

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 2025 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Підсистема розподілу кандидатів на вакансії у складі
інформаційної системи підтримки рекрутингу»

Виконала:

студент IV курсу, групи ІС-12

Гоголь Софія Сергіївна _____

Керівник:

доцент кафедри ІСТ, кандидат технічних наук,

Жданова Олена Григорівна _____

Рецензент:

професор каф. ОТ, д.т.н., професор

Кулаков Юрій Олексійович _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій
Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 126 «Інформаційні системи та технології»
Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студентці

Гоголь Софії Сергіївни

1. Тема проєкту «Підсистема розподілу кандидатів на вакансії у складі інформаційної системи підтримки рекрутингу», керівник проєкту Жданова Олена Григорівна, к.т.н., доцент, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: «09» червня 2025 року
3. Вихідні дані до проєкту: мова програмування C#, мова програмування TypeScript, фреймворк ASP.NET Core, фреймворк Entity Framework, фреймворк Angular, бібліотека Material Design.
4. Зміст пояснювальної записки: опис предметної області, аналіз готових рішень, постановка задачі, формування вимог до системи: вимоги до системи в цілому, вимоги до функціональних характеристик, вимоги до видів забезпечення, вибір технологій розробки, розробка інформаційної системи, реалізація програмної частини, тестування системи, оформлення пояснювальної записки.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма варіантів використання, модель бази даних, діаграма послідовності, діаграма класів.
6. Дата видачі завдання: «15» квітня 2025 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Опис предметної області	16 квітня 2025	
2	Аналіз готових рішень	19 квітня 2025	
3	Постановка та формалізація задачі	24 квітня 2025	
4	Формування вимог до системи	27 квітня 2025	
5	Вибір технологій розробки	1 травня 2025	
6	Опис математичного забезпечення	5 травня 2025	
7	Розроблення інформаційної системи	14 травня 2025	
8	Тестування системи	17 травня 2025	
9	Подання ДП на попередній захист	29 травня 2025	
10	Подання ДП на основний захист	9 червня 2025	

Студентка

Софія ГОГОЛЬ

Керівник проєкту

Олена ЖДАНОВА

АНОТАЦІЯ

Гоголь С.С. Підсистема розподілу кандидатів на вакансії у складі інформаційної системи підтримки рекрутингу. КПІ ім. Ігоря Сікорського, Київ, 2025.

Пояснювальна записка до дипломного проєкту складається з 61 сторінки та 7 розділів, що в свою чергу містять 8 рисунків, 17 таблиць, 1 додаток, 4 кресленики та 17 джерел.

АЛГОРИТМ МАКА, ДИСКРЕТНА ОПТИМІЗАЦІЯ, ЗАДАЧА ПРО ПРИЗНАЧЕННЯ, РОЗПОДІЛ КАНДИДАТІВ НА ВАКАНСІЇ, УГОРСЬКИЙ АЛГОРИТМ.

Об'єктом розробки є підсистема розподілу кандидатів на вакансії у складі інформаційної системи підтримки рекрутингу.

Мета розробки — підвищення обґрунтованості рішень у процесі військового найму.

У дипломному проєкті розроблено підсистему розподілу кандидатів на вакансії. За допомогою застосунку користувач може керувати даними про кандидатів, вакансії і заявки, запускати алгоритму автоматичного розподілу кандидатів на посади з урахуванням пріоритетів, квот і рівня відповідності кожного кандидата, а також переглядати ефективність розподілу. Функція розподілу кандидатів на вакансії реалізована у вигляді розв'язання задачі дискретної оптимізації, який враховує профілі кандидатів та вимоги до посад.

Було визначено можливості для подальшого покращення системи: реалізація автоматизованої системи пріоритезації вакансій на основі критичності бойових завдань; підтримка повторного розподілу кандидатів у разі змін кадрової ситуації.

Отримані результати можуть бути корисними при подальшій розробці та покращенні системи, зокрема у впровадженні цифрових рішень для управління кадровим потенціалом Сил оборони України.

SUMMARY

Hohol S.S. Candidate Allocation Subsystem of Recruitment Process Support Information System. Igor Sikorsky KPI, Kyiv, 2024.

The explanatory note to the diploma project consists of 62 pages and 7 sections, which in turn contain 10 figures, 17 tables, 1 annex, 4 draftsmen and 17 sources.

Keywords: assignment problem, candidate-to-position assignment problem, discrete optimization problem, Hungarian algorithm, Mack's algorithm.

The object of the development is a candidate-to-position allocation subsystem within a recruitment support information system.

The aim is to enhance decision-making in the military recruitment process.

The diploma project presents the implementation of a subsystem for the allocation of candidates to vacancies. The application enables users to manage data on candidates, vacancies, and applications, initiate an automated candidate-to-position allocation algorithm that considers assignment priorities, quota constraints, and individual suitability scores, as well as review the efficiency of the resulting allocations. The function of allocating candidates to vacancies is implemented as a solution to a discrete optimization problem that takes into consideration candidate profiles and job requirements.

Potential directions for further system improvement have been identified, including the development of an automated vacancy prioritization mechanism based on the urgency and criticality of operational tasks, as well as the implementation of functionality to support candidate reassignment in response to changes in personnel demands or organizational structure. The outcomes of this project may serve as a foundation for future system development and improvement, particularly in the context of deploying digital solutions for human resource management within the Defence Forces of Ukraine.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер ексем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC12.100БАК.005 ПЗ	Пояснювальна записка	61		
6	A3	IC12.100БАК.005 Д1	Підсистема розподілу	1		
7			кандидатів на вакансії			
8			у складі інформаційної			
9			системи підтримки			
10			рекрутингу. Діаграма			
11			варіантів використання			
12	A3	IC12.100БАК.005 Д2	Підсистема розподілу	1		
13			кандидатів на вакансії			
14			у складі інформаційної			
15			системи підтримки			
16			рекрутингу. Модель			
17			бази даних			
18	A3	IC12.100БАК.005 Д3	Підсистема розподілу	1		
19			кандидатів на вакансії			
20			у складі інформаційної			
21			системи підтримки			
22			рекрутингу. Діаграма			
23			послідовності			
24	A3	IC12.100БАК.005 Д4	Підсистема розподілу	1		
25			кандидатів на вакансії			
26			у складі інформаційної			
27			системи підтримки			
28			рекрутингу. Діаграма класів			

IC12.100БАК.005 ТП

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Гоголь С.С.			Підсистема розподілу кандидатів на вакансії у складі інформаційної системи підтримки рекрутингу. Відомість проекту	Літ.	Аркуш	Аркушів
Керівн.		Жданова О.Г.				Т	1	1
Затв.						КПІ ім. Ігоря Сікорського Група IC-12		

**Пояснювальна записка
до дипломного проєкту
на тему: «Підсистема розподілу кандидатів на вакансії у
складі інформаційної системи підтримки рекрутингу»**

Київ – 2025 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	4
ВСТУП	5
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Опис процесу діяльності	7
1.2 Постановка задачі	9
1.2.1. Призначення системи	9
1.2.2. Цілі та задачі розробки	9
Висновок до розділу 1	10
2 АНАЛІЗ ГОТОВИХ РІШЕНЬ.....	11
2.1 Загальні характеристики і типи існуючих рішень.....	11
2.2 Програмне рішення «CleverStaff»	11
2.3 Програмне рішення «HURMA».....	13
2.4 Програмне рішення «PeopleForce».....	14
Висновок до розділу 2	15
3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ	16
3.1 Вимоги до системи в цілому	16
3.2 Вимоги до функціональних характеристик.....	17
3.3 Вимоги до видів забезпечення.....	17
3.3.1 Математичне забезпечення	17
3.3.2 Інформаційне забезпечення.....	17
Висновок до розділу 3	18
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ	19
4.1 Мова програмування С#.....	19
4.2 SQL Server та Entity Framework.....	20
4.3 Платформа ASP.NET Core	20

					IC12.100БАК.005 ПЗ				
Зм.	Аркуш	№ докум.	Підпис	Дата	Підсистема розподілу кандидатів на вакансії у складі інформаційної системи підтримки рекрутингу. Пояснювальна записка	Літ.	Аркуш	Аркушів	
Розроб.	Гоголь С.С.						2	61	
Керівн.	Жданова О.Г.					КПІ ім. Ігоря Сікорського Група IC-12			
Затв.									

4.4 Мова програмування TypeScript.....	21
4.5 Фреймворк Angular	21
4.6 Бібліотека Material Design.....	22
Висновок до розділу 4	23
5 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	24
5.1 Структура системи.....	24
5.2 Функціональна модель системи	25
5.4 Передавання та обробка даних	28
5.5 Архітектура програмного забезпечення	29
Висновок до розділу 5	33
6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	35
6.1 Змістовна постановка задачі	35
6.2 Математична постановка задачі	35
6.3 Обґрунтування методу розв'язання	38
6.4 Опис методів розв'язання	39
6.4.1 Модифікований алгоритм Мака	40
Висновок до розділу 6	46
7 ТЕСТУВАННЯ СИСТЕМИ	47
7.1 Мета випробувань	47
7.2 Загальні положення.....	48
7.3 Результати випробувань	48
Висновок до розділу 7	58
ВИСНОВКИ	59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТОК А	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface – прикладний програмний інтерфейс.

ATS – Applicant Tracking System – система обліку кандидатів.

HRM – Human Resource Management – система управління персоналом.

REST – REpresentational State Transfer – передача репрезентативного стану.

SPA – single-page application – односторінковий застосунок.

ORM – Object-relational mapping – об’єктно-реляційна проєкція.

JWT – JSON Web Token – стандарт токена на основі JSON.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			4

ВСТУП

У межах збройного спротиву України повномасштабній агресії ворога цифровізація стала не просто інструментом, а стратегічною необхідністю, яка визначає здатність Сил оборони адаптуватися до умов гібридної війни [1]. У динамічному бойовому середовищі інтенсивність інформаційного потоку зростає експоненційно: від даних розвідки про розташування ворожих підрозділів до оперативної інформації про технічний стан техніки, логістичні потреби та кадрові резерви. Традиційні процедури пошуку, відбору та розміщення особового складу при цьому потребують надто багато часу й людських ресурсів, що неприпустимо, коли кожна хвилина може вартувати втрати життів або позицій на фронті.

Класичні методи управління людськими ресурсами у військових формуваннях передбачають значний обсяг паперової роботи, багатоетапні узгодження та формальні процедури затвердження кандидатур. У мирний час це може бути виправданим через відносно стабільне середовище, але в умовах постійних обстрілів, раптових атак та частих переміщень фронту такі підходи виявляються неефективними. Кадрові резерви потребують негайного перерозподілу залежно від пріоритетності завдань, а без єдиної цифрової платформи інформація про наявні кадри, їхню кваліфікацію, мотивацію та фізичний стан залишається розпорошеною по різних підрозділах і паперових архівах [2]. Через це виникає значний ризик дублювання зусиль, непродуктивних затримок і помилок під час призначення особового складу на критичні позиції.

Ефективність військового управління тепер безпосередньо залежить від здатності у реальному часі формувати боєздатні підрозділи, забезпечуючи їх повним спектром інформації про рівень підготовки, стан здоров'я та мотивацію кожного бійця. У цьому контексті рекрутинг виступає ключовим елементом, який передбачає не лише швидкий відбір кандидатів із відповідними фаховими навичками, а й їх оперативне оцінювання та розподіл на актуальні бойові чи тиллові позиції за критеріями пріоритетності, квотування й персональних побажань.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			5

Проте традиційні підходи до добору персоналу, що застосовуються в цивільних HRM-системах, виявилися недостатніми для умов воєнного часу, де кадрові рішення мають прийматися швидко, точно та з урахуванням багатьох факторів — від кваліфікацій і мотивації до пріоритетності закриття окремих вакансій.

У таких умовах процес розподілу кандидатів на посади перетворюється на задачу дискретної багатокритеріальної оптимізації, що вимагає використання математичних моделей і спеціалізованих алгоритмів. Необхідність одночасно враховувати як вимоги військових підрозділів, так і особисті характеристики кандидатів формує запит на створення інтелектуальних систем, які здатні забезпечити зважене призначення особового складу.

