

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

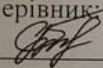
Пояснювальна записка

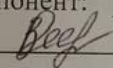
до магістерської кваліфікаційної роботи

на тему: "ІНФОРМАЦІЙНА СИСТЕМА ОБЛІКУ ТА ПІДГОТОВКИ
ДОБРОВОЛЬЦІВ ТЕРИТОРІАЛЬНОЇ ОБОРОНИ"
08-54.МКР.021.00.000 ПЗ

Виконав: магістрант групи 1КІ-23м
Спеціальності 123 – Комп'ютерна інженерія
(шифр і назва напрямку підготовки, спеціальності)

 Черній Д.В.
(прізвище та ініціали)

Керівник:  Городецька О.С.
(прізвище та ініціали)
к.т.н., доц. каф.ОТ

Опонент:  Салієва О.В.
(прізвище та ініціали)
д.ф., доц.каф. МБІС

Допущено до захисту

д.т.н., проф. Азаров О.Д.

" "

2024 р.

Вінниця 2024

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

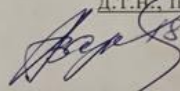
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Спеціальність 123 – «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., професор Азаров О.Д.

18.09. " " 2024 року



З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Чернія Дмитра Вікторовича

1 Тема роботи: «Інформаційна система обліку та підготовки добровольців територіальної оборони», керівник роботи Городецька Оксана Степанівна к.т.н., доцент кафедри ОТ, затверджені наказом вищого навчального закладу від "17.09.2024" 2024 року № 310.

2 Строк подання студентом роботи: 15.12.2024

3 Вихідні дані до роботи: інформаційна система з інтегрованим модулем штучного інтелекту.

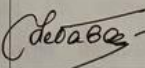
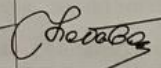
4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз сучасних методів та технологій розробки інформаційних систем обліку та підготовки добровольців територіальної оборони, розробка методу інтеграції інформаційної системи зі штучним інтелектом, розробка клієнтської та серверної частини інформаційної системи, тестування інформаційної системи з модулем штучного інтелекту, розрахунок економічної доцільності створення інформаційної системи обліку та підготовки добровольців територіальної оборони з інтеграцією штучного інтелекту.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) у магістерській роботі подано структурну схему інформаційної системи обліку та підготовки добровольців територіальної оборони, яка відображає основні її функціональні можливості. Представлено ілюстративні матеріали, що відображають процес вводу та обробки інформації за допомогою штучного

інтелекту. Окремо подано схему алгоритму маршрутизації по інформаційній системі, а також діаграму алгоритму роботи та UML-діаграму розгортання інформаційної системи обліку та підготовки добровольців територіальної оборони

6 Консультанти розділів роботи представлено в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Городецька О.С. к. т. н., доцент каф. ОТ		
5	Небава М.І., к.е.н., професор каф. ЕПВМ		

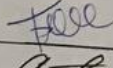
7 Дата видачі завдання:

8 Календарний план роботи зазначений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів дипломної роботи	Строк виконання	Примітка
1	Огляд сучасних методів і технологій розробки інформаційних систем	04.10.2024	Виконано
2	Аналіз технологій для розробки серверної та клієнтської частини інформаційної системи	18.10.2024	Виконано
3	Проектування та розробка API для інтеграції зі штучним інтелектом	01.11.2024	Виконано
4	Тестування та перевірка працездатності системи	12.11.2024	Виконано
5	Оформлення пояснювальної записки та підготовка звіту	25.11.2024	Виконано

Студент

 Черній Д.В.

Керівник роботи

 Городецька О.С.

АНОТАЦІЯ

УДК 004.9

Чернія Д.В. Інформаційна система обліку та підготовки добровольців територіальної оборони. Магістерська кваліфікаційна робота зі спеціальності 123 – Комп'ютерна Інженерія, Вінниця: ВНТУ, 2024 – 111с.

На укр. мові. Бібліогр.: 20 назв; рис.: 19

У магістерській роботі розглянуто сучасні методи розробки інформаційних систем, проаналізовано технології інтеграції штучного інтелекту та архітектурні рішення для створення систем обліку та підготовки добровольців.

Вдосконалено метод інтеграції інформаційної системи зі штучним інтелектом шляхом реалізації автоматизованого формування та надсилання запитів до API ChatGPT на основі користувацьких даних. Це дозволяє підвищити швидкодію, точність обробки запитів і забезпечити надання релевантних рекомендацій для підбору військових спеціальностей.

Ключові слова: інформаційна система, штучний інтелект, API, рекомендаційна система, обробка даних.

ABSTRACT

УДК 004.9

Chernii D.V. Information system for accounting and training of volunteers of territorial defense. Master's qualification work in the specialty 123 – Computer Engineering, Vinnytsia: VNTU, 2024 – 111p.

In Ukrainian. Bibliography: 20 titles; Fig.: 19

The master's work considers modern methods of developing information systems, analyzes technologies for integrating artificial intelligence and architectural solutions for creating systems for accounting and training of volunteers. The method of integrating an information system with artificial intelligence is improved by implementing automated generation and sending of requests to the ChatGPT API based on user data. This allows to increase the speed, accuracy of processing requests and ensure the provision of relevant recommendations for the selection of military specialties.

Keywords: information system, artificial intelligence, API, recommendation system, data processing.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ ТА ПІДГОТОВКИ ДОБРОВОЛЬЦІВ ТЕРИТОРІАЛЬНОЇ ОБОРОНИ	10
1.1 Аналіз психологічного стану добровольців в умовах воєнного часу	10
1.2 Огляд сучасних підходів до розробки інформаційних систем	12
1.3 Методи збору, обробки та аналізу даних в інформаційних системах.....	16
1.4 Архітектура та компоненти інформаційних систем для управління ресурсами	19
1.5 Системи для автоматизації процесів підготовки та обліку персоналу	22
1.6 Інструменти та технології для розробки інформаційних систем	25
1.7 Безпека та захист даних в інформаційних системах	27
2 МЕТОД ІНТЕГРАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗІ ШТУЧНИМ ІНТЕЛЕКТОМ	31
2.1 Аналіз можливостей штучного інтелекту для оцінки і підбору військових спеціальностей.....	33
2.2 Вдосконалення методу інтеграції штучного інтелекту з інформаційною системою	36
2.2.1 Використання методів OpenAI API	36
2.2.2 Розробка модуля для інтеграції зі штучним інтелектом	38
2.2.3 Розробка архітектури модулю інтеграції з штучним інтелектом.....	41
3 РОЗРОБКА КЛІЄНТСЬКОЇ ТА СЕРВЕРНОЇ ЧАСТИНИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТА ПІДГОТОВКИ ДОБРОВОЛЬЦІВ ТЕРИТОРІАЛЬНОЇ ОБОРОНИ	44
3.1 Реалізація клієнт–серверної архітектури інформаційної системи	48
3.2 Реалізація API для взаємодії з ChatGPT.....	50
3.3 Створення системи авторизації та обробки даних користувачів	53

					08–54.МКР.021.00.000 ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розробив	Черній Д.В.				Інформаційна система обліку та підготовки добровольців територіальної оборони. Пояснювальна записка	Літ.	Аркуш	Аркушів	
Керівник	Городецька О.С.						6	111	
Опонент	Салієва О.В.					ВНТУ, гр. 1КІ-23м			
Н.контр.	Швець С.І.								
Затвердж.	Азаров О.О.								

3.4 Підключення та налаштування API для обробки даних користувачів	55
4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ З МОДУЛЕМ ШТУЧНОГО ІНТЕЛЕКТУ	59
4.1 Перевірка працездатності системи	59
4.2 Огляд аналогів розробленої системи.....	64
5 ЕКОНОМІЧНА ЧАСТИНА.....	70
5.1 Комерційний та технологічний аудит науково-технічної розробки.....	70
5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	73
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	78
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	87
ДОДАТОК А Технічне завдання	89
ДОДАТОК Б Діаграма алгоритму роботи інформаційної системи.....	94
ДОДАТОК В Алгоритм маршрутизації по інформаційній системі.....	95
ДОДАТОК Г Лістинг питань для перевірки психологічної готовності добровольця	96
ДОДАТОК Д UML-діаграма розгортання інформаційної системи	99
ДОДАТОК Е Лістинг файлу Basephp	100
ДОДАТОК Ж Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	111

					08–54.МКР.021.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Сучасні виклики, пов'язані з військовою агресією та гібридними загрозами, потребують створення ефективної системи організації територіальної оборони, яка забезпечить швидку мобілізацію та підготовку громадян для захисту держави. У цьому контексті розробка інформаційної системи для обліку та підготовки добровольців територіальної оборони.

У ситуації, коли кількість добровольців постійно зростає, традиційні методи обліку стають неефективними. Одним із підходів подолання цього недоліку є впровадження передових технологій, які будуть використовувати штучний інтелект для автоматизації обробки даних і надання рекомендацій. Такі системи будуть сприяти підвищенню ефективності навчання при виборі військової спеціалізації на основі освітньої, психологічної, фізичної та професійної інформації кандидатів.

Виходячи із розглянутого, задачі подальшого розроблення та вдосконалення методів на теперішній час є **актуальними**.

Метою роботи є розширення функціональних можливостей інформаційної системи завдяки інтеграції з штучним інтелектом, що дасть можливість збільшити швидкодію обробки запитів штучним інтелектом.

Задачі дослідження:

- проаналізувати сучасні методи розробки інформаційних систем та можливості впровадження штучного інтелекту для обліку та підбору військових спеціальностей;
- вдосконалити метод інтеграції штучного інтелекту із інформаційною системою;
- розробити структуру інформаційної системи;
- виконати програмну реалізацію клієнтської та серверної частини інформаційної системи;

— реалізувати API для взаємодії з ChatGPT для введення інформації користувачем і персональної рекомендації військової професії;

— провести тестування інформаційної системи.

Об’єктом дослідження є процес інтеграції інформаційної системи з штучним інтелектом.

Предметом дослідження сучасні методи передачі інформації між серверною та клієнтською частинами інформаційної системи та методи інтеграції зі штучним інтелектом.

Новизна одержаних результатів — вдосконалено метод інтеграції інформаційної системи із штучним інтелектом, в якому, на відміну від існуючих, реалізовано побудову та надсилання запитів, створених на основі користувацької інформації, до штучного інтелекту з використанням його API, що дозволило розширити функціональні можливості та збільшити швидкість обробки запитів штучним інтелектом.

Практична цінність роботи полягає в отриманні українською армією інформаційної системи, яка ефективно управляє ресурсами, забезпечить високий рівень підготовки добровольців, сприятиме швидкій мобілізації в умовах кризи.

Апробація результатів роботи — зроблено доповіді на науково-технічній конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)».

Публікація: Інформаційна система обліку та підготовки добровольців територіальної оборони/ Д.В. Черній, О.С. Городецька, Н.В. Добровольська. Матеріали Молодь в науці: дослідження, проблеми, перспективи (МН-2025), 2024 р. [Електронний ресурс] —

<https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/22476/18595>[1]

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ ТА ПІДГОТОВКИ ДОБРОВОЛЬЦІВ ТЕРИТОРІАЛЬНОЇ ОБОРОНИ

В сучасних умовах загроз національній безпеці постає гостра необхідність розробки ефективних організаційно—управлінських інструментів у процесі реєстрації та підготовки добровольців до захисту Вітчизни[2]. Сучасні інформаційні системи мають автоматизувати процеси збору, обробки, аналізу даних про волонтерів і рекомендації на основі даних для них. З цією метою в цьому розділі узагальнено основну методологію проектування систем, існуючі підходи до штучного інтелекту та машинного навчання, їх потенційне застосування в територіальній обороні.

1.1 Аналіз психологічного стану добровольців в умовах воєнного часу

Однією з ключових детермінант мотивації добровольців до участі в оборонній війні є їх психологічний стан. Цей психологічний стан зазвичай виникає внаслідок змішення зовнішніх і внутрішніх факторів, які викликають непостійне мислення. Таким чином, психологічна мотивація військової служби добровольцями не завжди може бути прив'язана до абстрактних патріотів-ідеалістів і свого національного захисника.

У деяких випадках особисті психологічні переживання чи внутрішні кризи, бажання змінити життєву ситуацію можуть очолювати список причин для прийняття рішення стати добровольцем.

Коли людина вирішує вступати до територіальної оборони чи до будь-яких інших збройних сил, то найчастіше не за бажанням, а тому, що в ній глибоко закладені якісь психологічні фактори. Несвідомо чи підсвідомо деякі особисті кризові ситуації сприяють вирішенню піти добровольцем до збройних сил.

Як правило, особисті психологічні стани визначаються різними емоційними переживаннями, серед яких можуть бути тривалий стрес, депресія, втрата дорогої людини та особисті проблеми в повсякденному житті.

Можна розглянути випадок, коли індивід думає про вступ до військових формувань після емоційної кризи, наприклад, після розриву саме з коханою людиною. Це може виглядати як найбільш нелогічна та ірраціональна річ, яка може спасти комусь на думку, але людині, яка переживає емоційну кризу, це може здатися реакцією на надзвичайно похмурий внутрішній стан.

З іншого боку, приєднання до будь-яких військових формувань на ранній стадії може трактуватися як втеча від проблем або доведення собі та іншим своєї хоробрості та здатності впоратися.

Таким чином вступ до батальйону є способом зняти психічну напругу та уникнути реальності, хоча на практиці таке ухилення під час служби може призвести до далекосяжних психологічних проблем.

У цьому випадку мотиваційний фактор, який спонукає людину зробити цей крок, може бути внутрішнім і без зовнішнього тиску, тобто ефективність і стабільність у такому недружньому середовищі може бути важко передбачити. Досить часто люди приходять на військову службу не через свідоме бажання служити, а через бажання знайти новий сенс життя, позбутися емоційної порожнечі чи спробувати забути особистий досвід. Вони не готові до психологічного навантаження; вони можуть емоційно впасти, розвинути всілякі стресові реакції, а в деяких випадках – посттравматичний стресовий розлад (ПТСР).

Слід також зазначити, що вплив психологічних факторів не обмежується ситуативними емоційними кризами особистісного характеру. Іноді під зовнішнім тиском: у політичній атмосфері, існуючих соціальних чи економічних проблемах люди стають «безнадійними» і залучаються до засобу повторного пошуку сенсу. Такі заходи, як територіальна оборона чи участь у бойових діях, слід тлумачити як виконання нових ролей, або отримання заохочення, або відчуття відповідності.

У таких ситуаціях люди зазвичай не уявляють, наскільки складною буде служба або з якими психологічними потрясіннями вони можуть зіткнутися.

Ще один фактор, який необхідно враховувати, обговорюючи психологічний стан добровольців — це естетизований образ війни, заданий суспільними стереотипами та культурними канонами. На формування цих мотивацій суттєво впливають ЗМІ, які загалом романтизують військову службу, представляючи військових супергероями.

Все це може створити враження, що військова кар'єра — це шлях до мужності, шлях до честі та самореалізації, щось привабливе для «інтроспективної» або емоційно «спустошеної» особистості.

Таким чином, це може стати, мабуть, одним із тих спрощених уявлень про те, чому тоді хтось сам добровільно прагне вступити до армії, не враховуючи різноманітність і тип небезпеки, у яку він буде залучений.

Широкий психологічний аналіз волонтерів спрямований на розуміння мотивації та дій індивідів на різних етапах служби. Таким чином, слід знати, що навіть у разі кризового особистого рішення психологічні фактори матимуть довгострокові наслідки для психічного здоров'я волонтера. Це вимагає, щоб у воєнний час були доступні системи психологічної підтримки та лікування для тих, хто вступає до військових формувань на інших підставах, крім патріотизму, і хто діє імпульсивно.

Це забезпечить постійну психологічну рівновагу волонтерів і посилить їхню діяльність у важких умовах, знизить ризики патології та закріпить стабільність військових формувань у цілому.

1.2 Огляд сучасних підходів до розробки інформаційних систем

Поширення сучасних методів побудови інформаційних систем дійсно має вирішальне значення для будь-якої практичної побудови інформаційних систем [3]. Приклад СК: навчання та облік учасників територіальної оборони. Цей швидкий розвиток технологій, швидкі зміни в інформаційних технологіях і

створення інформаційних систем мають значний вплив на якість, продуктивність і безпеку впровадження інформаційних систем. У пошуках практичних підходів до розробки інформаційних систем ми змушені обговорити кілька фундаментальних підходів до розробки інформаційних систем, які б забезпечили ефективну, надійну та безпечну роботу в середовищах із високим ступенем місії, таких як військові.

Найбільш популярною методологією розробки інформаційних систем є системний аналіз. Він включає аналіз вимог, функціональних і нефункціональних специфікацій, проектування системи, програмування, тестування, впровадження та обслуговування. Особливу увагу необхідно приділити стабільності та надійності системи організації та підготовки волонтерів, яка має бути стабільною з достатньою кількістю робочих годин протягом року, незалежно від того, чи є війна чи якась інша надзвичайна ситуація.

Підходи до розробки інформаційної системи базуються на SDLC (рисунок 1.1).

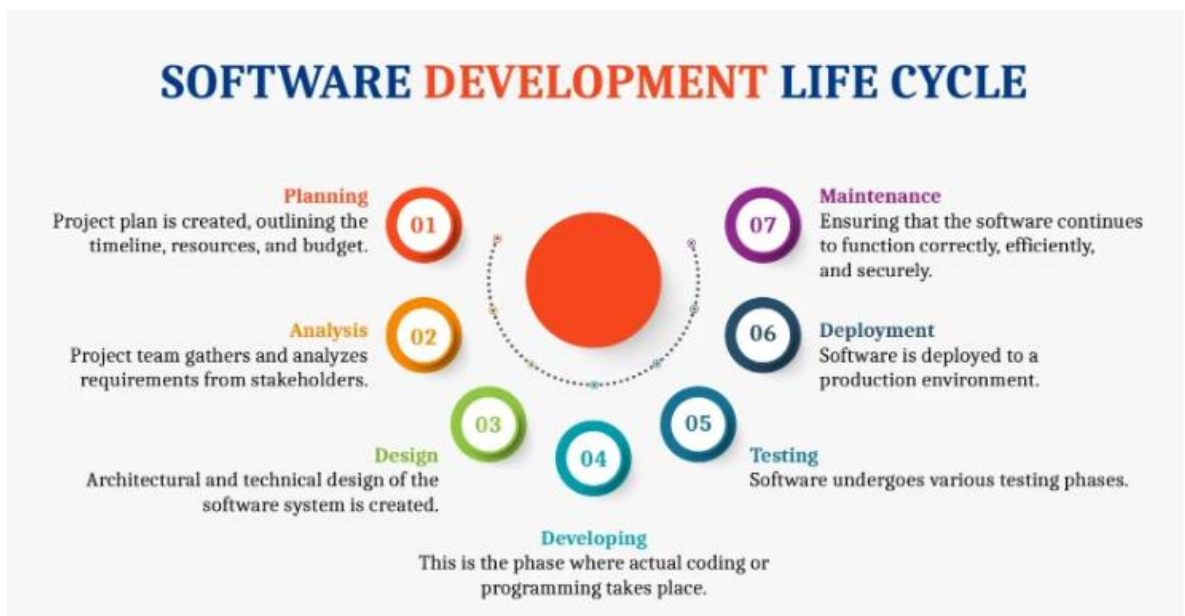


Рисунок 1.1 — Software Development Lifecycle

Це переважний спосіб розробки програмного продукту, вимоги до специфікацій, аналіз технічного обслуговування, а також розповсюдження та

методологія. Зазвичай застосовуваними моделями є каскадна модель, спіральна модель розробки, ітераційна модель тощо. Спіральна модель розробки: як правило, використовується в інформаційні технологічні системи, що мають значний компонент взаємодії. Це найкращий спосіб раннього виявлення помилок.

Власне, передбачається використання інноваційних методів розробки програмного забезпечення. Наприклад, за допомогою ООП (об'єктно-орієнтованого програмування) можна створити більш гнучку та розширювану систему.

Іншими альтернативами є системи розподіленої структури [4], з відмовостійкістю та масштабованістю для інформаційних систем у складних середовищах. Рекомендується, щоб програмне забезпечення охоплювало мікросервіси, які поділяють величезні програмні продукти на невеликі окремі одиниці, що розробляються, оновлюються та обслуговуються автономно.

Контейнерна система, наприклад Docker (рисунок 1.2), буде більш стабільною та працюватиме краще, тоді як користувачі стикатимуться з меншою кількістю конфліктів версій у середовищах.

