційних систем.
4. Розроблено рекомендаційну систему управління підбором персоналу ІТ-підприємств.

Методи дослідження - методи математичної статистики і математичного аналізу, загальнонаукові методи, теорія графів.

Наукова новизна одержаних результатів полягає у наступному:

уточнено:

– принципи проектування та етапи розробки рекомендаційних систем управління підбором персоналу;

– інструментарій покращення функціональних можливостей рекомендаційних систем;

дістали подальшого розвитку:

– теоретичні та методичні засади розробки рекомендаційних систем управління підбором персоналу ІТ-підприємств.

Результати теоретичного аналізу проблеми розробки рекомендаційних систем висвітлено на Міжнародній науково-практичній конференції «Європейський вектор модернізації інженерної та економікоуправлінської освіти в умовах сталого розвитку промислового регіону» та XIV університетській науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених «Молода наука-2021»[1, 2].

РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1. Теоретичні основи рекомендаційних систем

Рекомендаційні системи націлені на те, щоб передбачити інтереси користувачів, які, швидше за все, їм цікаві. Це одні з найпотужніших систем машинного навчання.

Рекомендації зазвичай прискорюють пошук і полегшують користувачам доступ до цікавого для них контенту, а також дивують їх пропозиціями, які вони ніколи б не шукали.

Більш того, компанії можуть залучати й утримувати клієнтів, розсилаючи електронні листи з посиланнями на нові пропозиції, що відповідають інтересам одержувачів. Користувач починає відчувати, що його знають і розуміють, і він з більшою ймовірністю споживе більше контенту. Знаючи, чого хоче користувач, компанія отримує конкурентну перевагу, і загроза втрати клієнта на користь конкурента зменшується. Забезпечення цієї додаткової цінності для користувачів шляхом включення рекомендацій в системи і продукти є привабливим. Крім того, це дозволяє компаніям випереджати своїх конкурентів і в кінцевому підсумку збільшувати свої доходи [3].

Більш того рекомендаційні системи - корисна альтернатива пошуковим алгоритмам, оскільки вони допомагають користувачам знаходити елементи, які інакше вони могли б не знайти. Слід зазначити, що рекомендаційні системи часто реалізуються з використанням пошукових систем, індексуючих нетрадиційні дані.

Рекомендаційні системи були вперше згадані в технічному звіті як «цифрова книжкова полиця» в 1990 році Юссі Карлгрену з Колумбійського університету, а з 1994 року збільшувалися в масштабах і опрацьовувалися в технічних звітах і публікаціях Юссі Карлгрену, потім в SICS, і дослідницькими

групами на чолі з Патті Мейс з Массачусетського технологічного інституту, Уіллом Хіллом з Bellcore, і Полом Резником з Массачусетського технологічного інституту, чия робота з GroupLens була удостоєна нагороди ACM Software Systems Award 2010 року [4].

М. Монтанер, Беатріс Лопес, Естеве в «Таксономія рекомендаційних агентів в Інтернеті» 2003 року подали перший огляд рекомендаційних систем з точки зору інтелектуального агента. А. Тужилін, Г. Адомавічус в своїй роботі «До наступного покоління рекомендаційних систем: огляд сучасного стану та можливих розширень» представили новий альтернативний огляд рекомендаційних систем. Дж. Л. Херлокер в 2004 році надав додатковий огляд методів оцінки рекомендаційних систем [4].

Однією з подій, які стимулювали дослідження в області рекомендаційних систем, стала премія Netflix. З 2006 по 2009 рік Netflix спонсорував конкурс, пропонуючи головний приз в розмірі 1 млн доларів команді, яка могла взяти запропонований набір даних з більш ніж 100 мільйонів оцінок фільмів і видати рекомендації, які були на 10% точніше, ніж ті, які пропонує існуюча система рекомендацій компанії. Цей конкурс стимулював пошук нових і більш точних алгоритмів. 21 вересня 2009 року головний приз був присуджений команді BellKor Pragmatic Chaos за правилами тай-брейка.

Найточніший алгоритм 2007 року використовував метод ансамблю з 107 різних алгоритмічних підходів, об'єднаних в один прогноз.

Завдяки проекту Netflix інтернет отримав безліч переваг. Деякі команди взяли свої технології і застосували їх на інших ринках. Деякі члени команди, що зайняла друге місце, заснували Gravity R&D, механізм рекомендацій, який бере активну участь в співтоваристві RecSys. 4-Tell, Inc. створила рішення на основі проекту Netflix для веб-сайтів електронної комерції.

У зв'язку з набором даних, запропонованим Netflix для конкурсу Netflix Prize, виник ряд проблем з конфіденційністю. Хоча набори даних були анонімними, щоб зберегти конфіденційність клієнтів, в 2007 році два дослідники з Техаського університету змогли ідентифікувати окремих користувачів,

зіставивши набори даних з рейтингами фільмів в базі даних Internet Movie Database. В результаті в грудні 2009 року анонімний користувач Netflix подав до суду на Netflix в справі Доу проти Netflix, стверджуючи, що Netflix порушила закони США про справедливу торгівлю і Закон про захист конфіденційності відео, опублікувавши набори даних. Це, а також побоювання Федеральної торгової комісії, привели до скасування другого конкурсу Netflix Prize у 2010 році [5].

Зазвичай дослідження рекомендаційних систем спрямовані на пошук найбільш точних алгоритмів рекомендацій. Однак є ряд чинників, які також важливі.

Різноманітність - користувачі, як правило, більше задовольняються рекомендаціями, коли існує більша різноманітність всередині списку.

Наполегливість рекомендацій - в деяких ситуаціях більш ефективно повторно показати рекомендації або дозволити користувачам повторно оцінити елементи, ніж показувати нові елементи. На це є кілька причин. Користувачі можуть ігнорувати елементи, коли вони відображаються вперше, наприклад, тому що у них не було часу уважно вивчити рекомендації.

Конфіденційність - рекомендаційним системам зазвичай доводиться мати справу з проблемами конфіденційності, тому що користувачі повинні розкривати конфіденційну інформацію. Створення профілів користувачів з використанням спільної фільтрації може бути проблематичним з точки зору конфіденційності. У багатьох європейських країнах існує сильна культура конфіденційності даних, і кожна спроба ввести будь-який рівень профілювання користувачів може привести до негативної реакції клієнтів. У цій області було проведено багато досліджень з поточних питань конфіденційності. Рамакришнан і ін. провели широкий огляд компромісів між персоналізацією і конфіденційністю і виявили, що поєднання слабких зв'язків (несподіване поєднання, яке дає випадкові рекомендації) та інших джерел даних можна використовувати для розкриття особистості користувачів в анонімізованому наборі даних.

Демографічні дані користувачів - Джоран Біл та ін. виявили, що демографічні дані користувачів можуть впливати на те, наскільки користувачі задоволені своїми рекомендаціями. У своїй статті вони показують, що літні користувачі, як правило, більше зацікавлені в рекомендаціях, ніж молоді.

Надійність - коли користувачі можуть брати участь в рекомендаційній системі, необхідно вирішити проблему шахрайства [6].

Інтуїція - це міра того, «наскільки дивні рекомендації». Наприклад, система рекомендацій, яка рекомендує молоко покупцеві в продуктовому магазині, може бути абсолютно точною, але це не дуже хороша рекомендація, тому що це очевидний товар для покупця. «Інтуїція» служить двом цілям: по-перше, зменшується ймовірність того, що користувачі втратять інтерес через занадто однакового набору варіантів. По-друге, ці елементи необхідні для того, щоб алгоритми вчилися і удосконалювалися [6].

Довіра - рекомендаційна система не має великого значення для користувача, якщо він не довіряє системі. Довіру можна створити за допомогою рекомендаційної системи, пояснивши, як вона генерує рекомендації і чому рекомендує будь-який елемент.

Маркування - на задоволеність користувачів рекомендаціями може впливати маркування рекомендацій. Так, Біл, Джоран, і ін. в «Рекламні і органічні (дослідні роботи) рекомендації і вплив маркування» в роботі кліків (CTR) для рекомендацій помічені як «рекламні» були нижче (CTR = 5,93%), ніж CTR для ідентичних рекомендацій позначені як «органічний» (CTR = 8,86%). Рекомендації без маркування показали себе з найкращого боку (CTR = 9,87%) в цьому дослідженні [7].

1.2. Класифікація рекомендаційних систем

У рекомендаційних системах зазвичай використовується або колаборативна фільтрація, або фільтрація на основі вмісту (також відома як індивідуальний підхід), а також інші системи, такі як системи, засновані на

знаннях. Підходи до колаборативної фільтрації будують модель на основі минулої поведінки користувача (раніше придбані або обрані предмети і/або числові рейтинги, присвоєні цим елементам), а також аналогічні рішення, прийняті іншими користувачами. Потім ця модель використовується для прогнозування елементів (або рейтингів елементів), які можуть бути цікаві користувачеві. Підходи до фільтрації на основі вмісту використовують серію дискретних, попередньо позначених характеристик елемента, щоб рекомендувати додаткові елементи з аналогічними властивостями. Сучасні рекомендаційні системи зазвичай об'єднують один або кілька підходів в гібридну систему.

Коллаборативна фільтрація (collaborative filtering)- рекомендації засновані на історії оцінок як самого користувача, так і інших. У цьому випадку системи розглядають споживачів, оцінки або інтереси яких схожі на ваші [4].

Коллаборативна фільтрація заснована на припущенні, що люди, які погоджувалися в минулому, погодяться в майбутньому, і що їм сподобаються предмети того ж типу, що і в минулому. Система генерує рекомендації, використовуючи тільки інформацію про профілі рейтингу для різних користувачів або товарів. Виявляючи тимчасових користувачів/елементи з історією рейтингів, аналогічної активного користувача або елемента, вони генерують рекомендації, використовуючи це оточення [6].

Методи коллаборативна фільтрація поділяються на засновані на пам'яті і засновані на моделі.

Ключовою перевагою підходу колаборативної фільтрації є те, що він не покладається на контент, аналізований машиною, і, отже, він здатний точно рекомендувати складні елементи, такі як фільми, не вимагаючи «розуміння» самого елемента [6]. Багато алгоритмів використовувалися для вимірювання подібності користувачів або елементів в рекомендаційних системах. Наприклад, підхід k-найближчого сусіда і кореляція Пірсона, вперше реалізована Алленом.

При побудові моделі на основі поведінки користувача часто розрізняють явні і неявні форми збору даних.[7]

Приклади явного збору даних включають наступне:

- просити користувача оцінити елемент по ковзній шкалі;
- просити користувача про пошук;
- просити користувача оцінити колекцію елементів від обраних до найменш улюблених;
- надати користувачеві два предмета і просити його обрати кращий з них;
- попросити користувача створити список предметів, які йому подобаються.

Приклади неявного збору даних включають наступне:

- спостереження за товарами, які користувач переглядає в інтернет-магазині;
- аналіз часу перегляду елемента/користувача;
- ведення обліку товарів, які користувач купує в інтернеті;
- отримання списку елементів, які користувач слухав або дивився на своєму комп'ютері;
- аналізуючи соціальну мережу користувача і виявляючи схожі симпатії і антипатії.

Підходи до колаборативної фільтрації часто страждають від трьох проблем: холодного запуску, масштабованості і розрідженості [8].

Холодний старт: для нового користувача або елемента недостатньо даних, щоб давати точні рекомендації. Одним з найбільш часто використовуваних рішень цієї проблеми є алгоритм «Багатурукий бандит».

Масштабованість: у багатьох середовищах, в яких ці системи дають рекомендації, є мільйони користувачів і продуктів. Таким чином, для розрахунку рекомендацій часто потрібна велика обчислювальна потужність.

Рідкість: кількість товарів, що продаються на великих сайтах електронної комерції, надзвичайно велике. Найактивніші користувачі оцінили тільки невелику частину всієї бази даних. Таким чином, навіть найпопулярніші товари мають дуже мало оцінок.

Один з найвідоміших прикладів коллаборативної фільтрації - це фільтрація за елементами (люди, які купують x , також купують y), алгоритм, популяризованому системою рекомендацій Amazon.com [8].

Багато соціальних мереж спочатку використовували коллаборативну фільтрацію, щоб рекомендувати нових друзів, групи та інші соціальні зв'язки, досліджуючи мережу зв'язків між користувачем і їх друзями. коллаборативна фільтрація як і раніше використовується як частина гібридних систем.

Засновані на контенті (content-based). Цей тип лежить в основі багатьох рекомендаційних систем. На відміну від коллаборативної фільтрації, етап знайомства з користувачем опускається.

Товари та послуги рекомендуються на основі знань про них: жанр, виробник, конкретні функції і т.п. Загалом, застосовують будь-які дані, які можна зібрати.

За таким принципом працюють системи інтернет-магазинів, онлайн-кінотеатрів і інших сервісів. Наприклад, Ivi вибудовує рекомендації по жанрам, країнам-виробникам фільмів, акторів і т.п [9].

Творці платформ використовують цей тип систем, щоб не втратити нових користувачів, даних про яких ще немає. Звідси ж впливають два недоліки: перший час системи діють неточно і потрібно більше часу на реалізацію.

Засновані на знаннях (knowledge-based). Методи фільтрації на основі знань засновані на описі елемента і профілі переваг користувача. Ці методи найкраще підходять для ситуацій, коли є відомі дані про елемент (ім'я, місце розташування, опис і т. д.), але не про користувача.[4]

Засновані на знаннях системи розглядають рекомендацію як проблему класифікації, специфічну для користувача, і вивчають класифікатор для симпатій і антипатій користувача на основі характеристик елемента.

У цій системі ключові слова використовуються для опису елементів, і створюється профіль користувача, щоб вказати тип елемента, який йому подобається. Іншими словами, ці алгоритми намагаються рекомендувати елементи, схожі на ті, які користувач любив в минулому або вивчає в даний час.

Він не покладається на механізм входу користувача в систему для створення цього часто тимчасового профілю. Зокрема, різні елементи-кандидати порівнюються з елементами, раніше оціненими користувачем, і рекомендуються найбільш підходящі елементи. Цей підхід йде корінням в пошук інформації та дослідження фільтрації інформації [7].

Для створення профілю користувача система в основному фокусується на двох типах інформації:

1. Модель переваг користувача.
2. Історія взаємодії користувача з рекомендаційною системою.

По суті, ці методи використовують профіль елемента (набір дискретних атрибутів і функцій), що характеризує елемент в системі. Щоб абстрагуватися від характеристик елементів у системі, застосовується алгоритм представлення елементів.

Широко використовуваним алгоритмом є уявлення $tf - idf$ (також зване представленням у векторному просторі). Система створює профіль користувачів на основі вмісту за допомогою зваженого вектора характеристик елемента. Ваги позначають важливість кожної функції для користувача і можуть бути обчислені на основі індивідуально оцінених векторів контенту з використанням різних методів. Прості підходи використовують середні значення вектора оцінених елементів, в той час як інші складні методи використовують методи машинного навчання, такі як байєсовські класифікатори, кластерний аналіз, дерева рішень і штучні нейронні мережі для оцінки ймовірності того, що користувачеві сподобається товар.

Ключовою проблемою фільтрації на основі вмісту є те, чи може система дізнатися переваги користувача по діям користувачів щодо одного джерела вмісту і використовувати їх в інших типах вмісту. Коли система обмежена рекомендацією контенту того ж типу, який вже використовує користувач, значення з системи рекомендацій значно менше, ніж коли можна рекомендувати інші типи контенту з інших служб. Наприклад, рекомендувати новинні статті на основі перегляду новин корисно, але було б набагато корисніше, коли можна

рекомендувати музику, відео, продукти, обговорення і т. д. з різних служб на основі перегляду новин [10]. Щоб подолати це, більшість рекомендаційних систем, заснованих на утриманні, тепер використовують деяку форму гібридної системи.

Рекомендаційні системи, засновані на змісті, також можуть включати рекомендаційні системи, засновані на думках. У деяких випадках користувачам дозволяється залишати текстові огляди або відгуки про товари. Ці створені користувачем тексти є неявними даними для рекомендаційної системи, оскільки вони потенційно є багатим ресурсом як за характеристиками/аспектам елемента, так і за оцінкою/настрою користувачів по відношенню до елементу. Функції, витягнуті з оглядів, створених користувачами, є поліпшеними метаданими елементів, оскільки вони також відображають такі аспекти елемента, як метадані. Думки, витягнуті з оглядів, можна розглядати як оцінки по відповідним функціям. Популярні підходи рекомендаційної системи, заснованої на думках, використовують різні методи, включаючи інтелектуальний аналіз тексту, пошук інформації, аналіз тональності і глибоке навчання [11].

Насправді кількість підтипів обмежена фантазією творців. При реалізації нового проекту в залежності від сфери діяльності в рекомендаційну систему можна закласти будь-яку предметну область і ранжувати за нею.

Рекомендаційні системи на основі сеансів. Ці рекомендаційні системи використовують взаємодії користувача в рамках сеансу для вироблення рекомендацій. Рекомендаційні системи на основі сеансів використовуються на Youtube і Amazon. Вони особливо корисні, коли історія (наприклад, минулі кліки, покупки) користувача недоступна або не актуальна в поточному сеансі користувача. Області, в яких рекомендації на основі сеансів особливо актуальні, включають відео, електронну комерцію, подорожі, музику і багато іншого. Більшість екземплярів рекомендаційних систем на основі сеансу покладаються на послідовність недавніх взаємодій в рамках сеансу, не вимагаючи будь-яких додаткових відомостей (історичних, демографічних) про користувача [11].