Дипломний проєкт складається з наступних розділів: вступ, основні розділи, висновки, список використаних джерел із 17 найменувань, 1 додатку. Графічна частина включає 4 кресленики формату А3. Загальний обсяг — 61 сторінка.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			6

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис процесу діяльності

Створення сучасних інформаційних систем у Силах оборони стало критично важливим елементом удосконалення ведення бойових дій. У контексті цифровізації оборонного сектору розробка інтегрованих рішень спрямована на забезпечення комплексної автоматизації ключових процесів — від ситуаційної обізнаності до управління ресурсами та підтримки прийняття рішень.

Для ефективного управління особовим складом в умовах сучасної війни необхідно впроваджувати систему, що охоплює всі аспекти кадрової роботи — від обліку до призначення. Така система має бути поділена на спеціалізовані підсистеми, кожна з яких виконує окрему функцію, але водночас інтегрована у загальний інформаційний контур. Це забезпечить цілісність даних, оперативність рішень та гнучкість у реагуванні на зміни бойової ситуації. Прикладом таких підсистем виступають:

- система обліку персональних і медичних даних;
- система оцінювання бойової та фахової підготовки;
- система формування ротаций і графіків служби;
- система первинного оцінювання кандидатів;
- система розподілу кандидатів на вакансії;
- система аналітики кадрових рішень.

У сучасних реаліях військового рекрутингу ключовим завданням є оперативне та ефективне закриття термінових вакансій за рахунок призначення кандидатів із найбільш відповідним рівнем кваліфікації. Відповідність навичок та мотиваційна складова відіграють все більшу роль під час призначення кандидатів на посади, що у довготривалій перспективі забезпечує стійкий результат, зменшує ризик швидкої ротации та підвищує ефективність команди в цілому [3]. Тому процес, що може виглядати, на перший погляд як простий добір персоналу, насправді перетворюється на нетривіальну задачу оптимізації, де враховуються й терміни, і компетенції претендентів, і доступні ресурси. Саме тому виникає потреба

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			7

в розробці алгоритмів, здатних з достатньою точністю та в прийнятних часових рамках визначати найкращі кандидатури для кожної позиції.

За імплементацією реформи [4] у сфері військового рекрутингу з'явилась можливість обирати бажану посаду у війську. У такій концепції, армія, як загальна структура, виконує роль працедавця, що має відкриті позиції з конкретними обов'язками, вимогами та умовами, залежно від спеціальності. Зі свого боку, особа, яка виявила бажання долучитися до служби у війську, може обрати бажану посаду та відправити своє резюме через спеціальні платформи, де розміщується вакансія.

Військові рекрутери отримують від кожного підрозділу набір вимог (критеріїв) до кандидатів на відповідні вакансії та здійснюють так званий прескринінг, визначаючи кандидатів, найбільш релевантних до зазначених позицій. При цьому зазвичай враховується декілька критеріїв, що відображають конкретні потреби підрозділу.

У цій роботі пропонується використовувати наступний підхід. Оскільки оперувати із багатьма критеріями зазвичай важко, то замість використання декількох критеріїв пропонується використовувати комбіновану оцінку кандидата, яка би враховувала відповідність кандидата до вимог кожної вакансії. У контексті цієї роботи процес розрахунку такої оцінки розглядається як «чорний ящик» (тобто механізм розрахунку не деталізується), і вважається, що результати доступні у вигляді кінцевої оцінки відповідності кандидата кожній вакансії. Ці дані можливо отримати з підсистеми первинного оцінювання кандидатів, що згадувалась раніше.

Крім того, кожна вакансія має числову характеристику — пріоритетність, що зумовлена як особливостями комплектування підрозділу, так і фактором нестачі кадрів. Це передбачає, що вакансії з вищим пріоритетом мають закриватися швидше або за участі найбільш відповідних кандидатів.

З огляду на наявність подібної оцінки та необхідності врахування пріоритетності закриття вакансії, врахуємо ці характеристики під час формування задачі для розподілу кандидатів на вакансії. Імплементація розв'язку такої задачі стає підґрунтям до розробки автоматизованої підсистеми, що становить собою частину цілісної системи підтримки рекрутингового процесу.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			8

1.2 Постановка задачі

1.2.1. Призначення системи

Підсистема призначена для розподілу кандидатів на вакансії з врахуванням числових оцінок відповідності кандидатів та визначеної пріоритетності вакансій, яка є складовою загальної системи підтримки військового рекрутингу.

1.2.2. Цілі та задачі розробки

Метою розробки є підвищення обґрунтованості рішень у процесі військового найму за рахунок використання математичних методів оптимізації, які враховують відповідність кандидатів вимогам до конкретних посад. Система покликана продемонструвати ефективність застосування цифрових технологій у сфері кадрового управління та слугувати прототипом для подальших рекомендацій щодо впровадження подібних рішень у практику Сил оборони.

Задля реалізації поставленої мети необхідно реалізувати такі функції:

- а) авторизація у системі;
- б) ведення кандидатів;
- в) ведення вакансій;
- г) розподіл кандидатів на вакансії;
- д) оцінка результату розподілу;
- е) перегляд результатів розподілу.

Розподіл кандидатів на вакансії має враховувати профілі кандидатів, вимоги до посад та обмеження ресурсів.

Оцінка результатів розподілу має відображати ступінь відповідності кандидатів до вакансій, на які вони призначені.

Розроблювана система є прототипом, що має також включати механізм генерації випадкових даних для перевірки її працездатності та демонстрації основної логіки розподілу.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			9

Висновок до розділу 1

У цьому розділі було обґрунтовано актуальність цифровізації процесів управління особовим складом у Силах Оборони та визначено роль підсистеми розподілу кандидатів на вакансії в загальній архітектурі інформаційної системи підтримки рекрутингу.

Проаналізовано специфіку сучасного військового найму, в якому ключову роль відіграють пріоритетність посад, релевантність навичок та потреба в оперативному прийнятті рішень.

Сформульовано призначення майбутньої системи, її мета та функції, а також окреслено підхід до побудови прототипу з можливістю генерації тестових даних, що дозволяє перевірити її працездатність і закласти основу для подальшого впровадження подібних рішень у сфері оборонного управління.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			10

2 АНАЛІЗ ГОТОВИХ РІШЕНЬ

2.1 Загальні характеристики і типи існуючих рішень

Загалом для роботи з кадрами використовуються різні типи застосунків, що націлені під певний аспект задач або етап найму кандидата.

Серед найбільш поширених виділяють наступні:

- Applicant Tracking System (ATS): CleverStaff, HURMA та PeopleForce;
- Human Resources Information System / Human Resource Management System (HRIS / HRMS): One Click, Remote;
- CRM для рекрутингу: Notion, HireFire.

Під час процесу найму, кандидата можуть супроводжувати різні команди рекрутингу та HR-відділів, що мають для своєї роботи різного типу застосунки та інструменти. Або ж всередині однієї команди використовується комбінація систем для виконання різних задач: пошуку та відстеження профілів, трекінг етапів комунікацій, процес найму та онбордингу тощо.

На ринку присутні як готові рішення, що адаптуються під потреби команди, зокрема за рахунок додавання модулів, так і ті, що створені та підтримуються власне самими компаніями, що забезпечує ще більшу гнучкість у процесах рекрутингу та ведення кандидатів.

Кожен із зазначених вище типів може бути обраний та застосований як система для оптимізації найму робітників, але надалі детальніше увагу зосереджено на розборі інструментів для відстеження кандидатів Applicant Tracking System (ATS) через відповідність функціям, виконання яких має забезпечувати розроблена система.

2.2 Програмне рішення «CleverStaff»

«CleverStaff» – це ATS для рекрутингових агентств і внутрішніх рекрутерів. Система пропонує гнучке налаштування воронки підбору (можна створювати власні етапи та посади), імпорт/експорт даних, розсилку листів кандидатам і

					IC12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			11

інтеграцію із LinkedIn (через браузерне розширення). Є можливість автоматично додавати кандидата в базу одним кліком (CV-парсер), система «Recommendations» підказує найвідповідніші резюме, автоматизований підбір кандидатів здійснюється представленням кандидатів з найбільшою відповідністю резюме до полів, що відстежуються.

Сторінка з виглядом системи зображена на рисунку 2.1.

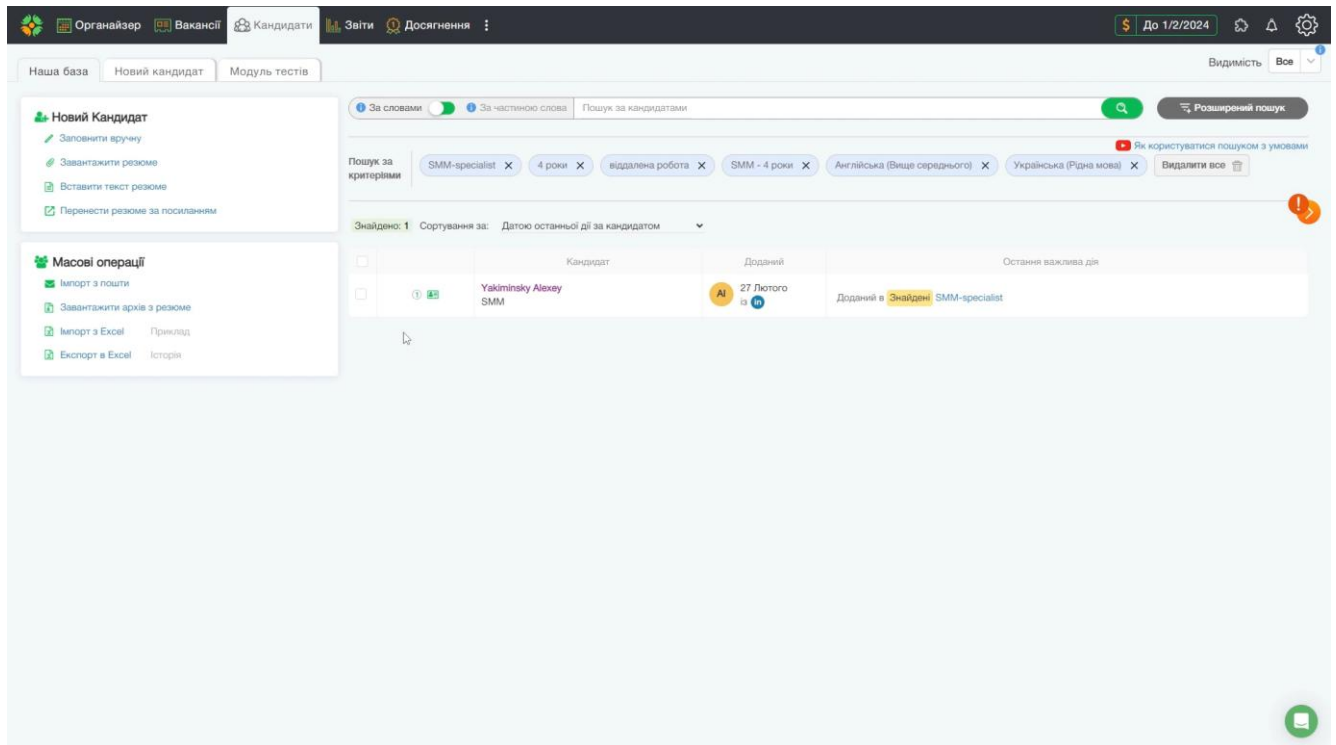


Рисунок 2.1 – Інтерфейс програмного рішення «CleverStaff»

Переваги:

- інтуїтивний інтерфейс;
- і просунуті інструменти пошуку та імпорту резюме.

Недоліки:

- повільна робота веб-інтерфейсу;
- дизайн і UX застарілий у порівнянні з конкурентами.

2.3 Програмне рішення «HURMA»

«HURMA» – комплексна HRM/ATS-система. З її допомогою можна створювати вакансії та публікувати їх на популярних порталах. Вбудований парсер резюме підтримує різні формати – кандидата швидко додають у базу й прив'язують до вакансії. Спеціальне браузерне розширення (AI Fox) дозволяє «забирати» кандидатів з інших сайтів у пару кліків і одразу додавати їх у воронку.

Сторінка з виглядом системи зображена на рисунку 2.2.