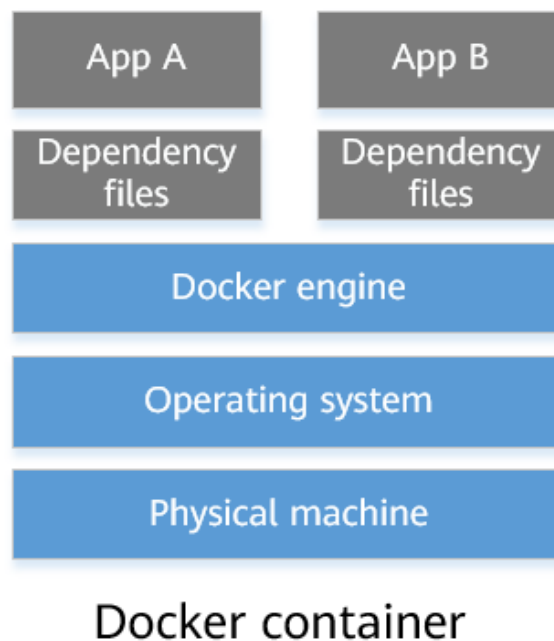


Рисунок 1.2 — контейнерна система Docker

При створенні інформаційної системи територіальної оборони було б надзвичайно важливо співпрацювати з іншими національними/військовими платформами чи базами даних або системами моніторингу та аналізу архітектури системи в слід брати до уваги зв'язок із відкритими API та впровадженням стандартів обміну даними.

Значну увагу слід приділяти безпеці разом із штучним інтелектом[5], а також великими даними, які підвищують точність і швидкість обробки інформації. Усе це виходить із даних про волонтерів, їхній фізичний та психологічний стан, а також готовність до захисту.

Таким чином, оскільки ефективність і безпека в першу чергу пов'язані з сучасними способами розробки ІС, ймовірність незалежного безперервного та ефективного тестування зростає в майбутньому.

Автоматизоване тестування також може бути дуже ефективним у цій операції, оскільки більшість помилок у цій великій системі буде виявлено під час циклу розробки.

Таким чином, стандартні проблеми встановлення системи будуть зведені до мінімуму завдяки цій великій установці ІС.

Це також робить скринінг настільки ж функціональним, безпечним і ефективним на необхідному рівні надійності в дуже складних ситуаціях ІС, коли це може знадобитися як захід на всякий випадок.

Звичайно, впровадження методології DevOps (рисуюнок 1.3), посилить зв'язок між командами розробки та підтримки, майже всі процеси життєвого циклу програмного забезпечення автоматизовані, а програмне забезпечення розгортається та модифікується відповідно до мінливих потреб і ситуацій.

Ці технології відомі як «безперервна інтеграція» або «розгортання» і мають великий вплив на розвиток; вони прискорюють процес оновлення та підтримки систем. Всю ідею системи можна легко зруйнувати зовнішньою загрозою. Мабуть, найважливішою з них є неоднозначність зовнішньої загрози, яка може

призвести до збою всієї системи за лічені хвилини. Прикладом цього є розподілені атаки на відмову в обслуговуванні.

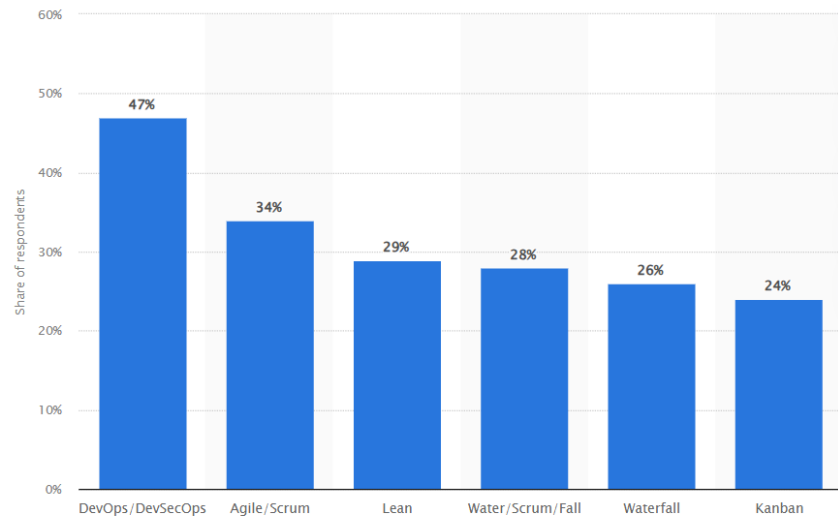


Рисунок 1.3 — Графік поширення використання DevOps методології

Таким чином, можна бути впевненим, що лише різні підходи до системного аналізу та проектування гарантуватимуть лише безпеку та функціональність підключеної до системи технології. Це дуже важливий момент, який особливо актуальний при реєстрації та навчанні добровольців територіальної оборони. Система працюватиме в складних середовищах, і її правильна робота є критичною.

1.3 Методи збору, обробки та аналізу даних в інформаційних системах

Система зберігає, обробляє та перевіряє інформацію, щоб приймати розумні рішення. Раніше було визнано, що ці методи набувають все більшого значення в розвитку інформаційних систем, присвячених реєстрації добровольців і підготовці національної оборони. Форма може включати проблеми зі здоров'ям, психологічні проблеми та іншу особисту інформацію, яка є конфіденційною та має бути розкрита. Таким чином, збір, обробка та аналіз даних повинні бути організовані з урахуванням характеру завдань, пов'язаних з інформаційними системами.

Усі існуючі дані мають бути задокументовані та збережені у відповідному форматі. Щоб досягти цього, реєстрація волонтерів і навчання спеціалізованих груп визнані одними з найголовніших задач для пошуку прямих, конкретних і релевантних методів збору інформації про різні аспекти з багатьох джерел.

Форми можуть являти собою листи про реєстрацію[6], звіти про аналіз історії хвороби, запитання та відповіді та іншу інформацію, яка додається або включається в програму на мобільному телефоні чи комп'ютері.

Іншими важливими посиленнями в дослідженні є інші національні реєстри, які документують освіту, професійні здібності та досвід, пов'язаний із несприятливою роботою чи військовою службою. Процедура збору даних повинна проводитися ретельно, щоб підвищити її точність, повноту та надійність, а також зменшити ймовірність помилок.

Інформацію можна отримати за допомогою онлайн та офлайн-опитувань або інтерв'ю, які використовують Інтернет як засіб проведення опитування чи інтерв'ю. Автоматизація збору даних повинна поєднуватися з існуючими джерелами даних і системами, без надмірних або неправильно витлумачених результатів.

Крім того, інформація, отримана під час відповідей на анкети чи інтерв'ю, є більш правдивою та має більший психологічний і соціальний вплив. Крім основної функції обробки даних, він також виконує завдання агрегації, оскільки всі дані з різних джерел матимуть єдиний формат, що дозволить аналізувати інформацію з різних джерел одночасно.

Деякі методи очищення даних використовуються для усунення або виправлення, наскільки це можливо, дублікатів або неправильних записів. Це дуже важливо, оскільки неправильні або неповні дані можуть призвести до неправильного уявлення або помилкового висновку, що негативно вплине на систему.

Безпека є одним із найважливіших факторів обробки даних, тому необхідно зберігати конфіденційність. Оскільки інформація про здоров'я також може

передаватися від донорів до тренерів разом із особистою інформацією, надзвичайно важливо мати сучасні методи зберігання цієї інформації та знань.

Це може включати просте шифрування даних на базі даних або такі технології, як блокчейн, усі вони обертаються навколо легітимності та прозорості обробки інформації. У цьому відношенні дані мають вирішальне значення. Вони отримують інформацію[7], а статистичні методи та прилади виводять закономірності, тенденції та зв'язки, які в них «не виявляються». Поширені аналітичні методи — це регресійний аналіз, класифікація, кореляція та машинне навчання, які оцінюють ефективність навчальних програм і передбачають ймовірне майбутнє програми. Приклад регресійного аналізу та методу класифікації зображено на рисунку 1.4.

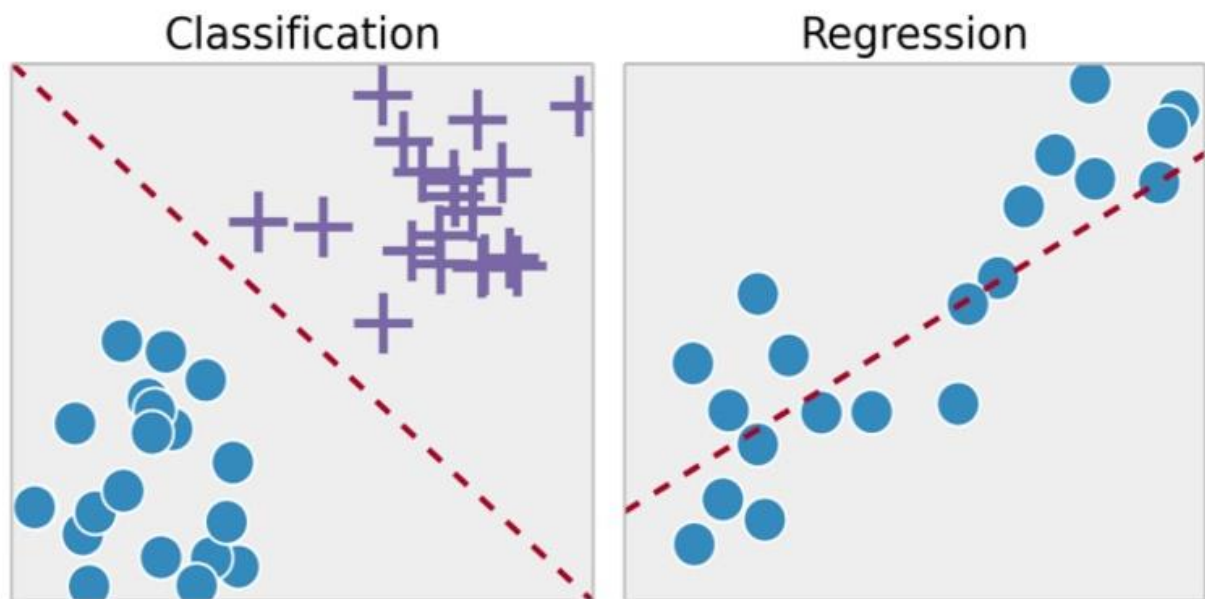


Рисунок 1.4 — Приклад класифікації та порівняння з регресією

Як правило, щоб керувати потоком інформації та збирати велику кількість даних, наявних в організації, сховища даних вимагають передових методів і технологій, таких як Hadoop або Spark, для обробки та зберігання інформації в розподіленому середовищі.

Ефективний візуальний аналіз простий для розуміння та часто базується на

важливому рішенні. Це можна зробити за допомогою графіків, таблиць, інформаційних панелей або будь-якого іншого методу, який швидко та легко показує «статус/підготовленість», «емоційність» або інші якості волонтера.

Зв'язок між різними джерелами даних, аналітикою та мобільними додатками, призначеними для збору інформації, пов'язаної зі здоров'ям, матиме вирішальне значення, особливо це стосується інформації щодо реєстрації та навчання волонтерів. Зрештою, йдеться про процес використання методів збору та аналізу даних для прийняття рішень та інтерпретації отриманих результатів. Таким чином, ці інструменти повинні створювати не лише звіти, але й рекомендації, отримані в результаті аналізу даних.

Це збільшує персональну взаємодію між системою та користувачем, що, як наслідок, підвищує її ефективність. Це приклад типового користувача, який використовує введення: enter[8]. Користувачі матимуть можливість приймати точні рішення щодо потенціалу для певної роботи, професійного розвитку та історії хвороби волонтера.

Відтак, саме порядок збору, обробки та інтерпретації даних стане основним поштовхом для будь-якої ефективної інформаційної системи щодо реєстрації та підготовки добровольців територіальної оборони. Це пояснюється тим, що ці методи не тільки забезпечують легітимність і надійність даних, але й мають конкретну мету, пов'язану з ними.

Поганий аналіз і обробка даних може негативно вплинути на інформацію, яка була спочатку зібрана, це може призвести до зниження продуктивності та відсутності безпеки, серед інших проблем.

1.4 Архітектура та компоненти інформаційних систем для управління ресурсами

Організації чи підприємства будь-якого профілю здатні забезпечити роботу системи, наприклад системи управління ресурсами територіальної оборони. Вони підтримують структурований процес організації інформації, пов'язаний із

функціональними можливостями системи, такими як реєстрація та відстеження системи, зустрічі та продуктивність серед іншого[9]. Це можуть бути людські ресурси, матеріальні ресурси, інформаційні потоки, системи фінансових даних, технологічні ресурси та інші ресурси, необхідні для виконання обов'язків внутрішньої безпеки.

Інструменти для планування ресурсів підприємства. Для побудови інформаційних систем управління ресурсами можна розглядати різні структури, але більшість із цих систем мають деякі спільні компоненти та принципи: архітектурні рівні, що утворюють систему, компоненти, які збирають і обробляють дані, компоненти, які керують ресурсами, а також компонент моніторингу, який аналізує та проектує.

Архітектура інформаційної системи має підтримувати швидку передачу даних, збереження інформації, швидкий доступ до важливих ресурсів і жорстку безпеку.

Основна архітектурна особливість вищезазначеного типу багаторівневої архітектури, яка включає в себе багато компонентів, що взаємодіють один з одним, створена для виконання вищезазначених функцій. Як правило, багаторівнева архітектура має певний розподіл обов'язків і завдань між її шарами або рівнями. Це підвищує керованість і корисність у підтримці та розробці нових функцій [10].

Перший рівень — це етап збору даних, який отримує інформацію про фізичний стан волонтерів, психічний стан, спеціальну медичну інформацію та інші параметри щодо систем внутрішньої безпеки. Методів збору даних може бути багато, наприклад, на основі анкетування, на основі медичних записів, на основі системи моніторингу, а також інноваційні методи для користувачів. Процедура збору даних має бути побудована таким чином, щоб інформація була доступна в дуже простому та швидкому режимі та була досягнута максимальна точність та надійність інформації. Інформація про волонтерів має бути

конфіденційною; отже, безпека даних і конфіденційність також повинні розглядатися на цьому рівні.

Другий рівень має архітектура, яка є рівнем обробки даних, який дозволяє збирати дані, а потім обробляє їх, виконує аналіз і структурування, щоб їх можна було використовувати далі. Така обробка може включати стандартизацію, очищення, виявлення помилок і статистичний аналіз або пошук тенденцій, шаблонів і аномалій за допомогою алгоритмів машинного навчання; все виконується автоматично. Отже, система повинна видавати хороші звіти, рекомендації чи прогнози на основі зібраної інформації. Крім того, він повинен мати можливість обмінюватися та поєднувати дані з іншими платформами чи програмами.

Третім компонентом архітектури є управління ресурсами. Цей рівень дозволяє доступ до ресурсів, контроль використання ресурсів або планування використання ресурсів. Наприклад, будь-яке задане завдання для волонтерів, необхідні навички та фізичні атрибути для даного завдання, а також доступні фізичні ресурси, будь то обладнання чи приміщення. Крім того, він повинен переглядати тимчасовий статус ресурсів, знати, в який час вони дефіцитні або в надлишку, а також оперативний аналіз зі змінними параметрами.

Четвертою важливою особливістю системи є зберігання даних та управління інформацією. Це означає, що будь-які введені дані повинні якимось чином зберігатися в базі даних системи в організованому чи невіпорядкованому порядку. Конструкція системи повинна забезпечувати безпечне зберігання інформації, легкий доступ до неї та швидку обробку. Цього можна досягти за допомогою різних технологій, згаданих у реляційних базах даних, сховищах NoSQL, а іноді і спеціальних платформах для зберігання великих обсягів даних, розміщених у хмарі. Сьогодні безпека даних є однією з головних проблем на даний момент часу; доступ до збережених даних здійснюється за допомогою деяких методів шифрування та інших відповідних методологій авторизації для цієї мети.

П'ятий рівень архітектури - це рівень аналізу та прогнозування. Тут виконується більш глибокий аналіз зібраних і оброблених даних, щоб передбачити поведінку системи, оцінити ефективність виконуваних завдань або визначити потребу в додаткових ресурсах. Ключовими аналітичними інструментами на цьому рівні є різні форми статистичного аналізу; машинне навчання та алгоритми на основі штучного інтелекту, переважно ефективні для прогнозування майбутніх потреб, оцінки ефективності волонтерів і виявлення прихованих ризиків або проблем під час роботи.

Нарешті, кожна інформаційна система управління ресурсами повинна включати забезпечення взаємодії користувачів. Повинен бути доступний через браузер та/або за допомогою мобільних додатків, щоб користувачі могли працювати з будь-якого місця.

Інтерфейс має бути розумним і простим; користувачі повинні мати можливість просто робити те, що потрібно, отримувати необхідну інформацію, а інколи швидко приймати рішення миттєво.

Архітектура інформаційної системи управління ресурсами повинна забезпечувати повний взаємозв'язок усіх цих компонентів і рівнів для забезпечення її ефективної роботи, а також швидкої та безпечної поведінки.

1.5 Системи для автоматизації процесів підготовки та обліку персоналу

Підготовка особового складу та автоматизація облікових процесів є найважливішими складовими сучасних інформаційних систем, що використовуються в управлінні персоналом різних сфер, у тому числі військ територіальної оборони. Усі ці системи мають простий у використанні інтерфейс, який полегшує збір, обробку, зберігання та аналіз даних про персонал, що дає змогу планувати, контролювати та контролювати навчальні та фінансові процеси, а також максимально ефективно використовувати ресурси.

Деякі з функцій[11], які роблять ці системи важливими з точки зору територіальної оборони та підготовки волонтерів, — це швидка реєстрація

користувачів, сортування та планування оцінювання підготовки та інші освітні програми для волонтерів. Такі системи допоможуть не лише створювати списки волонтерів, а й ефективно керувати даними щодо особистих здібностей, історій хвороби, знань і підготовки осіб. Це допоможе належним чином спрогнозувати необхідну підготовку та ресурси, необхідні для виконання завдань.

Також автоматизація обліку особового складу може забезпечити умовне спостереження за добровольцями щодо необхідності участі у фізичних і розумових заняттях, надавати їм інформацію про останню участь у навчаннях, а також під час бойових дій. Завдяки цьому він не лише регулярно оцінюватиме ступінь забезпечення, а й реагуватиме на відповідні зміни умов, спричинені бойовими діями, відпустками, медичними оглядами чи іншими факторами. Іншою важливою характеристикою є його здатність підключатися до інших платформ. В результаті інформація передається автоматично, що зводить до мінімуму ймовірність помилок.

Системи автоматизації людських ресурсів зазвичай мають великий обсяг даних, які необхідно обробити. Це більше стосується територіальної оборони через збільшення кількості учасників та різноманітні вимоги, включаючи фізичну підготовку, кваліфікацію, досвід, спеціальні здібності тощо, які необхідно враховувати. ручний збір даних у цих випадках повільний, громіздкий і не забезпечує бажаних результатів; в результаті автоматизація може прискорити процес і повністю виключити втручання людини.

Крім того, модуль аналізу та прогнозування можна інтегрувати в сучасні системи автоматизації бухгалтерського обліку та розвитку персоналу[12]. Він матиме можливість не тільки регулювати поточну реальність, але й передбачати майбутні вимоги щодо додаткового навчання чи спеціалізованого персоналу в майбутньому. Ці аналітичні інструменти на основі накопичених даних дозволяють коригувати навчальні плани відповідно до поточної ситуації, реагувати на зміни на території чи передовій, розробляти довгострокову стратегію щодо розвитку персоналу.

Що сприяє створенню чіткої структури, яка враховує всі аспекти діяльності персоналу, зокрема поділ на підрозділи, які дають контрольні завдання щодо їх виконання та оцінку результатів. Це сприяє підвищенню ефективності управління персоналом, а також швидкому реагуванню на ситуації, що вимагають швидкого переведення конкретного контингенту в інший підрозділ або забезпечення додаткової підготовки чи ресурсів. Це сприяє автоматизації обох процесів і їх оптимізації.

Останній метод стосується прив'язки цих систем до решти стрічки інформаційної системи. Такий сценарій може містити волонтерську інформацію, прив'язану до системи моніторингу бою, автоматичного оновлення медичних записів, а також фінансових систем, які належним чином виплачують гроші. Це додатково зменшує складність управління та відстеження процесів і знижує ймовірність помилок, пов'язаних із введенням і обробкою вручну.