Багатокритеріальні рекомендаційні системи. Багатокритеріальні рекомендаційні системи (MCRS) можна визначити як рекомендаційні системи, які включають інформацію про переваги по безлічі критеріїв. Замість розробки методів рекомендацій, заснованих на єдиному значенні критерію, загальній перевазі користувача u для елемента i , ці системи намагаються передбачити рейтинг для невивчених елементів u , використовуючи інформацію про переваги по безлічі критеріїв, які впливають на це загальне значення переваги.

Системи рекомендацій з урахуванням ризиків. Більшість існуючих підходів до рекомендаційних систем зосереджені на тому, щоб рекомендувати користувачам найбільш релевантний контент з використанням контекстної інформації, але при цьому не беруть до уваги ризик потурбувати користувача небажаними повідомленнями. Важливо враховувати ризик засмутити користувача, висуваючи рекомендації при певних обставинах, наприклад, під час професійної зустрічі, рано вранці або пізно вночі. Отже, ефективність рекомендаційної системи частково залежить від ступеня, в якій вона врахувала ризик в процесі рекомендацій. Одним з варіантів вирішення цієї проблеми є система DRARS, яка моделює контекстно залежну рекомендацію як проблему бандита. Ця система поєднує в собі контентну техніку і контекстний алгоритм бандитів [12].

Мобільні рекомендаційні системи. Мобільні рекомендаційні системи використовують смартфони з доступом в Інтернет, щоб пропонувати персоналізовані, контекстно-залежні рекомендації. Це особливо складна область дослідження, оскільки мобільні дані більш складні, ніж дані, з якими часто доводиться мати справу рекомендаційним системам. Вони неоднорідні, зашумлені, вимагають просторової і тимчасової автокореляції і мають проблеми з валідацією і спільністю.

Є три фактори, які можуть вплинути на мобільні рекомендаційні системи і точність результатів прогнозування: контекст, метод рекомендації і конфіденційність. Крім того, мобільні рекомендаційні системи страждають від проблеми трансплантації - рекомендації можуть застосовуватися не у всіх

регіонах (наприклад, було б нерозумно рекомендувати рецепт в області, де всі інгредієнти можуть бути недоступні).

Одним із прикладів мобільного рекомендаційної системи є підходи, що застосовуються такими компаніями, як Uber і Lyft, для створення маршрутів руху для водіїв таксі в місті. Ця система використовує дані GPS про маршрути, які таксисти обирають під час роботи, включаючи місце розташування (широта і довгота), позначки часу і робочий стан (з пасажиром або без них). Він використовує ці дані, щоб рекомендувати список пунктів зупинок на маршруті з метою оптимізації часу зайнятості і прибутку.

Гібридні рекомендаційні системи. У більшості рекомендаційних систем зараз використовується гібридний підхід, що поєднує колаборативну фільтрацію, фільтрацію на основі вмісту і інші підходи. Немає ніяких причин, за якими не можна було б гібридизувати кілька різних методів одного і того ж типу. Гібридні підходи можуть бути реалізовані кількома способами: шляхом роздільного прогнозування на основі контенту і на основі колаборативної фільтрації з подальшим їх об'єднанням; шляхом додавання можливостей, заснованих на утриманні, до підходу колаборативної фільтрації (і навпаки); або шляхом об'єднання підходів в одну модель. Кілька досліджень, які емпірично порівнюють продуктивність гібрида з чистими методами спільної роботи і методами, заснованими на утриманні, та продемонстрували, що гібридні методи можуть дати більш точні рекомендації, ніж чисті підходи. Ці методи також можуть використовуватися для подолання деяких загальних проблем в рекомендаційних системах, таких як холодний запуск і проблема розрідженості, а також вузьке місце інженерії знань в підходах, заснованих на знаннях.

Netflix - хороший приклад використання гібридних рекомендаційних систем. Веб-сайт дає рекомендації, порівнюючи звички перегляду і пошуку схожих користувачів, а також пропонує фільми, які мають спільні характеристики з фільмами, які отримали високу оцінку користувача (фільтрація на основі контенту) [5].

Деякі методи гібридизації включають:

Зважений: чисельне об'єднання оцінок різних компонентів рекомендації.

Перемикання: вибір серед рекомендованих компонентів і застосування обраного.

Змішаний: рекомендації від різних рекомендувачів представлені разом, щоб дати рекомендацію.

Комбінація функцій: функції, отримані з різних джерел знань, об'єднуються і передаються в єдиний алгоритм рекомендацій.

Розширення функцій: обчислення функції або набору функцій, які потім є частиною вихідних даних для наступного методу.

Каскад: рекомендаціям віддається строгий пріоритет, при цьому ті, які мають більш низький пріоритет, розривають нічию при підрахунку очок з більш високими [13].

Мета-рівень: застосовується один рекомендаційний метод і створюється свого роду модель, яка потім використовується в наступній техніці.

1.3. Показники ефективності рекомендаційних систем

Процес оцінювання важливий для ефективності алгоритмів рекомендацій. Для вимірювання ефективності рекомендаційних систем і порівняння різних підходів доступні три типи оцінок: дослідження користувача, онлайн-оцінки (A/B-тести) і офлайн-оцінки.

Зазвичай використовуються метрики - це середньоквадратична помилка і коренева середньоквадратична помилка, остання з яких використовувалася в конкурсі Netflix Prize. Метрики пошуку інформації, такі як точність і відгук або DCG, корисні для оцінки якості методу рекомендацій. Різноманітність, новизна і охоплення також вважаються важливими аспектами оцінки. Однак багато класичних методів оцінки піддаються різкій критиці [14].

Оцінка продуктивності алгоритму рекомендацій на фіксованому наборі тестових даних завжди буде надзвичайно складним завданням, оскільки неможливо точно передбачити реакцію реальних користувачів на рекомендації.

Отже, будь-яка метрика, яка обчислює ефективність алгоритму автономних даних, буде неточною.

Дослідження користувачів досить малі. Кільком десяткам або сотням користувачів представлені рекомендації, створені за допомогою різних підходів до рекомендацій, а потім користувачі вирішують, які рекомендації є кращими.

В А/В-тестах рекомендації зазвичай показуються тисячам користувачів реального продукту, і рекомендаційна система випадковим чином вибирає принаймні два різних підходи до рекомендацій для генерації рекомендацій. Ефективність вимірюється неявними заходами ефективності, такими як коефіцієнт конверсії або рейтинг кліків [14].

Автономні оцінки засновані на історичних даних, наприклад на наборі даних, який містить інформацію про те, як користувачі раніше оцінювали фільми. Потім ефективність рекомендаційних підходів вимірюється на основі того, наскільки добре рекомендаційний підхід може передбачити рейтинги користувачів в наборі даних. Хоча рейтинг є явним виразом того, чи сподобався фільм користувачеві, така інформація може бути недоступною в доменах.

Наприклад, в області систем рекомендацій по цитуванню користувачі зазвичай не оцінюють цитування або рекомендовану статтю. У таких випадках офлайн-оцінки можуть використовувати неявні заходи ефективності. Наприклад, можна припустити, що ефективна рекомендаційна система, здатна рекомендувати якомога більше статей, що містяться в списку посилань на дослідницьку статтю. Однак багато дослідників вважають подібні офлайн-оцінки критичними. Наприклад, було показано, що результати автономних оцінок мають низьку кореляцію з результатами призначених для користувача досліджень або А/В-тестів. Було показано, що набір даних, популярний для автономної оцінки, містить дані, що повторюються і, таким чином, призводить до неправильних висновків при оцінці алгоритмів[15].

Часто результати так званих автономних оцінок не корелюють з фактично оціненою ступенем задоволеності користувачів. Ймовірно, це пов'язано з тим, що автономне навчання сильно зміщене в бік високодоступних елементів, а дані

автономного тестування сильно залежать від результатів роботи модуля онлайн-рекомендацій. Дослідники прийшли до висновку, що до результатів офлайн-оцінок слід ставитися критично.

Як відомо, рекомендаційні системи складно оцінити в автономному режимі, і деякі дослідники стверджують, що це призвело до кризи відтворюваності в публікаціях рекомендаційних систем. Недавній огляд невеликого числа вибраних публікацій, які застосовують методи глибокого навчання або нейронних методів для вирішення топ-к проблем рекомендацій, опублікованих на провідних конференціях (SIGIR, KDD, WWW, RecSys, IJCAI), показав, що в середньому менше 40% статті могли бути відтворені авторами опитування, на деяких форумах їх частка становила всього 14%. В цілому дослідження виявили 26 статей, тільки 12 з них могли бути відтворені авторами, а 11 з них могли перевершити набагато старіші і більш прості, правильно налаштовані вихідні дані про показники автономної оцінки [16].

У статтях також висвітлюється ряд потенційних проблем в сучасній дослідницькій науці і міститься заклик до вдосконалення наукової практики в цій галузі. Аналогічна стаття тієї ж групи була опублікована про рекомендаційні системи, які враховують послідовність. Пізніша робота з тестування набору тих же методів призвела до якісно дуже різних результатів результати чого нейронні методи були визнані одними з найефективніших.

Глибоке навчання і нейронні методи для рекомендаційних систем використовувалися в успішних рішеннях в декількох недавніх завданнях рекомендаційних систем, WSDM, RecSys Challenge. Крім того, нейронні методи і методи глибокого навчання широко використовуються в промисловості, де вони ретельно тестуються.

Тема відтворюваності не нова в рекомендаційних системах. До 2011 року Екстранд, Констан і ін. критикували, що «в даний час важко відтворити і розширити результати досліджень рекомендаційних систем», і що оцінки «не обробляються послідовно». Констан і Адомавічус приходять до висновку, що «дослідницьке співтовариство рекомендує систем стикається з кризою, коли

значна кількість статей представляє результати, які мало сприяють накопиченню колективних знань [...] часто через те, що дослідженню не вистачає [...] оцінки для правильної оцінки і, отже, для внесення значимого вкладу «Як наслідок, багато досліджень рекомендаційних систем можна вважати невідтворюваними. Отже, оператори рекомендаційних систем знаходять в поточних дослідженнях мало вказівок для відповіді на питання, які рекомендаційні підходи використовувати в рекомендаційних системах [17].

Said & Bellogín провели дослідження статей, опублікованих в цій області, а також провели порівняльний аналіз деяких з найбільш популярних структур для рекомендацій і виявили великі невідповідності в результатах, навіть коли використовувалися одні і ті ж алгоритми і набори даних. Деякі дослідники продемонстрували, що незначні зміни в алгоритмах рекомендацій або сценаріях призводять до сильних змін в ефективності рекомендаційної системи.

Висновки до розділу 1

Рекомендаційні системи - системи фільтрації інформації, яка буде рейтинг об'єктів корисних користувачу.

У рекомендаційних системах використовують колаборативну фільтрацію, фільтрацію на основі вмісту та інші системи.

Встановлено, що при дослідженні рекомендаційних систем на предмет більш точних алгоритмів рекомендацій важливу роль грають такі чинники як: різноманітність, наполегливість, демографічні дані користувача, конфіденційність, надійність, довіра, інтуїція.

Коллаборативна фільтрація будує модель на основі поведінки користувача та аналогічних рішень прийнятих іншими користувачами. Коллаборативна фільтрація заснована на припущенні, що люди через деякий час погодяться, що їм сподобаються предмети того ж типу, що і в минулому.

Перевагою такого підходу є те, що він не покладається на контент аналізований машиною, а тому здатний рекомендувати складні елементи не вимагаючи «розуміння» самого елемента.

Однак, підхід колаборативної фільтрації часто страждає від 3 проблем : холодний старт, масштабованість, розрідженість. Відомий приклад колаборативної фільтрації - фільтрація за елементами (люди які купують x, також купують y), алгоритм, популяризований системою рекомендацій Amazon.com.

Визначено, що фільтрація на основі вмісту використовує дискретні, попередні характеристики елемента, аби рекомендувати елементи з аналогічними властивостями. Ключовою проблемою фільтрації на основі вмісту є те, чи може система дізнатись переваги користувача по діям користувачів щодо одного джерела вмісту і використовувати їх в інших типах вмісту [18].

Коли система обмежена рекомендацією контенту того ж типу, який вже використовує користувач, значення з системи рекомендацій значно менше, ніж коли можна рекомендувати інші типи контенту з інших служб. Щоб подолати це, більшість рекомендаційних систем, заснованих на утриманні, тепер використовують форму гібридної системи.

У більшості рекомендаційних систем зараз використовується гібридний підхід, що поєднує колаборативну фільтрацію, фільтрацію на основі вмісту і інші підходи. Гібридні підходи можуть бути реалізовані кількома способами: шляхом роздільного прогнозування на основі контенту і на основі колаборативної фільтрації з подальшим їх об'єднанням. шляхом додавання можливостей, заснованих на утриманні, до підходу колаборативної фільтрації (і навпаки); або шляхом об'єднання підходів в одну модель.

РОЗДІЛ 2

РОЗРОБКА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПІДБОРОМ ПЕРСОНАЛУ ІТ-ПІДПРИЄМСТВ

2.1. Концепція рекомендаційної системи управління підбором персоналу ІТ-підприємств

Рекомендаційна система - це підклас системи фільтрації інформації, яка прагне передбачити «оцінку» або «перевагу», яку користувач надасть елементу.

Системи рекомендацій використовуються в різних сферах, причому загальновідомі приклади мають форму генераторів списків відтворення для відео- та музичних сервісів, рекомендацій продуктів для онлайн-магазинів або рекомендацій вмісту для платформ соціальних мереж і рекомендацій відкритого веб-контенту. Системи рекомендацій також були розроблені для вивчення дослідницьких статей і експертів,[17] співробітників[18] та фінансових служб[17].

Оскільки рекомендаційна система це велика програма, яка містить у собі багато файлів, кожен з яких відповідає за окремий елемент у системі, то потрібно розробити передусім концептуальну схему. Розроблену концептуальну схему рекомендаційної системи управління підбором персоналу ІТ-підприємств представлено на рис. 2.1.

Розглянемо детальніше зміст і функціональне призначення її основних структурних елементів.

В блоку «Збір даних про користувача» отримується інформація про користувача, його вподобання завдяки опитуванню та пошуку в інших джерелах. Потім певним чином формуються в первинну інформацію для подальшої обробки.

Блок «Модуль аналізу даних про користувача» обробляє інформацію отриману від попереднього блоку. «Пошук дублікатів» із первинної інформації

виключає інформацію яка дублюється або має схоже значення. Потім «Групування» структурує інформацію певним чином. Завдяки «Пошуку асоціативних правил» між інформацією про користувача створюються зв'язки та надається значення. В кінці якщо потрібно можна повторити процес.

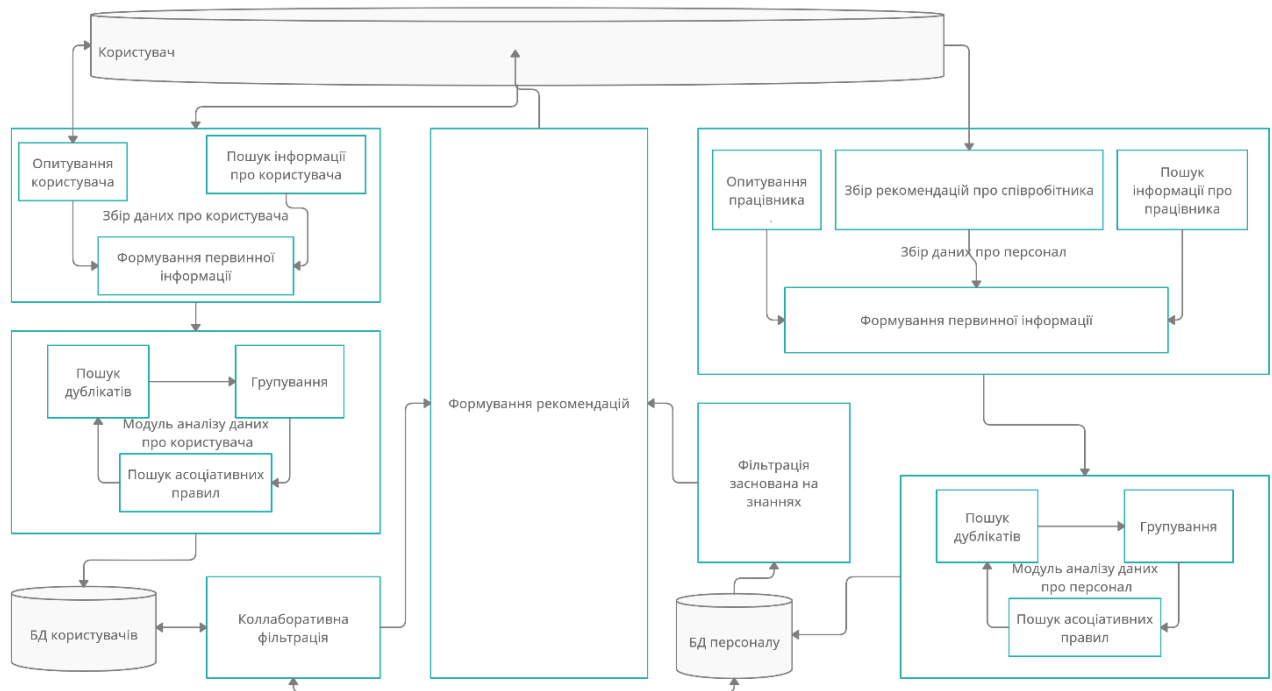


Рис. 2.1. Концептуальна модель рекомендаційної системи управління підбором персоналу ІТ-підприємств

Отримана інформація від блоку «Модуль аналізу даних про користувача» вноситься до бази даних користувачів для того, щоб в подальшому, завдяки колаборативній фільтрації, можна було запропонувати рекомендації щодо персоналу.

Блоки «Збір даних про персонал» та «Модуль аналізу даних про персонал» подібні до «Збір даних про користувача» і «Модуль аналізу даних про користувача» відповідно, але збирають і оброблюють інформацію про працівника ІТ-підприємства, а не про користувача даної програми.

