

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 126 «Інформаційні системи та технології»
(освітня програма «Системи бізнес-аналітики»)

на тему:

**«Інформаційна система внутрішньої мережі Управління
інформаційно-аналітичної підтримки ГУНП в
Житомирській області»**

Виконав студент групи ІСТ-20-1
РАЙКОВСЬКИЙ Владислав Андрійович

Керівник роботи:
ГРАФ Марина Сергіївна

Рецензент:
КУЗЬМЕНКО Олександр Вікторович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук
Марина ГРАФ
«25» січня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 126 «Інформаційні системи та технології»
(освітня програма «Системи бізнес-аналітики»)

Здобувач вищої освіти: РАЙКОВСЬКИЙ Владислав Андрійович

Керівник роботи: ГРАФ Марина Сергіївна

Тема роботи: «Інформаційна система внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області»,

затверджена наказом закладу вищої освіти від «25» січня 2024 р., №24/с

Термін здачі закінченої роботи: «10» червня 2024 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

об'єктом дослідження є інформаційні технології для оптимізації внутрішніх процесів УІАП ГУНП у Житомирській області.

Предметом дослідження виступає використання Web-орієнтованих технологій для розробки інформаційної системи для УІАП ГУНП у Житомирській області.

Проект: https://gitlab.com/ist201_rva/diploma-raikovskiyi-vlad-bachelor-20-1.git

Консультанти випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1,2	Граф М.С.		

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних рішень та їх особливостей. Формулювання технічного завдання до проекту. Опрацювання літературних джерел.	2 лютого 2024 – 5 березня 2024	Виконано
2	Проектування структури інформаційної системи	6 березня 2024 – 25 березня 2024	Виконано
3	Написання програмного коду	26 березня 2024 – 26 квітня 2024	Виконано
4	Тестування системи	27 квітня 2024 – 14 травня 2024	Виконано
5	Оформлення пояснювальної записки	15 травня 2024 – 1 червня 2024	Виконано
6	Попередній захист	1 червня 2024	Виконано

РЕФЕРАТ

Кваліфікаційна робота складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 76 сторінках друкованого тексту.

Пояснювальна записка має 18 сторінок додатків. Список використаних джерел містить 17 найменувань і займає 2 сторінки. В роботі наведено 43 рисунки. Загальний обсяг роботи – 76 сторінки.

Метою кваліфікаційної роботи є оптимізація та цифровізація внутрішніх процесів Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області шляхом створення інформаційної системи з використанням Web-орієнтованих технологій.

У роботі запропоновано використання інформаційних технологій для розробки та впровадження інформаційної системи в роботу Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області з метою оптимізації роботи управління.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, УПРАВЛІННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ ПІДТРИМКИ, ЗАХИСТ ІНФОРМАЦІЇ, PHP, MYSQL, CMS, CSS, JAVASCRIPT, HTML, BOOTSTRAP, WEB-САЙТ, MVC, СЕРВЕР.

					ІСТ.КР.Б – 126 – 24 – ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області	Літ.	Арк.	Аркушів	
Розроб.		В.А.РАЙКОВСЬКИЙ							
Керівник		М.С. ГРАФ					3	59	
Рецензент		О.В. КУЗЬМЕНКО				Житомирська політехніка, група ІСТ-20-1			
Н. контр.									
Зав. каф.		М.С. ГРАФ							
					Пояснювальна записка				

ABSTRACT

The qualification work consists of a list of abbreviations, an introduction, three chapters, conclusions, a list of references and appendices.

The text part is set out on 76 pages of printed text.

The explanatory note has 18 pages of appendices. The list of references includes 17 titles and occupies 2 pages. The paper contains 43 figures. The total volume of the work is 76 pages.

The purpose of the qualification work is to optimize and digitalize the internal processes of the Department of Information and Analytical Support of the Main Department of the National Police in Zhytomyr region by creating an information system using Web-oriented technologies.

The paper proposes the use of information technologies for the development and implementation of an information system in the work of the Department of Information and Analytical Support of the Main Department of the National Police in Zhytomyr region in order to optimize the work of the department.

KEYWORDS: INFORMATION SYSTEM, DEPARTMENT OF INFORMATION AND ANALYTICAL SUPPORT, INFORMATION PROTECTION, PHP, MYSQL, CMS, CSS, JAVASCRIPT, HTML, BOOTSTRAP, WEB-SITE, MVC, SERVER.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....	12
1.1 Постановка задачі	12
1.2 Вибір архітектури для побудови інформаційної системи.....	17
1.3 Інструментарій реалізації та вимоги до апаратних засобів для інформаційної системи.....	19
Висновки до першого розділу	21
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	22
2.1 Варіанти використання системи	22
2.2 Розробка структури інформаційної системи.....	25
2.3 Проектування архітектури інформаційної системи	27
2.4 Розробка бази даних системи	29
2.5 Проектування та розробка функціональних алгоритмів роботи інформаційної системи.....	37
2.6 Реалізація основних функціональних можливостей інформаційної системи.....	38
Висновки до другого розділу.....	44
РОЗДІЛ 3 ОГЛЯД ІНТЕРФЕЙСУ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	45
3.1 Огляд інтерфейсу інформаційної системи (користувач)	45
3.2 Тестування системи.....	53
Висновки до третього розділу	56
ВИСНОВОК.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	60
Додаток А	61
Додаток Б.....	62

Додаток В.....	63
Додаток Г.....	64
Додаток Г.....	71
Додаток Д.....	76

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ СКОРОЧЕНЬ І ТЕРМІНІВ

ІС — Інформаційна система

СУБД — Система управління базами даних

MVC — Model-view-controller, архітектурний шаблон, що використовується під час проєктування та розробки програмного забезпечення

MVVM — Model View View-model, архітектурний шаблон, що використовується під час проєктування та розробки програмного забезпечення

MVP — Model View Presenter, архітектурний шаблон, що використовується під час проєктування та розробки програмного забезпечення

УІАП — Управління інформаційно-аналітичної підтримки

ГУНП — Головне управління Національної поліції

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

Актуальність теми поляє в тому, що в наш час виникає необхідність оптимізації роботи підприємства шляхом цифровізації внутрішніх процесів. Найчастіше для вирішення проблем з оптимізацією роботи, підприємствами впроваджуються інформаційні технології в процес виробництва, управління та менеджменту. Середній та великий бізнес використовує інформаційні технології у вигляді інформаційних систем, web-сайтів, додатків для клієнтів та персоналу. Сучасні технології здатні вивести управління підприємством на новий рівень ефективності, забезпечити стабільність та надійність роботи.

Впровадження інформаційної системи у внутрішню мережу підприємства надасть можливість працівникам швидше та ефективніше виконувати повсякденні задачі.

Метою дослідження є оптимізація та цифровізація внутрішніх процесів Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області шляхом створення інформаційної системи

Встановлена мета обумовлює наступні завдання:

- Проведення аналізу роботи підприємства для визначення шляхів оптимізації роботи підприємства
- Визначення основних функціональних та системних вимог до інформаційної системи
- Визначення архітектури та загального алгоритму роботи системи
- Обґрунтування та вибір засобів реалізації системи
- Розробка інформаційної системи

Об'єктом дослідження є інформаційні технології для оптимізації внутрішніх процесів УІАП ГУНП у Житомирській області.

Предметом дослідження є використання Web-орієнтованих технологій для розробки інформаційної системи для УІАП ГУНП у Житомирській області.

В кваліфікаційній роботі використано: алгоритми оптимізації, SWOT-аналіз, проектування архітектури системи.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		9

Апробація результатів. Матеріали роботи доповідалися на всеукраїнській науково-технічній конференції, 29-30 листопада 2023 року, на Міжнародній науково-технічній конференції «Інформаційно-комп'ютерні технології – 2024» 28 – 29 березня 2024 року.

Публікації. За темою кваліфікаційної роботи було опубліковано 4 тези:

1. Граф М.С., Райковський В.А. Алгоритми оптимізації даних. Комп'ютерні технології: інновації, проблеми, рішення: тези VI Всеукраїнської науково-технічної конференції, 29-30 листопада 2023 року. Житомир : «Житомирська політехніка», 2023. С.143-144
2. Граф М.С., Райковський В.А. Розробка структури інформаційної системи внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області. Тези XIV Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології – 2024», м. Житомир, 28 – 29 березня 2024 р. – Житомир: Житомирська політехніка, 2024. С.119-121.
3. Граф М.С., Райковський В.А. Використання JavaScript в розробці сучасного програмного забезпечення. Тези III Всеукраїнської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих учених "Інформаційно-комп'ютерні технології: стан, досягнення та перспективи розвитку", 25-26 листопада 2021 року. Житомир : «Житомирська політехніка», 2021. С.111-113.
4. Райковський В.А., Вольський Р.А. Використання сучасних архітектурних підходів для розробки інформаційної системи для управління інформаційно-аналітичної підтримки. Тези Інтегровані інтелектуальні робототехнічні комплекси (ІРТК-2024). Сімнадцята міжнародна науково-практична конференція, 12-22 травня 2024 року. — Київ: НАУ, 2024. С.419-421.