Іншим прикладом є сучасні інформаційні системи обліку витрат праці, які можуть використовувати моделі планування та управління навчанням і поєднуватися з іншою інформацією щодо вдосконалення процесів навчання. Ці функції сприяють координації часу та ресурсів, пов'язаних із навчанням, з огляду на наявність конкретних волонтерів, кваліфікацію, спеціалізацію та рівні підготовки. В результаті своєчасне навчання окремих груп дозволить підвищити ефективність навчання в конкретному контексті або спеціалізації.

В результаті автоматизація облікових процесів разом з навчанням персоналу підвищує ефективність і точність прийняття управлінських рішень за рахунок зменшення складності збору та аналізу інформації та більш ефективного використання ресурсів. З цього моменту ця структура може слугувати великою системою підтримки територіальної оборони, забезпечуючи швидкий та ефективний механізм організації волонтерської роботи, яка є важливою складовою стратегії досягнення цілей та своєчасного реагування на зміни обставин.

1.6 Інструменти та технології для розробки інформаційних систем

Загалом, це досить складна багатокомпонентна розробка інформаційних систем, у якій потрібно вибрати правильне програмне забезпечення та інфраструктуру для найвищого рівня безпеки та масштабованості системи. Зокрема, при розробці інформаційної системи реєстрації та підготовки добровольців для захисту Батьківщини необхідно враховувати умови роботи та величезні дані, які необхідно обробляти, взаємозв'язки з іншими системами та зручність використання.

Таким чином, однією з найважливіших речей на цьому етапі є вибір відповідної технології для створення серверної та клієнтської частин, а також бази даних, яка може підтримувати безперебійну взаємодію між усіма частинами системи.

У цьому розділі розглядаються технології, які були застосовані для розробки системи: PHP, JavaScript, MySQL і Bootstrap.

PHP означає Hypertext Pre-processor[13]. Це одна з найпопулярніших серверних мов програмування для розробки веб-додатків. Це, у свою чергу, робить розробку дуже гнучкою, а також підвищує швидкість у розробці дуже інноваційних динамічних веб-сторінок і програм, які мають високу чутливість. Одна дуже велика перевага PHP полягає в тому, що він має відкритий вихідний код і існує велика кількість доступних рішень і бібліотек.

Поєднуючи багато інших баз даних, наприклад, MySQL, PHP може створювати високорозвинені та масштабовані інформаційні системи для роботи з величезними обсягами інформації про волонтерів та їхній стан здоров'я, рівень знань тощо. Фреймворк також дозволяє програмістам писати чітко закодовані, добре масштабовані, а також дуже посилені програми або веб-сайти.

В розробленій інформаційній системі сценарії на стороні сервера виконуються за допомогою PHP, і це допоможе забезпечити безперебійну роботу системи, швидку обробку запитів і підтримку майбутніх проєктів, які будуть

розроблені. Таким чином інформаційні системи для клієнта реалізуються переважно на JavaScript – додатках для роботи WS через веб-інтерфейси, зокрема інтерактивні веб-сторінки, які є зручними та зрозумілими. Це дає змогу застосовувати функціональні інтерфейси користувача через Інтернет, роблячи використання таких систем більш інтуїтивно зрозумілим та інтерактивним у роботі.

Такими функціями JavaScript[14] можуть бути, наприклад, перевірка пошуку вхідних даних; вміст, який оновлюється самостійно без мерехтіння сторінки, яка перезавантажується через AJAX; або взаємодії з іншими службами та ресурсами.

У поєднанні з PHP і MySQL JavaScript дозволяє розробляти інтерактивні інформаційні системи для відстеження та керування інформацією про волонтерів, представлення даних у реальному часі, а також можливість пошуку, фільтрації та сортування.

MySQL є однією з найважливіших технологій для зберігання та обробки інформації. Він пропонує ефективну платформу RDBMS щодо детальної інформації про все, що завгодно структуровано, наприклад особисту інформацію волонтерів, історію навчання, записи про ліки, результати навчання тощо та інші пов'язані функції системи[15].

Існують різноманітні вбудовані функції для ідеальної організації зберігання даних, а також швидкого запиту до бази даних, двох найважливіших аспектів в інформаційному центрі. Таким чином, використання MySQL забезпечить безпечне зберігання даних і ідеальну роботу в системах, які містять особисту інформацію користувача. Крім того, MySQL підтримує резервне копіювання, що допомагає повністю відновити систему після збою, зберігаючи безперебійну роботу.

Bootstrap — ще один фреймворк для розробки зручних інтерфейсів. Використовуючи цю технологію, ви можете отримувати добре розроблені веб-сторінки, зручні для пристроїв. Це дозволяє мінімізувати час розробки для

зовнішніх компонентів, тим самим скорочуючи час, витрачений на ручне написання довгих сценаріїв CSS або JavaScript для розробки адаптивних макетів[16].

Існує багато вбудованих бібліотек компонентів, таких як кнопки, форми, таблиці, навігаційні панелі, моди тощо, які ви можете просто стилізувати по-своєму та використовувати в розробці. Отримання ефективного інтерфейсу при використанні цього інструментарію не займе багато часу, тому він значно підвищує продуктивність роботи користувачів системи. Це дуже корисно, особливо у випадку систем реєстрації та навчання волонтерів, коли користувачі потребують доступу до найпростішої інформації, яку вони шукають, і хотіли б ефективно взаємодіяти з системою.

Хороший набір PHP, JavaScript, MySQL і Bootstrap утворює гарний стек розробки ІС, який забезпечує стабільну та безпечну роботу сервер-клієнт. Можливе впровадження зручної крос-функціональної інформаційної системи для процесів, які ведуть у протидії зазначеним заходам.

Таким чином, зазначені технології розробки системи обліку добровольців територіальної оборони є актуальними та оптимізують кожну роботу системи для надання дуже ефективної послуги.

1.7 Безпека та захист даних в інформаційних системах

Найважливіший аспект функціональності кожної сучасної інформаційної системи та інформації, яку вона підтримує, особливо коли йдеться про чутливу або конфіденційну інформацію. У пропонованій системі реєстрації та підготовки волонтерів Вітчизняної оборони інформаційна безпека віддається пріоритету для захисту особистої інформації, а також медичних і психологічних даних, необхідних для військової служби.

Першим і найбільшим є конфіденційність, яка гарантує, що будь-яка інформація має залишатися секретною та обмеженою для тих осіб, яким дозволено або мають права переглядати чи отримувати до неї доступ.

Таким чином, дані можуть бути доступні, змінені або використані лише авторизованими користувачами з відповідними правами доступу. Технологія шифрування передбачає, що дані, зібрані від користувача/ів, залишаються недоступними при передачі через будь-яку мережу. При цьому використовуються симетричні (AES, DES), асиметричні (RSA, ECC) серед інших різноманітних методів залежно від реквізитів системи. Маркери SSL/TLS необхідно використовувати для забезпечення безпечного з'єднання між клієнтським і серверним компонентами системи.

Цілісність даних є другим за важливістю елементом безпеки; він гарантує, що інформація, яка зберігається або передається в системі, є точною та повною. Щоб запобігти несанкціонованим змінам даних під час обробки, зберігання чи передачі, і це передбачає «як» захисту даних.

Це функції перевірки відповідності та автентичності відповідно. Наприклад, якщо хтось зберігає особисту інформацію про користувача в базі даних, він може хешувати пароль, щоб навіть якщо дані бази даних підроблено, зловмисник все одно не зможе отримати до них доступ. Основою доступності даних є надання даних, коли і де вони потрібні користувачеві.

Це включатиме технології зберігання даних, розподілену архітектуру та відмовостійкість для забезпечення доступності даних. Через те, що конфіденційні дані є конфіденційними, необхідно використовувати суворі мережеві фільтри та механізми безпеки для захисту програмних компонентів, які обробляють такі дані, від атак DoS і DDoS.

Іншим основним компонентом безпеки даних є реалізація автентифікації та авторизації. За своєю суттю система автентифікації дозволяє визначити особу конкретного кінцевого користувача або організації за допомогою комбінації паролів, біометричних методів або обох.

Водночас авторизація передбачає надання доступу лише певним функціям та інформації належним чином автентифікованим особам. В даний час механізми багатофакторної автентифікації, засновані на одноразових паролях (OTP),

цифрових сертифікатах і біометричних характеристиках, стають звичайними для письмових даних інформаційних систем (введення, розпізнавання обличчя). Що ще важливіше, потрібно запобігти хакерству проти даних у русі та підробці атак типу "людина посередині", що спричиняє витік даних під час передачі, забезпечивши захист зв'язку між користувачем і сервером за допомогою SSL/TLS. Захист даних заслуговує системного аудиту та моніторингу.

Таким чином, будь-які потенційні загрози безпеці та зброї можна швидко виявити та відреагувати на них. За допомогою аудиту доступу та постійного моніторингу активності користувачів підозрілу поведінку та спроби незаконного доступу до важливих даних можна виявити та належним чином відреагувати на них.

З цією метою всі важливі операції, які виконують інформаційні системи, пов'язані з оновленням даних, створенням нових записів і доступом до конфіденційної інформації. Використання IPS/IDS, який може виявляти, а також запобігати підозрілій діяльності в режимі реального часу, також є дуже важливим. Другим найважливішим фактором є управління внутрішніми загрозами. Звичайно, оскільки зловмисниками часто є самі співробітники компанії, названа організація повинна зменшити ризик, пов'язаний із цими загрозами. Цього можна досягти, обмеживши привілеї, за яких більшість внутрішніх систем і даних мають доступ недружні користувачі, а потім будь-яку ненавмисну діяльність внутрішніх користувачів.

Інші заходи безпеки полягають у тому, що дані мають бути безпечними під час передачі через Інтернет. Це досягається за допомогою технологій, які зберігають конфіденційність і цілісність інформації. Наприклад, протоколи VPN, IPsec і TLS/SSL використовуються для передачі секретної інформації через Інтернет. Використання цих технологій не дозволяє зловмисникам перехопити або змінити дані.

Загалом безпека інформаційної системи та захист даних є досить тернистим і складним процесом, що включає як технічні, так і організаційні заходи. Не

применшуючи місця цих заходів у системах обліку та підготовці волонтерів територіальної оборони, необхідно зазначити, що хороша організація безпеки може запобігти багатьом зовнішнім, а також внутрішнім загрозам.

2 МЕТОД ІНТЕГРАЦІЇ ІНФОМАЦІЙНОЇ СИСТЕМИ ЗІ ШТУЧНИМ ІНТЕЛЕКТОМ

Штучний інтелект вважається міждисциплінарною сферою, яка сприяє використанню комп'ютерів, роботів та інших машин для виконання завдань, які потребують інтелекту. Зазвичай їх виконують люди. Сучасні інформаційні системи виграють ід аналізу великих обсягів даних, розпізнавання образів і здатності адаптуватися до нових умов, усіма цими володіє штучний інтелект, перш за все тому, що він є основним компонентом більшості сучасних інформаційних систем і має важливе значення до їхнього дизайну. Ці системи в першу чергу призначені для вирішення складних багатокомпонентних проблем, таких як управління військовими ресурсами або підбір і підтримка рішень експертів.

Поєднання інформаційних систем зі штучним інтелектом є засобом узагальнення процесів і підвищення їх ефективності. Основною метою інтегративного підходу є об'єднання звичайних інформаційних систем із сучасними алгоритмами штучного інтелекту для максимальної продуктивності. Таким чином, інформаційні системи працюватимуть як платформи, на яких зберігатимуться, оброблятимуться та передаватимуться дані, а штучний інтелект розширюватиме інтелектуальну обробку інформації, виявлення складних закономірностей та адаптацію до змін. Таким чином, процес інтеграції складається з кількох основних етапів, кожен з яких спрямований на встановлення ефективних зв'язків між компонентами системи.

Перший етап – це визначення вимог до інформаційної системи та функціональності. Він включає оперативне призначення, оперативні завдання або роботи, які має виконати система, необхідні дані або сценарії введення та використання. Наприклад, військово-професійний відбір та перевірка, особистісні характеристики волонтерів (вік, зріст тощо), фізичний стан, професійні навички,

психологічні характеристики. У цьому контексті штучний інтелект аналізуватиме ці дані та водночас формулюватиме рекомендації.

На другому етапі відбувається підготовка даних. «Чистий, структурований ані є єдиною вимогою для того, щоб система була готова до аналізу». Це включає методи обробки природної мови, що застосовуються для аналізу інформації в текстовій формі, алгоритми для обробки чисел і моделі глибокого навчання для аналізу дуже складні візерунки. Ці технології забезпечують відповідне представлення даних для подальшої обробки алгоритмом штучного інтелекту.

Третій крок – об'єднання модуля ШІ з інформаційною системою. Отже, це означає, що машинне навчання, глибоке навчання а алгоритми обробки природної мови пов'язані з ядром системи. На основі моделей машинного навчання можна аналізувати шаблони в даних, робити прогнози та давати рекомендації щодо розподілу волонтерів за військовими спеціальностями. Текстові дані, такі як анкети и звіти, сприйматимуться алгоритмами обробки природної мови; отже, кожен аспект кожного користувача буде враховано в *lethe.flow* ще одним атрибутом є забезпечення системи самонавчання завдяки інтеграції штучного інтелекту; чим швидше об'єкт штучного інтелекту навчиться сам, тим точнішим і ефективнішим він буде з часом. Методи активного навчання постійно оновлюють модель новими даними, отриманими від користувачів або зовнішніх джерел; Це буде дуже важливо для армії, тому що все змінюється і їм доведеться швидко адаптуватися до нових умов. Інтерфейс користувача (UI): останнє, але не менш важливе, це саме забезпечення простої у використанні системи для кінцевого користувача для взаємодії з системою. Це може включати інтуїтивне представлення даних; наприклад, він може відображати результати аналізу на графіках, діаграмах або інтерактивних інформаційних панелях. Найбільш складним у цій ситуації є те, як підтримувати кожну частину інформації, яку обробляє ШІ, у безпечному середовищі та ніколи не витікати чи не отримати до неї незаконний доступ. Це буде досягнуто за допомогою останніх методів шифрування, контролю доступу, серед інших технологій мережевої безпеки.

Штучний інтелект дає інформаційним системам великі переваги: автоматизацію процесів і кращу якість рішень. Дані можна швидко проаналізувати за допомогою штучного інтелекту; можна врахувати більше факторів, і рекомендації будуть більш точними. Це досягається за рахунок вдосконалення управління персоналом у військових питаннях, оптимізації ресурсів, а також підвищення адаптивності систем до мінливих умов. Таким чином, інформаційні системи в поєднанні зі штучним інтелектом розглядаються як інтегративний елемент у процесі створення сучасних технічно передових рішень складних проблем.

2.1 Аналіз можливостей штучного інтелекту для оцінки і підбору військових спеціальностей

Штучний інтелект (ШІ) відкриває широкі можливості для вирішення складних завдань, зокрема для оцінки та підбору військових спеціальностей. Завдяки своїм адаптивним і прогнозувальним можливостям, ШІ здатний обробляти великі обсяги інформації, аналізувати її з високою точністю та враховувати багатофакторні зв'язки між параметрами. Це дозволяє створювати ефективні системи, що здатні автоматизувати процеси, які раніше вимагали значних ресурсів і людського втручання.

Основною перевагою штучного інтелекту в такому контексті є його здатність до інтеграції різних типів даних. Сучасні алгоритми машинного навчання можуть працювати з різнорідними інформаційними джерелами, включаючи числові показники, текстові дані, зображення та навіть відео. Це особливо важливо в процесі оцінки військових спеціальностей, де враховуються фізичні, психологічні, когнітивні та професійні характеристики добровольців. Завдяки використанню багатовимірного аналізу, штучний інтелект здатен враховувати сотні параметрів одночасно, визначаючи найкращі варіанти відповідно до заданих критеріїв.

Можливості штучного інтелекту в оцінці військових спеціальностей базуються на використанні алгоритмів машинного навчання та глибокого

навчання, які дозволяють знаходити закономірності у великих наборах даних. Ці алгоритми здатні визначати приховані зв'язки між характеристиками добровольців і вимогами до конкретних спеціальностей, які не завжди очевидні при використанні традиційних підходів. Наприклад, ШІ може виявити, що певні особистісні риси або навички, які раніше не бралися до уваги, мають істотний вплив на успішність виконання певних військових завдань.

Застосування штучного інтелекту дозволяє здійснювати персоналізований підхід до підбору спеціальностей. Завдяки аналізу індивідуальних характеристик кожного добровольця, система може створити точну картину його потенціалу, виявити сильні сторони та обмеження. Це дозволяє максимально ефективно розподіляти ресурси та забезпечувати, щоб кожна людина займала посаду, яка відповідає її здібностям і професійним навичкам. Такий підхід також сприяє підвищенню морального духу добровольців, оскільки вони отримують можливість виконувати завдання, які відповідають їхнім інтересам і компетенціям.

Штучний інтелект також відкриває нові можливості для об'єктивної оцінки психологічного стану добровольців. Використовуючи алгоритми аналізу поведінкових даних і текстової інформації, система може ідентифікувати психологічні характеристики та стійкість до стресу. Це особливо важливо в умовах воєнного часу, коли добровольці стикаються зі значними психоемоційними навантаженнями. Своєчасна оцінка стану дозволяє уникнути потенційних проблем і забезпечити якісний підбір спеціальностей, які відповідають їхнім можливостям.

Крім цього, штучний інтелект надає можливість прогнозування ефективності добровольців у конкретних ролях. Завдяки використанню історичних даних та моделей машинного навчання, можна передбачити, наскільки успішно людина буде виконувати завдання у певних умовах. Це дозволяє мінімізувати ризики, пов'язані з неправильним розподілом обов'язків, і сприяє досягненню високої ефективності роботи військових підрозділів. Система також

може оцінювати потенціал для навчання добровольців новим навичкам і адаптації до складних умов, що є критично важливим у військовій сфері.

Ще однією важливою особливістю використання штучного інтелекту є його здатність до самонавчання. З кожною ітерацією алгоритми стають точнішими, оскільки використовують нові дані для вдосконалення своїх моделей. Це означає, що система може адаптуватися до змінних умов і забезпечувати постійне підвищення точності оцінок і рекомендацій. Такий підхід дозволяє створити динамічну систему, яка здатна працювати з великими обсягами даних у реальному часі, зберігаючи при цьому високу швидкість обробки.

Особливе значення має можливість інтеграції штучного інтелекту з іншими інформаційними системами, такими як бази даних добровольців або модулі навчання. Це дозволяє створювати комплексні рішення, які забезпечують повний цикл роботи з персоналом, від первинного відбору до моніторингу прогресу в навчанні та оцінки результатів. Завдяки інтеграції ШІ з такими системами, процес підбору спеціальностей стає прозорішим, структурованішим і менш залежним від суб'єктивних факторів.

Сучасні можливості штучного інтелекту також дозволяють автоматизувати процеси аналізу великих обсягів даних. Завдяки високій швидкості обробки інформації, ШІ може за короткий час аналізувати тисячі анкет добровольців, формуючи рекомендації на основі їхніх характеристик. Це значно знижує навантаження на кадрові служби, дозволяючи їм зосередитися на стратегічних завданнях. Автоматизація також сприяє мінімізації помилок, які можуть виникнути через людський фактор, забезпечуючи високу точність і надійність рішень.

Отже, використання штучного інтелекту для оцінки і підбору військових спеціальностей є надзвичайно перспективним напрямком, який дозволяє поєднати передові технології з практичними потребами. Завдяки своїм можливостям аналізу, прогнозування і адаптації, ШІ стає невід'ємним інструментом для

створення сучасних інформаційних систем, здатних забезпечувати високий рівень ефективності та точності в процесі роботи з добровольцями.

2.2 Вдосконалення методу інтеграції штучного інтелекту з інформаційною системою

Існує декілька способів реалізації інтеграції штучного інтелекту з інформаційною системою. Деякі розробники штучного інтелекту надають API, до якого можна надсилати запити для обробки даних. Такі API надають доступ до функціональних можливостей ШІ, які можуть бути адаптовані під конкретні завдання. Також існують спеціалізовані бібліотеки та фреймворки, розроблені для інтеграції штучного інтелекту з певними платформами чи мовами програмування. У деяких випадках штучний інтелект може працювати автономно, виконуючи задачі аналізу, обробки даних або оптимізації роботи системи без прямої інтеграції в інфраструктуру.