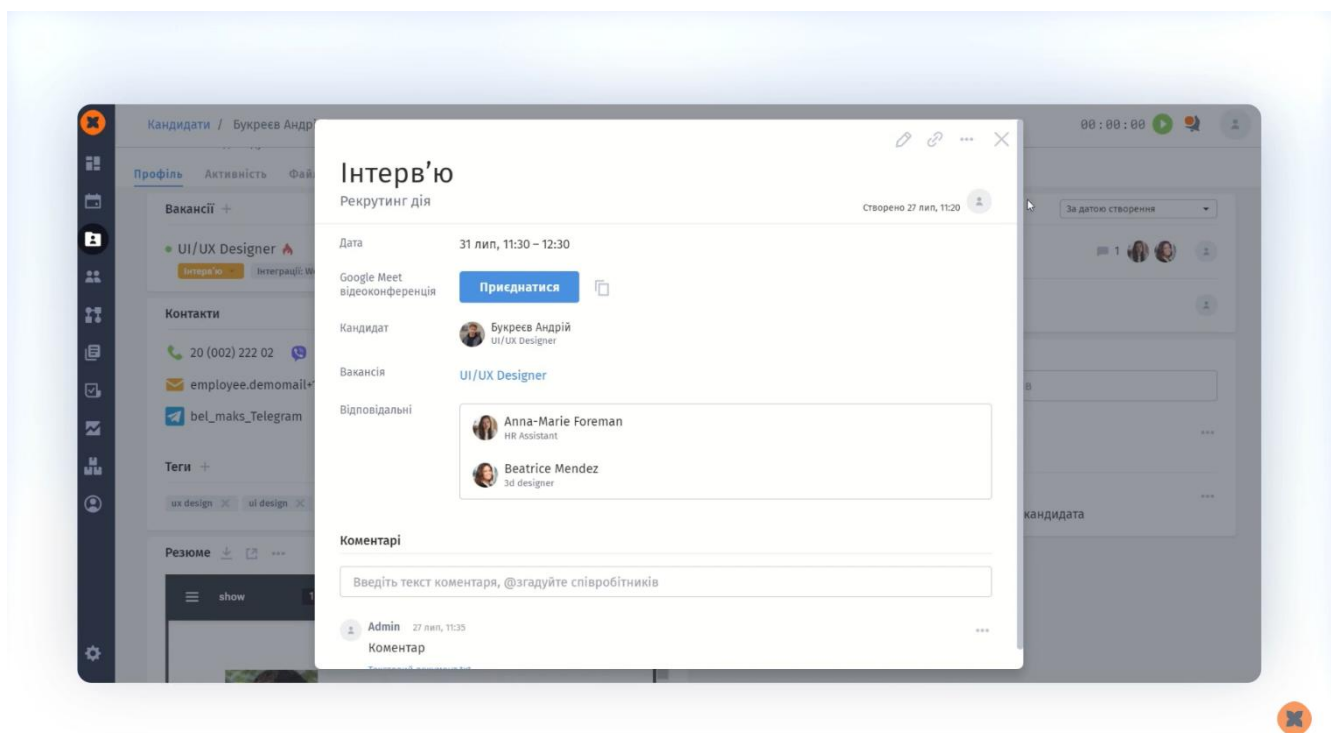


Рисунок 2.2 – Інтерфейс програмного рішення «HURMA»

Переваги:

- простота впровадження у робочий процес;
- наявність автоматизації повторюваних процесів (фільтри, статуси).

Недоліки:

- негнучкий підхід до кастомізації стадій відбору;
- висока вартість акаунту користувача.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			13

2.4 Програмне рішення «PeopleForce»

«PeopleForce» – модульна HRM-платформа, що має ATS-модуль («Recruit»), який дозволяє створювати вакансії та публікувати їх і на корпоративному сайті, і на дошках пошуку роботи з одного інтерфейсу. Система має Chrome-розширення PeopleForce Prospector, що швидко парсить резюме з інших джерел і додає їх у систему.

Сторінка з виглядом системи зображена на рисунку 2.3.

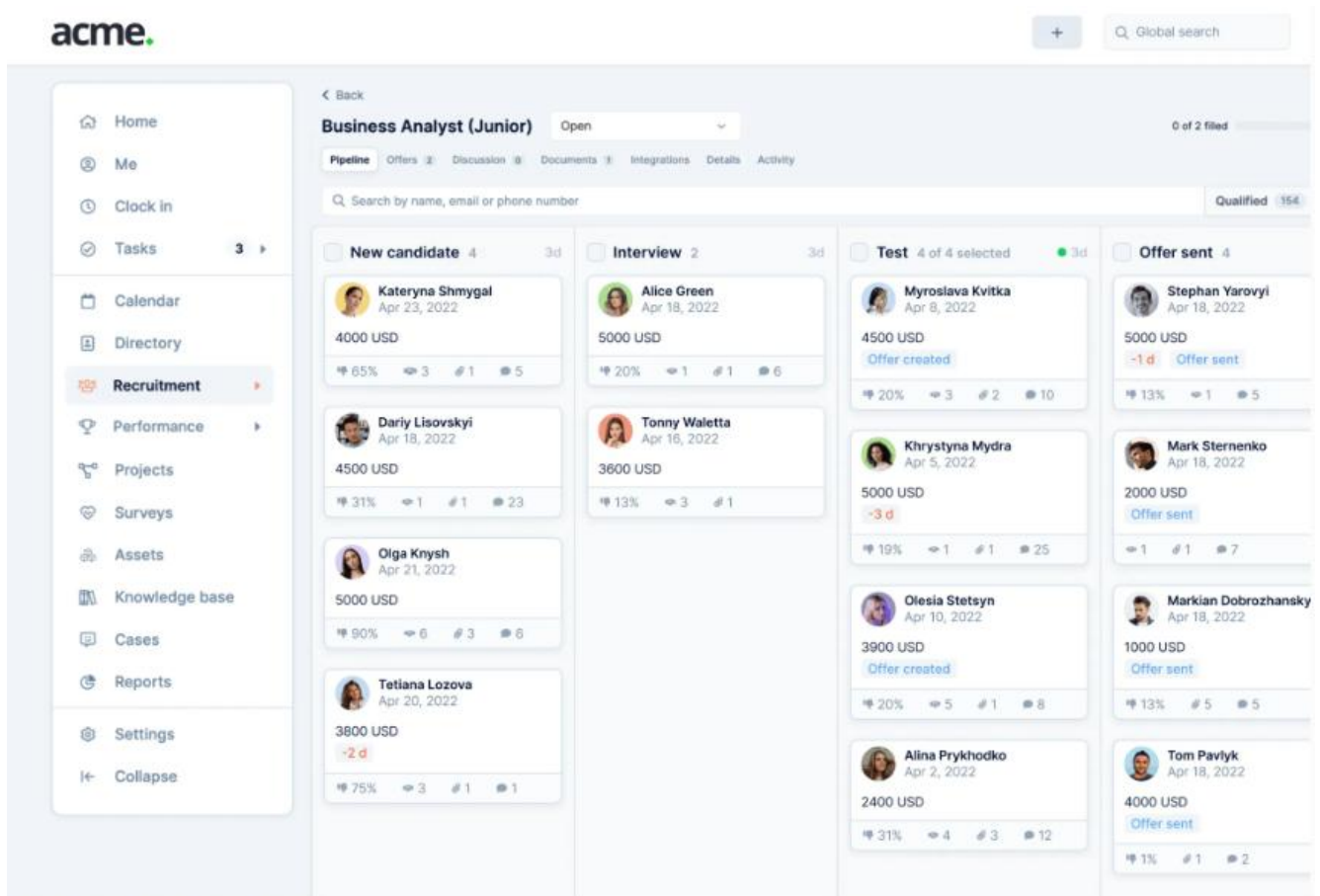


Рисунок 2.3 – Інтерфейс програмного рішення «PeopleForce»

Переваги:

- сучасний і приємний інтерфейс;
- можливість швидкої фільтрації й тегування кандидатів.

Недоліки:

- складна структура системи;

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			14

– значна частина функцій наявна тільки в додаткових платних модулях.

Проведений аналіз існуючих програмних рішень показав, що більшість із них є комерційними, мають високу вартість та недоступні для впровадження в державних установах через обмежені ресурси. Водночас жодна з розглянутих систем не підтримує функціональності інтелектуального розподілу кандидатів, що враховує математичні моделі, пріоритетність вибору та квотування (розподіл кількох кандидатів на одну вакансію).

Наразі наявні рішення, що представлені на ринку дають змогу проаналізувати тенденції розвитку подібних продуктів, виокремити функції, що є найчастіше використовуваними та виявити усталені підходи до проектування подібних систем.

Висновок до розділу 2

У результаті аналізу сучасних програмних засобів для роботи з кадрами було встановлено, що ринок представлений різними типами систем, орієнтованими на окремі етапи рекрутингового процесу — ATS, HRIS/HRMS та CRM-рішення. Попри функціональну різноманітність, ці продукти здебільшого є комерційними та фінансово недоступними для державного сектору.

Детальне вивчення рішень CleverStaff, HURMA та PeopleForce висвітлило їхні сильні сторони, як-от інтеграція з зовнішніми сервісами, автоматизація рутинних дій та підтримка роботи з базами кандидатів. Водночас виявлено низку обмежень, зокрема високу вартість, складність адаптації до специфічних процесів та відсутність підтримки механізмів інтелектуального розподілу кандидатів з урахуванням пріоритетів і квот.

Таким чином, огляд існуючих рішень дозволив визначити ключові функціональні можливості, сучасні практики та архітектурні підходи, що повинні бути враховані при розробці власної системи, орієнтованої на потреби автоматизованого та гнучкого розподілу кандидатів.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			15

3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

Інформаційна система, що супроводжує процес військового рекрутингу, та будь-яка її підсистема має бути побудована згідно з певними функціональними та нефункціональними вимогами [5], аби забезпечити усі зацікавлені сторони тими можливостями, яких вони потребують під час працевлаштування.

3.1 Вимоги до системи в цілому

З огляду на потенційне розповсюдження системи по усіх центрах рекрутингу української армії та впливаючу з цього необхідність у легкій масштабованості, а також передбачувану інтеграцію системи в загальнодержавний військовий ІТ-кластер, система має відповідати модульному підходу: клієнт та сервер мають складатися з простіших елементів-модулів, які матимуть свою зону відповідальності та будуть легко замінювані, коли виникне така потреба.

До клієнтської частини необхідно реалізувати такі модулі:

– модуль підключення до серверної частини — елемент системи, що «споживатиме» (англ. consume) API з сервера за допомогою HTTP-клієнта.

– модуль представлення даних та взаємодії з користувачем — елемент системи, що відповідатиме за відображення у браузері та опрацьовуватиме введення від користувача.

До серверної частини необхідно реалізувати такі складові:

– база даних;

– модуль підключення до бази даних та підготовки/обробки запитів до неї;

– модуль «бізнес-логіки» — ключовий елемент застосунку, де опрацьовуватиметься розподіл кандидатів та виконуватимуться так звані «бізнес правила» процесу, який підтримує система;

– модуль API — частина системи, яка надаватиме можливість підключення зовнішніх систем для передачі даних між ними, а також для передачі даних до клієнта.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			16

3.2 Вимоги до функціональних характеристик

Для користувачів системи мають бути доступні наступні дії:

- авторизуватися;
- вести перелік кандидатів;
- вести перелік вакансій;
- генерувати тестові дані вакансій, кандидатів та їхніх заявок на вакансії;
- провести розподіл кандидатів на вакансії, засновуючись на їхній оцінці за кожною заявкою;
- оцінити ступінь врахування бажань кандидатів;
- переглядати результати розподілу;
- переглядати ефективність розподілу.

3.3 Вимоги до видів забезпечення

3.3.1 Математичне забезпечення

У системі має бути реалізований ефективний алгоритм розподілу кандидатів на вакансії, який враховуватиме оцінку відповідності кандидата до кожної з вакансій, на яку він подавався. Також має бути реалізована можливість обчислення ступеню врахування бажань кандидатів, що свідчитиме про ефективність розподілу.

3.3.2 Інформаційне забезпечення

Для зберігання та обробки інформації про процес рекрутингу має бути використана база даних, основу якої складатимуть сутності «Вакансія», «Кандидат», «Заявки». Для реєстрації користувачів та їхньої авторизації у системі необхідно використати відповідні механізми мов програмування, що будуть відображені у розширенні структури бази даних допоміжними таблицями.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			17

Висновок до розділу 3

У цьому розділі було сформовано перелік вимог до інформаційної системи, що підтримує процес військового рекрутингу. Визначено архітектурні принципи побудови системи — зокрема модульний підхід, що забезпечує масштабованість, гнучкість і можливість інтеграції з іншими підсистемами в межах військового ІТ-кластера. Така архітектурна організація дозволяє гнучко адаптувати систему до змін у середовищі її використання, а також спрощує оновлення окремих компонентів без впливу на загальну працездатність.