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, 3 розділів, висновків, списку використаних джерел із 17 найменувань (2 стор.), 6

додатків (18 стор.), 43 ілюстрації по тексту. Повний обсяг роботи становить 76 сторінок, з них 59 сторінок основного тексту.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				11
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1 Постановка задачі

Завдання полягає у тому що потрібно:

1. Провести аналіз внутрішніх процесів Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області.
2. Розробити алгоритми та методи оптимізації процесів управління.
3. На основі проведеного аналізу сформулювати вимоги до системи та окреслити весь необхідний функціонал.
4. Розробити інформаційну систему для внутрішнього користування працівниками УІАП.

Розробка проекту здійснювалася від найпростішої до найпростішої версії, яка одразу тестувалася в реальних умовах працівниками управління. Тестування здійснювалося кожного процесу окремо, після відгуків користувачів до програмного продукту вносилися поліпшення. Після успішного тестування групи процесів, що являють собою модуль — окремий структурний компонент системи, створювався новий модуль інформаційної системи внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області.

Проаналізувавши джерела [1-3] було вирішено використовувати мову програмування PHP для серверної частини програмного продукту, а на стороні клієнта використовувати HTML, CSS. Спираючись на дослідження [4] та джерело [5] було обрано мову програмування JavaScript, фреймворк jQuery. Як середовище для розробки використовуватиметься PhpStorm [6].

Використовуватиметься вебсервер Apache [7] на базі ОС Microsoft Windows Server 2019 для розгортання інформаційної системи внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області.

Використовувався архітектурний шаблон MVC та патерн синглтон щоб забезпечити єдину точку входу в систему.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		12

Підчас розробки проєкту використовуватиметься об'єктно-орієнтований підхід програмування [8]. При розробці програмного продукту було дотримано основних принципів цього підходу: інкапсуляція, поліморфізм та наслідування.

Використовувались методи `get` та `set` щоб обмінюватися даними з сервером в межах клієнт-серверної архітектури.

Використовуватимуться різні методи та підходи щоб інтерфейс сайту був зручний, компактний та зрозумілий [9]. Сайт використовуватиметься у внутрішній мережі, отже в першу чергу має бути простим, зручним та мати весь необхідний функціонал.

Інформаційна система має підтримувати виконання наступних операцій:

1. Користувачі

- a. Аутентифікація користувача
- b. Додавання користувача
- c. Видалення користувача
- d. Зміна основних даних користувача

2. Новини

- a. Перегляд списку новин
- b. Перегляд новини
- c. Додавання новини
- d. Редагування
- e. Видалення
- f. Відкладена публікація
- g. Пошук

3. Календар

- a. Створення календаря (приватний, загальнодоступний)
- b. Видалення календаря
- c. Редагування календаря
- d. Поширення календаря між обраними користувачами
- e. Додавання події до календаря:
 - i. Перегляд

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		13

ii. Редагування

iii. Видалення

4. Пам'ятки

a. Додавання розділу пам'яток (приватний, загальнодоступний, доступний обраним користувачам):

i. Редагування створеного розділу

ii. Видалення створеного розділу

b. Додавання пам'ятки

c. Редагування пам'ятки

d. Видалення пам'ятки

e. Перегляд усіх пам'яток

f. Корзина

g. Друк

5. Оголошення

a. Додавання

b. Редагування

c. Видалення

d. Переміщення оголошення в неактуальні

6. Управління

a. Відділ:

i. Додавання

ii. Видалення

iii. Редагування

b. Керівництво:

i. Додавання

ii. Видалення

iii. Редагування

7. Моніторинг

a. Перегляд дій користувачів за обраний проміжок часу

b. Зміна форм відображення інформації

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		14

с. Очищення даних

8. Задачі

- а. Додавання
- б. Редагування
- с. Видалення
- д. Переміщення задачі в виконані
- е. Відновлення задачі
- ф. Створення копії задачі
- г. Можливість автоматичного додавання задачі в календар
- h. Пошук задач за ключовими словами

9. Працівники

- а. Особова інформація
 - i. Переміщення на нову посаду
 - ii. Звільнення
 - iii. Редагування інформації
 - iv. Пошук
 - v. Фільтрація
- б. Історія служби
 - i. Переміщення на нову посаду
 - ii. Звільнення
 - iii. Редагування інформації
 - iv. Пошук
 - v. Фільтрація

10. Події в системі

- а. При додаванні новити, оголошення і т.д. користувачеві надсилається сповіщення про зміни в інформаційному просторі системи

11. Документація

- а. Каталог:
 - i. Додавання
 - ii. Видалення

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		15

iii. Редагування

iv. Переміщення

b. Файл:

i. Додавання

ii. Видалення

iii. Редагування

iv. Переміщення

v. Завантаження

vi. Вивантаження

c. Можливість архівації

d. Пошук

12. Довідкова інформація

a. Перегляд довідкової інформації необхідної працівникам

Під час формування вимог до програмного продукту, повністю враховувалась думка працівників управління, потенційних користувачів, адже аналогів системи не існує, вона розробляється спеціально під потреби УІАП ГУНП в Житомирській області.

Інформаційна система для Управління інформаційно-аналітичної підтримки розробляється для оптимізації роботи управління шляхом цифровізації більшості рутинних процесів. Система забезпечить можливість електронного документообігу та пришвидшить процеси модернізації управління інформаційно аналітичної підтримки [10].

Для створення веб платформи було вирішено наслідувати поширений архітектурний шаблон — MVC. Завдяки такому підходу система поділяється на три взаємопов'язані частини: модель, вигляд (інтерфейс) та модуль керування. У представленому шаблоні передбачено відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу отримаю інформацією, а зміни в моделі даних відбувалися без змін інтерфейсу користувача. Завдяки використанню такого

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		16

підходу загальний алгоритм роботи програми буде гнучким та допускатиме подальші зміни та розширення функціоналу інформаційної системи.

1.2 Вибір архітектури для побудови інформаційної системи

Важливим етапом роботи над програмним продуктом є вибір архітектури інформаційної системи, ажде правильно спроектована архітектура — запорука успішного функціонування системи в майбутньому.

При виборі архітектури інформаційної системи внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області в першу чергу враховувалися функціональні вимоги, що були розроблені на основі думки працівників управління, що стануть пезпосередніми користувачами інформаційної системи.

При виборі архітектури розглядалися наступні архітектурні шаблони: MVC, MVP та MVVM. Розглянемо кожен з них окремо:

MVC — Model View Controller, складається з трьох головних компонентів: модель, представлення, контролер. У процесі роботи компоненти взаємодіють між собою, виконуючи покладені на себе функції, рисунку 1.1. Модель виражена у вигляді класу, що містить бізнес-логіку (модель даних). Також данна компонента містить правила та методи, що керують роботою з даними. Представлення — сукупність елементів інтерфейсу користувача. Виконує функцію відображення даних отриманих від контроллера. Контролер здійснює обробку даних що надходять від користувача через модель. Висупає з'єднуючою ланкою в роботі між представленням та моделлю.

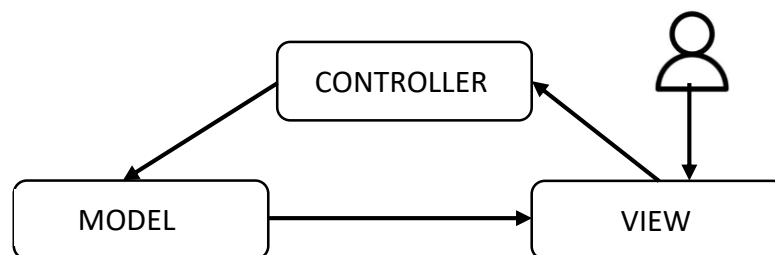


Рисунок 1.1 — Архітектурний шаблон MVC

MVP — Model View Presenter, даний патерн дуже схожий до попереднього, він так само передбачає три основних компоненти: модель, представлення, контролер. Модель та представлення виконують такі ж функції що і в попередньому архітектурному шаблоні. Єдина відмінність — це відсутність контроллера, замість нього тут виступає презентер. В презентер надходять данні від користувачів, використовуючи представлення. Обробка даних здійснюється моделлю, потім результат обробки передається назад в презентер, який вже візуалізує результат використовуючи компонент представлення, рисунок 1.2.