Компанія OpenAI, яка розробляє моделі штучного інтелекту, зокрема ChatGPT, надає доступ до платного API. Цей API пропонує широкий спектр функцій і методів, які дозволяють використовувати можливості штучного інтелекту для вирішення різноманітних завдань. Для реалізації інтеграції з інформаційною системою, що займається підбором військових спеціальностей, API OpenAI може забезпечити ефективний інструментарій для аналізу даних, генерації рекомендацій та автоматизації процесів.

2.2.1 Використання методів OpenAI API

OpenAI надає розробникам широкий спектр можливостей для інтеграції штучного інтелекту через API. ChatGPT можна налаштовувати для виконання різноманітних завдань, таких як пошук та виправлення помилок у коді, пояснення програмного коду різними мовами, перевірка текстів на граматичні помилки, виділення ключових слів із великих текстів тощо. Для реалізації цих функцій

OpenAI пропонує кілька ендпоінтів, хоча деякі з них наразі не підтримуються. Нижче наведено основні доступні методи.

Метод Speech-to-text перетворює аудіофайли на текст. Він дозволяє передавати аудіозапис у систему, яка повертає текстову розшифровку сказаного. Крім того, цей метод підтримує функцію перекладу тексту, отриманого з аудіо, на різні мови.

Метод Text-to-speech забезпечує озвучення будь-якого тексту. У відповідь на запит генерується аудіофайл, що містить озвучення переданого текстового фрагмента, що робить цей метод зручним для створення голосових інтерфейсів.

Метод Text generating відповідає за створення текстових відповідей на запити. Наприклад, сервер інформаційної системи може надсилати дані користувача до API, після чого ChatGPT генерує відповідь, яка може містити рекомендації, інструкції або будь-яку іншу релевантну інформацію.

Метод Image generation дозволяє створювати або модифікувати зображення на основі запитів користувача. Завдяки цьому методу можна, наприклад, генерувати візуалізації, що ілюструють різні аспекти тренувальних програм або специфіку професійних завдань, зокрема у військовій сфері. Методи Moderation використовуються для модерації контенту. Вони аналізують текст, що передається як параметр, і визначають такі показники, як:

- ненависть;
- погрози;
- харасмент;
- схильність до завдання собі шкоди;
- сексуалізація;
- жорстокість.

На основі числових значень цих показників можна оцінити, чи є контент прийнятним для публікації в інформаційній системі. Крім того, OpenAI пропонує вбудовані методи для аналізу текстових даних. Вони дозволяють:

- проводити пошук із впорядкуванням результатів за релевантністю;

- групувати текстові дані за подібністю;
- надавати рекомендації на основі схожості текстів;
- виявляти аномалії за низькою схожістю;
- аналізувати різноманітність текстів у вибірці;
- класифікувати текстові рядки за найбільш відповідними мітками.

На поточному етапі інтеграції в інформаційну систему для підбору військових спеціальностей використовується основний функціонал чат-бота, який забезпечує взаємодію з користувачами та відповіді на їхні запити. У майбутньому планується розширення можливостей системи, зокрема впровадження функції генерації візуальних матеріалів, таких як інфографіки, що відображають професійні навички або специфічні вимоги до різних спеціальностей.

Окрім цього, передбачається інтеграція функції озвучення тексту, яка дозволить користувачам із вадами зору отримувати інформацію в аудіоформаті. Така функція зробить систему більш доступною для різних категорій користувачів, забезпечуючи вищий рівень інклюзивності та зручності. Впровадження цих удосконалень значно підвищить ефективність і функціональність платформи.

2.2.2 Розробка модуля для інтеграції зі штучним інтелектом

Тригером для формування та відправки запиту до штучного інтелекту є спеціально розроблений ендпоінт `ask-for-specialization` (лістинг 2.1), який приймає ідентифікатор добровольця, для якого буде здійснюватись підбір військової спеціалізації.

Лістинг 2.1 — Приклад ендпоінту `ask-for-specialization`

```
[HttpGet("ask-for-specialization")]
public IActionResult AskForSpecialization(string userId)
{
    try
    {
```

```

        var personalData = _applicationContext.PersonalData.FirstOrDefault(c =>
c.UserId == userId);
        var skillsData = _applicationContext.SkillsData.FirstOrDefault(d => d.UserId ==
userId);
        if (personalData == null)
        {
            return NotFound($"No information about the volunteer with ID {userId}");
        }
        var specializationRequest = _requestBuilder.BuildSpecializationRequest(new
SpecializationRequestModel(personalData, skillsData));
        var chatResponse = _chatGptService.Ask(specializationRequest);
        return Ok(chatResponse);
    }
    catch (Exception e)
    {
        return StatusCode(500, e.Message);
    }
}

```

Даний метод отримує з бази даних персональні дані користувача, а також інформацію про його навички та досвід. Якщо необхідна інформація відсутня, ендпоінт повертає помилку з поясненням, що дані для підбору спеціалізації не були надані. У разі наявності інформації, вона використовується для створення запиту до штучного інтелекту, де ChatGPT буде налаштований на виконання ролі військового радника.

Приклад методу для побудови запиту представлено у лістингу 2.2. У цьому запиті описуються дані про добровольця, такі як вік, стать, рівень фізичної підготовки, професійні навички тощо.

Лістинг 2.2 — Приклад методу для побудови запиту до штучного інтелекту

```

public string BuildSpecializationRequest(SpecializationRequestModel
specializationRequestModel)
{
    var skillsData = specializationRequestModel.SkillsData;
    var specializationRequest = new StringBuilder(_messageBeginning);
    specializationRequest.AppendLine($"\"role\":    \"system\",    \"content\":    \"Ти
військовий радник.\"},");
    BuildBaseMessage(specializationRequestModel.PersonalData,
specializationRequest);
}

```

```

specializationRequest.Append("Мені потрібно, щоб ти підібрав спеціалізацію
згідно моїх даних.");
if (IsAnySkillsData(skillsData))
{
    AppendSkillsData(skillsData, specializationRequest);
}
specializationRequest.Append("Результат повинен бути представлений як
документ із описом рекомендованої військової спеціалізації, " +
    "а також необхідних кроків для підготовки до неї. Документ
має бути написаний чіткою, технічною мовою.");
var result = specializationRequest.ToString();
return result;
}

```

У сформованому запиті вказується завдання штучного інтелекту — надати рекомендацію щодо спеціалізації на основі даних про добровольця. Також, якщо користувач вказав додаткову інформацію про свої навички чи попередній досвід, вона додається до запиту (лістинг 2.3).

Лістинг 2.3 — Приклад методу для додавання інформації про навички добровольця

```

private void AppendSkillsData(SkillsData skillsData, StringBuilder
specializationRequest)
{
    specializationRequest.Append("Зверни увагу на наступну інформацію щодо моїх
навичок:");
    if (!string.IsNullOrEmpty(skillsData.ProfessionalExperience))
    {
        specializationRequest.Append($"* Мій професійний досвід:
{skillsData.ProfessionalExperience}.");
    }
    if (!string.IsNullOrEmpty(skillsData.PriorMilitaryExperience))
    {
        specializationRequest.Append($"* Попередній військовий досвід:
{skillsData.PriorMilitaryExperience}.");
    }
    if (!string.IsNullOrEmpty(skillsData.PhysicalSkills))
    {
        specializationRequest.Append($"* Мої фізичні навички:
{skillsData.PhysicalSkills}.");
    }
}

```

```
}
```

Після побудови запиту він надсилається на ендпоінт OpenAI API. Для цього використовується метод, наведений у лістингу 2.4, де HTTP-запит до API OpenAI містить підготовлені дані про добровольця.

Лістинг 2.4 — Метод для надсилання запиту штучному інтелекту

```
public string Ask(string chatRequest)
{
    var content = new StringContent(chatRequest, Encoding.UTF8, "application/json");
    HttpResponseMessage response =
    _chatGptClient.PostAsync(CHAT_POST_ADDRESS, content).Result;
    var responseString = response.Content.ReadAsStringAsync().Result;
    return responseString;
}
```

Для авторизації запиту використовуються токени доступу, які зберігаються у системі захищених конфігурацій (user-secrets). Приклади ініціалізації клієнта наведені у лістингу 2.5.

Лістинг 2.5 — Метод для ініціалізації HttpClient

```
private void InitHttpClient()
{
    _chatGptClient = new HttpClient();
    var token = _configuration.GetValue<string>(CHAT_GPT_TOKEN);
    _chatGptClient.DefaultRequestHeaders.Add("authorization", $"Bearer {token}");
}
```

Після надсилання запиту штучний інтелект генерує відповідь із рекомендацією щодо спеціалізації, яка передається на рівень представлення. У разі успішного виконання ендпоінт повертає статус «200 OK» разом із відповіддю.

Такий підхід до інтеграції OpenAI API дозволяє автоматизувати процес підбору військової спеціалізації, зробити його гнучким і ефективним, а також спростити взаємодію добровольців із інформаційною системою.

2.2.3 Розробка архітектури модулю інтеграції з штучним інтелектом

Інтеграція зі штучним інтелектом в інформаційній системі потребує чіткої структуризації та поділу функціональності на окремі компоненти. Зосередження

всієї логіки, такої як обробка запитів від клієнта, створення запиту до штучного інтелекту, відправлення цього запиту та обробка відповіді в одному методі було б порушенням принципів побудови гнучкої архітектури та ускладнювало б повторне використання коду. З метою дотримання принципу єдиної відповідальності (SOLID) було прийнято рішення розподілити функціональні можливості між різними класами-сервісами, кожен із яких відповідає лише за одну чітко визначену задачу.

Як видно з діаграми, кожен із класів відповідає за свою окрему функцію. Після надходження запиту до контролера на створення профілю добровольця чи аналізу його навичок, система передає інформацію до класу `ApplicationContext` для отримання необхідних даних про добровольця із бази даних.

Дані, отримані з бази, передаються в метод `BuildVolunteerAnalysisRequest` класу `AIRequestBuilder`, який відповідає за створення запиту до штучного інтелекту. Цей запит містить усю релевантну інформацію про добровольця: його особисті дані, військовий досвід, професійні навички, фізичний стан тощо.

Коли запит сформовано, контролер викликає метод `SendRequest` класу `AIIntegrationService`, який відповідає за надсилання запиту до API штучного інтелекту, обробку відповіді від OpenAI API та передачу її назад у систему.

Алгоритм роботи включає кілька ключових етапів:

- контролер отримує запит від клієнта та перевіряє, чи існує доброволець із зазначеним ідентифікатором у базі даних, якщо користувача не знайдено, повертається помилка 404 з відповідним описом;

- перевіряється, чи надані всі необхідні дані для аналізу, якщо інформація про добровольця неповна, система також повертає помилку 404, зазначаючи відсутність потрібної інформації;

- у разі успішної перевірки система передає дані до класу `AIRequestBuilder`, який формує запит для штучного інтелекту;

- сформований запит надсилається до API OpenAI за допомогою класу `AIIntegrationService`, де він обробляється штучним інтелектом;

— отримана відповідь повертається клієнту із статусом 200 OK.

Така архітектура забезпечує масштабованість, зручність у тестуванні та спрощує підтримку коду. Всі класи виконують строго визначені ролі, що мінімізує ризик появи помилок та полегшує майбутні модифікації системи.

3 РОЗРОБКА КЛІЄНТСЬКОЇ ТА СЕРВЕРНОЇ ЧАСТИНИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТА ПІДГОТОВКИ ДОБРОВОЛЬЦІВ ТЕРИТОРІАЛЬНОЇ ОБОРОНИ

Архітектура клієнтської частини інформаційної системи для обліку та підготовки добровольців територіальної оборони побудована на базі сучасних вебтехнологій, що забезпечують зручність, функціональність і масштабованість. Для реалізації цієї системи були використані PHP, HTML, CSS, MySQL, Bootstrap, JavaScript (зокрема бібліотека jQuery). Такий технологічний стек дозволяє створити динамічний і інтерактивний інтерфейс користувача, оптимізований для різних пристроїв та умов використання.

Основою клієнтської частини є вебінтерфейс, створений з використанням HTML для розмітки та CSS для стилізації. Використання Bootstrap як CSS-фреймворку дозволило забезпечити адаптивність інтерфейсу до різних екранів і мобільних пристроїв. Це особливо важливо для доступу до системи з планшетів і смартфонів у польових умовах. Структура інтерфейсу передбачає зручне меню навігації, форми для введення даних, таблиці для відображення списків добровольців і блоки для інтеграції навчальних матеріалів.

Динаміка та інтерактивність клієнтської частини забезпечуються за допомогою JavaScript та jQuery. Ці інструменти дозволяють реалізувати такі функції, як валідація форм, асинхронне завантаження даних через AJAX-запити та інтерактивні модулі, зокрема інтеграцію відеоуроків і тестів для оцінки підготовки добровольців. Завдяки асинхронним запитам клієнтська частина може отримувати або оновлювати дані без необхідності повного перезавантаження сторінки, що суттєво підвищує швидкодію системи.

Обробка даних у клієнтській частині передбачає інтеграцію з серверною частиною через PHP і MySQL. Клієнт надсилає запити на сервер, який обробляє їх за допомогою PHP-скриптів та зберігає або отримує дані з бази даних MySQL.

Наприклад, при реєстрації добровольця дані, введені у форму, проходять перевірку на клієнтському рівні за допомогою JavaScript, а потім передаються серверу для збереження. Це забезпечує як швидкість роботи, так і цілісність даних.

Для відображення великих обсягів інформації, таких як списки добровольців, використовуються таблиці з можливістю сортування та фільтрації. Ці функції реалізовані на JavaScript за допомогою бібліотек jQuery. Такі таблиці дозволяють швидко знаходити необхідні записи, наприклад, за регіоном чи рівнем підготовки. Крім того, таблиці інтегруються з серверною частиною, забезпечуючи динамічне оновлення вмісту при додаванні або редагуванні даних.

Особливу увагу приділено авторизації та безпеці. Клієнтська частина реалізує функції входу користувачів через захищені форми, які взаємодіють із сервером для перевірки облікових записів. PHP-скрипти виконують серверну перевірку автентичності, а дані передаються через захищений протокол HTTPS. Унікальні токени сесій запобігають несанкціонованому доступу до системи, а також забезпечують відповідність даних ролям користувачів.

Ще одним важливим компонентом клієнтської частини є модуль управління навчанням добровольців. Цей модуль дозволяє переглядати навчальні матеріали, проходити інтерактивні тести та оцінювати рівень підготовки. Інтерфейс розроблений таким чином, щоб бути інтуїтивно зрозумілим навіть для нових користувачів. Відео та текстові матеріали зберігаються на сервері, але завантажуються асинхронно через AJAX-запити, щоб уникнути перевантаження мережі.

Адаптивність і швидкодія клієнтської частини також досягаються завдяки оптимізації роботи з CSS і JavaScript. Використання мінімізації файлів CSS і JS зменшує час завантаження сторінки, а кешування даних на клієнті дозволяє прискорити повторний доступ до системи. Використання CDN для підключення зовнішніх бібліотек Bootstrap і jQuery підвищує доступність і надійність інтерфейсу.

У рамках тестування клієнтської частини проводилося відлагодження коду для усунення помилок, пов'язаних із відображенням і функціонуванням на різних браузерях та пристроях. Були враховані вимоги до роботи в умовах низької швидкості інтернет-з'єднання, зокрема через попереднє завантаження важливих компонентів.

Загалом клієнтська частина побудована як ефективна, інтерактивна і масштабована платформа для взаємодії з інформаційною системою. Її архітектура дозволяє швидко адаптувати систему до нових вимог, розширювати функціональність та інтегрувати додаткові модулі, забезпечуючи повний цикл управління обліком і підготовкою добровольців територіальної оборони.

Серверна частина інформаційної системи обліку та підготовки добровольців територіальної оборони побудована з використанням PHP та MySQL, що забезпечує ефективне зберігання і обробку даних. Архітектура серверної частини створена таким чином, щоб забезпечити швидку і надійну взаємодію з клієнтською частиною, реалізуючи запити від користувачів і обробляючи отримані дані в реальному часі. Серверна частина відповідає за управління базою даних, автентифікацію користувачів, виконання бізнес-логіки та забезпечення безпеки.

База даних MySQL використовується для зберігання всієї інформації, пов'язаної з добровольцями, включаючи їх персональні дані, інформацію про підготовку, статуси та результати тестів. Схема бази даних структурована так, щоб забезпечити логічне зберігання даних і мінімізувати дублювання. Всі таблиці пов'язані між собою через зовнішні ключі, що дозволяє підтримувати цілісність даних і легко виконувати складні запити. Для оптимізації продуктивності застосовуються індекси, які прискорюють доступ до ключових таблиць, наприклад, під час пошуку добровольців за регіоном або статусом.

PHP виступає основною мовою програмування для серверної частини, реалізуючи бізнес-логіку системи. Серверні скрипти забезпечують прийом запитів від клієнтської частини, їх обробку та генерацію відповідей. Ключовими

функціями PHP є авторизація користувачів, зберігання даних у базі, обробка запитів на зміну інформації, а також управління сесіями. Кожен запит від клієнта, наприклад, додавання нового добровольця або оновлення статусу, обробляється через контролери, які передають дані відповідним моделям бази даних.

Безпека серверної частини є одним із пріоритетів. Всі запити до сервера проходять перевірку автентичності через токени доступу. Використання підходів до підготовлених запитів (prepared statements) у PHP дозволяє уникнути SQL-ін'єкцій, забезпечуючи захист від несанкціонованого доступу до бази даних. Дані, які зберігаються у базі, шифруються, що підвищує рівень конфіденційності інформації, особливо у випадках зберігання персональних даних.

Сервер також обробляє асинхронні запити клієнтської частини через механізми AJAX. Це дозволяє забезпечити оновлення даних у режимі реального часу без повного перезавантаження сторінки. Наприклад, при зміні статусу добровольця відповідні зміни вносяться у базу, і клієнтська частина миттєво отримує оновлену інформацію. Це забезпечує плавний і швидкий користувацький досвід.

Окремо серверна частина відповідає за управління навчальними модулями системи. Вона забезпечує зберігання навчальних матеріалів, тестів та обробку результатів тестування. PHP-скрипти генерують унікальні запити для кожного користувача, враховуючи їх попередні результати, і зберігають нові результати в базі даних. Така логіка дозволяє ефективно відстежувати прогрес кожного добровольця.

Логіка серверної частини також реалізує аналітичні функції. Це включає генерацію звітів на основі зібраних даних про добровольців, їх активність у системі, статуси та результати підготовки. Звіти можуть бути представлені у вигляді таблиць або графіків, які інтегруються з клієнтською частиною. Ці дані допомагають адміністраторам системи відстежувати ефективність навчання та визначати ключові показники успішності.

Важливим аспектом є масштабованість серверної частини. Архітектура побудована так, щоб можна було легко інтегрувати нові модулі, додавати нові таблиці до бази даних та реалізовувати додаткові функції без шкоди для існуючої логіки. Це дозволяє системі відповідати на зростаючі потреби організації.

У підсумку серверна частина виконує роль основи всієї системи, забезпечуючи надійну взаємодію між клієнтським інтерфейсом та базою даних. Вона забезпечує виконання складних бізнес-операцій, гарантує безпеку та ефективність роботи, а також є гнучкою для майбутнього розвитку і розширення.

3.1 Реалізація клієнт–серверної архітектури інформаційної системи

Структура інформаційної системи обліку та підготовки добровольців територіальної оборони (рисунок 3.1) вміщає в себе дванадцять основних сторінок, кожна з яких має унікальні функціональні можливості, що змінюються залежно від ролі користувача. Ролі поділяються на добровольців, інструкторів, військову адміністрацію та адміністраторів, кожна з яких має власні повноваження та доступ до специфічних ресурсів системи.

Робота застосунку починається зі сторінки авторизації, де користувач може увійти до свого особистого кабінету, ввівши логін і пароль. Якщо обліковий запис існує, система перенаправляє користувача на сторінку двофакторної авторизації (2FA). У разі відсутності облікового запису є можливість перейти до реєстрації. Сторінка реєстрації дозволяє створити обліковий запис, ввівши особисті дані, як-от ПІБ, логін, пароль, а також обравши відповідну роль. Після успішної реєстрації користувач повертається до сторінки авторизації для входу в систему.

Двофакторна автентифікація забезпечується за допомогою сторінки, де нові користувачі отримують QR-код для додатку Google Authenticator. Додаток генерує одноразові паролі, які необхідно вводити разом із логіном і паролем для доступу до системи. Власне перевірка 2FA відбувається на окремій сторінці, де користувач вводить згенерований додатком код для підтвердження свого входу.