Сформульовано чіткі функціональні вимоги, які охоплюють повний спектр дій, необхідних для ефективного управління процесом розподілу: від авторизації користувача та ведення переліків кандидатів і вакансій до запуску алгоритму розподілу та перегляду його результатів. Такий підхід гарантує централізований контроль над процесом та узгодженість усіх дій адміністратора.

Окремо розглянуто вимоги до математичного та інформаційного забезпечення. Було зазначено, що система має включати ефективний алгоритм розподілу кандидатів з урахуванням їх відповідності кожній вакансії, а також механізми оцінки врахування інтересів кандидатів — це забезпечить прозорість, справедливість та ефективність кадрових рішень. Інформаційне забезпечення передбачає побудову реляційної бази даних, яка охоплює основні сутності, необхідні для збереження ключових рекрутингових даних і підтримки цілісності та надійності системи.

Таким чином, сформовані вимоги створюють надійну основу для подальшого проектування та розробки системи, яка відповідатиме сучасним стандартам військового управління та забезпечить автоматизацію одного з найважливіших процесів оборонного менеджменту — призначення особового складу на критично важливі посади.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			18

4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

Для реалізації інформаційної системи розподілу кандидатів на вакансії у складі підсистеми військового рекрутингу необхідно обрати технології, що забезпечать надійність, масштабованість та зручність розробки. З огляду на функціональні вимоги, архітектура системи поділяється на клієнтську та серверну частини, що взаємодіють через API. Обраний технологічний стек має підтримувати реалізацію математичних алгоритмів, обробку запитів до бази даних, авторизацію користувачів та зручний інтерфейс для роботи з даними.

4.1 Мова програмування C#

Для реалізації серверної частини інформаційної системи було обрано мову програмування C# [6]. Це сучасна, універсальна мова, яка ідеально підходить для створення функціонально насичених інформаційних рішень. C# має високу популярність серед розробників, що забезпечує доступ до великої кількості перевірених рішень, методологій та практик розробки.

Офіційна підтримка з боку компанії Microsoft гарантує стабільність та перспективність використання мови у довгостроковій перспективі. Це критично важливо для корпоративних застосунків, які зазвичай мають складну архітектуру і потребують постійного супроводу та оновлення протягом тривалого періоду.

Серед технічних переваг мови слід відзначити строгість типізації, що дозволяє виявляти помилки на етапі компіляції та забезпечує цілісність роботи з типами даних. Такий підхід підвищує стабільність і передбачуваність поведінки системи.

C# побудована на принципах об'єктно-орієнтованого програмування, що дозволяє структурувати програмний код у логічно розмежовані модулі, які легко масштабуються та обслуговуються в процесі еволюції системи.

Додатковою перевагою є наявність розвиненої екосистеми бібліотек і фреймворків, таких як ASP.NET Core і Entity Framework, які значно прискорюють

					IC12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			19

розробку сучасних веб-систем, забезпечуючи при цьому високу продуктивність та гнучкість.

4.2 SQL Server та Entity Framework

Для організації зберігання даних у системі було обрано реляційну СУБД Microsoft SQL Server у поєднанні з ORM-фреймворком Entity Framework для мови С#.

Microsoft SQL Server — це високопродуктивна, надійна та масштабована система управління базами даних, яка широко використовується у середовищі корпоративних застосунків. Її функціональні можливості, зокрема безпека та підтримка складних запитів, роблять її ефективним вибором для критичних бізнес-систем.

Entity Framework забезпечує зручний спосіб взаємодії з базою даних шляхом представлення таблиць у вигляді об'єктів С#. Це значно знижує потребу в написанні SQL-коду вручну, що пришвидшує розробку та покращує читаність програмного коду. Незважаючи на певне зниження продуктивності через додатковий рівень абстракції, на практиці це майже не позначається на користувацькому досвіді.

Однією з ключових переваг зв'язки Entity Framework із SQL Server є підтримка механізму міграцій [7]. Визначення структури бази даних безпосередньо у вигляді С#-коду дозволяє автоматично створювати та оновлювати схему бази через згенеровані SQL-скрипти. Це значно полегшує процес розгортання, супроводу та версіонування бази даних. Крім того, система міграцій дозволяє легко повернутись до попередніх версій схеми за потреби, що підвищує гнучкість і контроль над розвитком проекту.

4.3 Платформа ASP.NET Core

ASP.NET Core — це сучасний фреймворк з відкритим кодом, призначений

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			20

для створення високопродуктивних веб-застосунків на мові програмування C#. Його кросплатформенність дозволяє розгортати застосунки на різних операційних системах, що робить його гнучким і зручним для використання в різних середовищах.

Фреймворк підтримує модульну архітектуру та механізм впровадження залежностей (dependency injection), що спрощує структуру застосунку, полегшує його тестування та підвищує підтримуваність коду [8]. Завдяки вбудованій підтримці Web API ASP.NET Core дозволяє швидко реалізовувати RESTful-сервіси, які слугують інтерфейсом взаємодії як для клієнтської частини, так і для сторонніх сервісів.

Глибока інтеграція з Entity Framework Core надає доступ до повноцінного ORM-інструменту, який підтримує використання LINQ, управління схемою бази даних за допомогою міграцій та роботу з даними без необхідності писати SQL-код вручну. Усе це сприяє підвищенню продуктивності розробки та забезпечує зручність підтримки застосунку.

4.4 Мова програмування TypeScript

TypeScript — це мова програмування, яка є надмножиною JavaScript і додає до нього статичну типізацію та об'єктно-орієнтовані можливості. Вона була розроблена для покращення масштабованості та підтримуваності великих фронтенд-застосунків, що особливо актуально в умовах складних веб-систем.

Однією з ключових переваг TypeScript є наявність статичної типізації, яка дозволяє виявляти помилки ще на етапі компіляції, підвищуючи надійність та стабільність коду. Крім того, типізація полегшує навігацію в коді, автодоповнення та рефакторинг, що значно покращує досвід розробника.

4.5 Фреймворк Angular

Для реалізації клієнтської частини інформаційної системи було використано

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			21

Angular — сучасний фреймворк для розробки односторінкових веб-застосунків. Angular надає розробникам розвинений набір інструментів, серед яких двостороннє зв'язування даних, система компонентів і директив, що дозволяє створювати динамічні та інтерактивні інтерфейси користувача.

Angular заснований на мові TypeScript [9], яка забезпечує статичну типізацію та зручність розробки за рахунок кращої читаємості, автодоповнення та зменшення кількості помилок під час компіляції. Це дозволяє розробляти масштабовані застосунки з чіткою структурою.

Однією з ключових особливостей Angular є його модульна архітектура. Основним елементом є компонент, що містить власну логіку, шаблон, стилі та тести. Компоненти незалежні, легко повторно використовуються та об'єднуються через кореневий модуль застосунку.

Фреймворк також має гнучку систему маршрутизації (роутингу), яка дозволяє ефективно керувати переходами між сторінками. Завдяки використанню guard-механізмів можна реалізувати авторизацію доступу до окремих маршрутів, що підвищує безпеку застосунку та дозволяє налаштувати контроль доступу на рівні інтерфейсу.

4.6 Бібліотека Material Design

Material Design — це дизайн-система від Google, яка включає набір готових UI-компонентів та стилістичних рішень для створення сучасних, естетично привабливих і зручних у користуванні інтерфейсів. Її використання сприяє дотриманню єдиного стилю у застосунку та покращує взаємодію з користувачем.

Angular Material — офіційна реалізація Material Design для Angular — забезпечує легку інтеграцію компонентів у проєкт, а також надає детальну документацію та приклади використання. Це дозволяє значно скоротити час на розробку інтерфейсу та спростити його подальше розширення та підтримку.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			22

Висновок до розділу 4

У цьому розділі було обґрунтовано вибір технологій, необхідних для реалізації підсистеми розподілу кандидатів на вакансії у складі інформаційної системи підтримки військового рекрутингу. Враховуючи модульну архітектуру проєктованого рішення та вимоги до продуктивності, надійності й масштабованості системи, було сформовано цілісний технологічний стек, який забезпечує ефективну взаємодію клієнтської та серверної частин через RESTful API.

На рівні серверної частини було обрано мову програмування C#, фреймворк ASP.NET Core для реалізації логіки та інтерфейсів взаємодії, а також Entity Framework у поєднанні з Microsoft SQL Server для управління структурованими даними. Така комбінація дозволяє швидко розгортати серверну логіку, ефективно реалізовувати алгоритми обробки заявок та гарантувати стабільність зберігання даних із підтримкою механізму міграцій.

Клієнтську частину реалізовано на основі Angular та мови TypeScript, що забезпечує модульність та високий рівень інтерактивності користувацького інтерфейсу. Візуальна складова оформлена з використанням бібліотеки Material Design, що забезпечує естетичну послідовність, відповідність сучасним UI-практикам та покращену взаємодію з користувачем.

Загалом, обраний технологічний стек не лише відповідає сучасним вимогам до розробки інформаційних систем, а й забезпечує простоту підтримки, гнучкість для майбутнього розширення та можливість інтеграції з іншими державними або військовими інформаційними системами. Такий підхід створює надійну основу для реалізації подальших етапів розробки та впровадження системи в умовах оборонної цифрової інфраструктури.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			23

5 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

5.1 Структура системи

Для реалізації інформаційної системи розподілу кандидатів на вакансії було використано тришарову архітектуру, яка є одним із найпоширеніших підходів у розробці сучасних програмних рішень. Така архітектура забезпечує чітке розмежування відповідальності між окремими логічними рівнями, що забезпечує модульність, розширюваність, зручність супроводу та повторного використання компонентів системи. На рисунку 5.1 відображено графічне представлення описаного підходу.

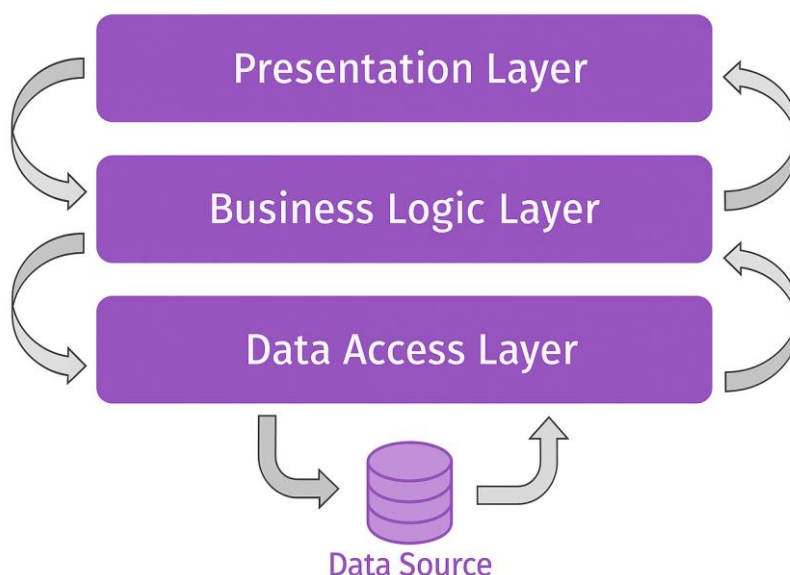


Рисунок 5.1 – Модель тришарової архітектури

У структурі системи виділено три основні шари: рівень доступу до даних, рівень бізнес-логіки та рівень представлення. На найнижчому рівні працює база даних, взаємодія з якою реалізована за допомогою об'єктно-реляційного фреймворку Entity Framework Core. Цей рівень відповідає за зберігання, вибірку та оновлення інформації, забезпечуючи при цьому прозорий доступ до даних для вищих рівнів.

У центрі архітектури розміщується рівень бізнес-логіки, що втілює основні функціональні процеси системи. Саме цей рівень забезпечує узгоджену роботу

Зм.	Лист	№ докум.	Підпис	

застосунку й організовує обробку даних, отриманих з бази.

Рівень представлення формує користувацький інтерфейс та відповідає за взаємодію з кінцевими користувачами. Веб-інтерфейс розроблено з використанням Angular та бібліотеки Material Design, що дозволяє створити сучасний, інтуїтивно зрозумілий і адаптивний дизайн. Обмін даними між клієнтом і сервером відбувається через RESTful API, реалізований за допомогою ASP.NET Core Web API.

Застосування тришарової архітектури дозволяє ефективно підтримувати логічну ізоляцію окремих компонентів системи, спрощує розгортання в умовах реального середовища, забезпечує підвищену безпеку й полегшує реалізацію змін або розширень без необхідності значного перероблення загальної структури.