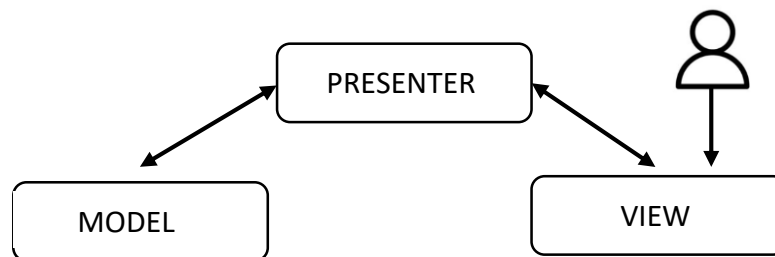


Рисунок 1.2 — Архітектурний шаблон MVP

MVVM — Model View View-model, схожий до попередніх архітектурних шаблонів своєю структурою. Головною його особливістю є наявність двохсторонньої прив'язки даних між представленням та моделлю представлення, це забезпечує кращий зв'язок моделі представлення та моделі, тобто збільшується швидкодія системи. View-model відповідає за керування представленням, управління моделлю, рисунок 1.3.

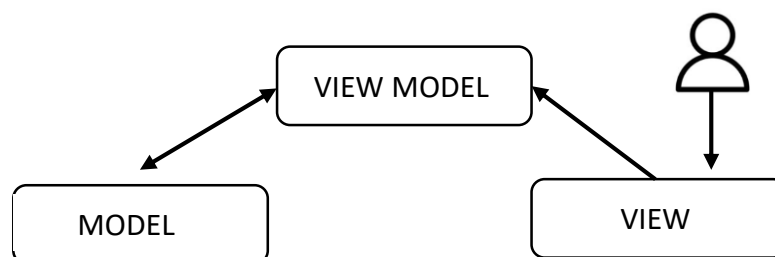


Рисунок 1.3 — Архітектурний шаблон MVVM

Використання сучасної layered архітектури в стилі MVC надасть ряд переваг:

- Гнучкість та масштабованість системи
- Поліпшення якості обслуговування

- Покращена безпека
- Більші можливості для тестування
- Стабільна робота системи

Структура інформаційної системи передбачає створення 12 контролерів, які відповідають за роботу окремих модулів системи та 12 моделей відповідно. До кожної пари контролер-модель відноситься від одного до декількох представлень.

Роботу контролера, моделі та представлення розглянемо на прикладі модуля новин.

Контролер News встановлює зв'язок між моделлю та представленнями, отримує відповідь від користувача на виконання однієї з наступних операцій: перегляд, додавання, видалення, редагування, пошук новин. На кожну дію користувача контролер містить відповідний метод. В методі контролера відбувається звертання до моделі, що відповідає за збереження даних та зв'язок з базою даних. Після того як відповідний метод в моделі здійснив взаємодію з базою даних, результат отримує контролер та візуалізує його за допомогою представлення [11].

1.3 Інструментарій реалізації та вимоги до апаратних засобів для інформаційної системи

1. Інструменти для розробки

1.1 Середовище розробки PhpStorm: Середовище PhpStorm було обрано з огляду на його функціональні можливості та широкий інструментарій для роботи з мовою програмування Php. Також PhpStorm має широкі можливості для відналадження та тестування.

1.2 Мови програмування та технології:

- Серверна частина: Php було обрано для реалізації серверної частини системи завдяки його широким можливостям для програмування саме на стороні сервера, популярність та простоту.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		19

- Клієнтська частина: Використання HTML та CSS для верстки сторінок системи та JavaScript, а саме JQuery для програмування на стороні клієнта, а саме для створення більш зручних та інтерактивних інтерфейсів.

2. Вимоги до Апаратних Засобів

2.1 Вебсервер Apache: Використання вебсервера Apache на базі операційної системи Microsoft Windows Server 2019 обрано з урахуванням його широкої підтримки та надійності. Apache забезпечує стабільну роботу серверної частини, підтримує велику кількість модулів та конфігурується зручно через конфігураційні файли.

2.2 Апаратні засоби сервера:

Одією з вимог до системи була масштабованість, тому варто передбачити таку можливість. Також система має бути готовою прийняти велику кількість користувачів тому що в подальшому кількість користувачів може зрости.

- Процесор: 8-ми ядерний, 16-ти поточний.

- Оперативна пам'ять: 32 ГБ.

- Диск: 16ТБ вільного місця для збереження файлів ІС та резервного копіювання.

Вказано мінімальні вимоги до апаратних засобів.

2.3 Апаратні засоби клієнтських пристроїв:

- Процесор: 4-р'юх ядерний

- Оперативна пам'ять: 4 ГБ.

Вказано мінімальні вимоги до апаратних засобів.

2.4 Безпека:

- ІС буде функціонувати у внутрішній захищеній мережі, що зменшить ризик зовнішніх атак.
- Використання протоколу HTTPS для забезпечення шифрування даних між сервером та клієнтом.
- Регулярні оновлення та моніторинг безпеки операційної системи, веб-сервера та інших компонентів системи.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				20
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до першого розділу

У ході виконання першого розділу кваліфікаційної роботи було описано завдання на кваліфікаційну роботу воно полягає в:

- Проведенні аналізу внутрішніх процесів організації
- Розробці алгоритмів оптимізації роботи організації
- Формулюванні вимог до системи
- Розробці інформаційної системи внутрішньої мережі для Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області

Було розглянуто варіанти можливої архітектури для інформаційної системи, проаналізовано їх переваги та недоліки, обґрунтовано вибір сучасної layered архітектури в стилі MVC.

Обрана архітектура забезпечить надійність та функціональність роботи системи роботи системи, модульність та покращення безпеки, так як розділення ускладнює несанкціонований доступ. Також враховано можливість легкого масштабування та рефакторингу.

Проведено аналіз інструментарію та обрано що буде використано в ході розробки інформаційної системи. Як середовище розробки було обрано PhpStorm. Серверна частина програмного комплексу буде написана на мові PHP, а клієнтська розроблятиметься з використанням HTML, CSS та мови програмування JavaScript. З урахуванням надійності та практичності в ролі сервера було обрано вебсервер Apache.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		21

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Варіанти використання системи

Спираючись на потреби, функціонал, проаналізувавши джерело [12] визначено, що інформаційна система налічуватиме 4 ролі: адміністратор, менеджер, користувач та не зареєстрований користувач. Розподілення обов'язків здійснюватиметься дотримуючись ієрархії: найвищий доступ має адміністратор, а найнижчий не зареєстрований в системі користувач. Користувачами виступатимуть працівники управління.

Для не зареєстрованого користувача будуть доступні наступні дії:

- Головна сторінка: перегляд галереї, останніх новин та оголошень, швидкий перехід за посиланнями, сповіщення про зміни що відбулися в системі
- Новини: перегляд новин та новини, пошук
- Довідкова інформація: перегляд довідкової інформації
- Перегляд керівництва та структури управління
- Оголошення: перегляд
- Пам'ятки: перегляд загальнодоступних пам'яток

Для звичайного користувача будуть доступні наступні дії:

- Головна сторінка: перегляд галереї, останніх новин та оголошень, швидкий перехід за посиланнями, сповіщення про зміни що відбулися в системі
- Новини: перегляд новин та новини, пошук
- Документація: дозволи в цьому модуль призначаються адміністратором окремо, в деяких користувачів вони можуть бути різними в залежності від зацманих посад
- Довідкова інформація: перегляд довідкової інформації
- Перегляд керівництва та структури управління
- Календар: створення календарів різних типів (загальнодоступні, приватні, доступні обраним користувачам), перегляд календарів

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				22
Змн.	Арк.	№ докум.	Підпис	Дата		

доступних користувачеві, додавання, видалення, редагування та дублювання подій

- Задачі: створення, видалення та редагування задач, пошук, додавання тегів, пошук за тегами, переміщення задач до відпрацьованих, поновлення відпрацьованих задач в списку актуальних, додавання, видалення та редагування тегів
- Оголошення: перегляд
- Пам'ятки: створення, видалення та редагування пам'яток, пошук за ключовими словами, створення та редагування розділів для пам'яток
- Користувач:перегляд особистої інформації, зміна паролю

Для менеджера будуть доступні дії:

- Головна сторінка: перегляд галереї, останніх новин та оголошень, швидкий перехід за посиланнями, сповіщення про зміни що відбулися в системі
- Новини: перегляд новин та новини, пошук, додавання, видалення та редагування, власноруч доданих новин.
- Документація: дозволи в цьому модуль призначаються адміністратором окремо, в деяких користувачів вони можуть бути різними в залежності від зазначених посад
- Довідкова інформація: перегляд довідкової інформації
- Перегляд керівництва та структури управління
- Календар: створення календарів різних типів (загальнодоступні, приватні, доступні обраним користувачам), перегляд календарів доступних користувачеві, додавання, видалення, редагування та дублювання подій
- Задачі: створення, видалення та редагування задач, пошук, додавання тегів, пошук за тегами, переміщення задач до відпрацьованих, поновлення відпрацьованих задач в списку актуальних, додавання, видалення та редагування тегів
- Оголошення: перегляд, додавання, видалення та редагування.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		23

- Пам'ятки: створення, видалення та редагування пам'яток, пошук за ключовими словами, створення та редагування розділів для пам'яток
- Користувач: Перегляд особистої інформації, зміна паролю

Для адміністратора будуть доступні дії:

- Головна сторінка: перегляд галереї, останніх новин та оголошень, швидкий перехід за посиланнями, сповіщення про зміни, що відбулися в системі
- Новини: перегляд новин та новини, пошук, додавання, видалення та редагування всіх новин.
- Документація: дозволи в цьому модуль призначаються адміністратором окремо, в деяких користувачів вони можуть бути різними в залежності від займаних посад
- Довідкова інформація: перегляд довідкової інформації
- Перегляд керівництва та структури управління, внесення змін в інформацію про керівництво, зміна структури: додавання, видалення, редагування управлінь/відділів.
- Календар: створення календарів різних типів (загальнодоступні, приватні, доступні обраним користувачам), перегляд календарів доступних користувачеві, додавання, видалення, редагування та дублювання подій
- Задачі: створення, видалення та редагування задач, пошук, додавання тегів, пошук за тегами, переміщення задач до відпрацьованих, поновлення відпрацьованих задач в списку актуальних, додавання, видалення та редагування тегів
- Оголошення: перегляд, додавання, видалення та редагування в системі.
- Пам'ятки: створення, видалення та редагування пам'яток, пошук за ключовими словами, створення та редагування розділів для пам'яток
- Користувач: Перегляд особистої інформації про всіх користувачів в системі, зміна паролю

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		24

- Повний доступ до системи обліку кадрів
- Моніторинг дій користувачів в системі
- Аналітика відвідуваності сторінок інформаційної системи

Для входу в систему як адміністратор використовувати логін: admin, пароль: adminpass. Як користувач, логін: vlad, пароль: vladpass123.

2.2 Розробка структури інформаційної системи

В наш час особливо є актуальною цифровізація та оптимізація даних у різних сферах нашого життя. Підприємства переходять на електронний документообіг, стабільно працює система дистанційної освіти, бізнес запроваджує інформаційні технології, котрі здатні оптимізувати внутрішні процеси підприємства.

Працівники сучасних підприємств майже не використовують архів, адже доступ до необхідної інформації можна отримати онлайн, або через спеціалізовані корпоративні платформи.

Інтеграція інформаційних систем в організації прогресувала протягом десятиліть.

Вдала розробка структури інформаційної системи внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області надасть можливість врахувати всі сформовані вимоги до системи та забезпечити стабільну роботу.

Система має на меті цифровізацію та оптимізацію внутрішніх процесів підприємства. Так як подібних систем на підприємстві не використовувалось та не було виявлено аналогів, що матимуть весь необхідних функціонал та відповідатимуть вимогам, то при визначенні вимог до функціоналу використовувалась думка працівників підприємства, що безпосередньо працюватимуть з системою. В результаті виконання опитування та аналізу результатів було виявлено наступні риси, якими повинна бути наділена система:

- гнучкість: система має бути доступною для використання на різних пристроях, що використовуються у внутрішній мережі;

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		25

- простота: інтерфейс системи має бути інтуїтивно зрозумілим для будь-якого користувача, а весь функціонал доступний без довгих блукань системою;
- модульність: інформаційна система має складатися з окремих модулів, що забезпечить легкість та простоту у подальшому розширенні функціоналу системи;
- функціональність: система має відповідати потребам працівників та спрощувати виконання основних внутрішніх задач для оптимізації роботи управління.

Інформаційна система передбачає створення 12 модулів. Кожен з модулів працюватиме окремо, виконуючи ті чи інші покладені на нього функції. Загальна структура модулів представлена на рисунку 2.1.

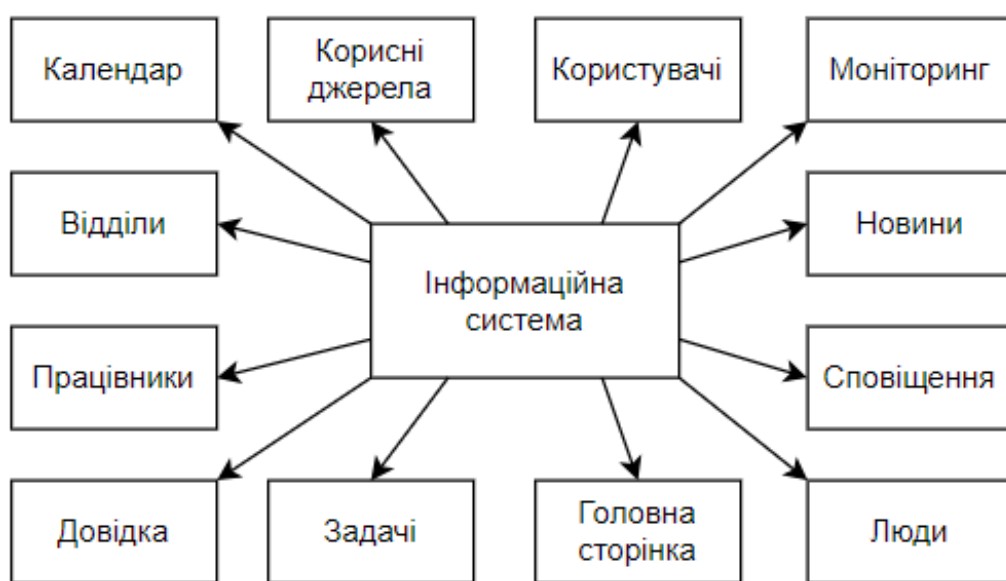


Рисунок 2.1 — Структура інформаційної системи

Розглянемо основні модулі детальніше. Календар – передбачає створення календарів, відмітки подій, поширення календарів між користувачами. Відділи – модуль відповідальний за збереження інформації про структуру управління. Працівники – забезпечення збереження інформації про кадровий склад. Довідка – модуль зберігає довідкову інформацію яка необхідна працівникам. Задачі – формування задач та тегів для них, прив’язка задач до календаря, пошук за тегами. Сповіщення – відображаються в правому верхньому кутку екрану,

сповіщають користувача про дедлайн задачі, тощо. Новини – модуль відповідає за створення, редагування, публікацію новин. Моніторинг – завдяки функціоналу модуля адміністратор має можливість переглядати дії інших користувачів. Користувачі – авторизація, реєстрація, збереження даних профілю користувача та прав доступу до системи. Корисні джерела – модуль надає можливість користуватися пам’ятками які стануть в нагоді при використанні інформаційної системи [10].

2.3 Проектування архітектури інформаційної системи

Для розробки інформаційної системи було обрано архітектурний шаблон MVC. Для роботи системи буде реалізовано ядро, клас для роботи з базою даних, контролери, моделі та представлення та інші класи необхідні для функціонування системи.

Діаграма класів інформаційної системи внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області розміщена в додатку Б та В.

Розглянемо основні класи системи.

Core — головний клас, являє собою ядро інформаційної системи знаходиться в неймспейсі core. В класі використовується патерн сингелтон, для забезпечення єдиного екземпляру ядра в системі.

DataBase — клас, що містить всі необхідні методи для роботи з базою даних, встановлення зв’язку, отримання даних, CRUD – операції.

Controller — базовий клас для всіх контролерів. Кожен створений контролер є нащадком цього класу.

Model — базовий клас для всіх моделей. Кожна створена модель є нащадком цього класу.

Template — клас шаблонізатора, відповідає за відображення необхідних представлень.