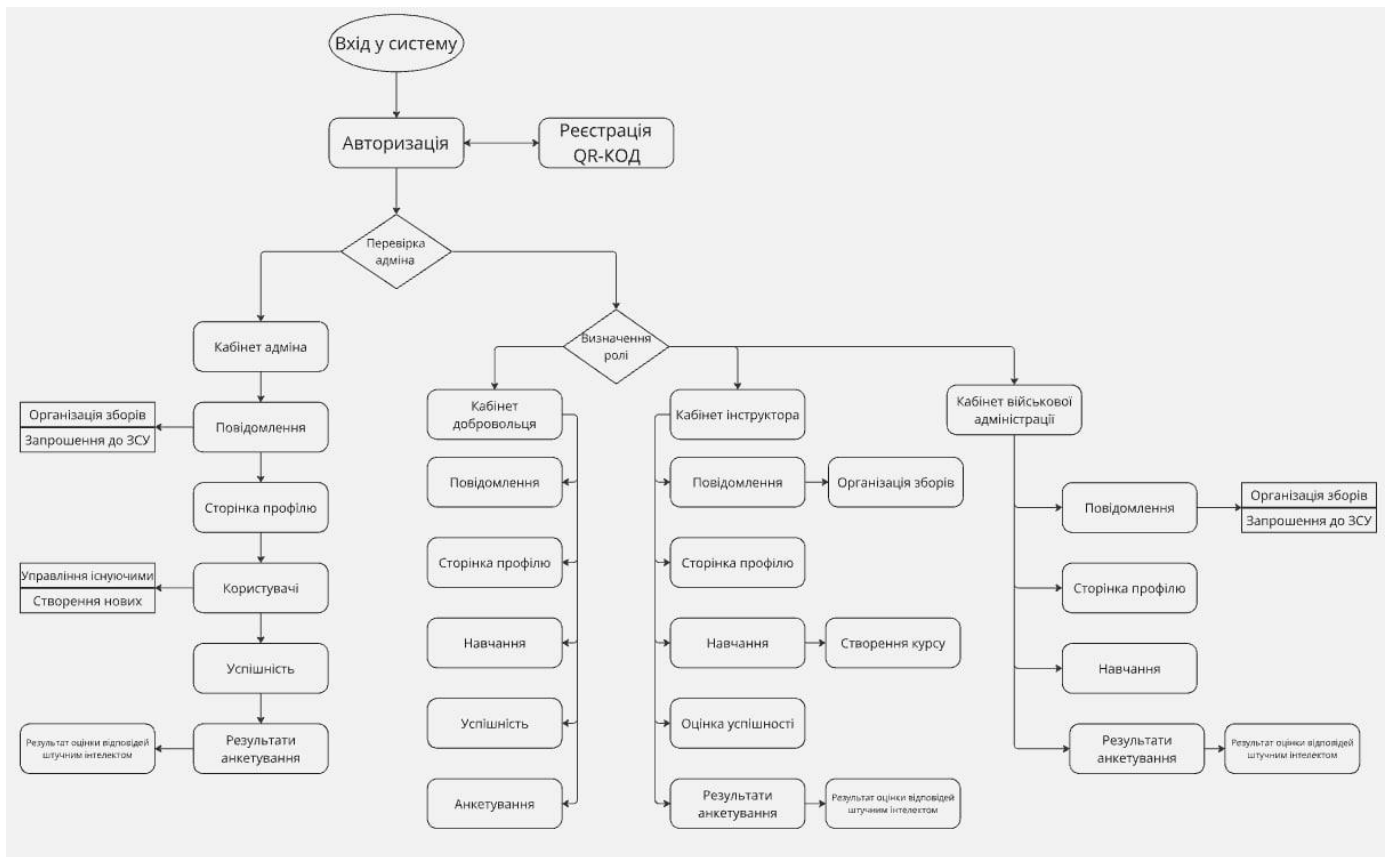


Рисунок 3.1 — Структура інформаційної системи обліку та підготовки добровольців територіальної оборони

Після успішної авторизації користувач потрапляє до кабінету, функціонал якого залежить від його ролі. Добровольці, інструктори, військова адміністрація та адміністратори можуть бачити схожі сторінки, але їх можливості значно відрізняються. Наприклад, на сторінці повідомлень добровольці отримують інформацію про військові збори від інструкторів, а також особисті повідомлення від військової адміністрації. У кабінеті також є доступ до профілю, де можна змінити фото, пароль, ПІБ, переглянути роль і підрозділ.

Також доброволець проходить анкетування де заповнює особисту інформацію про себе, таку як: загальна інформація (ріст, вага, освіту і тд.), дані про свою фізичну підготовку, а також проходить психологічний тест на готовність до військової служби.

Інструктор бачить результати анкетування, а також заключення штучного інтелекту, який проаналізував всю інформацію та відповіді на психологічний тест.

Навчання відбувається через окрему сторінку, де добровольці отримують матеріали відповідно до своєї спеціальності, а інструктори додають навчальні ресурси. Успішність відображається на іншій сторінці, де добровольці бачать оцінки, а інструктори їх виставляють на основі засвоєння матеріалу. Адміністрація та адміністратори мають доступ до перегляду цих даних і можуть приймати рішення про запрошення до ЗСУ.

Адміністратори також мають доступ до окремої сторінки для управління користувачами. Вона дозволяє змінювати ролі, підрозділи, видаляти або створювати нові облікові записи.

3.2 Реалізація API для взаємодії з ChatGPT

Розробка API для взаємодії з OpenAI на PHP є важливим завданням, що дозволяє інтегрувати потужні можливості штучного інтелекту, такі як генерація тексту або аналіз мовлення, у ваш веб-додаток. Цей процес передбачає кілька етапів: підготовку середовища, налаштування бібліотек, написання коду API, обробку запитів і відповідей, а також забезпечення безпеки та продуктивності.

Перш за все, необхідно створити обліковий запис на платформі OpenAI і отримати свій унікальний API-ключ. Цей ключ є основним інструментом для взаємодії з сервісом і його потрібно зберігати у конфігураціях, щоб уникнути витоку. Його можна помістити у .env файл, якщо використовується бібліотека vlucas/phpdotenv для керування середовищем:

```
OPENAI_API_KEY=your_openai_api_key
```

На сервері необхідно мати встановлений PHP версії 7.4 або вище, а також встановлений менеджер пакетів Composer для підключення сторонніх бібліотек. Використаємо Composer для встановлення бібліотеки HTTP-запитів, наприклад guzzlehttp/guzzle, оскільки вона спрощує роботу з REST API:

```
composer require guzzlehttp/guzzle
```

Наступним кроком є написання коду для взаємодії з OpenAI API. Починається все зі створення класу, який буде відповідати за відправку запитів.

```
<?php
require 'vendor/autoload.php';
use GuzzleHttp\Client;
class OpenAIClient {
    private $apiKey;
    private $httpClient;
    public function __construct($apiKey) {
        $this->apiKey = $apiKey;
        $this->httpClient = new Client(['base_uri' => 'https://api.openai.com/v1/']);
    }
    public function generateText($prompt, $maxTokens = 150) {
        try {
            $response = $this->httpClient->post('completions', [
                'headers' => [
                    'Authorization' => 'Bearer ' . $this->apiKey,
                    'Content-Type' => 'application/json',
                ],
                'json' => [
                    'model' => 'text-davinci-003',
                    'prompt' => $prompt,
                    'max_tokens' => $maxTokens,
                    'temperature' => 0.7,
                ],
            ]);
            $body = json_decode($response->getBody(), true);
            return $body['choices'][0]['text'] ?? "";
        } catch (\Exception $e) {
            return 'Error: ' . $e->getMessage();
        }
    }
}

$client = new OpenAIClient(getenv('OPENAI_API_KEY'));
echo $client->generateText("What is the capital of France?");
```

Цей клас OpenAIClient дозволяє вам надсилати запити до OpenAI API і отримувати відповіді. У прикладі показано, як створити HTTP POST-запит для виконання текстових завершень.

Далі, необхідно створити REST API для вашого сервісу, який обгортатиме запити до OpenAI. Використаємо фреймворк Slim для створення API. Код має такий вигляд:

```
<?php
require 'vendor/autoload.php';
use Psr\Http\Message\ResponseInterface as Response;
use Psr\Http\Message\ServerRequestInterface as Request;
use Slim\Factory\AppFactory;
$app = AppFactory::create();
$app->post('/generate', function (Request $request, Response $response) {
    $data = json_decode($request->getBody(), true);
    $prompt = $data['prompt'] ?? '';
    $client = new OpenAIClient(getenv('OPENAI_API_KEY'));
    $result = $client->generateText($prompt);
    $response->getBody()->write(json_encode(['response' => $result]));
    return $response->withHeader('Content-Type', 'application/json');
});
$app->run();
```

Тут /generate є маршрутом, який приймає запити з текстовим prompt у форматі JSON, звертається до OpenAI API і повертає результат.

Важливо обробляти помилки. Наприклад, якщо користувач надсилає порожній запит, API має повернути відповідне повідомлення. Для цього додамо валідацію:

```
if (empty($prompt)) {
    return $response->withStatus(400)->write(json_encode(['error' => 'Prompt is required']));
}
```

Щоб захистити API, додаємо автентифікацію за допомогою токенів. Передбачемо перевірку API-ключа у заголовках:

```
$apiKey = $request->getHeaderLine('Authorization');
if ($apiKey !== 'your_secure_api_key') {
    return $response->withStatus(403)->write(json_encode(['error' => 'Unauthorized']));
}
```

Важливим аспектом є логування запитів і відповіді для відслідковування роботи API та діагностики можливих проблем. Використаємо бібліотеку monolog/monolog.

Додатково, якщо планується значне навантаження, можна кешувати результати запитів за допомогою Redis або Memcached, щоб уникнути повторного звернення до OpenAI API для однакових запитів.

У фіналі API має бути задокументована для полегшення її використання іншими розробниками. OpenAPI Specification або Swagger можуть допомогти в цьому. Документація повинна включати інформацію про доступні маршрути, параметри запитів і приклади відповідей.

3.3 Створення системи авторизації та обробки даних користувачів

Створення системи авторизації та обробки даних користувачів на PHP передбачає кілька етапів: налаштування бази даних для зберігання інформації про користувачів, реалізацію реєстрації та входу, створення захищених сесій і обробку персональних даних. Така система повинна відповідати сучасним стандартам безпеки, зокрема захищати дані користувачів за допомогою хешування паролів і запобігати атакам типу SQL Injection.

На початку необхідно створити структуру бази даних. Для цього можна використати MySQL. Таблиця користувачів буде виглядати так:

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  password_hash VARCHAR(255) NOT NULL,
  email VARCHAR(100) NOT NULL UNIQUE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Паролі зберігаються у хешованому вигляді, для чого в PHP є функція `password_hash()`. Під час реєстрації новий користувач вводить ім'я користувача, пароль і електронну пошту, а система хешує пароль і зберігає дані у базу:

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
  $username = $_POST['username'];
  $password = $_POST['password'];
  $email = $_POST['email'];
  // Підключення до бази даних
```

```

$pdo = new PDO('mysql:host=localhost;dbname=mydatabase', 'root', 'password');
// Перевірка унікальності імені користувача
$stmt = $pdo->prepare('SELECT COUNT(*) FROM users WHERE username = ?');
$stmt->execute([$username]);
if ($stmt->fetchColumn() > 0) {
    echo "Ім'я користувача вже зайняте.";
    exit;
}
// Хешування пароля
$passwordHash = password_hash($password, PASSWORD_DEFAULT);
// Збереження даних користувача
$stmt = $pdo->prepare('INSERT INTO users (username, password_hash, email)
VALUES (?, ?, ?)');
$stmt->execute([$username, $passwordHash, $email]);

echo "Реєстрація успішна!";
}
?>

```

Для авторизації користувач вводить свої облікові дані (ім'я користувача і пароль), які перевіряються на відповідність записам у базі даних:

```

<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];
    // Підключення до бази
    $pdo = new PDO('mysql:host=localhost;dbname=mydatabase', 'root', 'password');
    // Пошук користувача
    $stmt = $pdo->prepare('SELECT id, password_hash FROM users WHERE username
= ?');
    $stmt->execute([$username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
    if ($user && password_verify($password, $user['password_hash'])) {
        // Ініціалізація сесії
        session_start();
        $_SESSION['user_id'] = $user['id'];
        echo "Вхід успішний!";
    } else {
        echo "Невірне ім'я користувача або пароль.";
    }
}
?>

```

Сесії використовуються для того, щоб зберігати стан авторизації користувача. Наприклад, після входу можна встановити ідентифікатор користувача у сесії, а потім перевіряти його на кожній сторінці, щоб визначити, чи авторизований користувач:

```
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header('Location: login.php');
    exit;
}
echo "Вітаємо, ви авторизовані!";
?>
```

Важливим аспектом є захист від SQL Injection. У цьому прикладі всі SQL-запити використовують підготовлені оператори, які знижують ризик атаки. Ще одна міра безпеки — регулярне оновлення хешу паролів за допомогою функції `password_needs_rehash()`.

Ця система на PHP є основою для створення надійного рішення, яке забезпечує авторизацію користувачів і захист їхніх даних.

3.4 Підключення та налаштування API для обробки даних користувачів

Підключення та налаштування API для обробки даних користувачів на PHP включає кілька ключових етапів, таких як створення точки входу для API, прийом запитів, обробка даних і відправка відповіді. Така система дозволяє зовнішнім програмам взаємодіяти із серверною частиною для отримання, збереження або обробки інформації про користувачів.

Спочатку створюється API-ендпоінт. Це файл на сервері, який обробляє запити, `api.php`. У цьому файлі реалізовано прийом HTTP-запитів і логіку обробки даних. У PHP для цього можна використовувати `$_GET`, `$_POST` або роботу з JSON через `php://input`. Зміст файла:

```
<?php
// Заголовки для API
header('Content-Type: application/json');
// Отримання даних запиту
```

```

$requestMethod = $_SERVER['REQUEST_METHOD'];
$input = json_decode(file_get_contents('php://input'), true);
// Підключення до бази даних
$pdo = new PDO('mysql:host=localhost;dbname=mydatabase', 'root', 'password');
// Обробка запиту залежно від методу
if ($requestMethod === 'GET') {
    // Отримання даних користувача
    if (isset($_GET['id'])) {
        $stmt = $pdo->prepare('SELECT * FROM users WHERE id = ?');
        $stmt->execute([$_GET['id']]);
        $user = $stmt->fetch(PDO::FETCH_ASSOC);
        echo json_encode($user);
    } else {
        // Отримання всіх користувачів
        $stmt = $pdo->query('SELECT * FROM users');
        $users = $stmt->fetchAll(PDO::FETCH_ASSOC);
        echo json_encode($users);
    }
} elseif ($requestMethod === 'POST') {
    // Створення нового користувача
    $username = $input['username'];
    $email = $input['email'];
    $passwordHash = password_hash($input['password'], PASSWORD_DEFAULT);
    $stmt = $pdo->prepare('INSERT INTO users (username, email, password_hash)
VALUES (?, ?, ?)');
    $stmt->execute([$username, $email, $passwordHash]);
    echo json_encode(['message' => 'User created successfully']);
} elseif ($requestMethod === 'PUT') {
    // Оновлення даних користувача
    $id = $input['id'];
    $username = $input['username'];
    $email = $input['email'];
    $stmt = $pdo->prepare('UPDATE users SET username = ?, email = ? WHERE id =
?');
    $stmt->execute([$username, $email, $id]);
    echo json_encode(['message' => 'User updated successfully']);
} elseif ($requestMethod === 'DELETE') {
    // Видалення користувача
    $id = $input['id'];
    $stmt = $pdo->prepare('DELETE FROM users WHERE id = ?');
    $stmt->execute([$id]);
    echo json_encode(['message' => 'User deleted successfully']);
} else {

```

```
// Непідтримуваний метод
http_response_code(405);
echo json_encode(['error' => 'Method Not Allowed']);
}
?>
```

Цей файл демонструє базову архітектуру API, яка підтримує методи GET, POST, PUT, і DELETE. У кожному випадку дані отримуються з запиту, перевіряються і обробляються.

При використанні API важливо дотримуватися стандартів безпеки. Один із ключових аспектів — аутентифікація та авторизація. Додамо механізм токенів, який перевіряє, чи має клієнт доступ до API:

```
// Перевірка токена
$headers = apache_request_headers();
if (!isset($headers['Authorization']) || $headers['Authorization'] !== 'Bearer your-api-token') {
    http_response_code(401);
    echo json_encode(['error' => 'Unauthorized']);
    exit;
}
```

Додатково створюємо спеціальну таблицю для журналювання запитів, щоб фіксувати дії користувачів і відслідковувати проблеми:

```
CREATE TABLE api_logs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    method VARCHAR(10),
    endpoint VARCHAR(255),
    request_data TEXT,
    response_data TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Важливо також обробляти помилки, використовуючи try-catch і створюючи користувацькі повідомлення:

```
try {
    $stmt = $pdo->prepare('INSERT INTO users (username, email, password_hash)
VALUES (?, ?, ?)');
    $stmt->execute([$username, $email, $passwordHash]);
    echo json_encode(['message' => 'User created successfully']);
}
```

```
} catch (PDOException $e) {  
    http_response_code(500);  
    echo json_encode(['error' => $e->getMessage()]);  
}
```

Результатом буде стабільна і безпечна API-система, яка дозволяє взаємодіяти із сервером і виконувати операції над даними користувачів. Цей підхід є основою для створення сучасних веб-додатків із динамічним обміном даними.

4 ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ З МОДУЛЕМ ШТУЧНОГО ІНТЕЛЕКТУ

4.1 Перевірка працездатності системи

Авторизуємось та пройдемо анкетування для добровольців.

На рисунку 4.1 вітальна карта. На цій карті користувача вітають та надають загальну інформацію про тест. Тут пояснюється призначення тестування, його значення та кінцева мета — підбір відповідної військової спеціальності або оцінка фізичного й психологічного стану добровольця. Ключовий елемент — кнопка "Розпочати", яка активує подальші блоки тестування.

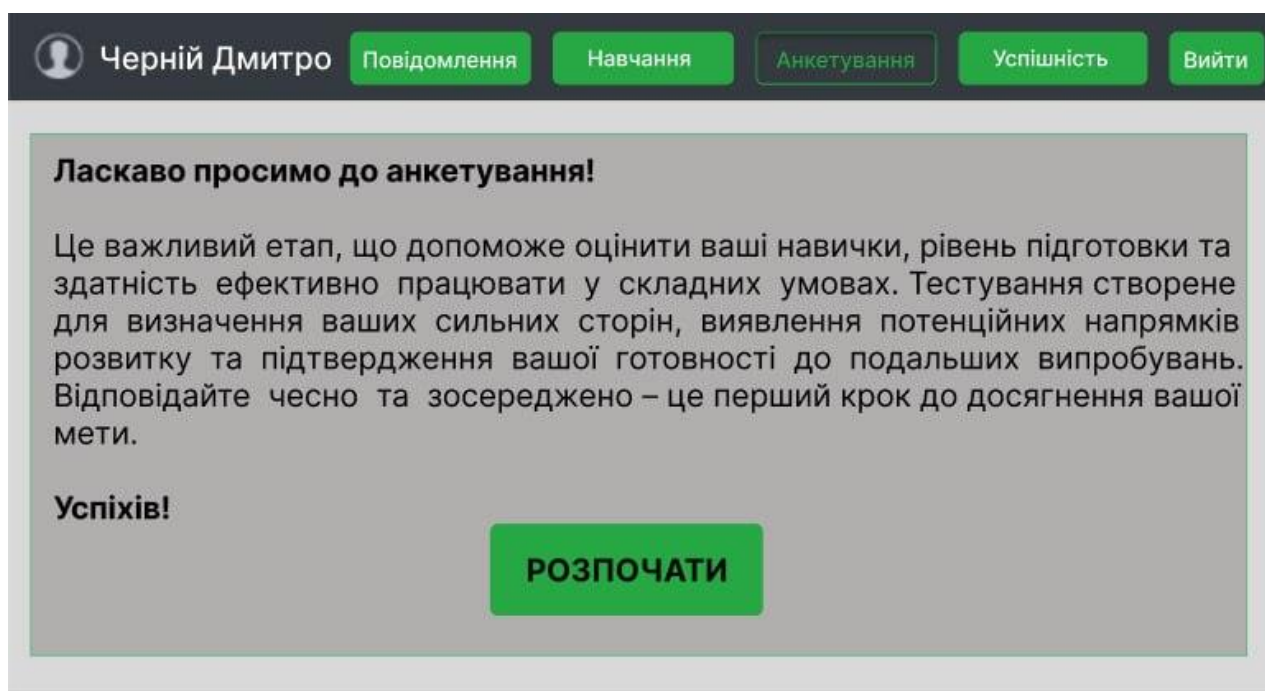


Рисунок 4.1 — Вітальний екран

Наступний екран зображено на рисунку 4.2. Користувач вводить особисті дані: зріст, вага, вік, базові фізичні характеристики, номер загальної військової спеціальності, розміри взуття, одягу та головного убору.