5.2 Функціональна модель системи

Функціональна модель системи описує основні дії користувачів та поведінку системи у відповідь на ці дії. У розробленій системі передбачено лише одну роль користувача — адміністратор, який виконує всі операції в межах рекрутингового процесу. Після проходження процедури авторизації, адміністратор отримує доступ до ключових функцій системи: ведення переліку кандидатів, керування базою вакансій та перегляду результатів автоматизованого розподілу кандидатів на наявні посади. Такий підхід дозволяє централізовано контролювати якість введених даних, забезпечує узгодженість дій та спрощує адміністрування системи в умовах обмеженого кола відповідальних осіб.

Користувач має наступні можливості:

- авторизуватися;
- вести перелік кандидатів;
- вести перелік вакансій;
- провести розподіл кандидатів на вакансії, засновуючись на їхній оцінці за кожною заявкою;
- переглядати результати розподілу;

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			25

– переглядати ефективність розподілу.

Діаграма варіантів використання представлена на кресленику IC12.100БАК.005 Д1.

5.3 Модель бази даних

У базі даних зберігаються відомості про вакансії, кандидатів, їхні подані заявки та результати вибору кандидатів алгоритмом. Крім того, структура бази передбачає збереження технічних даних, необхідних для функціонування алгоритму розподілу. Окремі таблиці призначені для забезпечення авторизації користувачів.

Модель бази даних представлена на кресленику IC12.100БАК.005 Д2.

У таблиці 5.1 висвітлено призначення таблиць бази даних.

Таблиця 5.1 – Опис таблиць бази даних

Таблиця	Призначення
Candidates	Збереження даних про кандидатів
Vacancies	Збереження даних про вакансії
Applications	Збереження заявок, поданих кандидатами. Ця таблиця містить також інформацію про те, чи було кандидата обрано на вакансію алгоритмом, та чи залишається вакансія, вказана в заявці, ймовірною для кандидата у процесі розподілу
AspNetUsers	Зберігає інформацію про користувачів системи: імена користувачів, електронну адресу, паролі
AspNetRoles	Зберігає інформацію про ролі користувачів. Згенерована автоматично та не використовується на даному етапі розробки продукту. Може бути використана при масштабуванні системи.
AspNetUserRoles	Таблиця, яка виконує функцію проміжної у

Зм.	Лист	№ докум.	Підпис	

Таблиця	Призначення
	зв'язку багато-до-багатьох між користувачами та ролями. Також наразі не використовується
AspNetRoleClaims	Зберігає інформацію для ролей, такі як привілеї або дозволи. Також наразі не використовується.
AspNetUserClaims	Зберігає інформацію для користувачів, такі як привілеї або дозволи.
AspNetUserLogins	Зберігає інформацію про сторонні підключені провайдери аутентифікації. Google, Facebook тощо.
AspNetUserTokens	Зберігає інформацію про токени для користувачів, які використовуються для аутентифікації або авторизації.
__EFMigrationsHistory	Зберігає інформацію про зроблені міграції, в базі даних, для підтримки версій та змін в схемі бази даних.

У таблиці 5.2 висвітлено опис полів бази даних.

Таблиця 5.2 – Опис полів бази даних

Таблиця	Поле	Опис
Candidates	Id	Первинний ключ
	FirstName	Ім'я
	LastName	Прізвище
	Email	Електронна адреса
	PhoneNumber	Контактний номер
Vacancies	Id	Первинний ключ
	Title	Назва
	Priority	Пріоритет

Таблиця	Поле	Опис
	Quota	Кількість необхідних кандидатів
	Region	Регіон несення служби
Applications	Id	Первинний ключ
	CandidateId	Зовнішній ключ таблиці Candidates
	VacancyId	Зовнішній ключ таблиці Vacancies
	Score	Оцінка кандидата за вказаною вакансією
	IsChosenByAlgorithm	Чи був кандидат обраний за цією заявкою
	WasFullyCheckedByAlgorithm	Технічне поле – чи було вже повністю розглянуто заявку (чи вона буде ще розглядатися для заміни)

5.4 Передавання та обробка даних

Взаємодія між клієнтською частиною інформаційної системи та її серверною логікою організована за допомогою Web API, яке побудовано згідно з принципами REST (REpresentational State Transfer). Це означає, що всі запити до сервера є незалежними, базуються на єдиному інтерфейсі та використовують стандартні методи протоколу HTTP (зокрема GET, POST, PUT, DELETE) для виконання операцій над ресурсами системи. У даному контексті ресурсами виступають об'єкти доменної області — кандидати, вакансії, заявки кандидатів.

Завдяки REST-архітектурі API забезпечується чітка структурованість запитів, прозорість взаємодії та легкість інтеграції клієнтського застосунку з серверною частиною. Формат обміну даними між клієнтом та сервером у вигляді

					IC12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			28

JSON дозволяє швидко обробляти запити як зі сторони Angular-застосунку, так і з боку сторонніх сервісів у разі потреби масштабування чи інтеграції.

Розглянемо основний бізнес-процес системи – розподіл кандидатів на вакансії – з огляду на передавання даних між компонентами:

- користувач викликає розподіл кліком на відповідну кнопку на фронтенді;
- Angular-компонент `vacancy-list.component` опрацьовує клік та викликає сервіс `vacancy.service`, передаючи йому запит на проведення розподілу;
- `vacancy.service` надсилає запит на бекенд, де цей запит обробляється контролером `VacancyController`;
- `VacancyController` передає запит до сервісу бізнес-логіки `VacancyService` для проведення розподілу;
- `VacancyService` виконує необхідні операції з базою даних, та опрацьовує дані відповідно до алгоритму;
- коли розподіл проведено, `VacancyController` видає результати, які йдуть зворотним шляхом до компонента `vacancy-list.component`, і користувачу відображаються результати розподілу.

Процес розподілу кандидатів на вакансії відображено на діаграмі послідовності, яка наведена у кресленику IC01.010БАК.005 ДЗ.

5.5 Архітектура програмного забезпечення

Як було згадано раніше, розроблена інформаційна система реалізована на основі тришарової архітектури, яка забезпечує логічне розділення відповідальностей між компонентами: рівнем доступу до даних (`Data Access Layer`), рівнем бізнес-логіки (`Business Logic Layer`) та рівнем представлення (`Presentation Layer`). Такий підхід сприяє зручному супроводженню, масштабуванню та повторному використанню компонентів системи. Загальна діаграма класів системи, у якій чітко відображено трирівневу архітектуру системи, наведена в кресленику IC12.100БАК.005 Д4.

У таблицях 5.3-5.6 висвітлено методи основних контролерів, які реалізовані

					IC12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			29

у проєкті WebAPI. Саме вони виступають точками входу для користувачів та забезпечують передачу запитів до бізнес-рівня та повернення відповідей. Для стислості у таблицях будуть наведені лише вхідні параметри методів, оскільки вихідні значення або очевидні з контексту виклику, або не мають суттєвого впливу на архітектурну логіку системи.

Таблиця 5.3 – Опис методів класу AuthController

Метод класу	Ендпоінт	HTTP метод	Призначення	Вхідні параметри
Login	api/auth/login	Post	Авторизація користувача	LoginModel model – авторизаційні дані користувача
GenerateJwtToken	-	-	Генерація JWT токена	ApplicationUser user – дані про користувача

Таблиця 5.4 – Опис методів класу VacanciesController

Метод класу	Ендпоінт	HTTP метод	Призначення	Вхідні параметри
GetAllVacancies	api/Vacancies	Get	Отримання списку вакансій	-
GetVacancyById	api/Vacancies/{id}	Get	Отримання однієї вакансії за її ідентифікатором	Guid id
CreateVacancy	api/Vacancies	Post	Створення нової вакансії	VacancyCreateDto vacancyDto
UpdateVacancy	api/Vacancies/{id}	Put	Оновлення	Guid id,

Зм.	Лист	№ докум.	Підпис	

Метод класу	Ендпоінт	HTTP метод	Призначення	Вхідні параметри
су	id}		існуючої вакансії	VacancyCreateDto vacancyDto
DeleteVacancy	api/Vacancies/{id}	Delete	Видалення вакансії	Guid id

Таблиця 5.5 – Опис методів класу CandidatesController

Метод класу	Ендпоінт	HTTP метод	Призначення	Вхідні параметри
GetAllCandidates	api/Candidates	Get	Повернення списку кандидатів	-
GetCandidateById	api/Candidates/{id}	Get	Отримання одного кандидата за його ідентифікатором	Guid id
CreateCandidate	api/Candidates	Post	Створення нового кандидата	CandidateCreateDto candidateDto
UpdateCandidate	api/Candidates/{id}	Put	Оновлення існуючої вакансії	Guid id, CandidateCreateDto candidateDto
DeleteCandidate	api/Candidates/{id}	Delete	Видалення кандидата	Guid id

Таблиця 5.6 – Опис методів класу ApplicationsController

Метод класу	Ендпоінт	HTTP метод	Призначення	Вхідні параметри
GetAllApplications	api/Applications	Get	Повернення списку всіх заявок	-
GetApplicationById	api/Applications/{id}	Get	Отримання однієї заявки за її ідентифікатором	Guid id
GetApplicationsByCandidateId	api/Applications/candidate/{candidateId}	Get	Отримання всіх заявок кандидата за його ідентифікатором	Guid candidateId
GetApplicationsByVacancyId	api/Applications/vacancy/{vacancyId}	Get	Отримання всіх заявок вакансії за її ідентифікатором	Guid vacancyId
GetApplicationByCandidateIdAndVacancyId	api/Applications/candidate/{candidateId}/vacancy/{vacancyId}	Get	Отримання однієї заявки за ідентифікатором кандидата та вакансії	Guid candidateId, Guid vacancyId
RunAssignmentAlgorithm	api/Applications/run-algorithm	Post	Проведення розподілу кандидатів на вакансії	-
ResetAssignments	api/Applications/reset-assignments	Post	Скасування розподілу кандидатів на	-

Зм.	Лист	№ докум.	Підпис	

Метод класу	Ендпоінт	HTTP метод	Призначення	Вхідні параметри
			вакансії	
GetAssignmentsEfficiencyScore	api/Application/s/efficiency	Get	Отримання ступеня ефективності призначення кандидатів	-
GetCandidateSatisfactionRate	api/Application/s/satisfaction	Get	Отримання ступеня задоволення кандидатів	-

Висновок до розділу 5

У результаті реалізації інформаційної системи розподілу кандидатів на вакансії було впроваджено тришарову архітектуру програмного забезпечення, яка забезпечила чітке логічне розмежування функцій між рівнями доступу до даних, бізнес-логіки та представлення. Така структура є перевіреним підходом у створенні масштабованих та підтримуваних рішень, особливо в умовах складних прикладних задач, що включають взаємодію між численними сутностями, алгоритмами та користувацькими інтерфейсами.

Рівень доступу до даних було реалізовано за допомогою Entity Framework Core, що дало змогу зручно й безпечно працювати з реляційною базою даних, використовуючи концепції об'єктно-реляційного відображення. Це суттєво пришвидшило розробку та зменшило кількість помилок під час міграції даних.

Бізнес-рівень системи акумулює основну логіку функціонування — від обробки вхідних даних користувача до виклику алгоритмів розподілу кандидатів на вакансії. Така централізація логіки забезпечує цілісність всієї системи та дає змогу легко масштабувати її в майбутньому.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			33

Рівень представлення розроблено на основі Angular із використанням бібліотеки компонентів Material Design, що дозволяє досягти сучасного й адаптивного вигляду інтерфейсу. Користувацький досвід забезпечено за рахунок логічно побудованої взаємодії та чіткого маршрутизованого переходу між функціональними модулями. Важливою складовою є реалізація взаємодії між клієнтською та серверною частинами через RESTful API, створене за допомогою ASP.NET Core. Така модель комунікації відповідає сучасним стандартам веб-розробки, дозволяє підтримувати гнучкий обмін даними у форматі JSON та відкриває можливості для подальшої інтеграції з іншими системами.

У межах розробленої системи було також реалізовано функціональну модель із підтримкою ключових операцій адміністратора: реєстрація та авторизація, створення та редагування кандидатів і вакансій, запуск алгоритму розподілу, перегляд результатів та оцінка ефективності. Для забезпечення надійності обчислень передбачено інтерфейси для тестування на згенерованих даних.