В блоку «Збір даних про персонал» отримується інформація про працівника, шляхом його опитування, пошуку інформації про нього в інших джерелах та завдяки оцінкам інших користувачів з якими він працював. Далі

інформація перетворюється в первинну інформацію для подальшого використання.

Блок «Модуль аналізу даних про персонал» обробляє первинну інформацію про працівника, яка була отримана від попереднього блоку. Завдяки «Пошуку дублікатів» із первинної інформації виключається інформація яка дублюється. «Групування» структурує інформацію певним чином. Потім «Пошук асоціативних правил» між інформацією про працівника створює зв'язки та надає їм значення.

На основі обробленої інформації про працівників створюється база даних персоналу і завдяки «Фільтрації заснованої на знаннях» отримуються певні дані для подальшого формування рекомендацій про підбор персоналу.

Блок «Формування рекомендацій» отримує дані від «Колаборативної фільтрації» і «Фільтрації заснованої на знаннях», обробляє, аналізує та видає рекомендації щодо працівника.

2.2. Обґрунтування необхідності використання програмного забезпечення для підбору кадрів

Управління персоналом - сфера діяльності керівного складу організації, керівників та спеціалістів підрозділів системи управління персоналом, спрямована на підвищення ефективності роботи організації за рахунок підвищення ефективності роботи з її співробітниками психологічними, правовими, економічними та іншими методами[19].

Одним із перших етапів у процесі управління персоналом є підбір кадрів. Від того, як підібрані люди для роботи в компанії, залежить ефективність їхньої подальшої діяльності, а також отриманий результат. Саме тому кожного роботодавця цікавить, як підібрати кваліфікаційного фахівця, за яким принципом проводити відбір, за якими критеріям його оцінювати і як зробити правильний вибір.

Першим та головним етапом підбору є визначення вимог до спеціалісту: досвід роботи, володіння тими чи іншими навичками, а також ряд необхідних компетенцій, якими має мати майбутній фахівець.

Основні етапи підбору персоналу:

1. Визначення потреб роботодавця у персоналі.
2. Визначення вимог до кандидата.
3. Підбір кандидата з резюме (аналіз анкет, співбесіда).
4. Пропозиція посад роботодавцю.
5. Укладання договору із претендентом.

Підбір персоналу є встановлення відповідності характеристик працівника та вимог організації, посади.

Відбір персоналу - це процес вивчення психологічних та професіональних якостей працівника з метою встановлення його придатності для виконня обов'язків на певному робочому місці або посади та вибору із сукупності претендентів найбільш відповідного з урахуванням відповідності встановлених вимог[20].

Підвищення ефективності процесів організації передбачає впровадження рекомендаційної системи.

Під час здійснення підбору персоналу фахівці обробляють велику кількість документів: заявки на підбір, резюме, анкети кандидатів, висновки за результатами співбесіди. Більшість роботи співробітники компаній виконують у стандартних офісних програмах MS Word, MS Excel.

Кількість заявок на підбір від підприємств-замовників може збільшуватись, що ускладнює роботу фахівців: збільшуються тимчасові ресурси на пошук спеціалістів з існуючої на підприємстві бази кандидатів, висновків за результатами співбесід з кандидатами, підвищуються фінансові витрати у зв'язку з розміщенням оголошень по повторюваних вакансій.

У зв'язку з цим, для підвищення ефективності необхідне програмне забезпечення (ПЗ) вузько орієнтоване на підбір персоналу.

Впровадження рекомендаційної системи на підприємстві значно покращить:

- швидкий пошук кандидатів;
- автоматизовані введення інформації за первинними даними;
- наочність подання інформації;
- зменшення ймовірності виникнення помилок у роботі;
- планування робіт та організацію взаємодії всіх членів колектива;
- формування звітності.

Таким чином, очевидні плюси впровадження системи дозволять оптимізувати систему підбору персоналу, залучити нових клієнтів для компанії.

В даний час існує велика кількість різних програмних продуктів, які спрямовані на оптимізацію діяльності компаній, незалежно від виробництва послуг.

Рекомендаційна система, яка має на увазі здійснення ефективної діяльності підприємств та бізнесу на всіх рівнях, має відповідати певним вимогам:

- зручне використання, що дозволяє контролювати всю інформацію підприємства;
- високий ступінь надійності, що надає можливість зберігати наявну інформацію;
- надійний захист інформації.

Потужним інструментом для роботи з інформацією є автоматизована рекомендаційна система. Система спрощує роботу підприємства та оперативно надає якісні отримані дані. Автоматизовані системи представляють функціонал, необхідний для виконання дій, а також для контролю, управління та аналізу роботи співробітників, підрозділів та підприємства в цілому[17].

Облік та управління є складним процесом підприємства, і відсутність рекомендаційної системи може призвести до цілого комплексу порушень та проблем. Саме впровадження автоматизованої системи дозволяє побудувати весь процес якісно.

2.3. Аналіз типових програмних рішень управління підбором персоналу ІТ-підприємств

Сьогодні часто шукачі та HR-менеджери використовують сайти для підбору персоналу. Пошук роботи через Інтернет сьогодні дуже популярний, так як дозволяє багаторазово скоротити час, уникнути простою в чергах та підготовці документів, так як можна знайти цікаву вакансію та відгукнутися на неї або написати листа в організацію, відправивши електронною поштою резюме і коротко розповівши про себе. З огляду популярності пошуку роботи через Інтернет, кожна організація сьогодні використовує сайти для пошуку роботи.

Work.ua. Work.ua - найбільший в Україні сайт для пошуку співробітників. Тут ви знайдете більше 66 000 вакансій та 2 468 500 резюме. У пошуку роботи щомісяця понад 3,5 млн осіб відвідують цей сайт, понад 115 тис. роботодавців розміщують тут свої вакансії. Після реєстрації рекрутер отримує безкоштовну пробну публікацію. За наступну вакансію доведеться заплатити за одне оголошення[21].

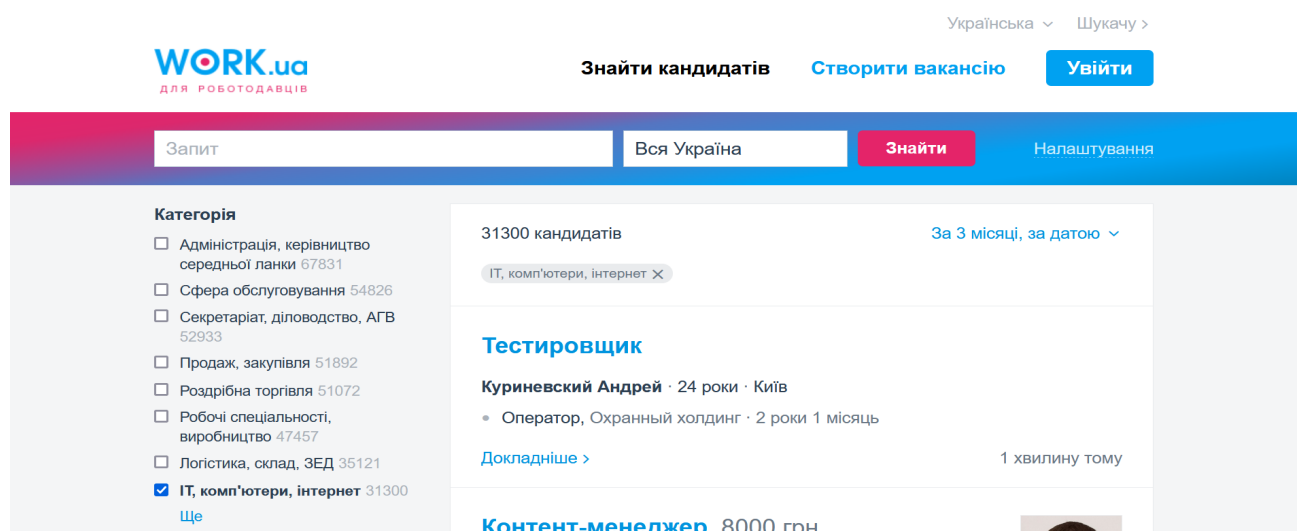


Рис. 2.2. Інтерфейс сайту work.ua

Rabota.ua. Rabota.ua - перший сайт пошуку персоналу, який 19 років тому почав працювати на просторах українського інтернету. 75% українських інтернет-користувачів обирають саме цей майданчик для пошуку роботи. Для

додавання першої вакансії рекрутеру необхідно також зареєструватися як роботодавець. Коли на таку тестову вакансію ви отримаєте 10 відгуків або mine один місяць з моменту її розміщення, вона автоматично завершиться[22].

Легко создавайте аккаунт
Для этого вам нужно лишь подтвердить номер телефона и указать краткую информацию о вашей компании

Публикуйте бесплатную вакансию
Каждый пользователь раз в месяц получает бесплатную публикацию, которая позволит лично оценить возможности roboto.ua

Получайте отклики кандидатов
Соискатели увидят вашу вакансию на сайте, в приложении и рассылках и начнут присылать свои резюме

101 934 актуальных вакансий от 42 115 компаний ◆ 3 917 410 кандидатов доверили нам собственное резюме

3,9 миллиона кандидатов ждут вас в базе резюме от roboto.ua

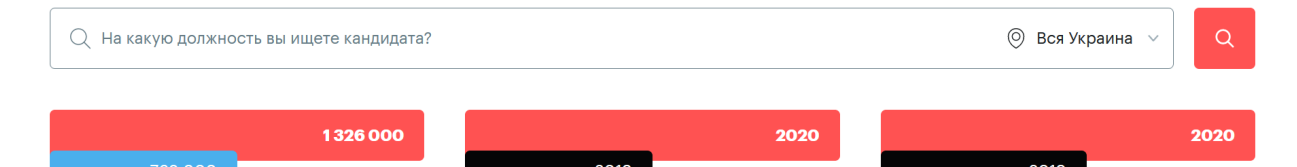


Рис. 2.3. Пошук на сайті Rabota.ua

Grc.ua. Grc.ua - щомісяця відвідує близько 1,6 млн. відвідувачів. Співпрацюють із понад 70 тисячами найкращих компаній у напрямку підбору персоналу у 28 професійних сферах. Тут ви знайдете понад 1,2 млн. резюме[23].

Doc.ua. Є спеціалізовані ресурси, головна ціль яких не стати сайтом пошуку персоналу. На них розміщуються колонки експертів, аналітичні статті, репортажі та додатково вакансії у вузьких сферах.

Doc.ua - найбільша українська спільнота програмістів. На сайті зареєстровано понад 272 тисячі користувачів. За місяць - близько 7 мільйонів переглядів сторінок видання. Вакансії можна знайти у розділі Робота. У середньому тут публікується 5000 вакансій на місяць[24].

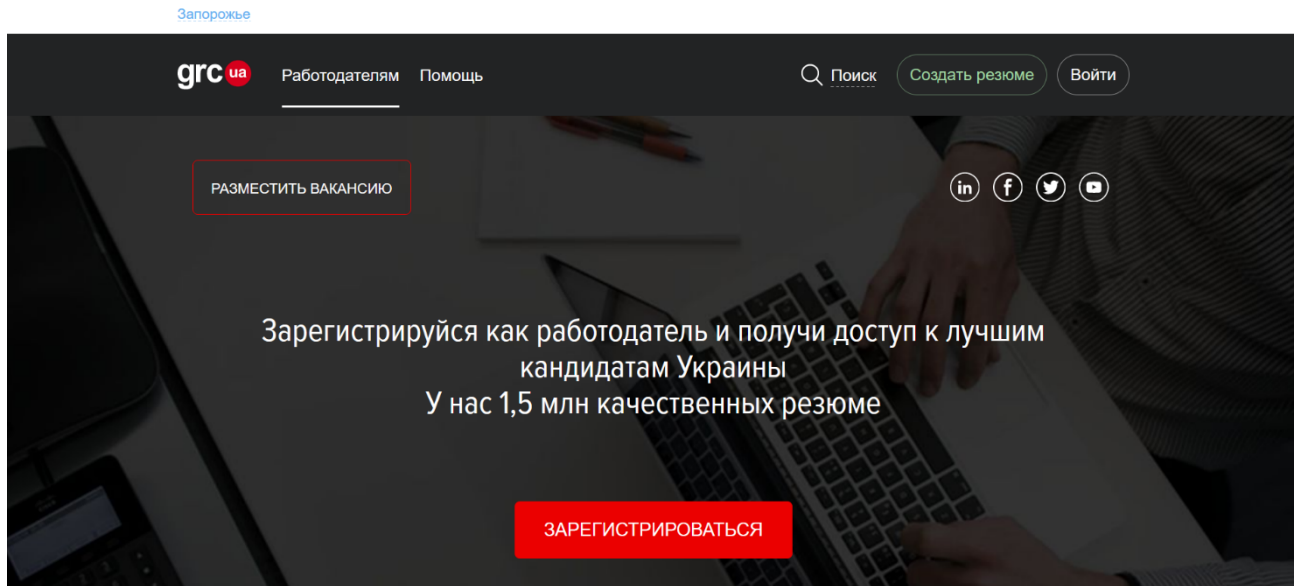


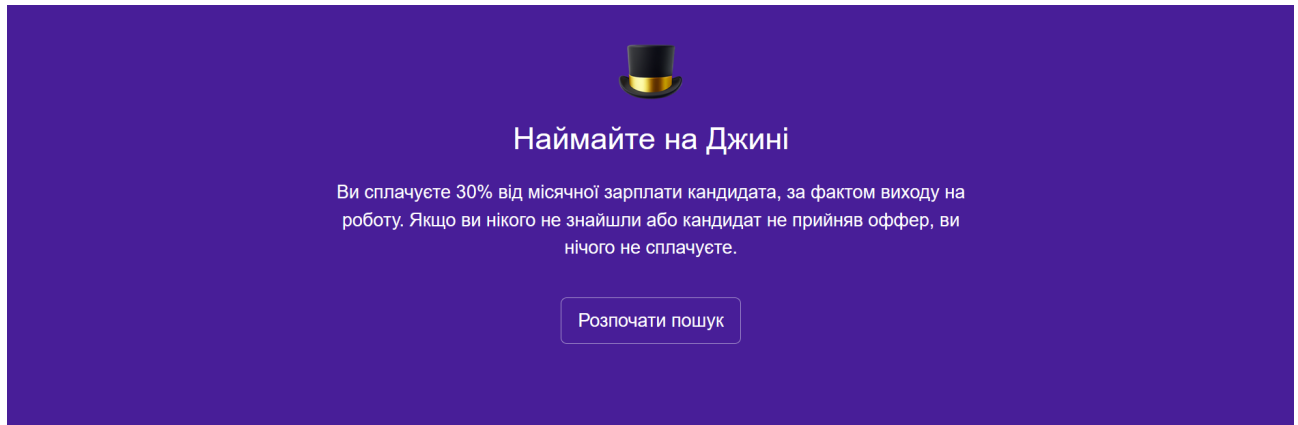
Рис. 2.4. Интерфейс сайта підбора персоналу Grc.ua



Рис. 2.5. Вигляд інтерфейса сайта Dou.ua

Djinni.co. Знайти спеціалістів у певних сферах можна також на не зовсім традиційних сайтах пошуку співробітників. Одним із таких є Djinni.co. Це обслуговування анонімного пошуку роботи для програмістів. Щомісяця сайтом користуються десятки тисяч програмістів та кілька тисяч роботодавців, від лідерів ринку до стартапів. Тут діє дуже цікава система взаємодії. Ось як автори сайту роз'яснюють умови розміщення: «Для кандидатів послуги порталу безкоштовні. Як роботодавець, ви платите бонус вже за фактом найму, тобто коли кандидат прийняв ваш оффер та вийшов на роботу. Цей бонус дорівнює

25% зарплатних очікувань кандидата із профілю на Джині. Наприклад, найм Android розробника, у якого в профілі вказана зарплата \$2000 на місяць, коштуватиме компанії \$500. Розмір бонусу завжди можна побачити на сторінці профілю кандидата у формі «Запропонувати вакансію». Сума бонусу фіксується у момент вашого першого контакту на Джині»[25].



magic@djinni.co

[Зарплати](#)

[Умови користування](#)

[Privacy](#)

Вакансії за містом

[Київ](#)

[Харків](#)

[Львів](#)

За спеціальністю

[iOS](#)

[Android](#)

[C++](#)

[JavaScript](#)

[PHP](#)

[Python](#)

Рис. 2.6. Інтерфейс сайту підбора персоналу Djinni.co

Інші сайти управління підбором персоналу. Це не лідери рейтингів, проте вони теж мають свою аудиторію і варто спробувати пошукати кандидатів там. Тим більше, що на цих сайтах пошуку співробітників безкоштовні опції трапляються частіше, ніж зазвичай.

<https://novarobota.ua/> - тут ви можете безкоштовно розмістити 5 вакансій[26].

<https://trud.ua/> - допускається безкоштовне розміщення 1 вакансії. Також є варіант публікації вакансії безкоштовно, але платити доведеться за можливість переглянути відгуки на неї або вносити оплату за розміщення вакансії щодня[27].