Задача цього екрану — збір базової інформації, яка згодом стане основою для формування індивідуальних рекомендацій.

Етап 1. Загальна інформація

Зріст:	170	Розмір взуття:	42
Вік :	22	Розмір одягу :	48
Сімейний стан:	Не одружений	Розмір шапки:	38
Номер ВОО:	186302		

Назад Далі

Рисунок 4.2 — Загальна інформація

Наступний екран зображено на рисунку 4.3. На цьому екрані користувач вводить інформацію про фізичний стан: кількість разів, які новобранець може присісти за певний проміжок часу, кількість підтягувань, результати бігу на 100 метрів (спринт) та 1000 метрів (марафон).

Задача цього екрану — оцінка фізичної витривалості, сили та швидкості добровольця.

Етап 2. Фізична підготовка

Кількість разів присідання за 1 хвилину:	50	Час бігу спринтом 100 метрів :	0 хв 14 с
Кількість разів підтягування за 1 хвилину:	20	Час бігу на витривалість 1000 метрів :	2 хв 30 с

Назад Далі

Рисунок 4.3 — Ввод інформації про фізичну підготовку

Для тесту було налаштовано 10 питань. Список питань буде наведено в лістингу в додатку Г. Приклад питання наведено на рисунку 4.4.

Рисунок 4.4 — Психологічний тест на визначення стану добровольця

При спробі пропустити питання користувачу буде відображено попередження, яке зображено на рисунку 4.5.



Рисунок 4.5 — Попередження, що потрібно обрати варіант відповіді

Після завершення всіх блоків тесту користувачу виводиться екран подяки (рисунок 4.6). Ключові елементи: текст про успішне завершення, кнопка «на головну».

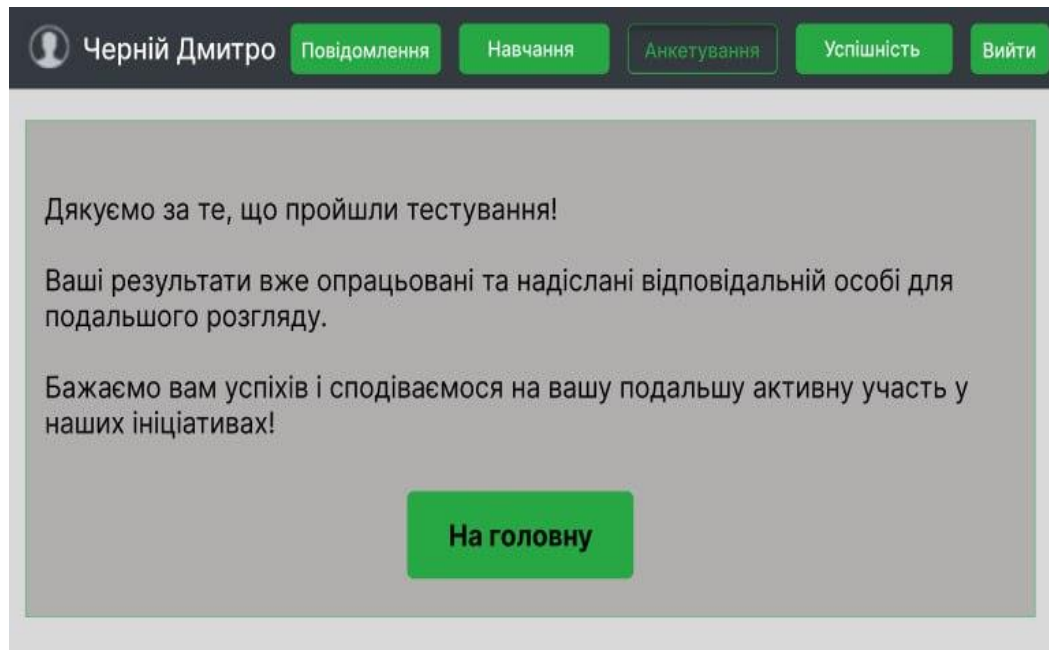


Рисунок 4.6 — Подяка за заповнення даних та проходження тесту

Після отримання усіх відповідей формується запит для ШІ та очікується відповідь. Результати анкетування та аналізу зберігаються в базі даних та доступні для перегляду лише особам з особливою роллю.

Цю роль може мати головний інструктор.

Для перегляду результатів інструктору або головному інструктору буде доступно спеціальну сторінку (рисунок 4.7).

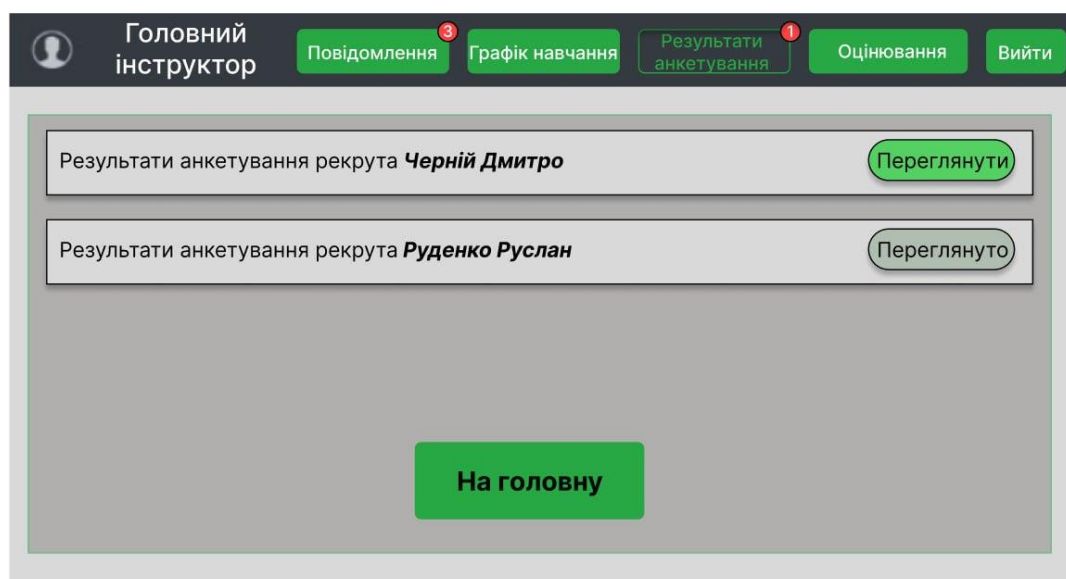


Рисунок 4.7 — Інструктор може переглянути результати тестування

При натисканні кнопки «переглянути» відкривається спеціальна форма для перегляду аналізу ШІ, висновку ШІ та зафіксованих відповідей (рисунок 4.8).

Головний інструктор Повідомлення Графік навчання Результати анкетування Оцінювання Вийти

Результати анкетування рекрута **Черній Дмитро** Відповіді

Аналіз відповідей:

Оцінюючи результати тесту, можна сказати, що моральний стан новобранця характеризується змішаними емоціями, певною напруженістю та нерівномірною здатністю до адаптації. Новобранець має сильні сторони, такі як здатність контролювати емоції в деяких ситуаціях (питання 2, 7) та зберігати спокій у певних обставинах (питання 8). Проте наявність тривожності, стресу та негативних думок (питання 4, 10) свідчить про емоційну вразливість і схильність до перевантаження.

Болісне сприйняття критики (питання 6) та відчуття труднощів із відпочинком (питання 9) вказують на високий рівень напруги, що може впливати на загальну продуктивність і здатність до швидкого відновлення. Новобранець демонструє готовність до ризиків і намагається діяти в складних ситуаціях (питання 3, 7), але потребує додаткової підтримки для подолання невпевненості та зменшення стресу.

Висновок:

Загалом, моральний стан новобранця можна охарактеризувати як нестабільний, із потенціалом для розвитку. Він здатний впоратися з певними викликами, але потребує підтримки та тренувань для підвищення впевненості, зменшення стресу та покращення стійкості в складних умовах.

Повернутись **На головну**

Рисунок 4.8 — Аналіз відповідей психологічного тесту за допомогою ШІ

У підсумку аналізу штучним інтелектом введених даних головному інструктору відображається загальний висновок по психологічним, фізичним та загальним вмінням. Також надаються рекомендації по розвитку бійця та спеціальності, які найвірогідніше йому підходять (рисунок 4.9).

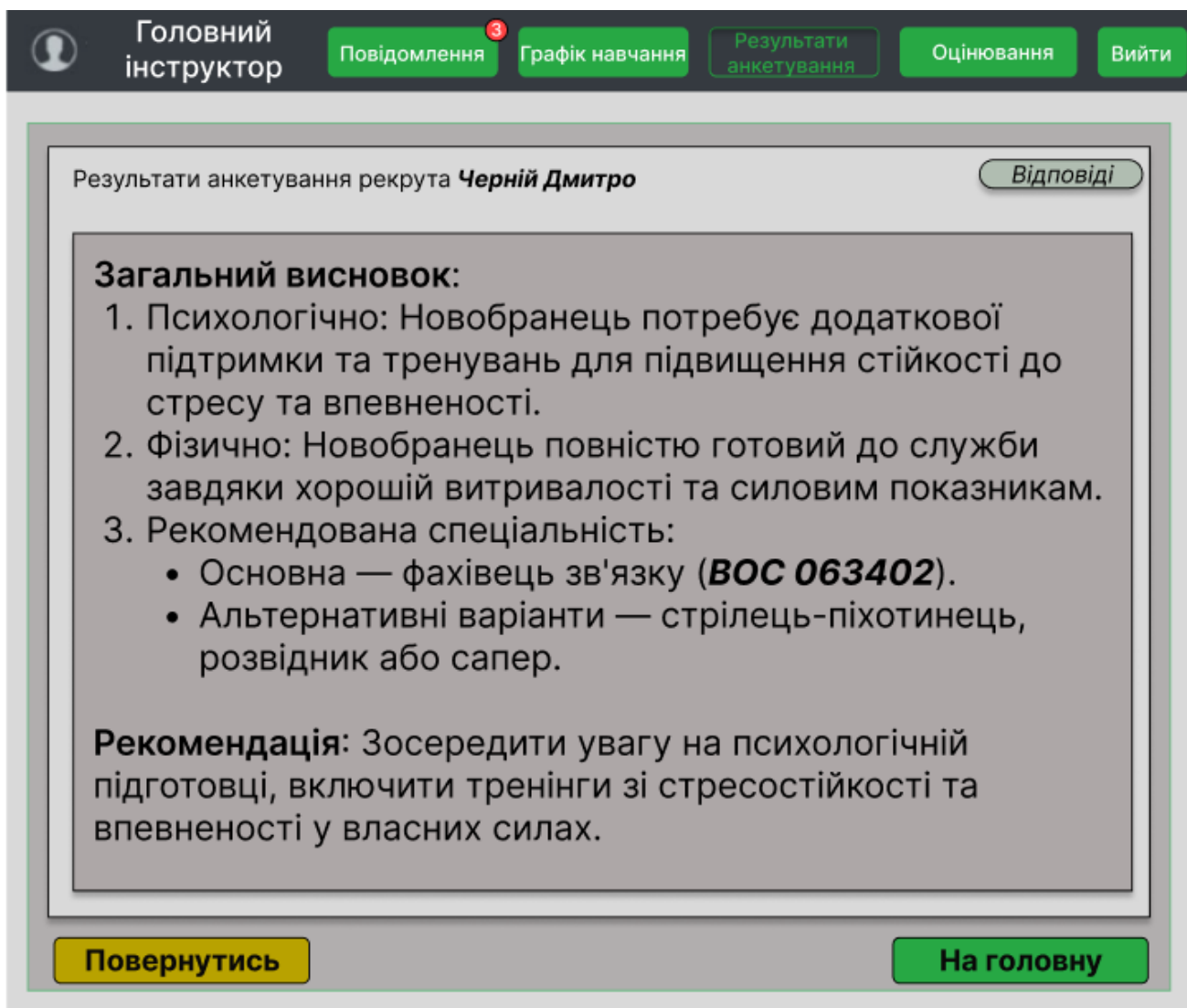


Рисунок 4.9 — Результат аналізу ІІІ даних, які ввів новобранець

4.2 Огляд аналогів розробленої системи

Центр рекрутингу української армії, представлений на сайті [17], є пілотним проектом Міністерства оборони України, спрямованим на створення прозорої та ефективної моделі залучення громадян на військову службу. Основна функція цього центру полягає в наданні інформації про вакансії у Збройних силах України та супроводі кандидатів на шляху до призначення на посаду у військовому підрозділі (рисунок 4.10).

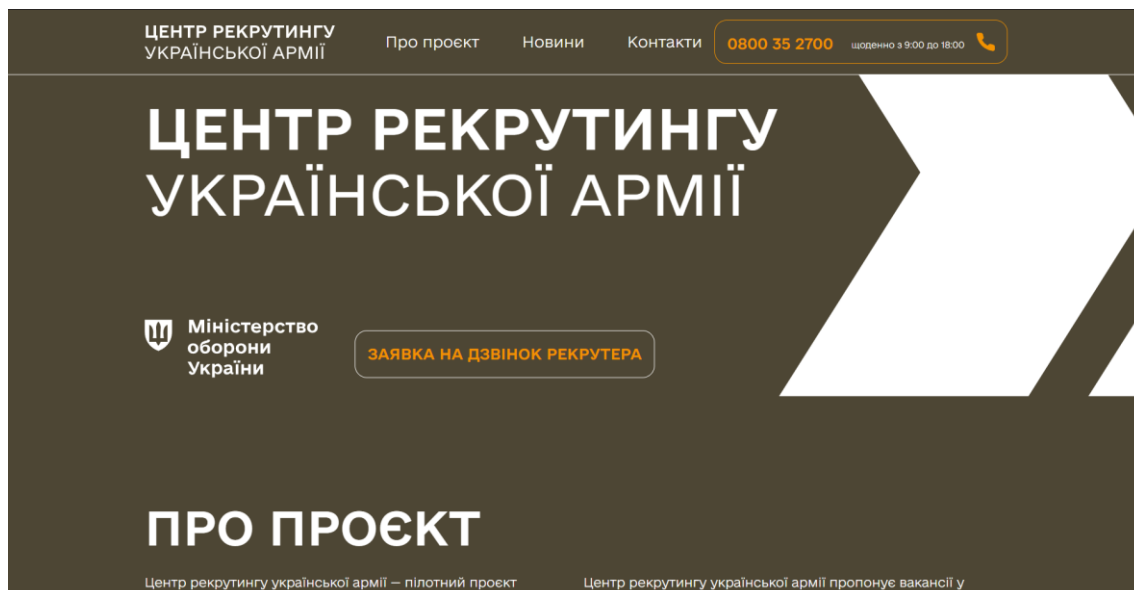


Рисунок 4.10 — Центр рекрутингу української армії

Дана інформаційна система є офіційною онлайн—платформою Міністерства оборони України, призначеною для залучення громадян до проходження військової служби у Збройних Силах України. Основна мета цього ресурсу — спростити процес рекрутингу, забезпечити прозорість та доступність інформації про можливості служби, а також сприяти формуванню ефективної кадрової політики у військових структурах.

Головними функціями платформи є інформування користувачів про поточні вакансії, умови служби, етапи відбору, а також надання інструкцій щодо подачі заявки та необхідних документів. Інтерфейс сайту є досить простим і зрозумілим, що дозволяє потенційним кандидатам швидко зорієнтуватися в доступних пропозиціях та почати процес подачі заявки.

Серед ключових переваг сайту варто відзначити його зручність у користуванні, доступність інформації та можливість звернення до спеціалістів для отримання консультації. Кандидати мають змогу дізнатися про вимоги до конкретних посад, ознайомитися з правилами проходження служби та заповнити анкету для участі у відборі.

Водночас функціональність сайту є обмеженою в контексті підтримки процесу підготовки рекрутів або забезпечення індивідуалізованого підходу до

навчання. Платформа зосереджена переважно на початкових етапах рекрутингу, таких як інформування та відбір, і не передбачає інтеграції з системами управління навчальним процесом або моніторингу компетенцій.

На відміну від цього, розроблена інформаційна система обліку та підготовки добровольців територіальної оборони має ширший спектр функціональних можливостей, що забезпечують більш комплексний підхід до управління процесом набору та підготовки добровольців.

По-перше, система забезпечує повний цикл управління даними добровольців, включаючи їх реєстрацію, облік особистих даних, відстеження прогресу в навчанні та підготовці, а також управління розкладом тренувань і заходів. Це дозволяє автоматизувати багато аспектів адміністрування та зменшити навантаження на персонал, відповідальний за організацію підготовки.

По-друге, система інтегрує модуль для оцінки компетенцій та навичок добровольців, що дає змогу індивідуально підходити до планування їх підготовки, враховуючи початковий рівень знань та вмінь. Це сприяє підвищенню ефективності навчального процесу та кращій адаптації добровольців до виконання завдань територіальної оборони.

По-третє, система надає можливість комунікації між добровольцями та інструкторами через вбудовані засоби обміну повідомленнями, що покращує координацію та оперативність у вирішенні організаційних питань. Це особливо важливо в умовах, коли швидкість реагування та злагодженість дій є критичними для успішного виконання завдань.

Крім того, система містить аналітичні інструменти для моніторингу та оцінки ефективності підготовки, що дозволяє керівництву приймати обґрунтовані рішення щодо вдосконалення навчальних програм та оптимального розподілу ресурсів. Це забезпечує постійне підвищення якості підготовки та готовності добровольців до виконання своїх обов'язків.

Таким чином, інформаційна система обліку та підготовки добровольців територіальної оборони не лише виконує функції, подібні до Центру рекрутингу

української армії, але й розширює їх, пропонуючи інтегровані рішення для управління, підготовки та комунікації з добровольцями, що в цілому підвищує ефективність та оперативність процесів, пов'язаних з територіальною обороною.

Другим аналогом інформаційної системи є платформа представлена на рисунку 4.11, яка створена для пошуку кадрів, здебільшого націленою на залучення фахівців до роботи в оборонній сфері[18]. Основною його метою є забезпечення зв'язку між роботодавцями, які шукають працівників для роботи в оборонних структурах або суміжних галузях, та потенційними кандидатами. Платформа пропонує користувачам можливість ознайомлення з актуальними вакансіями, надає базову інформацію про вимоги до кандидатів, а також спрощує подачу заявок на участь у конкурсному відборі.

Функціонал сайту орієнтований переважно на ринок праці, пов'язаний із військовою та оборонною діяльністю. Основний акцент зроблено на зручності подачі заявок, доступі до вакансій та швидкому пошуку роботи в оборонному секторі. Інтерфейс є адаптивним, що дозволяє користувачам швидко знайти необхідну інформацію про вакансії, умови праці та способи подачі заявок. Окрім цього, платформа має певну маркетингову привабливість завдяки інтеграції з популярними пошуковими системами, що сприяє залученню більшої кількості кандидатів.

Однак функціональні можливості сайту обмежуються переважно рекрутинговим напрямком і не охоплюють комплексних завдань, які стосуються підготовки, навчання або адаптації добровольців. Платформа не забезпечує інструментів для оцінювання фізичних, психологічних чи професійних характеристик кандидатів, які є важливими для ефективного комплектування підрозділів територіальної оборони. Крім того, система не передбачає інтеграції з модулями для проведення тренувань, відстеження прогресу підготовки добровольців чи аналізу їхньої відповідності військовим спеціальностям.