Таким чином, обраний підхід забезпечує технічну надійність, легкість подальшої підтримки та готовність системи до впровадження в реальне середовище військового рекрутингу.

					ІС12.100БАК.005 ПЗ	Арк.
						34
Зм.	Лист	№ докум.	Підпис			

6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

6.1 Змістовна постановка задачі

З огляду на задані умови, оптимізаційна задача сформульована наступним чином.

Існує m мобілізованих кандидатів K_i , $i = (\overline{1, m})$, які пройшли навчання.

Військовий підрозділ має n актуальних вакансій V_j , $j = (\overline{1, n})$ з квотою v_1, v_2, \dots, v_n відповідно, а їх пріоритетність становить p_1, p_2, \dots, p_n умовних одиниць.

Ефективність призначення кандидата K_i на вакансію V_j становить f_{ij} .

Необхідно скласти такий розподіл кандидатів на вакансії, при якому закриватиметься якнайбільше термінових вакансій і кандидатам з найбільш релевантним набором навичок для кожної окремої вакансії буде надаватись перевага.

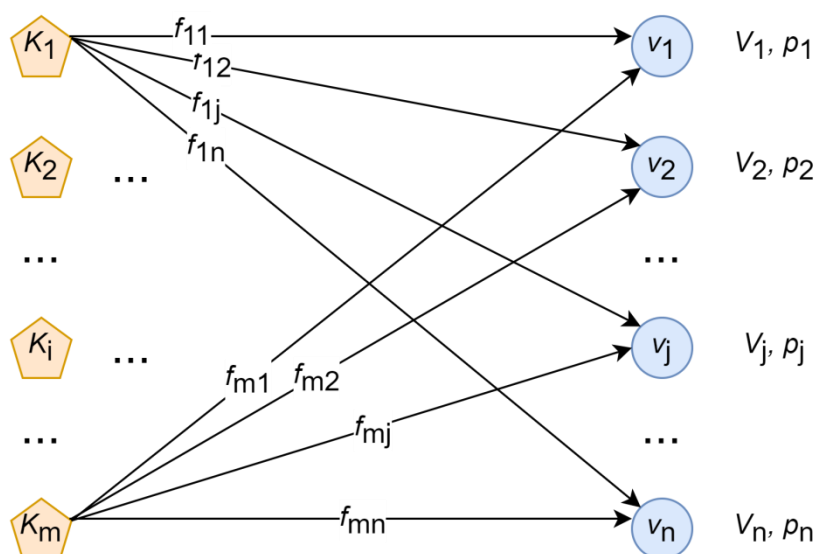


Рисунок 6.1 – Мережева модель задачі розподілу

6.2 Математична постановка задачі

Сформулюємо математичну модель задачі. Спочатку наведемо вхідні дані.

m – кількість кандидатів.

Зм.	Лист	№ докум.	Підпис	

n – кількість вакансій.

$v = (v_1, v_2, \dots, v_n)$ – необхідна кількість кандидатів для вакансій (v_j – необхідна кількість кандидатів для вакансії $V_j, j = (\overline{1, n})$).

$p = (p_1, p_2, \dots, p_n)$ – пріоритетність вакансій (p_j – пріоритетність вакансії $V_j, j = (\overline{1, n})$).

$f = \{f_{ij}\}, i = (\overline{1, m}), j = (\overline{1, n})$ – матриця ефективності призначення кандидата K_i на вакансію V_j .

$f' = \{p_j f_{ij}\}, i = (\overline{1, m}), j = (\overline{1, n})$ – матриця ефективності призначення f_{ij} з урахуванням пріоритетності вакансії p_j .

Змінні. Позначимо через булеву змінну $x_{ij}, i = (\overline{1, m}), j = (\overline{1, n})$ – фактичний розподіл кандидата K_i на вакансію V_j . Якщо кандидата i розподілено на вакансію $j, x_{ij} = 1$, інакше $x_{ij} = 0$.

Обмеження. Вищезазначений розподіл $x = \{x_{ij}\}$ повинен задовольняти умовам:

$$\sum_{i=1}^m x_{ij} \leq v_j, j = (\overline{1, n}), \quad (6.1)$$

$$\sum_{j=1}^n x_{ij} \leq 1, i = (\overline{1, m}), \quad (6.2)$$

$$x_{ij} \in \{0; 1\}, i = (\overline{1, m}), j = (\overline{1, n}). \quad (6.3)$$

Обмеження (6.1) гарантує, що на кожну вакансію може бути розподілено не більше необхідної кількості кандидатів. Обмеження (6.2) гарантує, що кожен кандидат буде розподілений не більше ніж на одну вакансію. Обмеження (6.3) гарантує існування булевих значень розподілу.

Цільова функція. Оскільки більша пристосованість кандидатів до вакансії f_{ij} сприяє більш ефективній роботі, знайдемо її як суму ефективностей призначених кандидатів

$$\sum_{j=1}^n \sum_{i=1}^m f_{ij} x_{ij}. \quad (6.4)$$

Також необхідно врахувати пріоритетність вакансії p_j (чим більший її пріоритет – тим важливіше розподілити туди кандидата):

$$\sum_{j=1}^n p_j \sum_{i=1}^m f_{ij} x_{ij}. \quad (6.5)$$

Для спрощення подання формули, використаємо величини $f'_{ij} = \{p_j f_{ij}\}$.

Таким чином, цільова функція розподілу кандидатів на вакансії набуває вигляду:

$$z = \sum_{j=1}^n \sum_{i=1}^m f'_{ij} x_{ij} \rightarrow \max. \quad (6.6)$$

Така задача називається задачею про призначення [10]. Вона належить до класу задач дискретної оптимізації [11].

Для дослідження наближення отриманих розв'язків до ідеальних з точки зору врахування інтересів кандидатів, уведемо другу цільову функцію — критерій врахування інтересів кандидата.

При обчисленні цього критерію не будемо враховувати терміновість вакансій, оскільки кандидат не керується цим при виборі майбутнього місця роботи.

Обчислимо врахування інтересів кандидата як частку від ділення суми ефективності призначення на вакансії, на які розподілили кандидатів:

$$\sum_{j=1}^n \sum_{i=1}^m f_{ij} x_{ij} \quad (6.7)$$

на суму найбільших ефективностей призначення для кожного кандидата

$$\sum_{i=1}^m \max_j f_{ij}. \quad (6.8)$$

Таким чином, друга цільова функція, яка враховує інтереси кандидатів набуває вигляду:

$$y = \frac{\sum_{j=1}^n \sum_{i=1}^m f_{ij} x_{ij}}{\sum_{i=1}^m \max_j f_{ij}} \rightarrow \max. \quad (6.9)$$

6.3 Обґрунтування методу розв'язання

Розподіл кандидатів на вакансії у військовому підрозділі належить до класу задач дискретної комбінаторної оптимізації та є розширенням класичної задачі на призначення.

Алгоритм, що не враховує індивідуальні характеристики кандидатів або пріоритетність вакансій, зводиться до жорсткого жадібного підходу: призначення кандидатів відбувається у порядку надходження або згідно з найпростішим критерієм сумісності. Такий підхід, хоч і простий у реалізації, часто не гарантує

оптимального використання наявних людських ресурсів, особливо в умовах обмежених квот і різнорівневих потреб.

Алгоритм, що враховує як ефективність кандидатів для конкретних вакансій, так і пріоритетність самих вакансій, реалізовано у модифікованому варіанті алгоритму Мака [10]. Він є представником ітеративних корегувальних алгоритмів, що поступово вдосконалюють попередній розв'язок, наближаючи його до допустимого та оптимального.

Ідея алгоритму полягає у побудові початкового розподілу на основі локально оптимальних призначень, після чого на кожній ітерації виконується аналіз порушених квот чи неефективних призначень, та виконується локальна заміна — переназначення кандидатів з мінімальними втратами ефективності, доти, доки не буде задоволено всі обмеження.

6.4 Опис методів розв'язання

Задача розподілу кандидатів на вакансії належить до класу задач дискретної оптимізації.

Без урахування пріоритетності вакансій, вона є типовою задачею на призначення. Задача на призначення є однією з класичних задач комбінаторної оптимізації, яка полягає в призначенні об'єктів до певних місць з метою оптимізації деякого критерію.

Для розв'язання задачі про призначення можуть бути застосовані різні методи. Від загальних методів розв'язання задач лінійного програмування до спеціальних методів розв'язування задач на графах. Як правило, спеціальні методи, що розроблені саме для цієї задачі, є значно швидшими оскільки враховують і використовують особливості структури задачі. Так, наприклад, Угорський алгоритм [12] є одним із перших алгоритмів, який був розроблений для розв'язання лінійної задачі про призначення. Іншими алгоритмами, що застосовуються для розв'язання задачі є адаптований симплекс метод [13], метод гілок та меж [14],

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			39

алгоритм аукціону [15] і менш відомий алгоритм Мака, який використовує алгоритм аукціону для вибору наступного значення.

Після аналізу вищезазначених алгоритмів для реалізації було обрано метод Мака. Врахування пріоритетів вакансій трансформує початкову задачу у розширену форму задачі на призначення, тому ефективним рішенням є адаптація обраного методу з урахуванням додаткових умов.

6.4.1 Модифікований алгоритм Мака

Метод Мака належить до класу ітеративних корегувальних алгоритмів. Метод спочатку знаходить оптимальний розв'язок, який може бути недопустимим (наприклад, не задовольняє всі обмеження). На кожному наступному кроці алгоритм корегує цей розв'язок, щоб зробити його більш допустимим, поки не буде досягнуто допустимого оптимального розв'язку.

Звичайний алгоритм Мака працює зі стовпчиками, але для зменшення обчислювальної складності алгоритму, будемо працювати з рядками вхідної матриці.

Також оригінальний алгоритм Мака передбачає, що на кожную вакансію буде призначено тільки 1 кандидата. Утім так, як за умовою задачі, вакансії можуть мати квоту більше 1 кандидата, то модифікуємо алгоритм під нашу задачу.

Псевдокод алгоритму, який реалізує вищенаведене, представлений нижче.

1 Вхід:

f – матриця ефективності призначення кандидатів на вакансії розмірності $n \times m$ (n – кількість вакансій, m – кількість кандидатів)

v – список квот вакансій

p – список пріоритетностей вакансій

2 Вихід:

f_{rez} – матриця призначень розмірності $n \times m$

z – рекордне значення ЦФ

y – рекордне значення відсотка врахування інтересів

					ІС12.100БАК.005 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис			

```

3  Допоміжні змінні:
   A – матриця, яка розглядається
   A' – матриця, яка не розглядається
   selected – список вибраних на вакансію кандидатів
   selected_A – список вибраних на вакансію кандидатів матриці A
   selected_A' – список вибраних на вакансію кандидатів матриці A'
   pretended – список кандидатів на максимальне значення
   difference – список різниць між поточним та потенційним найбільшим
   значенням
   is_ok – булева змінна допустимості розв'язку

4  is_ok = False
5  # Максимальні для кандидатів
6  for each кандидат j in матриця f:
7      for each вакансія i in кандидат:
8          selected[i][j] += max(f[i][j])
9          # Перевірка допустимості
10         if len(selected[i][j]) <= 1:
11             is_ok = True
12         endif
13     endfor
14 # Основний цикл
15 if not is_ok:
16     # Створення A та A'
17     A += кандидат f[j][i] у якого max(len(selected))
18     A' -= кандидат f[j][i] у якого max(len(selected))
19     # Максимальні елементи A
20     for each вакансія i in матриця A:
21         for each кандидат j in вакансія:
22             if A[i][j] is in selected:
23                 selected_A[j] += max(A[i][j])

```

```

24     max(A[i][j]) = 0
25     endif
26 endfor
27 # Максимальні елементи A'
28 for each кандидат j in матриця A':
29     for each вакансія i in кандидат:
30         selected_A'[j] = selected[i][j] in A'
31         # Немає претендента
32         if count(A' > 0 in кандидат який in selected_A) is 0:
33             # Скасування призначення
34             min(selected[j][i]) = 0
35         endif
36         # Є претендент
37     else:
38         pretended[i] += max(A'[i][j])
39         max(A'[i][j]) = 0
40         # Різниця selected - pretended
41         difference[i] = selected_A[i] - pretended[i]
42     endelse
43 endfor
44 # Оновлення A
45 for each вакансія i in матриця A:
46     for each кандидат j in вакансія:
47         # Немає претендента
48         if count(A' > 0 in кандидат який in selected_A) is 0:
49             break
50         endif
51         A[i][j] -= min(difference)
52     endfor

```

```

53  # Оновлення selected
54  for each кандидат  $j$  in матриця  $f$ :
55    for each вакансія  $i$  in кандидат:
56      # Немає претендента
57      if count( $A' > 0$  in кандидат який in  $selected\_A$ ) is 0:
58        break
59      endif
60      # Виконується квота
61      if len( $selected\_A'[i]$ ) < квота  $v[i]$ :
62         $selected[j][i] = selected[j][i] - selected\_A[\min(difference)]$ 
63      endif
64    endfor
65    # Перевірка допустимості
66    for each вакансія  $i$  in матриця  $A$ :
67      for each кандидат  $j$  in вакансія:
68        if len( $selected[i][j]$ ) <= 1:
69           $is\_ok = True$ 
70        endif
71      endfor
72    endif
73    # Визначення рекорду
74    for each вакансія  $i$  in матриця  $f$ :
75      for each кандидат  $j$  in вакансія:
76         $f\_rez[i][j] += selected[i][j]$ 
77         $z = \text{sum}(selected[i][j] \cdot p[i])$ 
78         $y = \text{sum}(selected[i][j] / \max(f[i]))$ 
79    endfor

```

Визначимо теоретичну оцінку трудомісткості модифікованого алгоритму Мака. Кожен крок основного циклу включає вкладені цикли по кандидатах та вакансіях з теоретичною складністю $O(n \times t)$. Основний цикл може виконуватися в найгіршому випадку $O(n \times t)$ разів, щоб забезпечити допустимий розв'язок.