Utils — допоміжний клас. Містить різні методи, що використовуються для обробки даних.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		27

Розглянемо класи відповідно модулям, роботу яких вони забезпечують. Класи представлено у зв'язці контролер-модель.

controller/Calendar — відповідає за забезпечення роботи модуля календар, а саме за обробку вхідних запитів користувачів, коли вони використовують модуль календар, в свою чергу *model/Calendar* відповідає за бізнес-логіку та управління даними модуля календар.

controller/Departments — відповідає за забезпечення роботи модуля структура управління, а саме за обробку вхідних запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/Departments* відповідає за бізнес-логіку та управління даними модуля структура управління.

controller/Employee — відповідає за забезпечення роботи модуля для управління кадрами, а саме за обробку вхідних запитів від менеджерів або адміністраторів, коли вони використовують цей модуль, в свою чергу *model/Employee* відповідає за бізнес-логіку та управління даними модуля для управління кадрами.

controller/Help — відповідає за забезпечення роботи модуля пам'ятки, а саме за обробку вхідних запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/Help* відповідає за бізнес-логіку та управління даними модуля пам'ятки.

controller/News — відповідає за забезпечення роботи модуля новин, а саме за обробку вхідних запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/News* відповідає за бізнес-логіку та управління даними модуля новини.

controller/Notices — відповідає за забезпечення роботи модуля оголошень, а саме за обробку вхідних запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/Notices* відповідає за бізнес-логіку та управління даними модуля оголошень, сповіщень.

controller/Persons — відповідає за забезпечення функціонування модуля для роботи з персональними даними працівників, а саме за обробку вхідних

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		28

запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/Persons* відповідає за бізнес-логіку та управління даними модуля.

controller/Reference — відповідає за забезпечення функціонування модуля для роботи з довідкою, корисними джерелами, а саме за обробку вхідних запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/Reference* відповідає за бізнес-логіку та управління даними модуля.

controller/Site — відповідає за забезпечення роботи модуля головної сторінки, а саме за обробку вхідних запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/Site* відповідає за бізнес-логіку та управління даними модуля.

controller/Tasks — відповідає за забезпечення роботи модуля задач, а саме за обробку вхідних запитів користувачів, коли вони використовують цей модуль, в свою чергу *model/Tasks* відповідає за бізнес-логіку та управління даними модуля.

controller/Users — відповідає за забезпечення роботи модуля користувачі, а саме за обробку вхідних запитів, коли використовується модуль, в свою чергу *model/Users* відповідає за бізнес-логіку та управління даними модуля.

Таким чином розроблене ядро, контролери, моделі, представлення та інші класи забезпечують роботу відповідних модулів, а ті в свою чергу — функціонування системи.

2.4 Розробка бази даних системи

Одна з ролей інформаційних систем полягає в тому, щоб брати дані і перетворювати їх на інформацію, а потім перетворювати цю інформацію в організаційні знання. Після років розвитку технологій ця роль стала основною, роблячи інформаційні системи невід'ємною частиною практично кожного бізнесу [13].

Важливим аспектом роботи з даними є їх оптимізація [14], вона забезпечить швидкодію та стабільну роботу системи.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				29
Змн.	Арк.	№ докум.	Підпис	Дата		

Неможливо уявити роботу інформаційної системи без даних, а робота з даними неможлива без бази даних.

Інформаційна система внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області працюватиме з базою даних MySQL.

MySQL був обраний так, як описана інформаційна система має швидко обробляти великі об'єми даних, бути надійною та зручною при розробці та використанні [15][16].

Розглянемо структуру бази даних для інформаційної системи — рисунок 2.2.

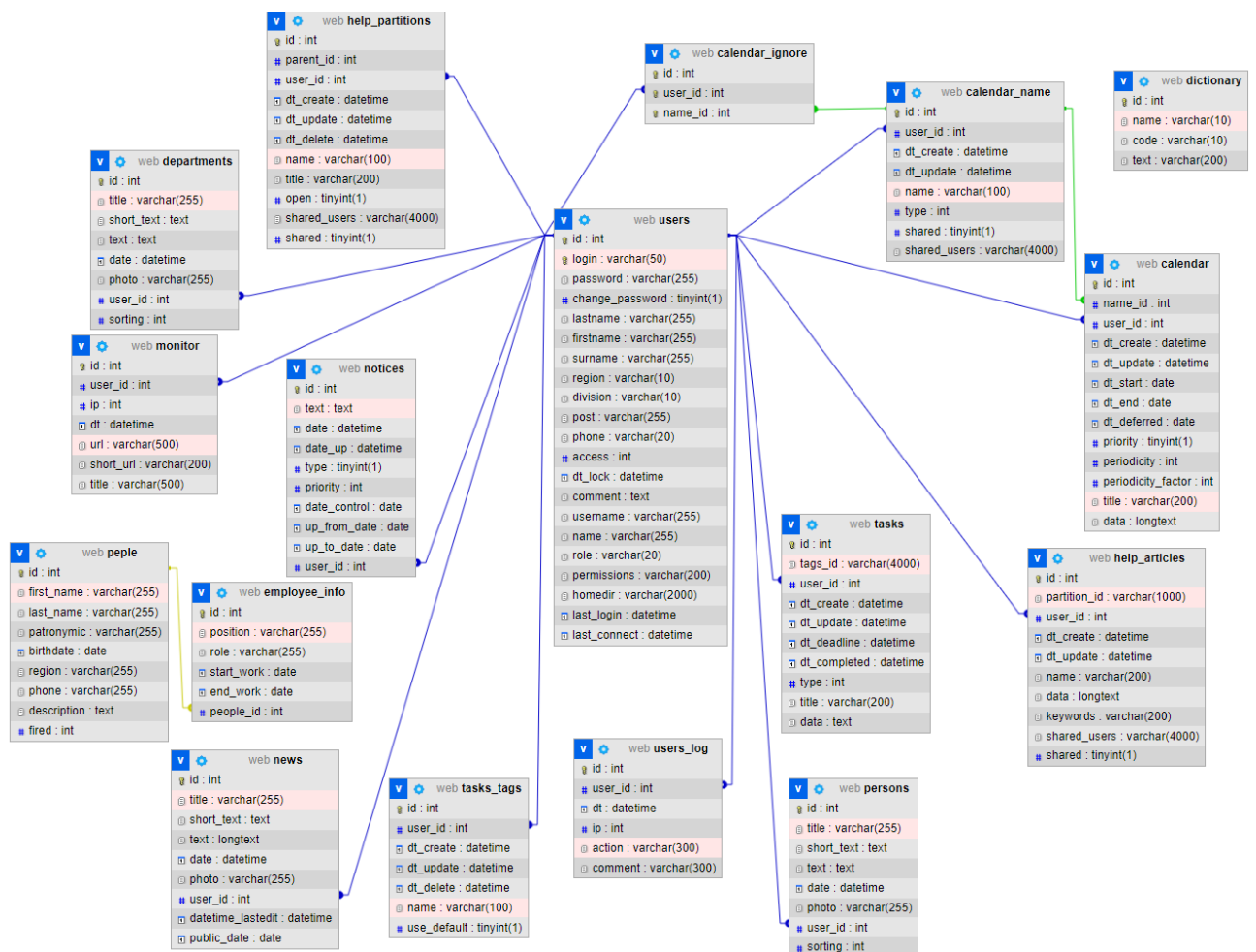


Рисунок 2.2 — Схема бази даних

База даних містить в собі 17 таблиць, присутні зв'язки: один до одного, один до багатьох, багато до одного, багато до багатьох.