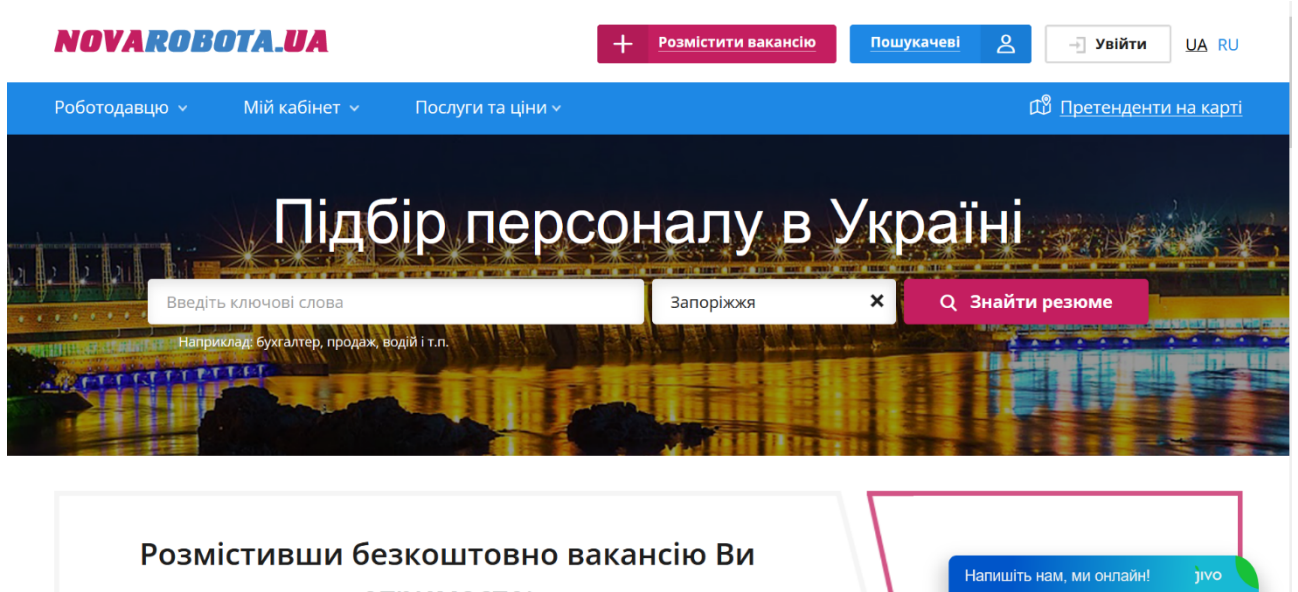


Рис. 2.7. Сторінка novarobota.ua

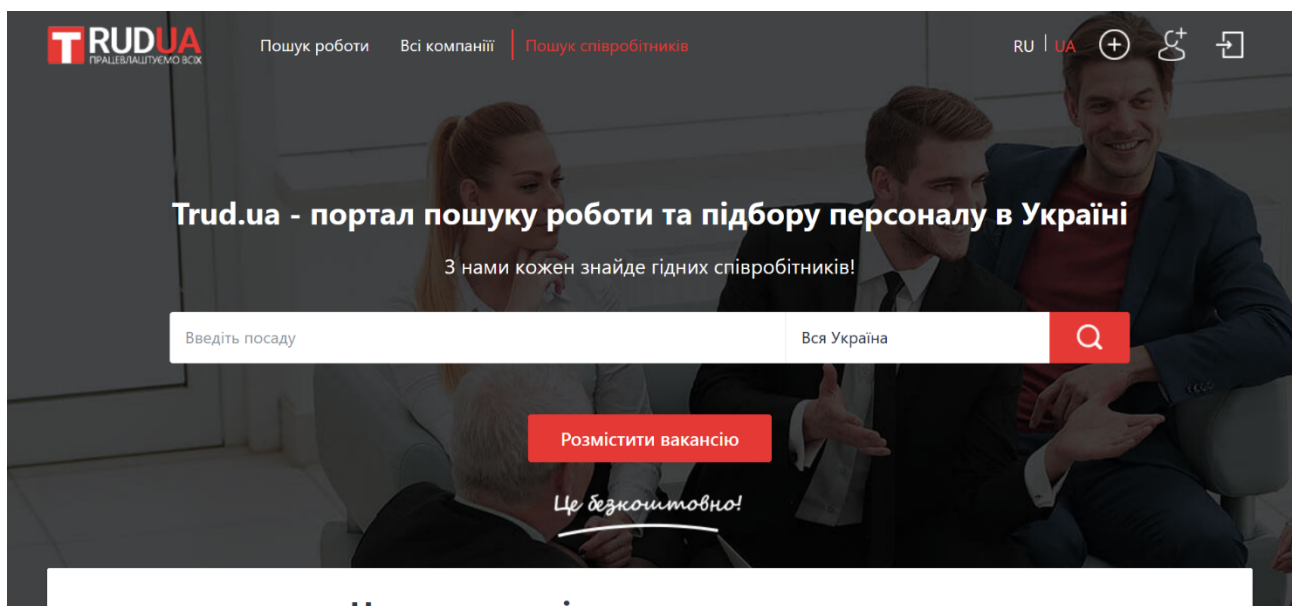


Рис. 2.8. Початкова сторінка сайту trud.ua

<https://ua.jooble.org/> - тут передбачено безкоштовне розміщення вакансій. Але є платний преміум-пакет, який збільшує охоплення і виводить ваше оголошення в топ пошуку. На Jooble ви платите не за розміщення, а за просування публікацій[28].

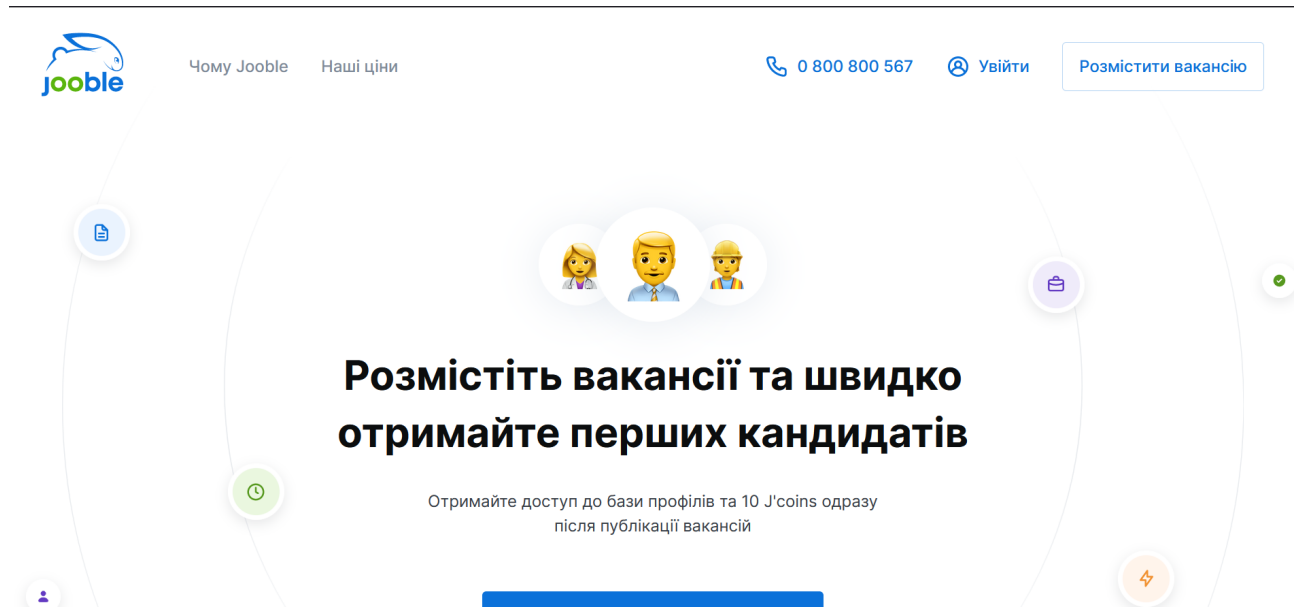


Рис. 2.9. Інтерфейс сайту з підбору персонала Jooble

Огляд систем для управління підбором персоналу. Сьогодні на ринку існує багато програм, так чи інакше пов'язаних із підбором персоналу та управлінням персоналом. Програмні рішення різняться за функціоналом, структурою.

Розглянемо кілька варіантів, проаналізуємо переваги та недоліки.

Програма «FriendWork Recruiter». «FriendWork Recruiter» - це сервіс для автоматизації підбору персоналу та спрощення роботи відділу кадрів. За словами розробників, «FriendWork Recruiter» дозволяє рекрутерам прискорити свою роботу до 30% і пропонує керівництву інструменти для кращого керування підбором персоналу, що так необхідно у великих компаніях[29].

«FriendWork Recruiter» дозволяє вирішити ряд наступних завдань:

- імпорт резюме «одним кліком» із сайтів з пошуку роботи, LinkedIn або із файлів;
- зберігання історії листування з кандидатами;
- керування базою кандидатів;
- отримання доступу до нового джерела претендентів на розширення бази;

– пошук профілів кандидатів «Facebook», «LinkedIn»; сервіси, з якими реалізована інтеграція: «GitHub», «Rabota.UA», «Linkedin», «Indeed», «ІС:Підприємство 8», «Facebook», «Avito».

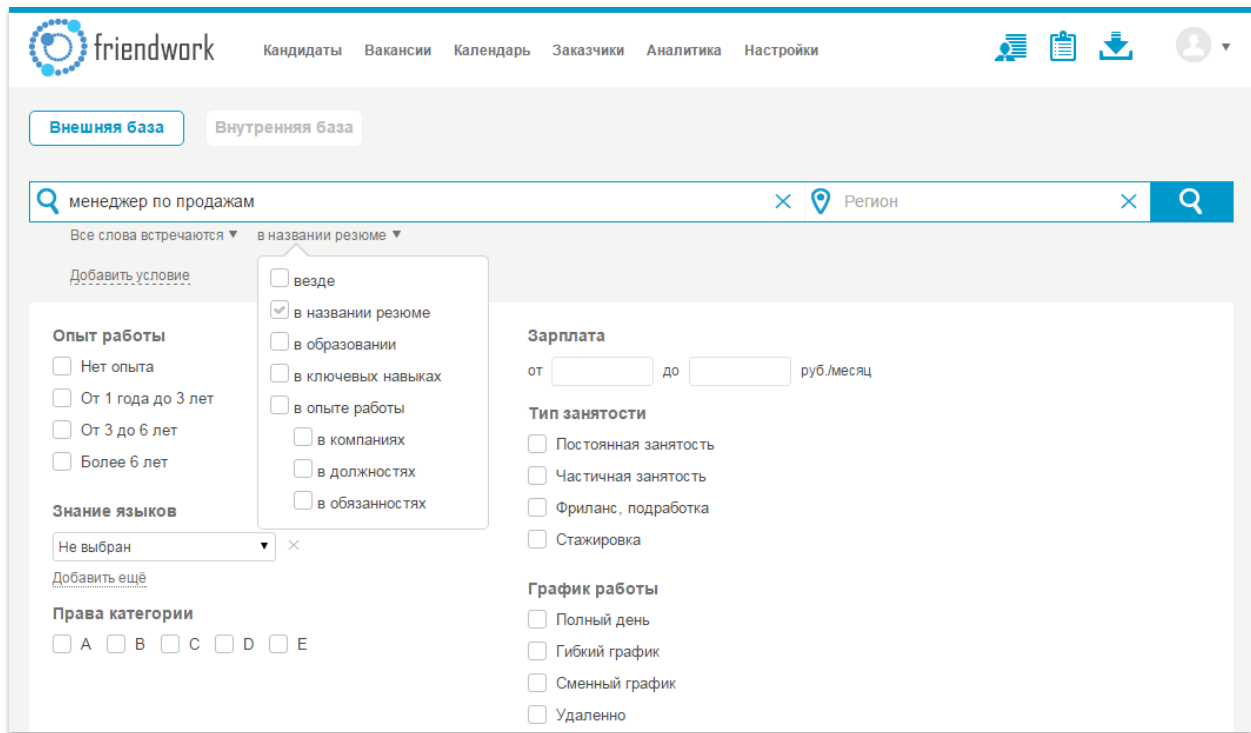


Рис. 2.10. Інтерфейс сервісу FriendWork Recruiter

Програма «Резюмекс». Призначення додатку - управління даними та відстеження інформації про організацію, професійний досвід та навички співробітників. Предметна область автоматизує підбір персоналу в корпорації або кадровому агентстві, що бажає покращити комунікацію між рекрутерами та начальниками відділів.

Можливості:

- вести облік кандидатів та вакансії;
- організація колективної роботи з електронною поштою;
- імпорт резюме, що приходять;
- нагадування про майбутні співбесіди;
- створення на веб-сайті розділу підбір персоналу.

Переваги:

- може бути одночасно використано у кількох офісах;
- автоматизує процес конкурсного відбору кандидатів;

Недоліки:

- складний інтерфейс;
- відсутність супровідної документації.

Програма «E-Staff Рекрутер». Призначення: автоматичне розміщення вакансій, пошук резюме на професійних порталах. Програму можна використовувати як комплексну програму для співробітника кадрового відділу, так і універсальний інструмент для великих кадрових агенцій[30].

Можливості програми:

- автоматичне відправлення резюме та повідомлень;
- ведення статистики щодо роботи з рекруторською компанією;
- ведення статистики щодо зайнятості, результатів діяльності працівників компанії;
- імпорт резюме кандидатів (здобувачів) з різних поштових сервісів;
- інтеграція з такими системами, як «Бос-Кадровик», «1С», «WebSoft», «WebTutor».

Переваги:

- оперативна технічна підтримка;
- зручний інтерфейс;
- база даних (БД) зберігається на комп'ютері, картки кандидатів зберігаються у вигляді *.xml файлів;
- збереження даних.

Недоліки:

- низька якість пошуку;
- відсутність здатності виключати оброблених у попередніх запросах кандидатів із результатів нового пошуку.

The screenshot shows the E-Staff Recruiter interface. At the top, there are candidate names: Савченко Ирина Геннадиевна, Хряпина Ольга Васильевна, and Данилова Анастасия. The main content area is divided into a sidebar and a main panel. The sidebar contains navigation menus for 'Org. структура', 'Вакансии', 'Кандидаты', and 'Контрагенты'. The main panel displays a table of job openings with columns: 'Наименование', 'Структурное подразделение', 'Кол-во позиц.', 'Кол-во открыт. позиц.', 'Дней в работе', 'Текущий статус', 'Дата', and 'Финальный к...'. Below the table, there are sections for 'Детализация' and 'Кандидаты в работе'.

Рис. 2.11. Веб-додаток E-Staff Рекрутер

Програма «CleverStaff». «CleverStaff» - єдина система для професійного підбору персоналу.

The screenshot shows the CleverStaff interface. At the top, there is a navigation bar with icons for 'Organizer', 'Vacancies', 'Candidates', 'Clients', and 'Account'. Below this, there is a header with 'Our database 9204', 'LinkedIn, VK, Job boards', and 'New candidate'. The main content area displays a candidate profile for Aizada Kydyrbekova, IT Security Advisor - Sales representative for Kazakhstan (Russian speaking). The profile includes a photo, contact information (Ireland, phone number, email), education (Казакский Национальный Технический Университет), current company (McAfee), and current position (IT Security Advisor - Sales representative for Kazakhstan). It also lists 'Core Skills' and a 'Description'.

Рис. 2.12. Інтерфейс CleverStaff

Основні можливості:

- керування базою даних: парсинг резюме;
- інтеграція з поштою та завантаження резюме з неї;
- масове завантаження даних;
- інтеграція з «LinkedIn» та сайтами пошуку роботи;
- індивідуалізований інтерфейс: етапи, що настроюються роботи та шаблони листів, інтеграція з календарями «Google» та «Outlook», управління ролями та доступом;
- автоматизація рекрутингу: пропозиції та рейтинги кандидатів, швидкі звіти з вакансій;
- просунуті функції: налаштування API, HR-модуль, збереження резервні копії бази, моментальні звіти;
- система інтегрована з «Microsoft Outlook», «Google»[31].

Висновки до розділу 2

У будь-якій організації, як великої, і малої, виникає проблема такого управління даними, яке забезпечило б найефективнішу роботу. Деякі організації використовують при цьому паперові носії, проте сучасні підприємства залучають комп'ютерні системи автоматизації, дозволяють ефективно зберігати, обробляти інформацію й управляти великими обсягами даних.

Інформаційні технології надають бізнесу більше можливостей, аніж традиційні носії інформації. Крім традиційно використовуваних плюсів застосування програмного забезпечення (оптимізація організаційних та адміністративних процесів на основі обліку та аналізу даних) підприємство, яке зуміло організувати свої інформаційні потоки, виводить послуги клієнтам на якісно новий рівень.

В результаті аналізу предметної області було сформовано уявлення про автоматизований бізнес-процес, про розроблювану систему та її призначення;

був проведений аналітичний огляд існуючих аналогів системи та популярних сайтів онлайн-рекрутменту.

Досліджено функціональні можливості, а також інтерфейсні особливості систем. В результаті було виявлено, що кожна з описаних систем має функціонал для спрощення роботи з підбору персоналу. Перевагами описаних систем є наявність великого функціоналу та інтеграції з безліччю різних сайтів онлайн-рекрутменту. Серед недоліків вивчених систем можна виділити високу цінову політику, відсутність функції формування кадрового резерву, а також наявність «зайвого» функціоналу.

Розроблено концептуальну модель рекомендаційної системи управління підбором персоналу ІТ-підприємств.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

3.1. Вибір програмних компонентів реалізації

Існують різні варіанти для вирішення завдання підбору персоналу: дані можна обробляти в існуючій повнофункціональній HRM-системі, іншим варіантом є розробка власних систем для зберігання, обробки та надання даних. При цьому другий шлях можна реалізувати у вигляді власної розробленої настільної рекомендаційної системи або у вигляді веб-програми.

Одним із варіантів створення системи є її розробка у вигляді настільного додатка. Цей підхід дозволяє надавати користувачам лише необхідний функціонал та предметно-орієнтований інтерфейс для вирішення конкретних завдань, пов'язаних з підбором персоналу. Проте недоліками такого варіанта є моральне старіння настільних додатків, необхідність встановлення та конфігурування певного програмного забезпечення на робітники машини працівників [19].