Таким чином, загальна теоретична складність модифікованого алгоритму Мака в найгіршому випадку: $O(n \times t) \times O(n \times t) = O(n^2 \times t^2)$.

Знайдемо розв'язок модифікованим алгоритмом Мака.

Крок 1. Визначимо максимальні елементи для кожного кандидата. Матриця A пуста. До матриці A' входять вакансії Медик, IT-спеціаліст, Штурмовик. Перевіримо, чи кожна квота виконується.

Так, як для вакансії Медика не виконується квота, занесемо цей рядок до матриці A .

Для кандидатів $k1, k2, k3$ у матриці A' немає вибраних елементів. Знайдемо максимальні елементи матриці A' і запишемо їх у список на заміну, як показано на рисунку 6.2. Обчислимо різницю максимальних елементів двох матриць A та A' . Найменша різниця і буде нашою потенційною заміною.

Квота	Пріоритетність	Кандидат \ Вакансії	k1	k2	k3	k4	k5	
2	0,5	Медик	0,96	0,91	0,8	0	0,97	A
1	0,3	IT-спеціаліст	0	0,7	0,51	0,72	0	A'
2	0,2	Штурмовик	0,5	0	0,6	0,6	0,99	

0.96	0.91	0.8
-0.5	-0.7	-0.6
0.46	0.21	0.2

Рисунок 6.2 – Модифікований Алгоритм Мака. Крок 1

Крок 2. Від елементів матриці A віднімемо найменшу різницю максимальних елементів двох матриць A та A' , як наведено на рисунку 6.3. Перевіримо квоти.

Так, як кожен кандидат призначений лише на одну вакансію і виконується квота вакансій – отримано допустимий розв’язок.

Квота	Пріоритетність	Кандидат\ Вакансії	k1	k2	k3	k4	k5	
2	0,5	Медик	0,76	0,71	0,6	0	0,77	A - 0.2
1	0,3	ІТ-спеціаліст	0	0,7	0,51	0,72	0	A'
2	0,2	Штурмовик	0,5	0	0,6	0,6	0,99	

Рисунок 6.3 – Модифікований Алгоритм Мака. Крок 2

Для розрахунку значення цільової функції використовуються вихідні значення відповідно до початкового варіанту, але зі збереженням призначених кандидатів.

Отриманий розв’язок наведено у таблиці 6.1.

Таблиця 6.1 – Отриманий за допомогою модифікованого алгоритму Мака розв’язок

Квота	Пріоритетність	Кандидат\ Вакансії	k1	k2	k3	k4	k5
2	0.5	Медик	0.96	0.91	0.8	0	0.97
1	0.3	ІТ-спеціаліст	0	0.7	0.51	0.72	0
2	0.2	Штурмовик	0.5	0	0.6	0.6	0.99

Розрахуємо значення цільової функції для розв’язку, що отриманий за модифікованим алгоритмом Мака:

$$0,5*(0,96 + 0,91) + 0,3*0,72 + 0,2*(0,6 + 0,99) = 1,469 \text{ одиниці ефективності.}$$

Інтереси кандидатів для розв’язку за модифікованим алгоритмом Мака враховано на:

$$(0,96 + 0,91 + 0,6 + 0,72 + 0,99) / (0,96 + 0,91 + 0,8 + 0,72 + 0,99) * 100\% = 95,434\%$$

Висновок до розділу 6

У цьому розділі було сформульовано формальну математичну модель задачі розподілу кандидатів на вакансії у військовому підрозділі, яка є варіантом задачі на призначення з урахуванням квот і пріоритетності вакансій. Модель враховує ефективність відповідності кандидатів певним вакансіям, а також дозволяє оптимізувати розподіл з огляду на потреби підрозділу та інтереси кандидатів.

Особливістю побудованої математичної моделі є наявність другої цільової функції, яка враховує інтереси кандидатів, а саме відповідність їх вимогам обраної посади.

На основі проведеного аналізу різних методів комбінаторної оптимізації для розв'язання поставленої задачі було обрано алгоритм Мака, що в подальшому було адаптовано до умов розподілу з урахуванням квот. Реалізація алгоритму дозволяє отримувати ефективні розподіли із достатньо високим рівнем відповідності інтересів кандидатів (понад 95 %) та зваженим урахуванням пріоритетності вакансій.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			46

7 ТЕСТУВАННЯ СИСТЕМИ

Тестування програмного забезпечення — це важливий етап життєвого циклу розробки, спрямований на забезпечення якості та надійності програмного продукту. Його мета полягає у виявленні потенційних помилок, логічних дефектів, відхилень від очікуваної поведінки та недоліків, які можуть негативно вплинути на функціональність системи під час реального використання. У процесі тестування програму запускають із різноманітними наборами вхідних даних, перевіряють її поведінку в типових і граничних умовах, а також аналізують результати виконання, щоб переконатися у відповідності фактичної роботи системи заданим технічним вимогам. Такий підхід дозволяє мінімізувати ризики, підвищити стабільність і надійність розробленого програмного забезпечення перед його впровадженням в експлуатацію.

7.1 Мета випробувань

Основною метою тестування є гарантування високої якості програмного продукту шляхом перевірки його відповідності визначеним функціональним і нефункціональним вимогам [16]. Це також включає виявлення і усунення дефектів на етапі розробки, щоб мінімізувати ймовірність помилок, які можуть бути помічені кінцевими користувачами під час реального використання системи. В контексті розробленої інформаційної системи, головною функціональною вимогою є точний та ефективний розподіл кандидатів на відповідні вакансії.

Для забезпечення надійності цього процесу необхідно чіткий набір тестових сценаріїв, що охоплюють ключові аспекти роботи системи. Після цього ці сценаріїв мають бути реалізовані та виконані з метою перевірки реакції системи на різні умови та вхідні дані. Такий підхід дозволяє переконатися в коректності реалізації логіки, стабільності роботи та здатності системи до адекватного функціонування у межах заданих вимог.

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			47

7.2 Загальні положення

Під час тестування розробленої інформаційної системи пріоритет буде надано перевірці функціональності, що відповідає її ключовому призначенню. Основний акцент зроблено на функціональному тестуванні, яке полягає у ретельному аналізі роботи основних компонентів системи та виявленні можливих невідповідностей між фактичною поведінкою реалізованих функцій і очікуваними результатами, сформульованими у вимогах до системи та технічній специфікації. Такий тип тестування орієнтований на імітацію реальних сценаріїв використання, наближених до поведінки кінцевого користувача, що дозволяє виявити критичні дефекти на рівні бізнес-логіки.

Додатково буде застосовано E2E-тестування (англ. end-to-end testing) — підхід, що охоплює перевірку роботи всієї системи в комплексі. Це охоплює перевірку інтерфейсу користувача, роботу API, а також взаємодію з сервером і базою даних. Мета цього виду тестування — переконатися, що всі елементи системи коректно працюють разом і забезпечують повноцінне функціонування з точки зору користувача.

Враховуючи, що поточна версія програмного забезпечення є мінімально життєздатним продуктом (англ. MVP — minimal valuable product), а ресурси на автоматизоване тестування обмежені, всі передбачені тестові сценарії будуть реалізовані вручну. Такий підхід дозволить на ранньому етапі виявити найбільш критичні помилки і забезпечити базову стабільність системи до подальшої розробки.

7.3 Результати випробувань

У рамках тестування програмного забезпечення було здійснено повну перевірку основних функціональних можливостей системи. Були розроблені сценарії взаємодії [17] із системою, що охоплюють ключові операції, зокрема створення та редагування записів, запуск алгоритму розподілу, перегляд

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			48

результатів, скасування розподілу та інші дії згідно з функціональними вимогами.

Усі тестові випадки виконано успішно — система стабільно реагувала на введення даних, коректно обробляла запити та повертала очікувані результати. Це свідчить про правильну реалізацію основної бізнес-логіки, належну інтеграцію між клієнтською частиною та серверною, а також ефективну взаємодію з базою даних.

Підсумкові результати проведених випробувань структуровано та наведено у таблицях 7.1–7.10, що містять опис кожного тестового сценарію, вхідні дані, очікувану та фактичну поведінку системи.

Таблиця 7.1 – Тестування успішної авторизації у системі

Ідентифікатор	001
Назва	Успішна авторизація у системі
Опис	Перевірка можливості увійти в акаунт
Передумови	<ul style="list-style-type: none">– користувач знаходиться на сторінці авторизації;– акаунт для користувача було створено.
Вхідні дані	Авторизаційні дані: ім'я користувача (admin), пароль (admin)
Кроки виконання	<ul style="list-style-type: none">– ввести дані у відповідні поля;– натиснути кнопку «Увійти».
Очікуваний результат	<ul style="list-style-type: none">– відкрилася сторінка усіх вакансій;– відображено перелік усіх вакансій;– на кожній вакансії відображено кнопки: деталі вакансії, заявки за вакансією, редагувати вакансію;– відображено кнопку для додавання вакансії;– відображено кнопку для проведення розподілу.– відображено кнопку перегляду всіх кандидатів.
Статус	Успішний

Зм.	Лист	№ докум.	Підпис	

Таблиця 7.2 – Тестування неуспішної авторизації у системі

Ідентифікатор	002
Назва	Неуспішна авторизація у системі
Опис	Перевірка неможливості увійти в акаунт з неправильними логіном та/або паролем
Передумови	Користувач знаходиться на сторінці авторизації
Вхідні дані	Авторизаційні дані (7 окремих тестових сценаріїв): <ul style="list-style-type: none"> – неіснуюче ім'я користувача, неіснуючий пароль; – неіснуюче ім'я користувача, існуючий пароль (admin); – існуюче ім'я користувача (admin), неправильний пароль; – існуюче ім'я користувача (admin), порожній пароль; – порожнє ім'я користувача, неіснуючий пароль; – порожнє ім'я користувача, існуючий пароль (admin); – порожнє ім'я користувача, порожній пароль.
Кроки виконання	<ul style="list-style-type: none"> – ввести дані у відповідні поля; – натиснути кнопку «Увійти».
Очікуваний результат	<ul style="list-style-type: none"> – користувач залишився на сторінці авторизації – відображено помилку «Введено некоректні дані авторизації».
Статус	Успішний

Таблиця 7.3 – Тестування створення нової вакансії

Ідентифікатор	003
Назва	Створення вакансії

Опис	Перевірка коректності збереження та відображення створеної вакансії	
Передумови	<ul style="list-style-type: none"> – користувач авторизований та знаходиться на сторінці усіх вакансій; – розподіл ще не було проведено. 	
Вхідні дані	<p>Авторизаційні дані: ім'я користувача (admin), пароль (admin).</p> <p>Параметри вакансії: назва (test drone operator), пріоритет (1), квота (2), регіон (Херсонська область)</p>	
Кроки виконання та очікувані результати	Натиснути кнопку «Додати вакансію»	Відображено модальне вікно додавання вакансії
	Ввести параметри вакансії у відповідні поля	Нову вакансію відображено у переліку всіх дисциплін
	Натиснути кнопку «Додати вакансію»	
Статус	Успішний	

Таблиця 7.4 – Тестування відображення деталей вакансії

Ідентифікатор	004
Назва	Деталі вакансії
Опис	Перевірка коректності відображення деталей вакансії
Передумови	<ul style="list-style-type: none"> – користувач авторизований та знаходиться на сторінці усіх вакансій; – вакансії мають заявки.
Вхідні дані	Відсутні

Кроки виконання та очікувані результати	Відкрити сторінку деталей будь-якої вакансії, яка має заявки, за кліком по її назві	Відкривається сторінка деталей вакансії; Відображається інформація про вакансію; Відображаються всі заявки вакансії; Відображається кнопка додавання заявки.
Статус	Успішний	

Таблиця 7.5 – Тестування відображення списку кандидатів

Ідентифікатор	005	
Назва	Список кандидатів	
Опис	Перевірка коректності відображення списку кандидатів	
Передумови	Користувач авторизований та знаходиться на сторінці усіх вакансій	
Вхідні дані	Відсутні	
Кроки виконання та очікувані результати		Кнопка «Всі кандидати» відображається і активна
	Натиснути кнопку «Всі кандидати»	Відкривається сторінка зі списком всіх кандидатів; Відображаються всі кандидати (або порожній список, якщо кандидатів ще не було додано); Відображається кнопка додавання кандидата.