Розглянемо створені таблиці:

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		30

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
name_id	int		No		
user_id	int		Yes	NULL	
dt_create	datetime		No		
dt_update	datetime		No		
dt_start	date		No		
dt_end	date		Yes	NULL	
dt_deferred	date		Yes	NULL	
priority	tinyint(1)		Yes	NULL	
periodicity	int		Yes	NULL	
periodicity_factor	int		Yes	NULL	
title	varchar(200)		No		
data	longtext		No		

Рисунок 2.3 — Таблиця calendar бази даних

Таблиця на рисунку 2.3 відповідає за збереження даних календаря.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
user_id	int		No		
name_id	int		No		

Рисунок 2.4 — Таблиця calendar_ignore бази даних

Таблиця на рисунку 2.4 містить данні для визначення яким користувачам, які календарі доступні.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
user_id	int		No		
dt_create	datetime		No		
dt_update	datetime		No		
name	varchar(100)		No		
type	int		Yes	NULL	
shared	tinyint(1)		Yes	NULL	
shared_users	varchar(4000)		Yes	NULL	

Рисунок 2.5 — Таблиця calendar_name бази даних

Таблиця на рисунку 2.5 зберігає назви календарів, дата їх створення та оновлення, тип та список користувачів яким доступний даний календар.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
title	varchar(255)		No		
short_text	text		No		
text	text		Yes	NULL	
date	datetime		No		
photo	varchar(255)		Yes	NULL	
user_id	int		No		
sorting	int		Yes	NULL	

Рисунок 2.6 — Таблиця departmentс бази даних

В таблиці на рисунку 2.6 зберігаються дані про відділи, структура управління.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
name	varchar(10)		No		
code	varchar(10)		Yes	NULL	
text	varchar(200)		No		

Рисунок 2.7 — Таблиця dictionary бази даних

На рисунку 2.7 зображена допоміжна таблиця що містить інформацію, що необхідна для функціонування інформаційної системи, наприклад посади, що існують.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
position	varchar(255)		No		
role	varchar(255)		No		
start_work	date		No		
end_work	date		Yes	NULL	
people_id	int		No		

Рисунок 2.8 — Таблиця employee_info бази даних

Таблиця на рисунку 2.7 зберігає робочу інформацію про працівників. Використовується в модулі, що містить всі необхідні інструменти для роботи з кадровим складом.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
first_name	varchar(255)		No		
last_name	varchar(255)		No		
patronymic	varchar(255)		No		
birthdate	date		No		
region	varchar(255)		No		
phone	varchar(255)		No		
description	text		Yes	NULL	
fired	int		No	0	

Рисунок 2.9 — Таблиця перле бази даних

Таблиця на рисунку 2.9 зберігає персональну інформацію про працівників. Як і попередня таблиця, використовується в модулі для роботи з кадрами.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
partition_id	varchar(1000)		No		
user_id	int		No		
dt_create	datetime		No		
dt_update	datetime		No		
name	varchar(200)		No		
data	longtext		No		
keywords	varchar(200)		Yes	NULL	
shared_users	varchar(4000)		Yes	NULL	
shared	tinyint(1)		Yes	NULL	

Рисунок 2.10 — Таблиця help_articles бази даних

Таблиця на рисунку 2.10 зберігає данні про пам'ятки.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
parent_id	int		No		
user_id	int		No		
dt_create	datetime		No		
dt_update	datetime		No		
dt_delete	datetime		Yes	NULL	
name	varchar(100)		No		
title	varchar(200)		Yes	NULL	
open	tinyint(1)		Yes	NULL	
shared_users	varchar(4000)		Yes	NULL	
shared	tinyint(1)		Yes	NULL	

Рисунок 2.11 — Таблиця help_partitions бази даних

Таблиця на рисунку 2.11 зберігає данні про розділи, до яких відносяться пам'ятки.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
user_id	int		Yes	NULL	
ip	int		Yes	NULL	
dt	datetime		No		
url	varchar(500)		Yes	NULL	
short_url	varchar(200)		Yes	NULL	
title	varchar(500)		Yes	NULL	

Рисунок 2.12 — Таблиця monitor бази даних

Таблиця на рисунку 2.12 містить дані для моніторингу та аналізу дій користувачів в системі. Зберігаються дані частоти відвідування сторінок інформаційної системи.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
title	varchar(255)		No		
short_text	text		No		
text	longtext		No		
date	datetime		No		
photo	varchar(255)		Yes	NULL	
user_id	int		No		
datetime_lastedit	datetime		Yes	NULL	
public_date	date		Yes	NULL	

Рисунок 2.13 — Таблиця news бази даних

Таблиця на рисунку 2.13 відповідає за збереження даних новин.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
text	text		No		
date	datetime		No		
date_up	datetime		No		
type	tinyint(1)		Yes	NULL	
priority	int		No	5	
date_control	date		Yes	NULL	
up_from_date	date		Yes	NULL	
up_to_date	date		Yes	NULL	
user_id	int		No		

Рисунок 2.14 — Таблиця notices бази даних

Таблиця на рисунку 2.14 містить данні оголошень.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
title	varchar(255)		No		
short_text	text		No		
text	text		Yes	NULL	
date	datetime		No		
photo	varchar(255)		Yes	NULL	
user_id	int		No		
sorting	int		Yes	NULL	

Рисунок 2.15 — Таблиця persons бази даних

Таблиця на рисунку 2.15 відповідає за збереження даних керівників.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
tags_id	varchar(4000)		Yes	NULL	
user_id	int		No		
dt_create	datetime		No		
dt_update	datetime		No		
dt_deadline	datetime		Yes	NULL	
dt_completed	datetime		Yes	NULL	
type	int		Yes	NULL	
title	varchar(200)		Yes	NULL	
data	text		Yes	NULL	

Рисунок 2.16 — Таблиця tasks бази даних

Таблиця на рисунку 2.16 зберігає дані задач.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
user_id	int		No		
dt_create	datetime		No		
dt_update	datetime		No		
dt_delete	datetime		Yes	NULL	
name	varchar(100)		No		
use_default	tinyint(1)		Yes	NULL	

Рисунок 2.17 — Таблиця tasks_tags бази даних

Таблиця на рисунку 2.17 містить теги для задач.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
login	varchar(50)		No		
password	varchar(255)		No		
change_password	tinyint(1)		Yes	NULL	
lastname	varchar(255)		No		
firstname	varchar(255)		No		
surname	varchar(255)		Yes	NULL	
region	varchar(10)		Yes	NULL	
division	varchar(10)		Yes	NULL	
post	varchar(255)		Yes	NULL	
phone	varchar(20)		Yes	NULL	
access	int		No	0	
dt_lock	datetime		Yes	NULL	
comment	text		Yes	NULL	
username	varchar(255)		No		
name	varchar(255)		No		
role	varchar(20)		Yes	NULL	
permissions	varchar(200)		Yes	NULL	
homedir	varchar(200)		Yes	NULL	
last_login	datetime		Yes	NULL	
last_connect	datetime		Yes	NULL	

Рисунок 2.18 — Таблиця users бази даних

Таблиця на рисунку 2.18 зберігає дані зареєстрованих користувачів системи.

Column	Type	Attributes	Null	Default	Extra
id	int		No		auto_increment
user_id	int		No		
dt	datetime		No		
ip	int		Yes	NULL	
action	varchar(300)		No		
comment	varchar(300)		Yes	NULL	

Рисунок 2.19 — Таблиця users_log бази даних

Таблиця на рисунку 2.19 містить данні про авторизацію користувачів в системі. Наприклад, тут зберігаються відмітки про неправильне введення паролю успішну авторизацію, дату та час авторизації та ір користувача, що виконував ці дії.

2.5 Проектування та розробка функціональних алгоритмів роботи інформаційної системи