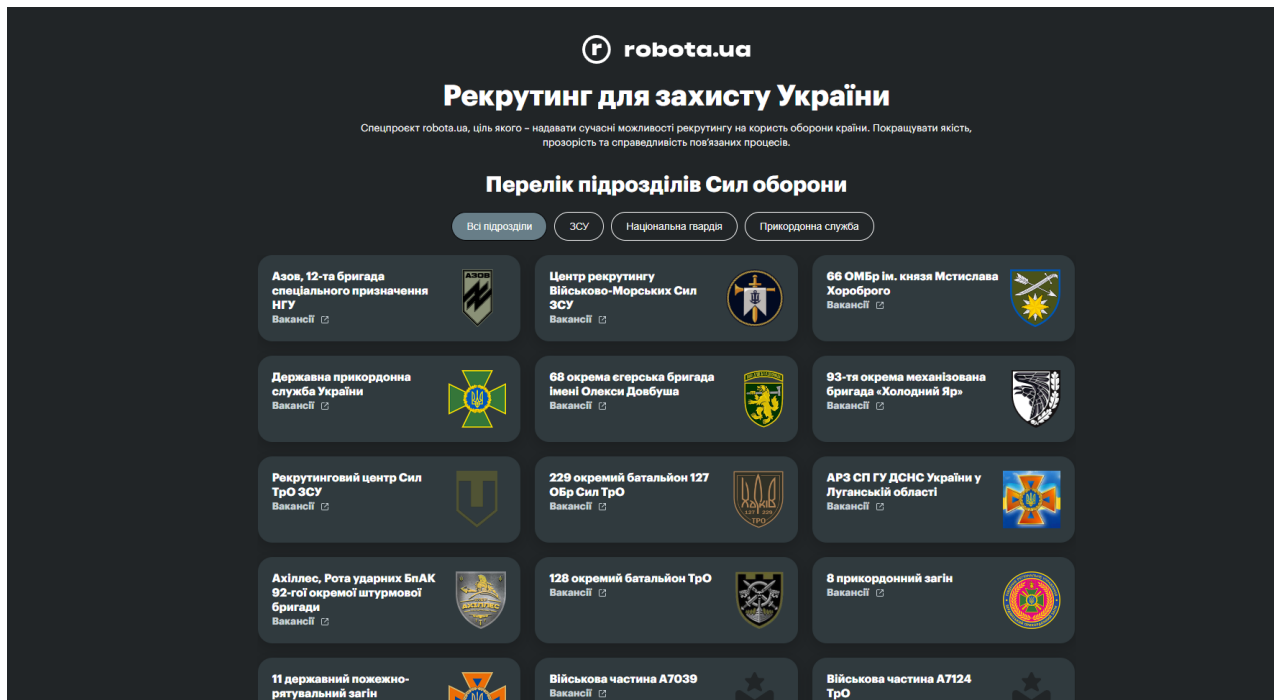


Рисунок 4.11 — Платформа рекрутингу для захисту України

Розроблена інформаційна система обліку та підготовки добровольців територіальної оборони значно перевершує функціонал цього сайту у кількох ключових аспектах. По-перше, вона має комплексний підхід до управління добровольцями, включаючи не лише процес рекрутингу, а й подальшу підготовку та розподіл кандидатів за військовими спеціальностями. Система інтегрує модулі для збору, аналізу та обробки даних, що дозволяє автоматизувати процес підбору кадрів, виходячи з їхніх фізичних, психологічних і професійних характеристик.

Значною перевагою є можливість використання інструментів штучного інтелекту, які дозволяють не тільки здійснювати аналіз відповідності добровольців конкретним військовим спеціальностям, а й формувати індивідуальні плани підготовки. Це забезпечує якісну і точну оцінку кожного кандидата, що особливо важливо для ефективного функціонування підрозділів територіальної оборони. Інформаційна система також включає засоби для інтерактивного навчання, модулі для контролю за виконанням тренувальних програм та оцінки результатів підготовки добровольців.

Крім цього, розроблена система здатна надавати індивідуалізовані рекомендації щодо подальшого навчання чи службового призначення, базуючись на аналізі даних про кожного кандидата. Це дозволяє мінімізувати людський фактор і забезпечити прозорість процесу підготовки, що не може запропонувати згаданий сайт, який має лише базові функції пошуку вакансій і подачі заявок.

Ще однією суттєвою відмінністю є впровадження механізмів для довготривалого моніторингу та підтримки добровольців після їх включення до лав територіальної оборони. Система передбачає регулярне оновлення інформації про їхній стан підготовки, можливості професійного зростання та потреби в додаткових тренуваннях. Це створює умови для постійного вдосконалення кадрового складу та підвищення бойової готовності підрозділів.

Таким чином, інформаційна система обліку та підготовки добровольців територіальної оборони значно перевершує можливості існуючого сайту за рівнем функціональності, адаптивності та інтеграції сучасних технологій. Вона орієнтована на комплексне вирішення завдань, пов'язаних із рекрутингом, підготовкою та розвитком добровольців, що робить її ефективним інструментом для підтримки діяльності територіальної оборони України.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково—технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку інформаційної системи обліку та підготовки добровольців територіальної оборони.

Особливістю розробки є вдосконалення методу інтеграції інформаційної системи із штучним інтелектом, в якому, на відміну від існуючих, реалізовано побудову та надсилання запитів, створених на основі користувацької інформації, до штучного інтелекту з використанням його API, що дозволило розширити функціональні можливості та збільшити швидкодію обробки запитів штучним інтелектом.

Аналогом може SAP SuccessFactors за ціною 4000 \$/рік або 160000 грн/рік.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів.

Оцінювання науково—технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження табл. 5.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до-рівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	матеріали для реалізації ідеї відомі та використов. виробництві

Продовження табл. 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	3
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	3	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	3	3	3
Сума	43	40	42
Середньоарифметична сума балів	$(43+40+42) / 3 = 41,37$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків	Рівень комерційного потенціалу розробки
0 - 10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок використання ІІІ дозволяє автоматизувати процеси оцінювання фізичних, психологічних, освітніх та професійних характеристик добровольців, забезпечуючи точний підбір військових спеціальностей.

5.2 Прогнозування витрат на виконання науково—дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де М — місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p — число робочих днів за місяць, 24 днів;

t — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного

програмного продукту.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	35500	1479,17	38	56208,333
Програміст	32000	1333,33	38	50666,667
Всього				106875,00

Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 12,5 % від основної заробітної плати розробників та робітників[19]:

$$З_д = З_о \cdot 12,5 \% / 100 \% \quad (5.2)$$

$$З_д = (106875,00 \cdot 12,5 \% / 100 \%) = 13359,38 \text{ (грн.)}$$

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22 \% / 100\% \quad (5.3)$$

$$Н_з = (106875,00 + 13359,38) \cdot 22 \% / 100 \% = 26451,56 \text{ (грн.)}$$

Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, що використовувалось для розробки в

спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T_{\theta}} \cdot \frac{t_{\text{вик}}}{12} \text{ [грн.]}. \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

T — термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ — термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 26999 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,58 міс.

$$A_{\text{обл}} = \frac{26999}{2} \times \frac{1,58}{12} = 1781,18 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення.

Розрахунки заносимо до таблиці 5.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{\text{нем.ак.}} = 11000 + 1000 = 12000$ грн.

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас).

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	26999	2	1,58	1781,184
Офісне обладнання (меблі)	25000	4	1,58	824,653
Приміщення	950000	20	1,58	6267,361
Всього				8873,20

Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства з ПДВ в 2024 році для Вінницької області за даними Енера-Вінниця, $V = 10,42$ грн./кВт;

P — встановлена потужність обладнання, кВт. $P = 0,25$ кВт;

Φ — фактична кількість годин роботи обладнання, годин.

K_{Π} — коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,25 \cdot 8 \cdot 38 \cdot 10,42 = 712,728 \text{ (грн.)}$$

Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_{\epsilon} = (3_o + 3_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.6)$$

де H_{ib} — норма нарахування за статтею «Інші витрати».

$$I_{\epsilon} = 106875,00 * 65\% / 100\% = 69468,75 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де $H_{нзв}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 106875,00 * 125 \% / 100 \% = 133594 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 106875,00 + 13359,38 + 26451,56 + 8873,20 + 12000 + 712,73 + 69468,75 + \\ + 133594 = 371334,36 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 371334,36 / 0,5 = 742669 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно[20]:

- вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.9)$$

де $\pm\Delta\Pi_0$ — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 — вартість програмного продукту у році до впровадження результатів розробки;

ΔN — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ — коефіцієнт, який враховує рентабельність продукту;

ϑ — ставка податку на прибуток, у 2024 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 65000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 3000 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 150 шт., протягом другого року – на 130 шт., протягом третього року на 100 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0 \cdot 3000 + (65000 + 3000) \cdot 150) \cdot 0,8333 \cdot 0,35 \cdot (1 - 0,18) = 2331874,907 \text{ грн.}$$

$$\Delta\Pi_2 = (0 \cdot 3000 + (65000 + 3000) \cdot (150 + 130)) \cdot 0,8333 \cdot 0,35 \cdot (1 - 0,18) = 4553733,151 \text{ грн.}$$

$$\Delta\Pi_3 = (0*3000 + (65000 + 3000) * (150 + 130 + 100) * 0,8333 * 0,35) * (1 - 0,18) = 6180066,419 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 13065674,48 грн.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\Pi\Pi = (2331874,907 / (1 + 0,1)^1) + (4553733,151 / (1 + 0,1)^2) + (6180066,419 / (1 + 0,1)^3) = 2119886,28 + 3763415,827 + 4643175,371 = 10526477,48 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{inv} * 3B, \quad (5.11)$$

де k_{inv} – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{inv}=2...5$, але може бути і більшим;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 742669 = 1485337,45 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV, \quad (5.12)$$

$$E_{abc} = 10526477,48 - 1485337,45 = 9041140,02 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно[21].

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_{ϵ} . Для цього використаємо формулу:

$$E_{\epsilon} = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.13)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_{\epsilon} = \sqrt[3]{(1 + 9041140,02/1485337,45) - 1} = 0,921$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_{\epsilon} > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_{\epsilon}}, \quad (5.15)$$

$$T_{ок} = 1 / 0,921 = 1,09 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 1,09 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 742669 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 1,09 роки.

ВИСНОВКИ

Проведений аналіз сучасних методів розробки інформаційних систем дозволив виявити найактуальніші технології та підходи, які відповідають потребам автоматизації та оптимізації процесів обліку й підготовки добровольців. Окремо розглянуто можливості використання штучного інтелекту для аналізу даних, автоматизації підбору військових спеціальностей, а також персоналізованих рекомендацій. Це стало теоретичною основою для вибору інструментів та підходів, що були застосовані у розробці інформаційної системи.

Удосконалено метод інтеграції штучного інтелекту із інформаційною системою. На основі обраного підходу створено механізм, який дозволяє ефективно використовувати API ChatGPT для взаємодії користувачів із системою. Це включає збирання та обробку вхідних даних, генерацію персоналізованих рекомендацій щодо військових спеціальностей і передачу результатів користувачам у зручному форматі. Такий підхід дозволив підвищити точність і швидкість роботи системи.

Розроблено структуру інформаційної системи. Це включає серверну та клієнтську частини, базу даних для зберігання інформації про добровольців, а також окремий модуль інтеграції зі штучним інтелектом. Структура побудована відповідно до принципів SOLID, що забезпечує легкість розширення функціоналу в майбутньому.

Виконано програмну реалізацію клієнтської та серверної частин інформаційної системи. Клієнтська частина побудована на сучасному фреймворку, що забезпечує високу продуктивність і зручність взаємодії користувачів із системою. Серверна частина реалізована з використанням надійних технологій, які забезпечують швидкість обробки даних та інтеграцію з іншими компонентами.

Реалізовано API для взаємодії з ChatGPT. Це API дозволяє користувачам вводити необхідну інформацію про себе, яка обробляється системою і передається

до ChatGPT для генерації рекомендацій щодо відповідних військових спеціальностей.

Проведено тестування інформаційної системи. У процесі тестування перевірено роботу функціонування API з інтеграцію з ChatGPT. Результати тестування підтвердили коректність роботи системи, її високу продуктивність та відповідність заданим функціональним і технічним вимогам.

Таким чином всі поставлені завдання виконано, а поставлена мета роботи щодо розширення функціональних можливостей інформаційної системи завдяки інтеграції з штучним інтелектом досягнута.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Інформаційна система обліку та підготовки добровольців територіальної оборони/ Д.В. Черній, О.С. Городецька, Н.В. Добровольська. Матеріали Молодь в науці: дослідження, проблеми, перспективи (МН-2025), 2024 р. [Електронний ресурс] — Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/22476/18595>
2. Поняття добровольчого формування територіальної громади [Електронний ресурс] // Pravo — Режим доступу до ресурсу: <https://pravo-ua.com/dftg/pravovyj-status-dobrovolczya-dobrovolchogo-formuvannya-terytorialnoyi-gromady/>.
3. Інформаційні системи [Електронний ресурс] // Step — Режим доступу до ресурсу: <https://step.org.ua/konspekt/inform/tema1>
4. Архітектура ПО [Електронний ресурс] // Wezom — Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/arhitektura-programmnogo-obespecheniya>.
5. Клієнт-серверна архітектура [Електронний ресурс] // Qatestlab — Режим доступу до ресурсу: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>.
6. Apache HTTP Server [Електронний ресурс] // Wikipedia — Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Apache_HTTP_Server#:~:text=Web-сервер%20Apache%20є%20самостійним,які%20розширюють%20основні%20функціональні%20можливості.
7. PHP [Електронний ресурс] // Wikipedia — Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/PHP>.
8. OpenAI [Електронний ресурс] // OpenAI — Режим доступу до ресурсу: <https://openai.com/api/>
9. Platform OpenAI [Електронний ресурс] // Platform — Режим доступу до ресурсу: <https://platform.openai.com/docs/models>.

10. API Reference [Електронний ресурс] // Gptengineer — Режим доступу до ресурсу: https://gpt-engineer.readthedocs.io/en/latest/api_reference.html.
11. What is Java? [Електронний ресурс] // Amazon — Режим доступу до ресурсу: https://aws.amazon.com/what-is/java/?nc1=h_ls.
12. Типи сучасних баз даних [Електронний ресурс] // Senior — Режим доступу до ресурсу: <https://senior.ua/articles/11-tipv-suchasnih-baz-danih-korotkiy-opis-shemi--prikladi-bd>.
13. Що таке MySQL? [Електронний ресурс] // Freehost — Режим доступу до ресурсу: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-mysql/>.
14. JavaScript [Електронний ресурс] // Developer.Mozilla — Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
15. jQuery [Електронний ресурс] // Builtin — Режим доступу до ресурсу: <https://builtin.com/software-engineering-perspectives/jquery>.
16. Bootstrap [Електронний ресурс] // Getbootstrap — Режим доступу до ресурсу: <https://getbootstrap.com>.
17. Центр рекрутингу Української армії [Електронний ресурс] // Recruiting — Режим доступу до ресурсу: <https://recruiting.mod.gov.ua>
18. Рекрутинг для захисту України [Електронний ресурс] // Defence — Режим доступу до ресурсу: <https://defence.robota.ua>
19. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. — Вінниця : ВНТУ, 2021. — 42 с.
20. Витрати на оплату праці як складова собівартості продукту [Електронний ресурс] — Режим доступу до ресурсу: https://osvita.ua/vnz/reports/econom_pidpr/21918/
21. Терміни амортизації основних засобів у податковому обліку [Електронний ресурс] — Режим доступу до ресурсу: <https://if.tax.gov.ua/media-ark/news-ark/437911.html>

ДОДАТОК А**Технічне завдання**

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. О.Д. Азаров

“18” вересня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«Інформаційна система обліку та підготовки добровольців територіальної
оборони»

08–54.МКР.021.00.000 ПЗ

Науковий керівник к.т.н., доц., каф. ОТ

Городецька О.С.

Магістрант групи 1КІ–23м

Черній Д.В.

Вінниця 2024

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

1.1 У нинішніх умовах загроз національній безпеці виникає гостра потреба у створенні дієвих організаційно-управлінських рішень для реєстрації та підготовки добровольців до захисту країни. Сучасні інформаційні системи повинні забезпечувати автоматизацію збору, обробки та аналізу даних про добровольців, а також надавати персоналізовані рекомендації на основі цих даних.

1.2 Підставою для розробки даної магістерської кваліфікаційної роботи є наказ ВНТУ №_310_ від «17» 09 2024 року та рішення засідання кафедри обчислювальної техніки (протокол №__ від «____»__ 2024 року).

2 Мета і призначення МКР

2.1 Мета роботи — розширення функціональних можливостей інформаційної системи завдяки інтеграції з штучним інтелектом, що дасть можливість збільшити швидкодію обробки запитів штучним інтелектом.

2.2 Призначення розробки — інформаційна система обліку та підготовки добровольців територіальної оборони створена для збору, обробки та аналізу даних про добровольців, а також забезпечення ефективної організації підбору кандидатів на основі їхніх фізичних, психологічних і професійних характеристик. Система сприяє інтеграції сучасних технологій штучного інтелекту для персоналізованого підходу у визначенні військових спеціальностей і розробці індивідуальних планів підготовки.

3 Вихідні дані для виконання МКР

3.1 Проведення аналізу існуючих принципів оцінки психологічного стану добровольців

3.2 Проведення аналізу технологій для реалізації серверної та клієнтської частини.

3.3 Аналіз та вдосконалення методу інтеграції штучного інтелекту з системою за допомогою OpenAI API. Це дозволило реалізувати модуль для автоматизованого аналізу даних про кандидатів, формування персоналізованих рекомендацій щодо військових спеціальностей, а також створення ефективних інструментів навчання та підготовки.

3.4 Проведення реалізацію клієнтської та серверної частин системи. Клієнтська частина надає зручний інтерфейс для користувачів, що дозволяє швидко вводити дані та отримувати результати. Серверна частина забезпечує швидку та надійну обробку запитів, взаємодію з базою даних і модулем штучного інтелекту.

3.5 Проведення тестування розробленої системи. У процесі тестування перевірено працездатність клієнтської та серверної частин, функціонування API для взаємодії з ChatGPT, а також точність і швидкість обробки даних. Результати тестування підтвердили відповідність системи поставленим завданням і її готовність до використання в реальних умовах.

4 Вимоги до виконання МКР

Головна вимога — використовувати методи обробки даних із застосуванням алгоритмів штучного інтелекту для автоматизованого аналізу характеристик добровольців та їхнього підбору до військових спеціальностей. Система повинна забезпечувати ефективну інтеграцію даних, отриманих від користувачів, з аналітичними модулями для генерації персоналізованих рекомендацій та створення індивідуальних планів підготовки.

Реалізовані алгоритми мають враховувати багатофакторний підхід до аналізу, включаючи фізичні, психологічні та професійні характеристики добровольців. Це забезпечує точність та адаптивність рекомендацій для ефективного розподілу ресурсів і часу на навчання.

Усі розробки повинні відповідати сучасним вимогам стабільності роботи та забезпечувати інтеграцію з основними компонентами інформаційної системи,

включаючи API для взаємодії з ChatGPT.

5. Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	2	3	4	5
1	Огляд і аналіз джерел інформації	20.09.2024	04.10.2024	Розділ 1
2	Теоретичні дослідження можливостей інтеграції штучного інтелекту для оцінки і підбору військових спеціальностей	05.10.2024	18.10.2024	Розділ 2
3	Розробка клієнтської та серверної частини інформаційної системи, інтеграція API для взаємодії з ChatGPT	19.10.2024	01.11.2024	Розділ 3
4	Тестування інформаційної системи з модулем штучного інтелекту	02.11.2024	16.11.2024	Розділ 4
5	Економічна частина	17.11.2024	25.11.2024	Розділ 5

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації

до МКР українською та іноземною мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора

8. Вимоги до оформлювання та порядок виконання МКР

8.1 При оформлювання МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— міждержавний ГОСТ 2.104—2006 «Єдина система конструкторської документації. Основні написи»;

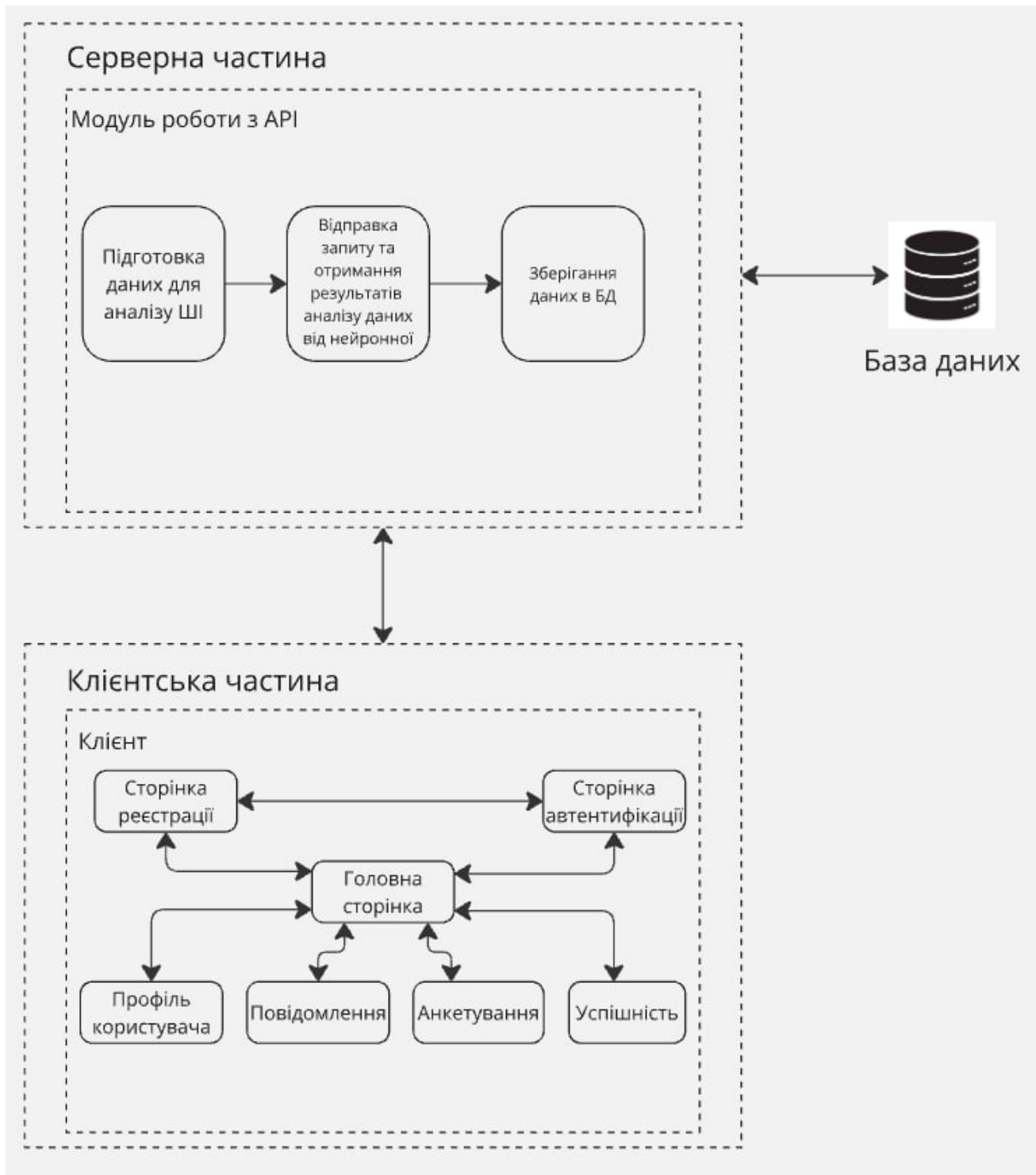
— Методичні вказівки до виконання бакалаврських кваліфікаційних робіт зі спеціальності 123 «Комп'ютерна інженерія» (освітня програма «Комп'ютерна інженерія»). Кафедра обчислювальної техніки ВНТУ 2022;

— документами на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ—03.02.02—П.001.01:21».

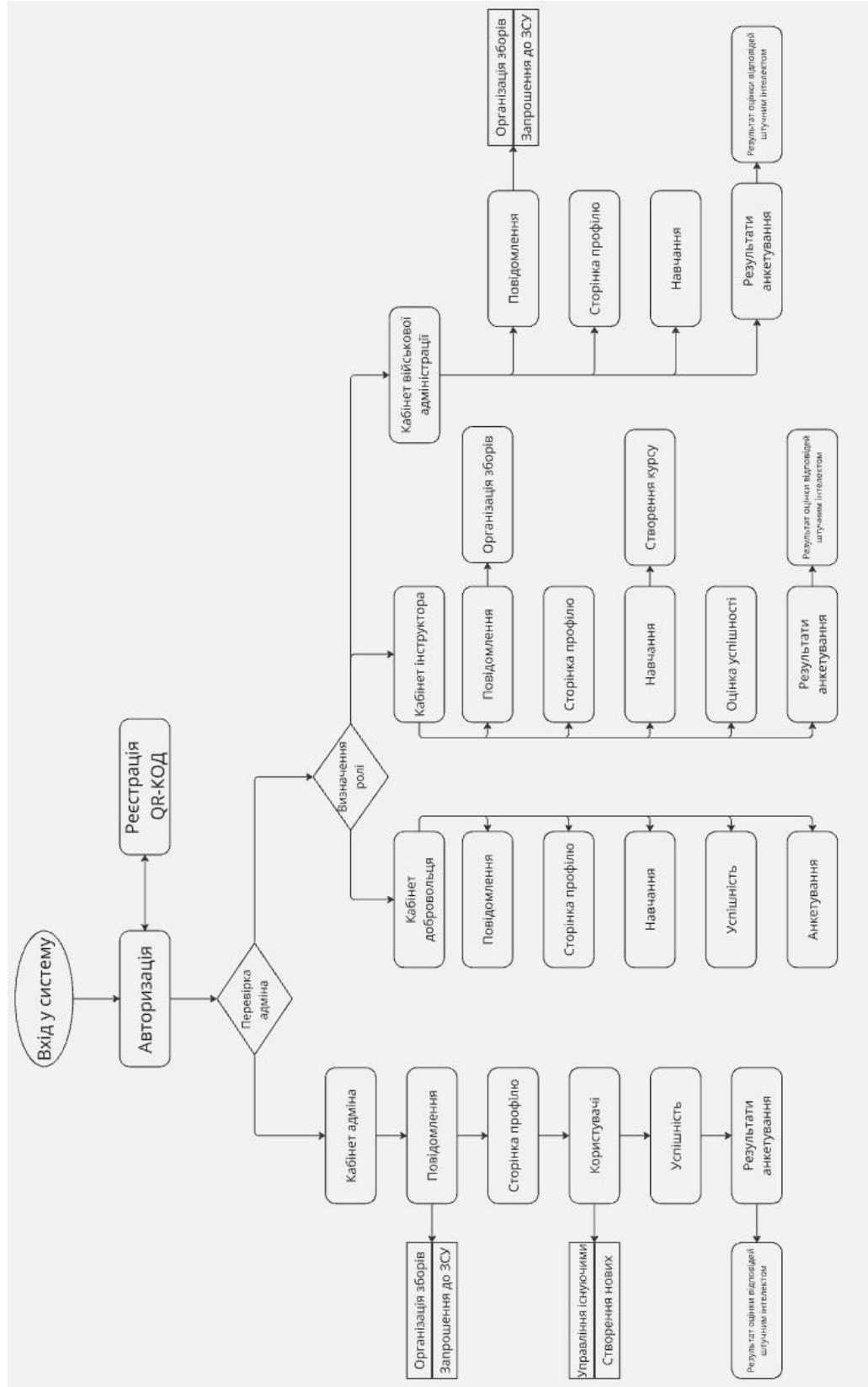
ДОДАТОК Б

Діаграма алгоритму роботи інформаційної системи обліку та підготовки
добровольців територіальної оборони



ДОДАТОК В

Алгоритм маршрутизації по інформаційній системі обліку та підготовки
добровольців територіальної оборони



ДОДАТОК Г

Лістинг питань для перевірки психологічної готовності добровольця

```
{
  "questions": [
    {
      "id": 1,
      "question": "Як ви зазвичай відчуваєте себе після важкого дня або емоційно насиченої ситуації?",
      "answers": {
        "A": "Мені вдається швидко заспокоїтись і відновитися.",
        "B": "Я відчуваю втому, але можу функціонувати.",
        "C": "Я довго не можу повернутися до нормального стану.",
        "D": "Втома та напруга повністю вибивають мене з колії."
      }
    },
    {
      "id": 2,
      "question": "Чи виникають у вас неконтрольовані спалахи гніву, страху або інші сильні емоції без очевидної причини?",
      "answers": {
        "A": "Ніколи, я завжди контролюю свої емоції.",
        "B": "Рідко, але я вмію їх швидко приборкати.",
        "C": "Час від часу, і мені важко з ними впоратися.",
        "D": "Постійно, це впливає на моє життя."
      }
    },
    {
      "id": 3,
      "question": "Що ви зазвичай відчуваєте, коли стикаєтеся з новою та невідомою проблемою?",
      "answers": {
        "A": "Інтерес і бажання вирішити її.",
        "B": "Легке занепокоєння, але я намагаюся діяти.",
        "C": "Сильний стрес, який впливає на мої рішення.",
        "D": "Я відчуваю безпорадність і уникаю проблем."
      }
    },
    {
      "id": 4,
      "question": "Які думки переважають у вас останнім часом?",

```

```

"answers": {
  "A": "Я відчуваю мотивацію і готовий до нових викликів.",
  "B": "У мене є певні сумніви, але я намагаюся не зациклюватися на них.",
  "C": "Я часто думаю про невдачі чи можливі небезпеки.",
  "D": "Мене постійно переслідують негативні думки."
},
{
  "id": 5,
  "question": "Як часто ви відчуваєте фізичні прояви стресу (серцебиття, тремтіння, пітливість тощо)?",
  "answers": {
    "A": "Майже ніколи, я добре контролюю свої реакції.",
    "B": "Інколи, але ці симптоми швидко проходять.",
    "C": "Часто, і це заважає мені діяти.",
    "D": "Постійно, навіть у звичайних ситуаціях."
  },
  {
    "id": 6,
    "question": "Як ви сприймаєте критику від інших?",
    "answers": {
      "A": "Я сприймаю її конструктивно і намагаюся вдосконалюватися.",
      "B": "Інколи мені це неприємно, але я вмію аналізувати сказане.",
      "C": "Критику я сприймаю болісно, це мене демотивує.",
      "D": "Я уникаю критики і сприймаю її як напад."
    },
    {
      "id": 7,
      "question": "Коли ви думаєте про можливі ризики під час служби, що ви відчуваєте?",
      "answers": {
        "A": "Я розумію ризики, але впевнений у своїй готовності.",
        "B": "Я відчуваю тривогу, але намагаюся її контролювати.",
        "C": "Я постійно турбуюся і не знаю, чи впораюсь.",
        "D": "Ці думки викликають у мене паніку та бажання уникати таких ситуацій."
      },
      {
        "id": 8,

```

"question": "Як ви реагуєте, коли чуєте різкі звуки або бачите щось несподіване?",

"answers": {

"A": "Я залишаюся спокійним і зосередженим.",

"B": "Я злегка напружуюся, але швидко заспокоююся.",

"C": "Я часто лякаюся і тривалий час залишаюся в напрузі.",

"D": "У мене починається паніка, і я не можу діяти."

}

},

{

"id": 9,

"question": "Як ви оцінюєте свою здатність відпочивати та розслаблятися в складні періоди?",

"answers": {

"A": "Я легко знаходжу способи відновлення.",

"B": "Мені потрібно докладати зусиль, щоб заспокоїтися.",

"C": "Я рідко можу повноцінно відпочити через стрес.",

"D": "Відпочинок не допомагає, я завжди напружений."

}

},

{

"id": 10,

"question": "Що найкраще описує ваш нинішній емоційний стан?",

"answers": {

"A": "Я відчуваю впевненість та енергію.",

"B": "У мене є певний рівень тривоги, але я можу працювати з цим.",

"C": "Я часто відчуваю стрес і невпевненість.",

"D": "Я відчуваю повну емоційну виснаженість."

}

}

]

}

ДОДАТОК Д

UML-діаграма розгортання інформаційної системи



ДОДАТОК Е

Лістинг файлу Base.php

```
<?php
Base::$debug = 0;
$data = [
    'host' => 'localhost',
    'user' => 'root',
    'pass' => '',
    'base' => 'teroborona',
    'charset' => 'utf8'
];
$pdo = Base::connect('mysql', $data) or die('Нет подключения.');
```

```
Base::query('set names utf8mb4');
```

```
class Base
{
    public static $pdo = false, // Объект PDO
        $debug = 0; // Уровень отчётности
    protected static $transaction = false;
    public static function transaction()
    {
        if (self::$pdo && !self::$transaction)
        {
            try {
                $bool = self::$pdo->beginTransaction();
                self::$transaction = 0;
                self::info('Transaction included', 2);
                return $bool;
            }
            catch (\PDOException $e) {
```

```

        self::info('Transaction error'. "\n".$e->getMessage(), 1);
        return false;
    }
}
return false;
}
public static function commit()
{
    if (self::$pdo && is_int(self::$transaction))
    {
        $count = self::$transaction;
        self::$transaction = false;
        if ($count == 0)
        {
            self::$pdo->commit();
            self::info('Commit successfully', 2);
            return true;
        }
        else
        {
            self::$pdo->rollBack();
            self::info('Commit rollback', 1);
        }
    }
    return false;
}
public static function get($table, $mixed_var = [], $param = [], $select = '*')
{
    list($part_sql, $param) = self::genPartSql($mixed_var, $param, ' AND ');

```

```

        return self::query('SELECT '.$select.' FROM `'.$table.` WHERE
'.$part_sql.' LIMIT 1', $param);
    }

    public static function count($table, $mixed_var = [], $param = [])
    {
        return self::get($table, $mixed_var, $param, 'COUNT(`id`)');
    }

    public static function remove($table, $mixed_var = [], $param = [])
    {
        list($part_sql, $param) = self::genPartSql($mixed_var, $param, ' AND ');
        return self::query('DELETE FROM `'.$table.` WHERE '.$part_sql,
$param);
    }

    public static function update($table, $mixed_var = [], $param = [])
    {
        list($sel_sql, $sel_params) = self::genPartSql($mixed_var, [], ' AND ',
'sel_');

        list($sup_sql, $sup_params) = self::genPartSql($param, [], ', ', 'up_');
        return self::query('UPDATE `'.$table.` SET '.$sup_sql.' WHERE '.$sel_sql,
array_merge($sel_params, $sup_params));
    }

    public static function add($table, $param = [])
    {
        $str_column = "";
        $str_values = "";
        $array = [];
        foreach ($param as $column => $value)
        {
            $name_column = explode('/', $column)[0];

```

```

        $str_column .= (! empty($str_column) ? ', ' : null).'`.$name_column.`';
        $str_values .= (! empty($str_values) ? ', ' : null).'`.$name_column`;
        $array[$column] = $value;
    }
    return self::query('INSERT INTO ``.$table.`' (``.$str_column.`) VALUES
(`.$str_values.`)', $array);
}

public static function query($str_sql, $param = [], $all_type = false)
{
    if (self::$pdo)
    {
        $time = microtime(true);
        $info = "";
        if (self::$debug > 0)
        {
            $param_info = [];
            foreach ($param as $key => $value)
            {
                $param_info[] = '['.$key.'] => '.$value;
            }
            $info .= $str_sql."\n";
            $info .= (! empty($param) ? implode("\n", $param_info)."\n" : null);
        }
        try {
            if (! empty($param))
            {
                $stm = self::$pdo->prepare($str_sql);
                foreach ($param as $key => $value)
                {

```

```

        $exp = explode('/', $key);
        $value = (! empty($value) ? $value : "");
        $set_type = \PDO::PARAM_STR;
        if (isset($exp[1]) && $exp[1] == 'int')
        {
            $value = (int) $value;
            $set_type = \PDO::PARAM_INT;
        }
        $stm->bindValue(':'.$exp[0], $value, $set_type);
    }
    $stm->execute();
}
else
{
    $stm = self::$pdo->query($str_sql);
}
self::info($info.'Time: '.number_format( microtime(true) - $time, 5).'
sec', 2);
}
catch (\PDOException $e) {
    if (is_int(self::$transaction)) {
        self::$transaction++;
    }
    self::info('Query error '.$e->getMessage()."\n".trim($info), 1);
    return false;
}
if ($stm)
{
    preg_match('~^(/\*.*+?\*/ *)\s*~i', trim($str_sql), $preg);

```

```

$comand = mb_strtolower($preg[2], 'UTF-8');
if ($comand == 'select')
{
    if ($all_type == 'arr')
    {
        return $stm->fetchAll(\PDO::FETCH_ASSOC);
    }
    elseif ($all_type == 'gen')
    {
        return self::queryGenerator($stm);
    }
    elseif ($array = $stm->fetch(\PDO::FETCH_ASSOC))
    {
        if (count($array) == 1)
        {
            if (substr_count( array_keys($array)[0], '(' ) > 0)
            {
                return array_values($array)[0];
            }
        }
        return $array;
    }
    return false;
}
elseif ($comand == 'insert')
{
    return self::$pdo->lastInsertId();
}
return true;

```

```

    }
}
return false;
}

public static function connect($type = 'mysql', $param = [])
{
    switch ($type)
    {
        case 'mysql':
            $dsn = 'mysql:host='.(! empty($param['host']) ? $param['host'] :
'localhost').';' . (! empty($param['port']) ? 'port=' . $param['port'] . ';' :
null) . 'dbname=' . $param['base'] . ';charset=' . (! empty($param['charset']) ?
$param['charset'] : 'utf8');
            break;
        case 'postgresql':
            $dsn = 'pgsql:host='.(! empty($param['host']) ? $param['host'] :
'localhost').';dbname=' . $param['base'] . ';options="--client_encoding=' . (!
empty($param['charset']) ? $param['charset'] : 'utf8') . '";';
            break;
        case 'oracle':
            $dsn = 'oci:dbname='.(! empty($param['host']) ? $param['host'] :
'localhost').'/' . $param['base'] . ';charset=' . (! empty($param['charset']) ? $param['charset'] :
'utf8');
            break;
        case 'sqlite':
            $dsn = 'sqlite:' . $param['path'];
            break;
    }
    if (isset($dsn))

```

```

    {
        try {
            self::$pdo = new \PDO($dsn, (! empty($param['user']) ? $param['user']
: 'root'), (! empty($param['pass']) ? $param['pass'] : ''), [
                \PDO::ATTR_ERRMODE =>
\PDO::ERRMODE_EXCEPTION,
                \PDO::ATTR_DEFAULT_FETCH_MODE =>
\PDO::FETCH_ASSOC,
            ]);
            self::info('Successfully connected to the '.$type.' database', 2);
            return self::$pdo;
        }
        catch (\PDOException $e) {
            self::info('Connect '.$type.' error'. "\n". $e->getMessage(), 1);
        }
    }
    return false;
}

public static function safely($string)
{
    return addslashes($string);
}

public static function end()
{
    self::info('End', 2);
    self::$pdo = null;
}

protected static function info($text, $level = 1)
{

```

```

if (self::$debug >= $level)
{
    $trace = (new \Exception)->getTraceAsString();
    $trace = preg_replace_callback(
        "~#\d+ ((.+)\(\d+\):)(.+)(\n)~i",
        function($matc){
            static $num = -1;
            if (empty($matc[1]) || $matc[2] == __FILE__){
                return "";
            }
            $path = str_replace("\\", '/', $matc[1]);
            if (isset($_SERVER['DOCUMENT_ROOT'])) {
                $path = str_replace($_SERVER['DOCUMENT_ROOT'], "",
$path);
            }
            $num++;
            return '#'.$num.' '.$path . $matc[3]."\n";
        },
        $trace
    );
    echo "\n
```

```

    }
}

protected static function genPartSql($mixed_var, $param = [], $glue = ', ',
$prefix = "")
{
    if (is_numeric($mixed_var))
    {
        $part_sql = `id` = :id';
        $param = [
            'id' => $mixed_var
        ];
    }
    elseif (is_array($mixed_var))
    {
        $part_sql = [];
        $param = [];
        foreach ($mixed_var as $column => $value)
        {
            $name_column = explode('/', $column)[0];
            $part_sql[] = ``.$name_column.` = :'. $prefix . $name_column;
            $param[ $prefix . $column ] = $value;
        }
        $part_sql = implode($glue, $part_sql);
    }
    else
    {
        $part_sql = $mixed_var;
    }
    return [

```

```
        $part_sql,  
        $param  
    ];  
}  
}
```

ДОДАТОК Ж

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна система обліку та підготовки добровольців територіальної оборони

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Turnitin

Оригінальність 95 % Схожість 5 %

Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи _____ Черній Д.В.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Городецька О.С.
(підпис) (прізвище, ініціали)