Інше рішення, яке і було обрано, передбачає розробку необхідної системи на веб-основі. Взаємодія між веб-браузером користувача та веб-програмою відбувається за протоколом HTTP. Веб-додаток відповідає за формування інтерфейсу користувача і не займається будь-якою обробкою даних. Всі запити на обробку даних передаються від веб-програми на спеціалізований веб-сервер, реалізує всю логіку системи як веб-сервісів. В свою черга, веб-сервіс звертається за даними для своїх розрахунків до окремої СУБД, що зберігає цю інформацію.

Переваги такого підходу полягають у тому, що веб-додатки є платформонезалежними для кінцевого користувача, не вимагають установки, налаштування та адміністрування додаткового програмного забезпечення, а працювати з програмою можна фактично з будь-якої точки, де є доступ до Інтернету.

На початковому етапі створення рекомендаційної системи підбору персоналу проведемо вибір мов програмування для подальшого використання в процесі розробки.

На даний момент існує безліч мов програмування, і не ясно яка з них є найбільш відповідною. Тому є необхідність розгляду найбільш популярних з них, виявити переваги та недоліки і в підсумку зробити вибір.

C#. C# є об'єктно-орієнтованим і в цьому плані багато перейняв у Java і C++. Наприклад, C# підтримує поліморфізм, успадкування, перевантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід дозволяє вирішити завдання з побудови великих, але в той же час гнучких, масштабованих і розширюваних додатків. І C# продовжує активно розвиватися, і з кожною новою версією з'являється все більше цікавих функціональностей, як, наприклад, лямбда, динамічне зв'язування, асинхронні методи і т.д [32].

Переваги:

- підтримує автоматичний збір сміття;
- має високу надійність та захищеність;
- є повністю об'єктно-орієнтованою мовою програмування;
- має обширну бібліотеку класів, що спрощує реалізацію багатьох функцій;
- середній поріг входження. Синтаксис схожий на C, C++ або Java;
- присутній функціональне програмування (F #).

Недоліки:

- не є гнучкою мовою програмування;
- необхідність компілювати код кожного разу при будь-яких змінах в ньому;
- зберегли оператор go to;
- орієнтованість, в основному, тільки на .NET (на Windows платформу).

Java. Java - мова програмування загального призначення. Відноситься до об'єктно-орієнтованих мов програмування, до мов з сильною типізацією. Творці реалізували принцип WORA: write once, run anywhere або «пиши один раз,

запускай скрізь». Це означає, що написаний на Java додаток можна запустити на будь-якій платформі, якщо на ній встановлено середовище виконання Java (JRE, Java Runtime Environment)[33].

Переваги:

- Java - мова високого рівня з простим синтаксисом і плавною кривою навчання;
- підтримка багатопоточності;
- висока стабільність;
- безпека;
- незалежність від платформи;
- мова для розподіленого програмування і комфортною віддаленої спільної роботи;
- автоматичне управління пам'яттю; розробник створює об'єкти, а java за допомогою збирача сміття очищає пам'ять, коли об'єкти перестають використовуватися;
- стабільність і співтовариство.

Недоліки:

- підтримує тільки одну парадигму програмування;
- низька продуктивність;
- відсутність нативного дизайну;
- багатослівний і складний код.

Python. Python - високорівнева мова програмування загального призначення з динамічною строгою типізацією, орієнтований на підвищення продуктивності розробника, читання коду і його якості, а також на забезпечення переносимості написаних на ньому програм. Мова є повністю об'єктно-орієнтованим. Незвичайною особливістю мови є виділення блоків коду пробільними відступами. Синтаксис ядра мови мінімалістичний, за рахунок чого на практиці рідше виникає необхідність звертатися до документації[34].

Переваги:

- низький поріг входження. Синтаксис Python більш зрозумілий для початківця;
- логічний, лаконічний і зрозумілий. У порівнянні з багатьма іншими мовами Python має легкочитасий синтаксис;
- багатоплатформовий: підходить для різних платформ: і Linux, і Windows;
- є реалізація інтерпретаторів для мобільних пристроїв і непопулярних систем;
- широке застосування. Використовується для розробки веб-додатків, ігор, зручний для автоматизації, математичних обчислень, машинного навчання, в області інтернету речей;
- у світі Python багато якісних бібліотек. Для навчання є багато книг, в першу чергу англійською мовою. Сьогодні багато навчальних матеріалів на Youtube: відео блоги, записи вебінарів і конференцій.

Недоліки:

- Python відрізняється суворою вимогою до написання коду (вимагає відступи);
- низька швидкість виконання програм, у порівнянні з іншими мовами;
- погано взаємодіє з БД, порівняно з іншими мовами програмування;
- копіювання коду. При копіюванні коду з іншого ресурсу, в деяких випадках, він може скопіювати без збереження відступів. Тому код буде невалідним, а доведеться довго додавати табуляцію в кожен рядок;
- конвертація програми на python в exe. Програми на python мають розширення *.py. Для використання без інтерпретатора, наприклад, на Windows, його потрібно конвертувати в файл з розширенням *.exe.

C++. C++ - компільована, статично типізована мова програмування загального призначення. Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування. Мова має багату стандартну бібліотеку, яка включає в себе поширені контейнери і алгоритми, введення-виведення, регулярні вирази,

підтримку багатопоточності і інші можливості. С++ поєднує властивості як високорівневих, так і низькорівневих мов. Область його застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудованих систем, високопродуктивних серверів, а також ігор[35].

Переваги:

- висока продуктивність (швидкість роботи програм, але не їх написання);
- кросплатформеність (в основному, за рахунок qt);
- багатопотоковість;
- підтримує різні парадигми програмування (ооп, процедурне і т;д);
- можливість керування пам'яттю, що дозволяє оптимізувати програмні продукти;
- доступність літератури, документації;
- безкоштовність (опенсорс);
- універсальність.

Недоліки:

- відсутність системи збирання сміття;
- наявність показників, які потребують велику кількість пам'яті та при неправильному використанні призводять до різних сбоїв в програмних продуктах;
- через доступність дружніх функцій, глобальних змінних та покажчиків можуть виникати проблеми безпеки та захищеності програмних продуктів;
- при програмуванні можуть зустрічатися властивості, характерні для низькорівневого програмування, навіть там, де вони не потрібні;
- малозастосовний для роботи з Web;
- високий поріг входження;
- заплутаний і громіздкий синтаксис;

- для портіруємості додатків потрібно дотримуватися вимог стандарту, щоб не виникло помилок при роботі з іншим компілятором;

- складність знаходження помилок. Якщо, наприклад, для C# є такі інструменти як ReSharper, які шукають помилки за вас, то в C++ для цього доведеться попідніти.

PHP. PHP - скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів[36].

Переваги:

- розробка за допомогою PHP дає багато можливостей. При належному рівні володіння, за допомогою шаблонізатора можна створювати не тільки сценарії для веб-додатків, але й повноцінні програми. Існують рішення, що дозволяють створювати мобільні додатки на PHP;

- вивчення PHP не вимагає багато часу;

- кросплатформеність. PHP може бути запущений в будь-якій операційній системі, включаючи юніксоїд;

- підтримка веб-серверів;

- безкоштовне розповсюдження;

- має достатню довільність для web-розробки. Звичайно, такі базові мови, як C-сімейство, працюють швидше, але для інтернету це не критично;

- наявність навчальних матеріалів;

- безперервний розвиток.

Недоліки:

- вузькопрофільність. І хоча можливості розширені різними реалізаціями, все ж він «заточений» під програмування для Інтернету;

- безпека. У PHP є засоби безпеки рівня системи і рівня web-додатки;

- протиріччя в коді. Код переповнений різними залишками з різних мов.

І всі вони можуть навіть бути сконцентровані в одному вираженні коду.

JavaScript. JavaScript - це динамічна мова комп'ютерного програмування. Він легкий і найчастіше використовується як частина веб-сторінок, чії реалізації дозволяють клієнтському сценарію взаємодіяти з користувачем і створювати динамічні сторінки. Це інтерпретована мова програмування з об'єктно-орієнтованими можливостями[37].

Переваги:

– швидкість. JavaScript має тенденцію бути дуже швидким, оскільки він часто запускається відразу в браузері клієнта. Поки він не вимагає зовнішніх ресурсів, JavaScript не сповільнюється через виклики до сервера. Крім того, всі основні браузери підтримують компіляцію JIT (вчасно) для JavaScript, що означає, що немає необхідності компілювати код перед його запуском;

– популярність. Існує незліченна кількість ресурсів для вивчення JavaScript. Як StackOverflow, так і GitHub демонструють дедалі більшу кількість проєктів, які використовують JavaScript, і привабливість, яку вона здобула за останні роки, очікується лише збільшення;

– сумісність. JavaScript можна використовувати в багатьох різних видах програм завдяки підтримці інших мов, таких як Pearl та PHP;

– навантаження сервера. JavaScript працює на стороні клієнта, тому загалом зменшує навантаження на сервери, а простим програмам сервер може взагалі не знадобитися;

– універсальність. Є багато способів використовувати JavaScript через сервери Node.js.

Недоліки:

– захист на стороні клієнта. Оскільки код JavaScript виконується на стороні клієнта, помилки та недогляди іноді можуть бути використані для зловмисних цілей;

– підтримка браузера. Хоча сценарії на стороні сервера завжди дають однакові результати, різні браузери іноді інтерпретують код JavaScript по-різному.

Дослідження різних мов програмування дозволило створити порівняльну характеристику, яка наведена у табл. 3.1

Таблиця 3.1

Порівняльна характеристика мов програмування

Критерії порівняння	C#	Java	Python	JavaScript	C++	PHP
Незалежність від платформи	-	+	+	+	+	+
Можливість обробляти велику кількість даних	+	+	+	+	+	-
Наявність додаткових бібліотек та пакетів	+	+	+	+	+	+
Висока швидкість роботи	+	-	-	+	+	+
Простота вивчення та розробки	+	+	+	+	+	+

Проаналізувавши мови програмування вибір зупинився на JavaScript, оскільки він легкий в освоєнні, написанні коду і має досить багато бібліотек з готовими рішеннями.

Середовище розробки. Для ефективної розробки програмного продукту треба обрати середовище розробки, яке буде підходити під обрану мову програмування.

Для мови програмування JavaScript підходить декілька середовищ розробки. Для того, щоб обрати одне з них було вирішено дослідити їх та виділити основні переваги та недоліки.

Microsoft Visual Studio. Microsoft Visual Studio є середовищем розробки від фірми Microsoft. Дає змогу розробляти консольні та графічні програмні продукти, має вбудований відладчик коду, підтримує велику кількість додаткових бібліотек для розширення базового функціоналу, в тому числі для візуального проектування[38].

Переваги:

- можливість безкоштовного використання;

- підтримка платформи .NET.

Недоліки:

- складність освоєння через неочевидне розташування функцій в меню;
- при переході на іншу версію можуть виникати помилки та збої;
- необхідність зберігати зміни в коді вручну.

Eclipse. Eclipse є інтегрованим середовищем розробки від компанії Eclipse Foundation. Воно орієнтоване для розробки програмних продуктів на мові програмування Java. Використовується для розробки модульних програмних продуктів[39].

Переваги:

- компілятор та відладчик працюють ефективніше ніж в інших середовищах розробки;
- функціонал дозволяє кастомізувати середовище розробки для зручного використання;
- безкоштовність та відкритий вихідний код.

Недоліки:

- додаткові плагіни не є офіційними, тому при їх використанні немає гарантій надійності;
- використовує велику кількість апаратних ресурсів та займає багато місця на диску.

WebStorm. WebStorm від JetBrains прекрасний в обох своїх проявах: як IDE він підтримує роботу з системами контролю версій, дозволяє віддалено розгорнути код, як редактор - стандартні зручності, на зразок підсвічування синтаксису, автодоповнення, навігації[40].

Переваги:

- LiveEdit - перегляд внесених до коду змін без необхідності його зберігати;
- взаємодія з фреймворками, наприклад, React, Angular, Meteor;
- більше сотні інтегрованих тестів для виявлення помилок;
- інтегрування з Mocha, Protractor, Jest, Karma для юніт-тестів;

- повномасштабний дебаггер для налагодження коду на серверній та клієнтській сторонах;

- навігація для одночасної роботи з кількома файлами;
- автодоповнення коду, підсвічування синтаксису.

Недоліки:

- для кодерів-початківців функціонал надмірний.

Sublime Text. Зручний і перевірений часом кросплатформовий редактор, з інтерфейсом, що настроюється, і можливістю здійснювати тривіальні дії за допомогою гарячих клавіш[41].

Переваги:

- гарячі клавіші;
- навігація за кодом у вигляді міні-картки;
- можливість змінити візуальну тему;
- підтримка сніпетів;
- підсвічування, автодоповнення змінних та синтаксису;
- перевірка синтаксису безпосередньо під час введення;
- купа плагінів;
- автозбереження.

Недоліки:

- відсутність аналізатора коду для розміщення посилань.

Atom. Atom - текстовий редактор з багатьма налаштуваннями. У Atom вбудований менеджер пакетів, за допомогою якого можна знайти, встановити та навіть створити власні пакети. Передбачено чотири призначені для користувача інтерфейси та вісім синтаксичних тем у темних та світлих тонах. Також доступні теми, створені співтовариством[42].

Переваги:

- зручний та приємний інтерфейс;
- розумне автозавершення;

- файловий менеджер, за допомогою якого легко переглядати як окремі файли, так і цілі проекти;
- мультипанельний інтерфейс дозволяє розділити інтерфейс для зручності порівняння та редагування коду в декількох файлах;
- функція пошуку та заміни, попередній перегляд та заміна тексту в одному файлі або в проекті;
- автодоповнення та підсвічування коду;
- гнучкі налаштування редактора, пакетів, що підключаються, тим інтерфейсу;
- редагування та навігація за допомогою гарячих клавіш.

Недоліки:

- невисока продуктивність;
- порожня комплектація «з коробки».

Visual Studio Code. Visual Studio Code - легкий, але потужний редактор коду. У первинній конфігурації використовується для редагування коду на JavaScript, TypeScript і Node.JS, а за допомогою розширень підтримує C++, C#, Python і PHP.

Visual Studio Code не просто виконує автодоповнення, а робить це з розумом: за допомогою технології IntelliSense дописує назви оголошених змінних, функцій і модулів, а також робить посилання на відповідний розділ документації. Можливе налагодження коду безпосередньо з редактора, запуск приладу для налагодження та приєднання до запущених додатків[43].

Переваги:

- контекстне автодоповнення, як синтаксису, так і використовуваних змінних, модулів, функцій тощо;
- дебаггер з точками зупинки, стеком викликів, інтерактивною консоллю;
- підтримка сніпетів та шаблонів;
- зручний та простий інтерфейс.

Недоліки:

– мало плагінів.

Під час дослідження різних середовищ розробки було створено порівняльну характеристику, яка представлена в табл. 3.2.

Таблиця 3.2

Порівняльна характеристика середовищ розробки

Критерії порівняння	Microsoft Visual Studio	Eclipse	WebStorm	Sublime Text	Atom	Visual Studio Code
Можливість використання в різних системах	+	+	+	+	+	+
Підтримка контролю версій	+	+	+	-	-	-
Наявність додаткових модулів	+	+	+	+	+	+
Простота в освоєнні	-	-	+	+	-	-
Можливість безкоштовного використання	+	+	+	+	+	+
Зручний інтерфейс	-	+	+	+	+	-
Використання великої кількості апаратних ресурсів	-	+	-	-	-	-

В табл. 3.2 можна побачити, що середовище розробки WebStorm має більш високу надійність порівняно з іншими середовищами, не потребує великої кількості апаратних ресурсів та має безкоштовну версію з усім необхідним функціоналом. Саме тому для розробки програмного продукту було обране середовище розробки WebStorm.

Система управління базами даних. На сьогоднішній день існують різні СУБД: Microsoft SQL Server, Oracle, MySQL, PostgreSQL та багато інших. Всі

вони схожі по характеристика, дозволяють вирішувати схожі завдання та підходять для управління базою даних системи, що розробляється.

Для зберігання інформації було обрано СУБД Microsoft SQL Server 2019, оскільки дана система є дуже популярною та легкою в освоєнні та настройці.

ActiveDirectory. Active Directory (AD) - служба каталогів корпорації Microsoft. Спочатку створювалася як LDAP (Lightweight Directory Access Protocol)-сумісна реалізація служби каталогів, проте сьогодні включає можливості інтеграції з іншими службами авторизації, виконуючи для них інтегруючу та об'єднуючу роль. AD дозволяє використовувати групові політики для забезпечення одноманітності налаштування користувальницького робочого середовища, встановлення оновлень програмного забезпечення на всіх комп'ютерах у мережі. Зберігає дані та налаштування середовища у централізованій базі даних. Для системи, що розробляється служба каталогів AD буде використовуватися як джерело даних користувача, необхідні для аутентифікації користувачів за допомогою протоколу LDAP (LDAP-автентифікація).

Технологія доступу до даних. Найбільш поширеними технологіями доступу до даних БД вибраному стеку технологій (JS React- .NET - ASP.NET - MS SQL) є: Entity Framework (EF), NHibernate, Dapper. Об'єктно-орієнтована технологія ADO.NET Entity Framework (EF) є стандартною технологією вибраного стека, повністю підходить для вирішення поставлених завдань, тому як технологія доступу до даних із бази даних буде використана вона. Для роботи з базою даних буде використано підхід «Code First» та будуть організовані міграції для бази даних: на основі проекту бази даних будуть створені необхідні доменні моделі, які в надалі будуть використані в початковій (ініціуючій) міграції для створення бази даних. За необхідності внесення зміни до схеми бази - зміни проводяться в коді і створюється нова міграція. Даний підхід дозволяє легко змінювати схему БД та ділитися цими змінами з командою. Таким чином, міграції можна назвати системою контролю версій бази даних.