Зм.	Лист	№ докум.	Підпис	

Статус	Успішний
--------	----------

Таблиця 7.6 – Тестування створення нового кандидата

Ідентифікатор	006	
Назва	Створення кандидата	
Опис	Перевірка коректності збереження та відображення створеного кандидата	
Передумови	<ul style="list-style-type: none"> – користувач авторизований та знаходиться на сторінці усіх кандидатів; – розподіл ще не було проведено. 	
Вхідні дані	Параметри кандидата: ім'я, прізвище, email, номер телефона	
Кроки виконання та очікувані результати	Натиснути кнопку «Додати кандидата»	Відображено модальне вікно додавання кандидата
	Ввести параметри кандидата у відповідні поля	Нового кандидата відображено у переліку всіх кандидатів
	Натиснути кнопку «Додати»	
Статус	Успішний	

Таблиця 7.7 – Тестування відображення одного кандидата

Ідентифікатор	007
Назва	Профіль кандидата
Опис	Перевірка коректності відображення профілю кандидата
Передумови	– користувач авторизований та знаходиться на сторінці усіх вакансій;

	– кандидати мають заявки.	
Вхідні дані	Відсутні	
Кроки виконання та очікувані результати	Відкрити сторінку профіля будь-якого кандидата, який має заявки, за кліком по його імені	Відкривається сторінка профілю кандидата; Відображається інформація про кандидата; Відображаються всі заявки кандидата (або порожній список, якщо кандидат ще не подавав заявку); Відображається кнопка додавання заявки.
Статус	Успішний	

Таблиця 7.8 – Тестування створення нової заявки зі сторінки профілю кандидата

Ідентифікатор	008
Назва	Створення заявки: профіль кандидата
Опис	Перевірка коректності збереження та відображення заявки, створеної зі сторінки профілю кандидата
Передумови	<ul style="list-style-type: none"> – користувач авторизований та знаходиться на сторінці усіх кандидатів; – існує принаймні один кандидат; – існує принаймні одна вакансія; – розподіл ще не було проведено.
Вхідні дані	Параметри заявки: вакансія, кандидат (визначається протягом тесту), оцінка відповідності кандидата до вказаної вакансії

Зм.	Лист	№ докум.	Підпис	

Кроки виконання та очікувані результати	Вибрати будь-якого кандидата зі списку (запам'ятати ім'я) та відкрити його профіль	Відкривається сторінка профілю кандидата;
	Натиснути кнопку «Додати заявку»	Відображено модальне вікно додавання заявки; Поле «Кандидат» передзаповнене ім'ям обраного кандидата та недоступне до зміни.
	Ввести параметри заявки у відповідні поля	Нову заявку відображено у переліку всіх заявок кандидата
	Натиснути кнопку «Додати»	
	Перейти на сторінку деталей вакансії, що була обрана	Відкривається сторінка деталей вакансії; Заявка, що була створена, відображається у списку всіх заявок вакансії;
Статус	Успішний	

Таблиця 7.9 – Тестування створення заявки зі сторінки деталей вакансії

Ідентифікатор	009
Назва	Створення заявки: деталі вакансії
Опис	Перевірка коректності збереження та відображення заявки, створеної зі сторінки деталей вакансії
Передумови	– користувач авторизований та знаходиться на сторінці усіх кандидатів;

	<ul style="list-style-type: none"> – існує принаймні один кандидат; – існує принаймні одна вакансія; – розподіл ще не було проведено. 	
Вхідні дані	Параметри заявки: вакансія (визначається протягом тесту), кандидат, оцінка відповідності кандидата до вказаної вакансії	
Кроки виконання та очікувані результати	Вибрати будь-яку вакансію зі списку (запам'ятати назву) та відкрити її деталі	Відкривається сторінка деталей вакансії;
	Натиснути кнопку «Додати заявку»	Відображено модальне вікно додавання заявки; Поле «Вакансія» передзаповнене назвою обраної вакансії та недоступне до зміни.
	Ввести параметри заявки у відповідні поля	Нову заявку відображено у переліку всіх заявок вакансії
	Натиснути кнопку «Додати»	
	Перейти на сторінку профіля кандидата, що був обраний	Відкривається сторінка профіля кандидата; Заявка, що була створена, відображається у списку всіх заявок кандидата;
Статус	Успішний	

Таблиця 7.10 – Тестування розподілу кандидатів на вакансії

Ідентифікатор	010	
Назва	Розподіл кандидатів	
Опис	Перевірка можливості здійснити розподіл кандидатів і скасувати розподіл	
Передумови	<ul style="list-style-type: none"> – користувач авторизований у системі та знаходиться на сторінці усіх вакансій; – розподіл ще не було проведено. 	
Вхідні дані	Відсутні	
Кроки виконання та очікувані результати	Натиснути кнопку «Провести розподіл»	Відображено модальне вікно «Завантаження. Будь ласка, зачекайте»
		Модальне вікно неможливо зачинити
	Почекати зникнення модального вікна	Відображено значення ефективності розподілу
		Відображено кнопку «Скасувати розподіл»
		На кожній вакансії приховано кнопки для редагування
		Приховано кнопку для додавання вакансій
		Зміни збережено в таблиці Applications
Натиснути кнопку «Скасувати розподіл»	Відображено перелік усіх вакансій	

Зм.	Лист	№ докум.	Підпис	

		На кожній вакансії відображено кнопки для редагування
		Відображено кнопку для додавання дисципліни
		Відображено кнопки для проведення розподілу
		Зміни збережено в таблиці Applications
Статус	Успішний	

Висновок до розділу 7

У цьому розділі було визначено ключову мету тестування — перевірка відповідності функціональності інформаційної системи встановленим вимогам та очікуванням кінцевого користувача. З цією метою розроблено загальний підхід до побудови тестових сценаріїв, що охоплюють основні функціональні можливості системи. Особливу увагу приділено функціональному тестуванню, яке дозволяє виявити помилки у реалізації логіки, а також E2E-тестуванню, спрямованому на перевірку коректності роботи системи в цілому — від інтерфейсу до обробки даних на сервері.

Система успішно проходить всі ключові перевірки, реалізовані функції працюють відповідно до технічних та бізнес-вимог. Отже, можна зробити висновок, що розроблена інформаційна система на даному етапі забезпечує необхідний рівень якості та готова до подальшого вдосконалення або впровадження в робоче середовище.

ВИСНОВКИ

У результаті виконання дипломного проєкту було досягнуто оптимізації одного з етапів військового рекрутингу шляхом створення автоматизованої інформаційної системи, яка забезпечує розподіл кандидатів на вакансії.

Було детально проаналізовано предметну область, що включає специфіку військового рекрутингу, виявлено основні процеси діяльності та визначено можливості для автоматизації. Розроблена система забезпечує можливість роботи з вакансіями, кандидатами і здійснює розподіл з урахуванням пріоритетності посад та ефективності кандидатів.

Функціональність системи охоплює авторизацію користувачів, ведення баз кандидатів, вакансій та заявок, проведення алгоритмічного розподілу кандидатів, а також перегляд ефективності отриманих результатів. Це дозволяє централізовано та прозоро керувати процесом кадрового призначення в умовах високої динаміки військової діяльності.

Для забезпечення високої точності та ефективності розподілу було розроблено та реалізовано модифікований алгоритм Мака. Основною перевагою цього методу є його здатність враховувати багатокритеріальність задачі, квотування вакансій та пріоритетність призначень.

Використання сучасного технологічного стека, зокрема C#, ASP.NET Core, Entity Framework на серверній частині та Angular з Material Design на клієнтській, забезпечує зручність, швидкість та масштабованість розробки, дозволяючи легко розширювати функціонал і підтримувати систему в майбутньому.

Результати проведених функціональних випробувань підтвердили відповідність системи усім поставленим вимогам, що засвідчує її готовність до практичного впровадження.

Отримані результати є корисними для подальшого розвитку автоматизованих систем у сфері військового рекрутингу та можуть бути рекомендовані для широкого впровадження в діяльність Сил оборони України.

					IC12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			59

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про внесення змін до деяких законодавчих актів України щодо окремих питань проходження військової служби, мобілізації та військового обліку: Закон України від 11.04.2024 р. № 3633-IX. URL: <https://zakon.rada.gov.ua/laws/show/3633-20> (дата звернення: 25.05.2024)
2. Про затвердження Концепції військової кадрової політики в системі Міністерства оборони України на період до 2028 року. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/rada/show/v0637322-23#Text> (дата звернення: 15.05.2025)
3. Про затвердження змін до положення про військово-лікарську експертизу в збройних силах України. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/z0691-25#Text> (дата звернення: 01.05.2025)
4. Про мобілізаційну підготовку та мобілізацію. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/3543-12#Text> (дата звернення: 08.05.2025)
5. Wiegers K., Beatty J. Software requirements. 3rd ed. Redmond, WA: Microsoft Press, 2013. 640 p. (date of access: 09.05.2025)
6. C# docs - get started, tutorials, reference. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (date of access: 01.05.2024)
7. Microsoft SQL Server documentation. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/sql/sql-server/> (date of access: 03.05.2025)
8. ASP.NET Core documentation. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/aspnet/core/> (date of access: 03.05.2025)

					ІС12.100БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			60

9. Angular documentation. Angular - The modern web developer's platform. URL: <https://angular.io/docs> (date of access: 03.05.2025)

10. Burkard, M. Dell'Amico, S. Martello: Assignment Problems (Revised reprint). SIAM, Philadelphia (PA.) 2012. ISBN 978-1-61197-222-1, <https://doi.org/10.1137/1.9781611972238>, (дата звернення: 20.05.2025)

11. Конспект лекцій з дисципліни «Теорія розкладів» для студентів спеціальності “Інформаційні системи та технології” / Укл: О.Г. Жданова. К.: НТУУ «КПІ ім. Ігоря Сікорського». (дата звернення: 10.05.2025)

12. Jonker R., Volgenant T. Improving the Hungarian assignment algorithm. 1011 NH Amsterdam : University of Amsterdam. (date of access: 03.05.2025)

13. Hansen T. D., Zwick U. An Improved Version of the Random-Facet Pivoting Rule for the Simplex Algorithm. *STOC '15: Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*. P. 209–218. (date of access: 03.05.2025)

14. Narendra, Fukunaga. A Branch and Bound Algorithm for Feature Subset Selection. *IEEE Transactions on Computers*. 1977. Vol. C-26, no. 9. P. 917–922. URL: <https://doi.org/10.1109/tc.1977.1674939> (date of access: 08.05.2025)

15. J. Munkres, "Algorithms for the Assignment and Transportation Problems", *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957 (date of access: 08.03.2025)

16. Різниця між функціональним і нефункціональним тестуванням. QATestLab training centre. URL: <https://training.qatestlab.com/blog/technical-articles/difference-between-functional-and-non-functional-testing/> (дата звернення 01.06.2025)

17. Test case. ISTQB Glossary. URL: https://glossary.istqb.org/en_US/term/test-case-2 (дата звернення: 01.06.2025)