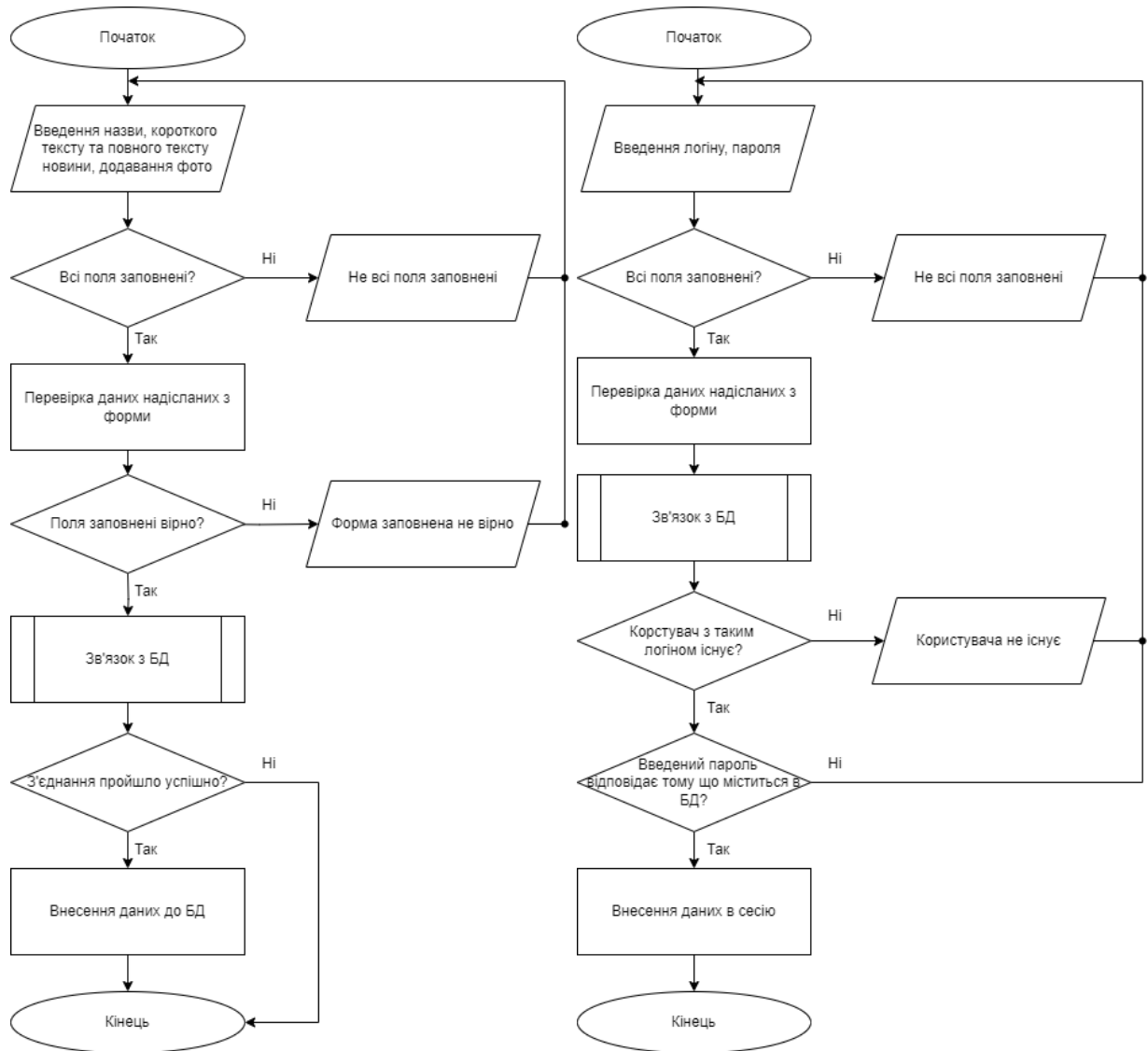
Після проектування, розробки структури та архітектури інформаційної системи варто приступити до розбиття програми на менші функціональні одиниці, для проектування роботи окремих алгоритмів. Окреме проектування алгоритмів роботи функцій дасть більш чітке уявлення про загальну роботу системи та більш реалістичне уявлення про необхідні алгоритми для вирішення технічних задач.

Розглянемо один з основних алгоритмів роботи системи — алгоритм додавання новини адміністратором рисунок 2.20(а) На першому етапі додавання новини адмін вводить основні дані, такі як заголовок, короткий текст, повний текст та фото новини. Після натискання користувачем кнопки «Додати» дані з форми за допомогою методу post передаються на сервер, де і проходить їх подальша обробка. Далі перевіряється чи всі поля є заповненими. У разі якщо перевірка виявить помилку користувача поверне на початковий етап — введення основних даних. Якщо перевірка пройшла успішно виконується з'єднання з БД та внесення до неї новини. Після всіх вище перелічених дій новина є доданою та відображається в списку новин.

Після згаданого вище алгоритму варто розглянути алгоритм входу адміністратора в свій обліковий запис рисунок 2.20(б).

В запропонованій формі входу у свій обліковий запис адмін вводить логін та пароль свого акаунту. Після заповнення відбувається перевірка чи всі поля заповнені. Якщо поля не заповнені адміністратора повертає на етап введення логіну та паролю. Якщо все заповнено — здійснюється зв'язок з БД та пошук у ній користувача з таким логіном. У випадку співпадіння даних, вони заносяться в сесію та користувач отримує повідомлення про успішний вхід до свого облікового запису. Якщо якась з перевірок не пройшла успішно, користувача повертає на форму введення з відповідним повідомленням.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		37



а) б)
Рисунок 2.20 — Блок- схеми алгоритмів

а) - додавання новин, б) – аутентифікації користувача

2.6 Реалізація основних функціональних можливостей інформаційної системи

При розробці інформаційної системи використовувався об'єктно-орінтований підхід. Було дотримано основних принципів підходу: інкапсуляція, поліморфізм, наслідування.

Було використано патерн синглтон для реалізації класа Core.php, для забезпечення функціонування єдиного екземпляру ядра в системі.

Розглянемо необхідний функціонал класу ядра системи.

Створено приватне поле для збереження єдиного екземпляру класу:

```
private static $instance = null;
```

Створено приватний конструктор для запобігання прямого створення об'єктів класу, а також метод для отримання єдиного екземпляру класу:

```
private function __construct() {
    // Ініціалізація класу
}

public static function getInstance()
{
    if (empty(self::$instance)) {
        self::$instance = new Core();
        return self::getInstance();
    } else return self::$instance;
```

Розглянемо клас Controller.php — базовий клас для всіх контролерів, а саме метод monitor(). Він відповідає за моніторинг відвідуваності сторінок сайту користувачами.

```
protected function monitor($title = null){
    if(!$this->access(true, 'monitor/donot')){
        $do_monitor = true;
        if(!empty($this->config->no_monitoring_uri)){
            foreach ($this->config->no_monitoring_uri as $uri){
                if(preg_match("/{ $uri }/i", $_SERVER['REQUEST_URI'])){
                    $do_monitor = false;
                    break;
                }
            }
        }
        if($do_monitor) {
            $monitor = new \models\Monitor($this->access);
            $monitor->add($this->user['id'] ?? null, $_SERVER['REQUEST_URI'],
            $title);
        }
    }
}
```

Метод є доступним тільки всередині класу, або його нащадків для забезпечення інкапсуляції. Він приймає параметр \$title, який за замовчуванням має значення null. На початку виконується перевірка прав доступу користувача до даного метода if(!\$this->access(true, 'monitor/donot')). Якщо користувач має відповідні права відбувається ініціалізація змінної \$do_monitor = true ('monitor/donot' - зазначається в конфігурації, є позначенням того, що моніторинг дій в системі для цього користувача відключено). Далі відбувається перевірка чи є порожнім масив no_monitoring_uri в конфігураційному файлі, якщо масив не порожній виконується цикл. В циклі виконується перевірка чи

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		39

відповідає значення `$uri` поточному запиту. Якщо відповідність існує, то змінюється значення флажка `$do_monitor` на `false`, цикл зупиняється.

Якщо значення `$do_monitor` залишається `true` створюється об'єкт класу `Monitor $monitor = new \models\Monitor($this->access)`, далі відбувається виклик `add()` — додавання сторінки в моніторинг.

Клас `DataBase.php` містить усі необхідні методи для роботи базою даних. Основною задачею класу є встановлення з'єднання з БД та взаємодія з нею. З'єднання реалізується функцією `__construct` за допомогою PDO:

```
public function __construct($type, $charset, $hostname, $database, $username,
    $password)
{
    try {
        $this->pdo = new
        \PDO("{ $type } : host={ $hostname }; dbname={ $database }; charset={ $charset }",
        $username, $password, array(\PDO::ATTR_ERRMODE => \PDO::ERRMODE_SILENT));
        $this->init();
        $this->init_last();
    } catch (\PDOException $e) {
        print "Error!: " . $e->getMessage(). "<br/>";
        die("Помилка підключення до бази даних!");
    }
}
```

Конструктор приймає: тип бази даних, кодування, ім'я хоста, базиданих та користувача, пароль для підключення. Далі відбувається створення об'єкта PDO, виклик методів `init()` та `init_last()` для подальшого налаштування роботи з БД. Для перехоплення винятків використовується блок `cath`.

Розглянемо ще декілька методів класу `DataBase.php`: `order_by()` — для констроювання запиту з оператором `order by`; `group_by` — для створення запиту з `group by`. Для можливості побудови більш складних запитів оптимально створено функцію `join_left()`:

```
public function join_left(string|array $from, string|array $on) :DataBase{
    $this->join('left', $from, $on);
    return $this;
}
```

Подібно представленому коду створено наступні методи: `join_right()`, `join_inner()`, `join_full()`, `join()`.

Розглянемо модель модуля новин — клас News.php. Клас містить всі необхідні методи для збереження моделі новин та роботи з відповідними даними.

Розглянемо метод update():

```
public function update($row, $id, $validate = true){

    $validateResult = $validate?$this->Validate($row):true;
    if(is_array($validateResult))
        return $validateResult;
    $fields = ['title', 'short_text', 'text', 'photo', 'public_date'];
    $rowFiltered = Utils::ArrayFilter($row, $fields);
    $rowFiltered['datetime_lastedit'] = date('Y-m-d H:i:s');
    if (empty($rowFiltered['public_date'])){
        $rowFiltered['public_date'] = null;
    }
    if($this->db->update($this->dbTable, $rowFiltered, ['id'=>$id])==false){
        return false;
    }
    return true;
}
```

Поданий метод використовується оновлення новини.