3.2. Проектування рекомендаційної системи управління підбором персоналу ІТ-підприємств

Проектування - один із найважливіших та відповідальних етапів у реалізації будь-якого проекту. Від якості створеного проекту системи залежить чи доведеться надалі вносити до неї принципові зміни, що завжди є трудомістким та ресурсозатратним завданням[16].

На етапі проектування рекомендаційної системи управління підбором персоналу ІТ-підприємств було виділено такі підетапи:

- проектування структури бази даних;
- проектування архітектури системи;
- проектування інтерфейсу системи.

Проектування структури бази даних рекомендаційної системи підбору персоналу ІТ-підприємств

Спроектвана база даних представлена на рис. 3.1 як ER-модель, що містить 9 сутностей із необхідними атрибутами та зв'язками між сутностями (у моделі вказані зв'язки, які після конвертації у фізичну модель стануть атрибутами таблиць):

– Applicants - сутність для зберігання даних про претендентів. Її атрибути представлені у таблиці 3.3.

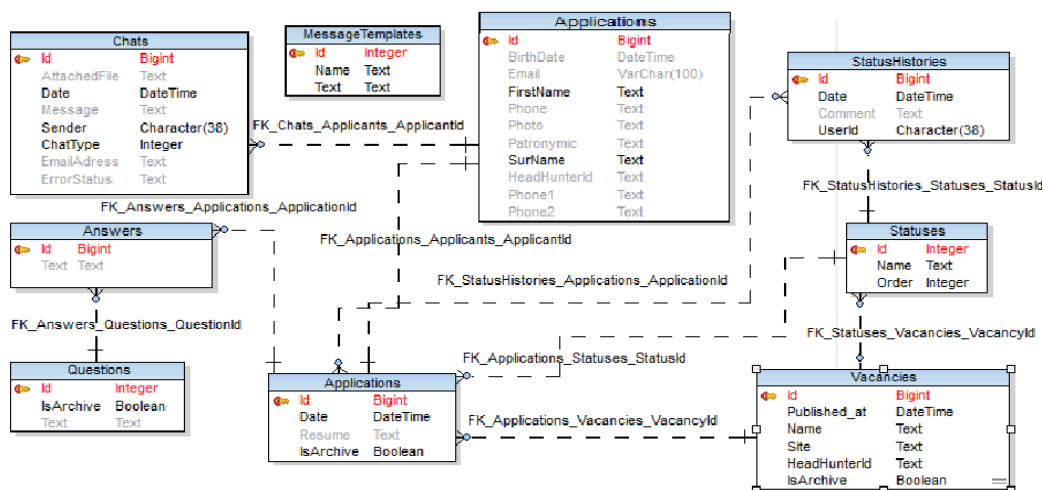


Рис. 3.1. Схема бази даних рекомендаційної системи підбору персоналу

Таблиця 3.3

Атрибути таблиці «Applicants»

Атрибут	Опис
Id	Ідентифікатор претендента
FirstName	Ім'я претендента
Surname	Прізвище претендента
Patronymic	По батькові здобувача
BirthDate	Дата народження претендента
Phone	Контактний телефон претендента
Phone1	Додатковий телефон No1 претендента
Phone2	Додатковий телефон No2 претендента
Email	Електронна пошта претендента
Photo	Посилання на фото претендента
SaitId	Ідентифікатор сайту та претендента на ньому

– Applications - сутність, що містить інформацію про заявки (Відгуки на вакансію). Її атрибути представлені у таблиці 3.4.

– Атрибут IsArchive необхідний для визначення активних заявок, з якими безпосередньо працює користувач, а також для зберігання неактивних заявок, що дозволить визначити архів заявок кандидатів.

– Status є таблицею-довідником, що зберігає список можливих станів заявки кандидата. Кожен статус належить конкретній вакансії, таким чином, буде доступна можливість встановити власні статуси кожної вакансії. У кожного статусу є свій порядок (поле «Order»), що визначає черговість проходження кандидата за статусами у поточній вакансії.

– StatusHistories призначена для зберігання архівної інформації про статуси, які кандидат пройшов за заявкою.

– Vacancies - таблиця, записи якої є однією вакансією. Її атрибути представлені в таблиці 3.5.

Таблиця 3.4

Атрибути таблиці «Applications»

Атрибут	Опис
Id	Ідентифікатор заявки
Applicant Id	Ідентифікатор претендента, який надіслав заявку (атрибут для зв'язку буде відображено у фізичній моделі)
Vacancy Id	Ідентифікатор вакансії, на яку надійшла заявка (атрибут для зв'язку, буде відображено у фізичній моделі)
Date	Дата отримання заявки
Resume	Резюме претендента на цю заявку
IsArchive	Булева змінна, що визначає стан заявки: true - активна заявка, false - заявка неактивна
StatusId	Ідентифікатор статусу (атрибут для зв'язку, буде відображено у фізичній моделі)

Таблиця 3.5

Атрибути таблиці «Vacancies»

Атрибут	Опис
Id	Ідентифікатор вакансії
Name	Найменування вакансії
Site	посилання на сайт, де вакансія розміщена
PublishedAt	Дата розміщення вакансії
HeadHunterId	Ідентифікатор вакансії на сайті hh.ru
IsArchive	Прапор, що визначає в архіві чи вакансія ні.

– Chats є таблицею, що зберігає листування. Листування може вестись між користувачем та кандидатом та між користувачем та іншим користувачем. Атрибути даної таблиці представлені у таблиці 3.6.

– Questions є таблицею-довідником, що зберігає список можливих питань, які задаються претендентам на сайтах.

– Answers призначена для зберігання інформації про відповіді претендентів на запитання.

– MessageTemplates призначена для зберігання шаблонів повідомлень.

Проектування архітектури системи. Розглянемо компонентну та модульну архітектуру системи. Основу системи складають 6 компонентів: сервер (Server), СУБД («DBMS»), сервіс-постачальник даних про вакансії («RabotaUA.WebApi»), поштові сервіси («MailService»), плагін для завантаження кандидатів та заявок («Plugin») та клієнти («JSClient»). Компонентне архітектурне представлення наведено на рис 3.2.

Компонент «DBMS» являє собою встановлену на комп'ютер-сервер СУБД для управління базою даних. Для зв'язку з базою надає інтерфейс SQLClient.

Компонент «Plugin» є встановленим у браузері клієнта плагін, що відправляє на сервер запити за протоколом HTTPS.

Компонент «MailService» є поштовим сервісом, який буде використовувати організація для оправки та отримання листів від претендентів.

Таблиця 3.6

Атрибути таблиці "Chats"

Атрибут	Опис
Id	Ідентифікатор повідомлення
ApplicantId	Ідентифікатор кандидата (атрибут для зв'язку, буде відображено у фізичній моделі)
AttachedFile	Рядок із посиланнями на прикріплені файли
Date	Дата надсилання повідомлення
Message	Текст повідомлення
Sender	Ідентифікатор відправника (або користувач, або здобувач)
ChatType	Ідентифікатор типу повідомлення (системне, від користувача, від кандидата)
EmailAddress	Електронна адреса відправника (якщо відправлення від користувача кандидату)
ErrorStatus	Поле, що зберігає повідомлення про помилку (на випадок, якщо листа не було відправлено)

Компонент «Сервер» є компонентом сервера, на якому розгортаються такі компонентні частини:

Компонент DataAccessLibrary - бібліотека для доступу до даних бази даних, що надає інтерфейс «DA» для роботи з нею.

Компонент «HR», який є програмою, що містить логіку взаємодії з клієнтами, що надсилає їм необхідну інформацію, що працюють безпосередньо з бібліотекою для доступу до даних. «HR» надає інтерфейс (протокол) HTTPS, завдяки якому дані можуть передаватися клієнтським компонентам.

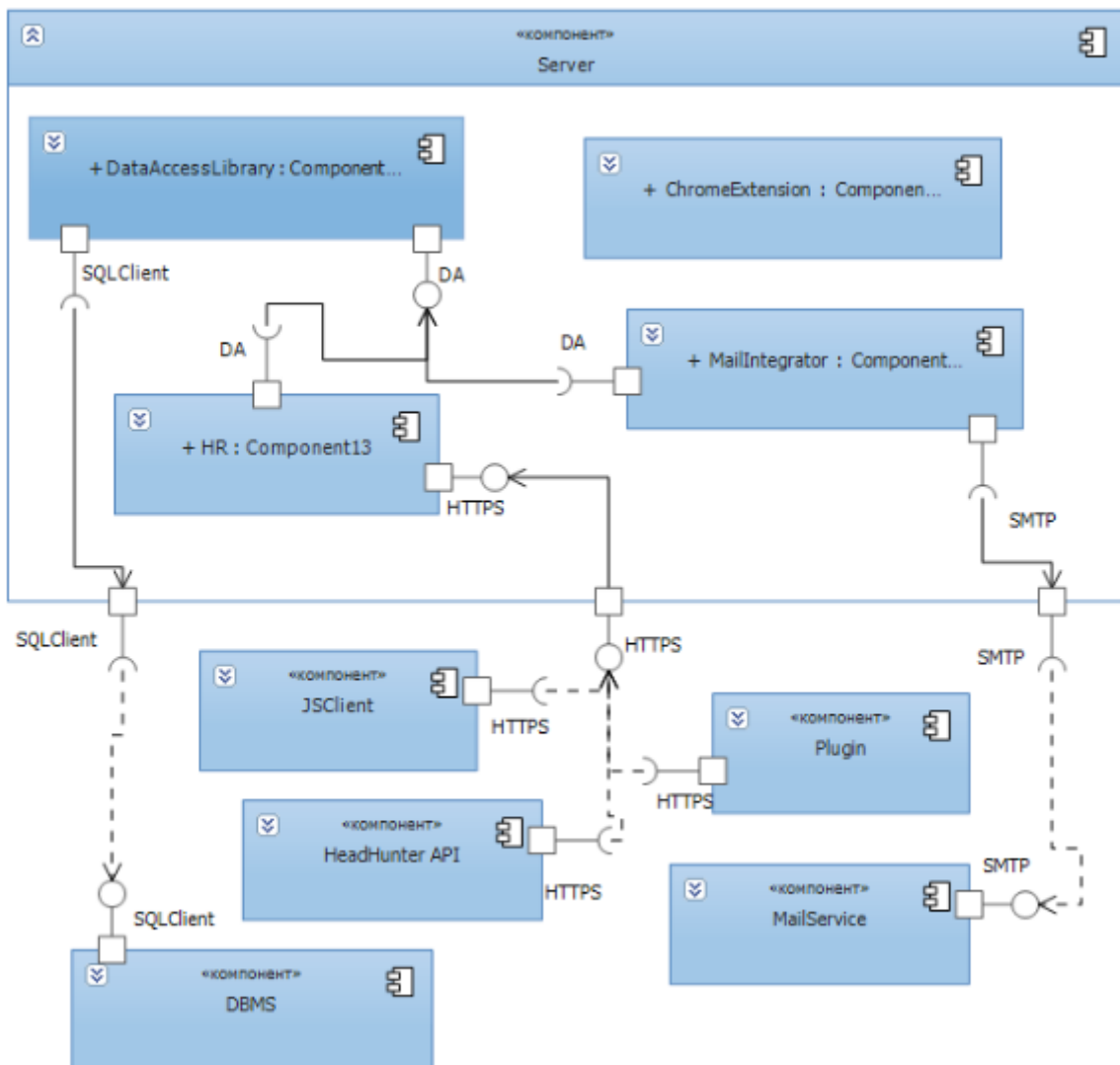


Рис. 3.2 Діаграма компонентів рекомендаційної системи підбору персоналу

Компонент ChromeExtension - бібліотека, в якій буде розроблятися плагін для браузера Chrome.

Компонент «MailIntegrator» - програма з веб-сервісної архітектури, яка містить логіку для інтеграції з поштою. Програма працює з бібліотекою для доступу до даних.

Як клієнтський компонент буде використовуватися браузер.

Архітектурне представлення розгортання. Результат архітектурного проектування представлений на діаграмі розгортання (рис. 3.3), що відображає конфігурацію вузлів, які виконуються системою, та компонентів, розміщених у цих вузлах.

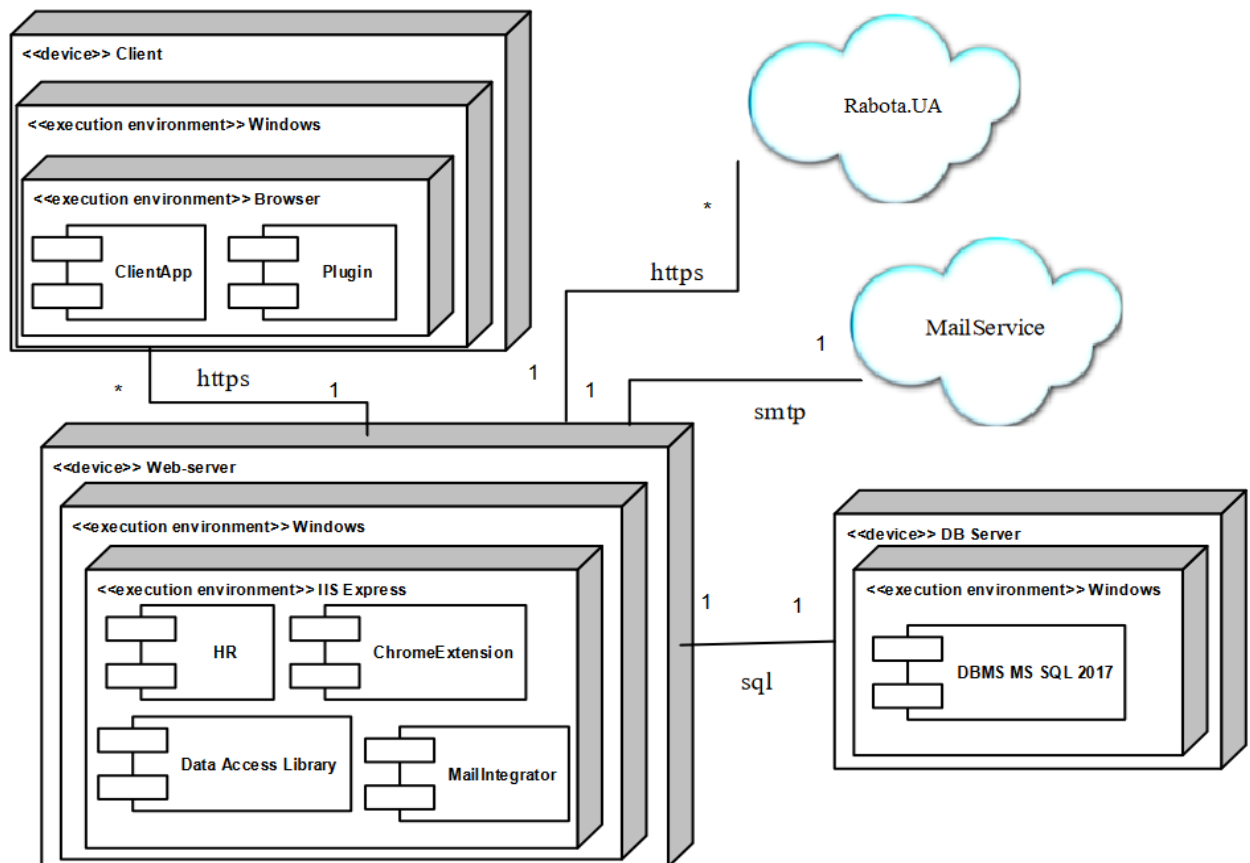


Рис. 3.3 Діаграма розгортання рекомендаційної системи підбору персоналу

На наведеній діаграмі веб-додаток та веб-сервіси поєднані на платформі одного веб-сервера, але вони можуть працювати і на різних серверах.

Як середовище виконання сервера може використовуватися операційна система сімейства Windows. Компонент «Web-application», є додатком MVC і

виконуються в середовищі IIS Express. Операційна система сімейства Windows також є середовищем виконання для компонента СУБД MS SQL Server.

Компонент «Клієнт» є браузерною програмою, тому середовищем його виконання є браузер, який у свою чергу виконується в операційній системі. Взаємодія між сервером та клієнтами здійснюється за протоколом HTTPS.

Як постачальник даних про претендентів, вакансії та відгуки на них можуть бути використані зовнішні по відношенню до системи веб-послуги. Для того щоб рекомендаційна система була актуальна та затребувана різними організаціями, потрібна інтеграція з різними сайтами онлайн-рекрутменту. Однак для початку роботи було обрано один сайт для інтеграції- «RabotaUA».

Для організації листування між кандидатом та користувачем буде реалізовано інтеграцію з поштовими сервісами.

Проектування сценаріїв роботи. Для показу всіх типових дій було обрано процес роботи з здобувачами. Було створено діаграму послідовностей, яка відображає типову послідовність роботи сервера (рис 3.4). Показано виклик анонімними повідомленнями (з клієнта) методів контролера «ApplicantsController». Відображено всі типові методи: «Index()», «Create()», «Edit()», «Delete()».

Докладно розглянемо типову дію клієнта - перегляд списку претендентів. Користувач опиняється на сторінці «Здобувачі», відображає список претендентів та елементи для їх керування. Для відображення даного списку при створенні сторінки був викликаний метод Index() у контролера «ApplicantsController», який у свою чергу викликає необхідний метод у об'єкта класу «ApplicantService» з бібліотеки доступу до даних DataAccessLibrary. Викликаний метод створює та надсилає запит до бази даних, певним чином обробляє відповідь та повертає необхідні дані.

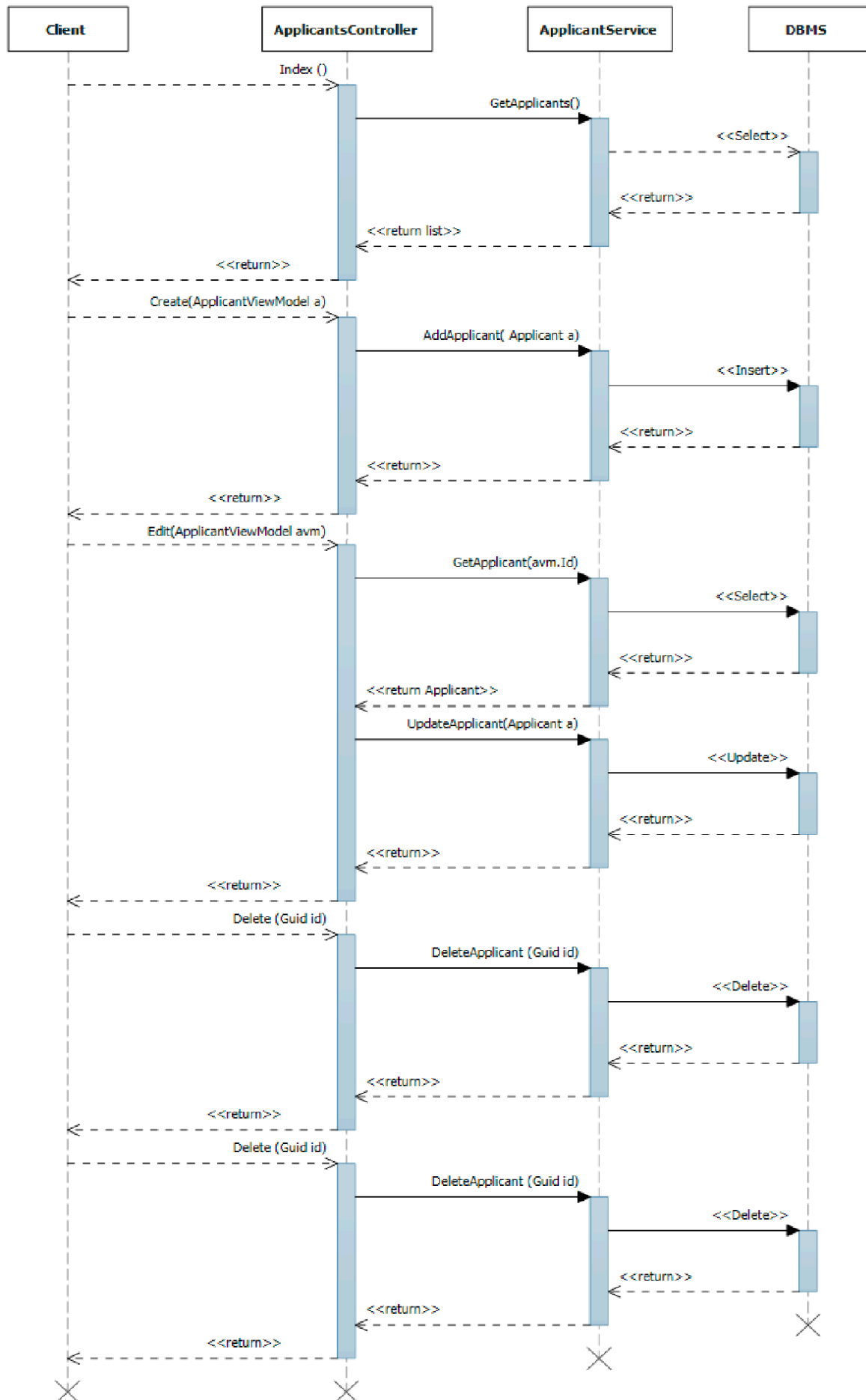


Рис. 3.4 Діаграма послідовностей

Коли користувач додає, видаляє або редагує претендента, цей алгоритм повторюється знову: викликається відповідний метод контролера «ApplicantsController», який викликає метод у об'єкта класу «ApplicantService», який у свою чергу надсилає запити до бази даних та обробляє відповідь.

3.3. Реалізація рекомендаційної системи управління підбором персоналу ІТ-підприємств

Реалізація бази даних. На основі спроектованої концептуальної схеми були розроблені доменні моделі, на основі яких за допомогою міграцій була створена база даних у СУБД MS SQL Server 2019. Фізична схема створеної БД представлена на рис 3.5.

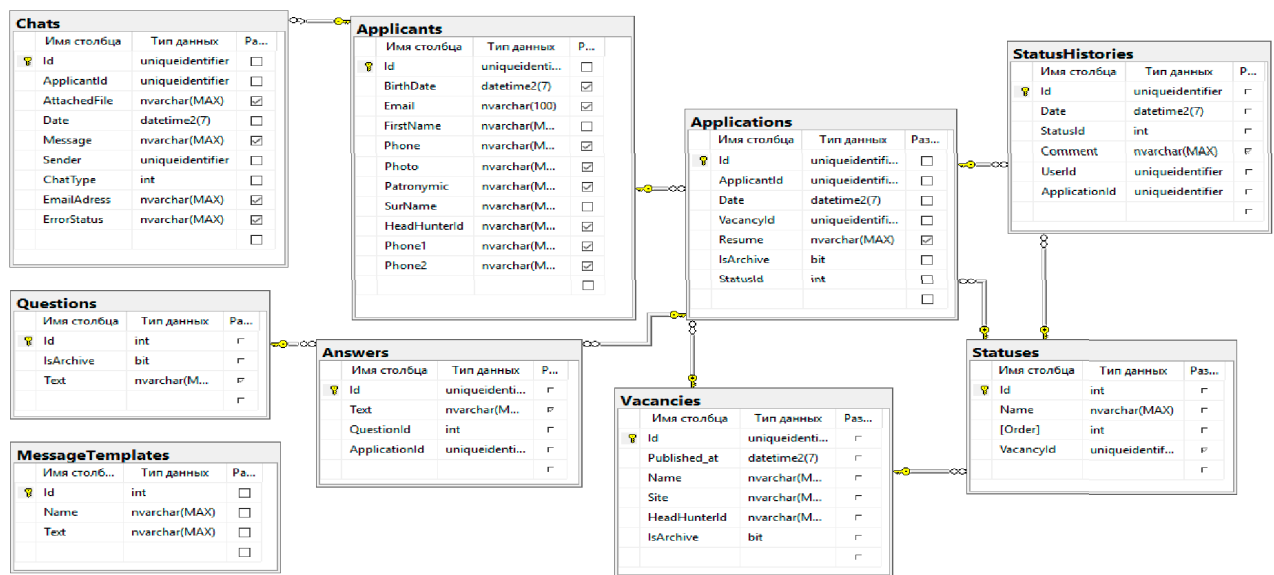


Рис. 3.5 Фізична схема бази даних рекомендаційної системи підбору персоналу

Реалізація архітектури системи

Основу структури системи складають наступні контроллери:

- «Account» для реалізації логіки авторизації і аутентифікація;
- «Applicants» для управління даними, які стосуються кандидата;
- «Applications» для виконання логіки роботи із заявками;

- «Statuses» для реалізації логіки роботи зі статусами;
- «Chat» для відображення чату на сторінці;
- «ChatApi» для виконання логіки чата (введення переписки користувача з кандидатами або іншими користувачами, загрузка повідомлення з поштового сервісу), є веб-сервісом;
- «ChromeExtensionApi» для обробки запитів із плагіном для браузера, є веб-сервісом;
- «Vacancies» для відображення списку вакансій та завантаження їх з ресурсу «RabotaUA»;
- «MessageTemplateController» для перегляду та керування шаблонами повідомлень;
- «RubiusApi» для додавання кандидатів у систему, які відкликаються по вакансіям через сайт організації.

Основні частини системи, що описують розроблену структуру, зображенні на діаграмі класів, представлено на рис. 3.6.

Контролери створюють уявлення відображення тих чи інших даних. Так, наприклад, контролер «Applicants» взаємодіє з чотирма «повними» уявленнями: «Index», «Board», «Create» і «Details». Ці уявлення представляють такі дані про претендентів як: список всіх претендентів («Index»); kanban-дошка («Board»), на якій відображено картки претендентів і можливість керувати їх статусами; інформацію про кандидата, його заявки, історію переміщень його за статусами («Details»). Подання «Create» описує сторінку для додавання здобувача.

Контролер «Account» взаємодіє з поданням «Login», необхідним для авторизації користувача.

Як підхід до побудови веб-інтерфейсу програми використовується технологія AJAX (Asynchronous Javascript and XML). За рахунок цього, при оновленні даних на веб-сторінці, вона не перезавантажується повністю, а відбувається оновлення потрібної частини контенту веб-сторінки. За рахунок цього швидкість роботи веб-програми збільшується. Так, наприклад, за допомогою технології AJAX на сторінці з детальною інформацією про здобувача

відображаються часткові уявлення, що містять інформацію за його активними та архівними заявками, історії переміщень за статусами, відповіді на тест; відображають чат на сторінці і т.д.

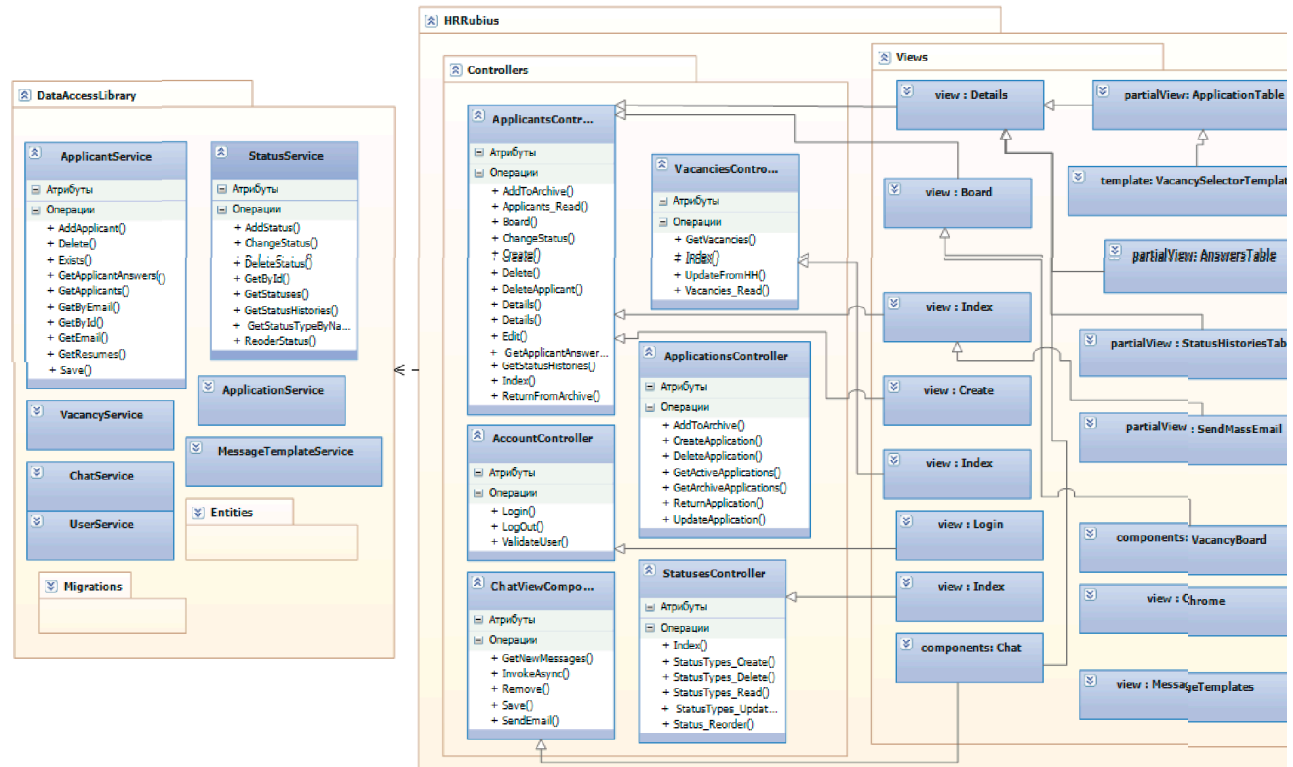


Рис. 3.6 Фрагмент структури класів рекомендаційної системи підбору персоналу

Реалізація інтерфейсу системи. Після реалізації функціональних можливостей системи, був розроблений запланований інтерфейс системи. В даному розділі буде наведено частину розроблених сторінок, що демонструють її функціональні можливості.

Співробітник кадрів розпочинає роботу зі сторінки вакансій (Рис 3.7). Вакансії завантажуються з сайту «rabota.ua» за допомогою API, що надається даним майданчиком онлайн-рекрутменту. При завантаженні була реалізована перевірка на існування вакансій на сайті. Якщо вакансія, завантажена раніше, була видалена із сайту - у системі вона стає архівною. На сторінці зі списком вакансій був реалізований перемикач для перегляду активних та архівних вакансій.

Після вибору певної вакансії система відображає kanban-дошку для обраної вакансії (Рис 3.8). Реалізовано фільтр на дошці вибору вакансії та завантаження кандидатів на дошку за допомогою технології AJAX без перезавантаження сторінки. Знизу на картці кандидата реалізовано відображення дати змін статусу кандидата за поточною заявці, а також прізвища співробітника, який переніс заявку на новий статус.

Recommended system					Вакації		Здобувачі		Статуси		Плагін		Шаблони		Стеценко Максим		Вийти				
																		<input checked="" type="checkbox"/> Активні		<input type="checkbox"/> Архівні	
Найменування	Отклики	Дата публікації	Сайт																		
Розробник C# (.NET, WPF, middle)	9	13.10.2021	https://rabota.ua/company6523733/vacancy8714419																		
Пресейл менеджер (Pre-sale)	2	13.10.2021	https://rabota.ua/company6523733/vacancy8714421																		
C# розробник під мобільні платформи	2	13.10.2021	https://rabota.ua/company6523733/vacancy8714430																		
Web-розробник (C#, ASP.NET)	2	13.10.2021	https://rabota.ua/company6523733/vacancy8714439																		
Build engineer/DevOps Engineer	3	13.10.2021	https://rabota.ua/company6523733/vacancy8714449																		
C# розробник під мобільні платформи	1	13.10.2021	https://rabota.ua/company6523733/vacancy8714479																		
Менеджер по роботі з персоналом	3	13.10.2021	https://rabota.ua/company6523733/vacancy8714485																		
Програміст C# (WPF, CAD, middle)	1	13.10.2021	https://rabota.ua/company6523733/vacancy8714490																		
Офіс менеджер	2	13.10.2021	https://rabota.ua/company6523733/vacancy8714495																		
Розробник C# (.NET, WPF, middle)	1	13.10.2021	https://rabota.ua/company6523733/vacancy8714504																		
Web-розробник (C#, ASP.NET)	1	13.10.2021	https://rabota.ua/company6523733/vacancy8714512																		

Рис. 3.7 Сторінка вакансій

Після натискання на ПІБ кандидата буде здійснено перехід на сторінку з детальною інформацією щодо кандидата (Рис 3.9-3.11).

Recommended system Вакансії Здобувачі Статуси Плагін Шаблони Стеценко Максим Вийти

Вакансія: Розробник C# (.NET, WPF, midd) ▾











Отклик	Резерв	Дод. питання	Тестове	На перевірі	Кандидати на співбесіду	Призначено співбесіду	Готові взяти	Випробувальний термін	Відмовити
+ Додати Іванова Марія Василівна  15.10.2021 Стеценко		Ульянов Ігор Борисович  15.10.2021 Стеценко		Савицький Володимир Олегович  15.10.2021 Стеценко		Бейм Ганна Олегівна  15.10.2021 Стеценко	Костенко Микита Юрійович  15.10.2021 Стеценко		Уотс Наомі  15.10.2021 Стеценко
Іванов Павло Сергійович  15.10.2021 Стеценко				Борухін Володимир Сергійович  15.10.2021 Стеценко			Петров Олег Юрійович  15.10.2021 Стеценко		

Рис. 3.8 Сторінка з kanban-дошкою по вакансії

У системі реалізований множинний вибір кандидатів для подальшого масового надсилання повідомлень на електронну пошту кандидатам (Рис 3.12).

Recommended system Вакансії Здобувачі Статуси Плагін Шаблони Стеценко Максим Вийти

Іванова Марія Василівна **ЧАТ** ОТКЛИКИ ТЕСТ



Дата народження: 30.05.1987
 Телефон: (096) 656-5656
 Пошта: mariya@gmail.com

17.11.2021 12:30:45 Приходила сьогодні. У розмові приємна, ввічлива, тактовна. На запитання відповідала добре. Очікування із зарплати надмірно високі.

17.11.2021 12:35:10 Зрозуміла

Статус змінено: Резерв 17.11.2021 12:36:30

Статус змінено: Тестове 17.11.2021 12:36:32

Статус змінено: Кандидати на співбесіду 17.11.2021 12:36:34

Статус змінено: Отклик 17.11.2021 12:36:36

Введіть повідомлення... ✎ ✉ ➤

Рис. 3.9 Сторінка здобувача Чат.

Recommended system Вакансії Здобувачі Статуси Плагін Шаблони Стеценко Максим Вийти

Іванова Марія Василівна ЧАТ ОТКЛИКИ ТЕСТ

+ Додати отклик

Вакансія	Дата отклика+	Резюме	Стан ...
Розробник C# (.NET, WPF, middle)	11.10.2021	немає даних	ОТКЛИКИ Активна
Офіс менеджер	29.10.2021	немає даних	ОТКЛИКИ Активна

Дата народження: 30.05.1987
Телефон: (096) 656-5656
Пошта: mariya@gmail.com

Рис. 3.10 Сторінка здобувача. Заявки.

Була реалізована функція вибору шаблону повідомлень під час відправлення повідомлень. Для шаблонів повідомлень були реалізовані методи їх читання, створення, редагування та видалення (Рис 3.13).

Recommended system Вакансії Здобувачі Статуси Плагін Шаблони Стеценко Максим Вийти

Іванова Марія Василівна ЧАТ ОТКЛИКИ ТЕСТ

Історія статусів за вакансією "Розробник C# (.NET, WPF, middle)"

Статус	Дата встановлення	Переніс
Отклик	11.10.2021 12:39	Стеценко Максим Костянтинович
Кандидати на співбесіду	11.10.2021 12:38	Стеценко Максим Костянтинович
Тестове	11.10.2021 12:37	Стеценко Максим Костянтинович
Резерв	11.10.2021 12:36	Стеценко Максим Костянтинович
Отклик	11.10.2021 12:29	Стеценко Максим Костянтинович

Дата народження: 30.05.1987
Телефон: (096) 656-5656
Пошта: mariya@gmail.com

Рис. 3.11 Сторінка претендента. Історія статусів.





Recommended system						Вакансії		Здобувачі		Статуси		Плагін		Шаблони		Стеценко Максим		Вийти		
Пошук																				
+ Додати претендента																				
✉ Відправити лист																				
✉ Очистити вибір																				
<input type="checkbox"/>	ПІБ	Вакансії	Дата народж...	Контакти																
<input checked="" type="checkbox"/>		Ульянов Ігор Борисович	Розробник C# (.NET, WPF, middle) Web-розробник (C#, ASP.NET)	09.12.1991	Телефон: (098) 565-6565 Пошта: ulian@gmail.com															
<input type="checkbox"/>		Іванов Павло Сергійович	Розробник C# (.NET, WPF, middle)	05.03.1993	Телефон: (054) 898-9989 Пошта: urist@gmail.com															
<input checked="" type="checkbox"/>		Мерченко Володимир Іванович	C# Розробник під мобільні платформи	20.04.1977	Телефон: немає даних Пошта: merV@ukr.ua															
<input type="checkbox"/>		Бейм Марія Олегівна	Менеджер по роботі з персоналом	30.05.1994	Телефон: (084) 131-3233 Пошта: mariya@gmail.com															

Рис. 3.12 Список кандидатів. Множинний вибір.

При надсиланні шаблонного повідомлення кандидат отримає на пошту лист із вставленими ПІБ кандидата та ПІБ співробітника, який відправив йому листа. Дані вставляються в плейсхолдери, що містяться у шаблоні повідомлення.







Recommended system		Вакансії		Здобувачі		Статуси		Плагін		Шаблони		Стеценко Максим		Вийти					
+ Додати шаблон																			
Найменування		Текст																	
Вакансія закрита		Здрастуйте, *Кандидат*! Спасибі за інтерес до нашої вакансії. Вибачте за затримку з відповіддю, ми отримали велику кількість відгуків і перед прийняттям рішення уважно вивчили кожне резюме. На жаль, в даний час ми не можемо запропонувати вам роботу, тому що зробили вибір на користь іншого кандидата. Проте, в нашій компанії регулярно з'являються нові вакансії, які, можливо, теж вас зацікавлять. Слідкуйте за оновленнями! З повагою, 																	
Недолік досвіду		Здрастуйте, *Кандидат*! Дякую за інтерес до нашої вакансії. Вибачте за затримку з відповіддю, ми отримали велику кількість відгуків і перед прийняттям рішення уважно вивчили кожне резюме. На жаль, в даний час ми не можемо запропонувати вам роботу, тому що на дану вакансію шукаємо досвідченішого кандидата. Проте, в нашій компанії регулярно з'являються нові вакансії, які, можливо, теж вас зацікавлять. Слідкуйте за оновленнями! З повагою, 																	
Повернемося до вашої кандидатури		Здрастуйте, *Кандидат*! Спасибі за інтерес до нашої вакансії. Вибачте за затримку з відповіддю, ми отримали велику кількість відгуків і перед прийняттям рішення уважно вивчили кожне резюме. На жаль, зараз ми не готові зробити вам цю пропозицію. Ми уважно ознайомилися з вашим резюме, і, можливо, повернемося до Вашої кандидатури, коли у нас виникне така потреба. З повагою, *Підпис*.																	

Рис. 3.13 Сторінка із шаблонами повідомлень

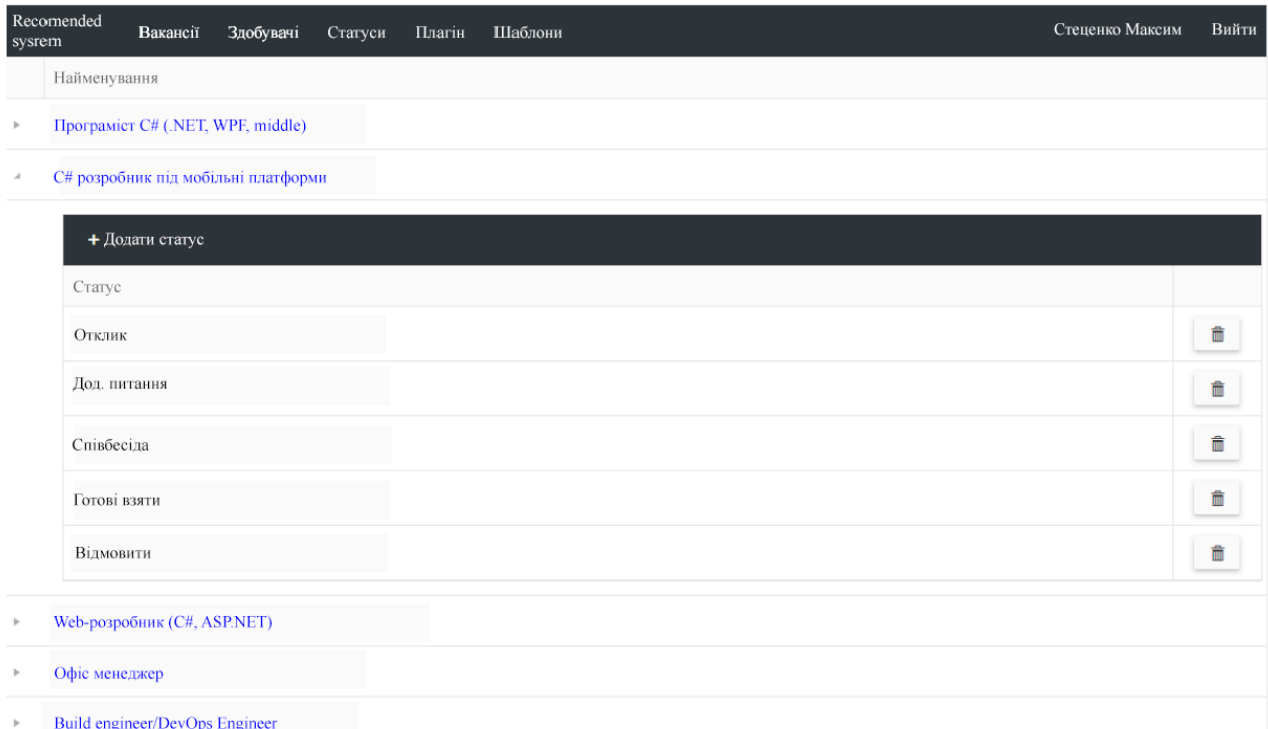


Рис. 3.14 Сторінка зі списком статусів по вакансіях

Для кожної вакансії передбачено власні статуси (Рис 3.14). Таким чином, на kanban-дошці у кожній вакансії будуть свої статуси. При додаванні нової вакансії з сайту «rabota.ua» на дану вакансію буде додано статуси, передбачені за замовчуванням.

Висновки до розділу 3

Спершу було обрано шлях вирішення поставлених завдань - розробка рекомендаційної системи управління підбором персоналу ІТ-підприємств на веб-основі. Переваги такого підходу полягають у тому, що веб-додатки є незалежними від платформи для кінцевого користувача, не вимагають установки, налаштування та адміністрування додаткового програмного забезпечення, а працювати з програмою можна фактично з будь-якої точки, де є доступ до Інтернету.

Розроблено рекомендаційну систему управління підбором персоналу ІТ-підприємств для автоматизації процесу найму персоналу. Розробка велася на основі проекту системи.

Реалізовано весь запланований функціонал системи, створено архітектуру системи та розроблено необхідні сценарії роботи. Система відповідає всім вимогам технічного завдання.

ВИСНОВКИ

За підсумками виконання магістерської дисертації розроблено рекомендаційну систему управління підбором персоналу ІТ-підприємств та супроводу їх до найму до організації. Для створення системи було проведено аналіз предметної галузі, проведено проектування інформаційної системи та бази даних, здійснено аналітичний огляд використовуваних технологій та реалізація запланованого функціоналу та інтерфейсу системи.

Створена система підтримує режим розрахований на багато користувачів, надаючи доступ авторизованим користувачам; надає дані щодо кандидатів, їх відгуків та вакансіями з бази даних та дозволяє керувати ними; дозволяє здійснювати пошук за кандидатами; зберігає архів кандидатів та вакансій. Система інтегрована з сайтом по підбору персоналу «rabota.ua», володіє можливістю ведення листування з здобувачами та співробітниками, дозволяє здійснювати множинний вибір кандидатів та масове відправлення повідомлень.

Розроблена інформаційна система є затребуваною на сьогоднішній день і має практичне значення. Автоматизація даного бізнес-процесу дозволяє підвищити швидкість прийняття рішень, забезпечила економію тимчасових, людських та матеріальних ресурсів. В результаті підвищується ефективність управління та продуктивність праці співробітників, які займаються підбором персоналу.

Описаний у роботі алгоритм створення інформаційної системи автоматизації процесу найму персоналу може бути використаний при розробці аналогічних систем накопичення та обробки інформації, автоматизації бізнес-процесів організацій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Клопов І.О., Стеценко М. Рекомендаційні системи як інструмент підвищення ефективності веб-сервісів. *Збірник наукових праць студентів, аспірантів, докторантів і молодих вчених «Молода наука-2021»* : у 5 т. / Запорізький національний університет. Запоріжжя : ЗНУ, 2021. Т.5. С. 228-230.
2. Клопов І.О., Стеценко М., Бобро Д.В. Методи рекомендаційних систем. *Європейський вектор модернізації інженерної та економіко-управлінської освіти в умовах сталого розвитку промислового регіону* : матеріали Міжнародної науково-практичної конференції (27-28 травня 2021 року, м. Запоріжжя). - Запоріжжя : Наук. ред. Н.Г. Метеленко. ЗНУ Інженерний навчально-науковий інститут, 2021. С. 115-118.
3. Practical Recommender Systems / Kim Falk. Shelter Island : Manning Publications Co., 2019. 432 с. URL: <https://www.twirpx.com/file/2737578/> (дата звернення: 19.09.2021)
4. Introduction to Recommender Systems // tryolabs : веб-сайт. URL: <https://tryolabs.com/blog/introduction-to-recommender-systems/> (дата звернення: 20.09.2021)
5. Netflix Prize – Wikipedia URL: https://en.wikipedia.org/wiki/Netflix_Prize (дата звернення: 20.09.2021)
6. Machine Learning for Building Recommender System in Python // towards data science : веб-сайт. URL: <https://towardsdatascience.com/machine-learning-for-building-recommender-system-in-python-9e4922dd7e97> (дата звернення: 21.09.2021)
7. Build a Recommendation Engine With Collaborative Filtering // realpython : веб-сайт. URL: <https://realpython.com/build-recommendation-engine-collaborative-filtering/> (дата звернення: 22.09.2021)

8. Introduction to recommender systems // towards data science: веб-сайт. URL: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada> (дата звернення: 23.09.2021)

9. Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries / за заг. ред Sachi Nandan Mohanty, Jyotir Moy Chatterjee, Sarika Jain, Ahmed A. Elngar and Priya Gupta. Scrivener Publishing LLC, 2020. - 448 с. URL: <https://www.twirpx.com/file/3205961/> (дата звернення: 23.09.2021)

10. Recommendation systems: Principles, methods and evaluation / за заг. ред F.O.Isinkaye, Y.O.Folajimi, B.A.Ojokoh: Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. Cairo University, 2015. URL: <https://reader.elsevier.com/reader/sd/pii/S1110866515000341?token=A275D895AED449837D47D64C51730ECFC81EE95C2CA0A94E5E56724B426910A3C3961AA3C7C34C4CA265F89A870A0113&originRegion=eu-west-1&originCreation=20210602172349> (дата звернення: 24.09.2021)

11. An In-Depth Guide to How Recommender Systems Work // builtin: веб-сайт. URL: <https://builtin.com/data-science/recommender-systems> (дата звернення: 24.09.2021)

12. Recommender Systems Handbook / за заг. ред Francesco Ricci, Lior Rokach, Bracha Shapira: Springer Science+Business Media New York 2011, 2015 1003 с. URL: <https://edyaaleh.files.wordpress.com/2016/02/recommendersystemshandbook.pdf> (дата звернення: 25.09.2021)

13. Social Network-Based Recommender Systems / Schall Daniel. Springer International Publishing Switzerland, 2015. 126 p URL: <https://www.twirpx.com/file/2418659/> (дата звернення: 25.09.2021)

14. Statistical Methods for Recommender Systems / за заг. ред Agarwal Deepak K., Chen Bee-Chung. Cambridge University : Press, 2016. 293 с. URL: <https://www.twirpx.com/file/2352271/> (дата звернення: 27.09.2021)

15. An Introduction to Recommendation Engines // dataconomy: веб-сайт. URL: <https://dataconomy.com/2015/03/an-introduction-to-recommendation-engines/> (дата звернення: 28.09.2021)

16. How Do Recommendation Engines Work? What are the Benefits? //maruti tech: веб-сайт. URL: <https://marutitech.com/recommendation-engine-benefits/> (дата звернення: 29.09.2021)

17. Recommender systems: from algorithmsto user experience / за заг. ред Joseph A. Konstan·John Springer Science+Business Media B.V., 2012 URL: <https://link.springer.com/content/pdf/10.1007/s11257-011-9112-x.pdf> (дата звернення: 03.10.2021)

18. Recommender systems, Part 1.Introduction to approaches and algorithms. Learn about the concepts that underlie web recommendation engines /M.Jones-2013. URL: <https://www.ibm.com/developerworks/library/os-recommender1/os-recommender1-pdf.pdf> (дата звернення: 07.10.2021)

19. How to Build a Recommendation Engine for Online Business // the app solutions: веб-сайт. URL: <https://theappsolutions.com/blog/development/recommender-systems-guide/> (дата звернення: 10.10.2021)

20. What Are Recommendation Systems in Machine Learning? // analytics steps: веб-сайт. URL: <https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning> (дата звернення: 15.10.2021)

21. Work.ua - сайт пошуку роботи №1 в Україні URL: <https://www.work.ua/> (дата звернення: 17.10.2021)

22. Работа в Украине. Трудоустройство и поиск работы на robotha.ua URL: <https://rabota.ua/> (дата звернення: 19.10.2021)

23. Работа в Украине, поиск персонала и публикация вакансий - grc.ua URL: <https://grc.ua/> (дата звернення: 19.10.2021)
24. Спільнота програмістів | DOU URL: <https://dou.ua/>(дата звернення: 20.10.2021)
25. Джин для пошуку роботи 🍀(°°🍀) URL: <https://djinni.co/> (дата звернення: 20.10.2021)
26. Работа в Украине. Трудоустройство и поиск работы на Нова Робота. URL: <https://novarobota.ua/> (дата звернення: 20.10.2021)
27. Работа в Україні. Свежие вакансии и трудоустройство в Україні на портале Trud.ua URL: <https://trud.ua/> (дата звернення: 22.10.2021)
28. Работа в Україні - 105.000+ актуальних вакансій | Jooble URL: <https://ua.jooble.org/> (дата звернення: 23.10.2021)
29. Онлайн-сервис для автоматизации рекрутинга FriendWork Recruiter URL: <https://friend.work/> (дата звернення: 24.10.2021)
30. E-Staff - Программа для HR и кадровых агентств URL: https://e-staff.ru/estaff_home (дата звернення: 25.10.2021)
31. Система для рекрутинга, ПО для подбора персонала. URL: <https://cleverstaff.net/ru/> (дата звернення: 27.10.2021)
32. C# docs - get started, tutorials, reference. | Microsoft Docs URL: <https://docs.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 30.10.2021)
33. Java Software | Oracle URL: <https://www.oracle.com/java/> (дата звернення: 01.11.2021)
34. Welcome to Python.org URL: <https://www.python.org/> (дата звернення: 03.11.2021)
35. Standard C++ URL: <https://isocpp.org/> (дата звернення: 03.11.2021)
36. PHP: Hypertext Preprocessor URL: <https://www.php.net/> (дата звернення: 05.11.2021)
37. JavaScript – Wikipedia URL: <https://en.wikipedia.org/wiki/JavaScript> (дата звернення: 06.11.2021)

38. Visual Studio: IDE and Code Editor for Software Developers and Teams
URL: <https://visualstudio.microsoft.com/> (дата звернення: 07.11.2021)
39. Eclipse IDE 2021-12 | The Eclipse Foundation URL:
<https://www.eclipse.org/eclipseide/>(дата звернення: 08.11.2021)
40. WebStorm — умная среда разработки для JavaScript, созданная в
JetBrains URL: <https://www.jetbrains.com/webstorm/>(дата звернення:
08.11.2021)
41. Sublime Text - Text Editing, Done Right URL:
<https://www.sublimetext.com/>(дата звернення: 10.11.2021)
42. Atom URL: <https://atom.io/> (дата звернення: 10.11.2021)
43. Visual Studio Code - Code Editing. Redefined URL:
<https://code.visualstudio.com/> (дата звернення: 10.11.2021)