Якщо параметр \$validate встановлено в true, метод викликає функцію validate() з переданими даними - \$row. Якщо валідація не вдається, метод повертає результат валідації.

Метод фільтрує вхідні дані - \$row, на основі зазначеного списку полів \$fields. У фільтрованих даних \$rowFiltered залишаються лише поля, зазначені в цьому списку (title, short_text, text, photo, public_date).

Метод встановлює поле 'datetime_lastedit' у фільтрованих даних на поточну дату і час. Якщо поле 'public_date' у фільтрованих даних порожнє, воно встановлюється в null.

Метод оновлює вказану таблицю бази даних \$this->dbTable фільтрованими даними за допомогою зазначеного ідентифікатора \$id. Якщо операція оновлення не вдається, метод повертає false. Якщо операція оновлення вдається, метод повертає true.

З метою оновлення даних, подібно представленому вище, реалізовано подібні методи update() у інших моделях, класах: Notices.php, Departments.php, Tasks.php.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		41

Розглянемо контролер модуля для обліку кадрів , а саме частину методу, яка відповідає за додання запису в історію служби працівника:

```
if ($this->isPost()) {

    $_POST['people_id'] = $id;
    if (!empty($_POST['id'])) {
        $result = $this->employeeModel->updateHistoryRow($_POST);
    } else {
        $result = $this->employeeModel->add($_POST);
    }

    if ($result['error'] === false) {

        return $this->renderInformMessage('success', 'Зміни збережено',
            [
                'redirect_url' => $this->url,
                'redirect_timeout' => $this->redirect_timeout
            ],
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    } else {
        $message = implode('<br/>', $result['messages']);
        return $this->render('form', ['model' => $_POST], [
            'MainTitle' => $title,
            'PageTitle' => $title,
            'InformText' => $message,
            'InformClass' => 'danger'
        ]);
    }
}
```

Умова `if ($this->isPost())` перевіряє, чи був викликаний запит методом POST. Метод `isPost()` визначений і перевіряє, чи запит є HTTP POST-запитом. У тілі умови встановлюється значення `$_POST['people_id']` рівним `$id`, це ідентифікатор даних особи, який призначений запису у базі даних. Далі відбувається подальша обробка форми. Якщо `$_POST['id']` не є порожнім (якщо існує значення ідентифікатора), то викликається метод `$this->employeeModel->updateHistoryRow($_POST)` для оновлення рядка в базі даних. Якщо `$_POST['id']` порожній (новий запис), викликається метод `$this->employeeModel->add($_POST)` для додавання нового запису у базу даних. Результат виклику методу (по замовчуванню повертається асоціативний масив) зберігається у змінну `$result`. Перевіряється, чи в результаті виклику була помилка `$result['error'] === false`.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		42

Якщо все пройшло успішно, відбувається перенаправлення користувача на заданий url із повідомленням про успіх. У протилежному випадку формується повідомлення про помилку. В останньому return відбувається виклик методу render() для відображення представлення.

Подібну структуру містять методи реалізовані в контролерах, що відповідають за обробку запиту користувача на додавання об'єкта до бази даних. Наприклад, метод actionAdd() в контролері News.php відповідає за додавання новини. Він містить подібну частину коду, але звертається в свою чергу до іншої моделі та має іншу валідацію.

Розглянемо частину коду контролера Users.php, а саме функції actionLogin(), яка відповідає за аутентифікацію користувача в системі:

```
if(!empty($user))
{
    if(empty($user['change_password'])){
        $_SESSION['user'] = $user;
        return $this->renderInformMessage("success", "Ви успішно увійшли на
сайт<script
type='text/javascript'>$('#btn_logout').addClass('disabled');</script>",
[
    'redirect_url' => $this->redirect_url,
    'redirect_timeout' => $this->redirect_timeout
],
[
    'MainTitle' => $title,
    'PageTitle' => $title
]);
    }
}
```

Спочатку виконується перевірка чи користувач вже не авторизований в системі, шляхом перевірки чи змінна \$user не порожня. Якщо в цій змінній містився б об'єкт це означало що користувач вже авторизований в системі. Далі виконується перевірка чи не змінював користувач пароль, перевіряється ключ 'change_password' асоціативного масиву \$user[]. У випадку якщо зміни паролю не відбувалося об'єкт користувача зберігається в сесію. На останок викликається

функція , яка виводить повідомлення про успішну авторизацію. Повертаються данні для сторінку та url.

Протилежно працює метод `actionLogout()`, він відповідає за вихід користувача з системи, здійснюється видалення даних користувача з сесії.

Висновки до другого розділу

В результаті виконання другого розділу кваліфікаційної роботи було визначено та описано варіанти використання інформаційної системи. Система передбачає чотири ролі: не зареєстрований користувач, зареєстрований користувач, менеджер, адміністратор. Розподілення можливостей буде наступне — найвищий доступ має адміністратор, а найнижчий не зареєстрований в системі користувач

Було розроблено структуру системи. Вона налічує 12 модулів, кожен з яких працює окремо. Спроектовано архітектуру, при розробці системи використано патерн MVC. Розглянуто основні класи: *Core*, *DataBase*, *Controller*, *Model*, *Template*, *Util*, моделі, контролери та представлення.

База даних налічує 17 таблиць та містить зв'язки: один до одного, один до багатьох, багато до одного, багато до багатьох.

Розроблено та представлено функціональні алгоритми, розглянуто алгоритми додавання новини та аутентифікації користувача.

Після виконання проектування системи, було почато розробку програмного коду. Для написання програмного продукту було обрано об'єкто-орієнтований підхід з дотриманням основних принципів: інкапсуляція, поліморфізм, наслідування.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				44
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ОГЛЯД ІНТЕРФЕЙСУ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Огляд інтерфейсу інформаційної системи (користувач)

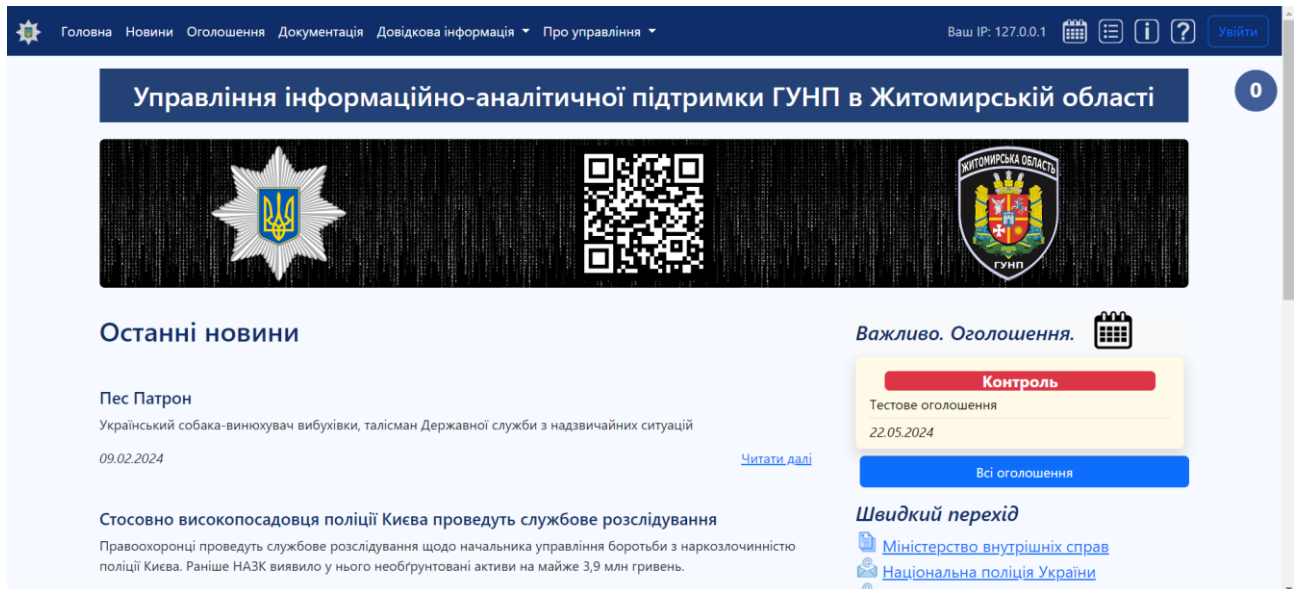


Рисунок 3.1 — Головна сторінка системи

На головній сторінці, що зображена на рисунку 3.1, користувач може переглянути 5 останніх новин, оголошення, галерею, основні посилання.

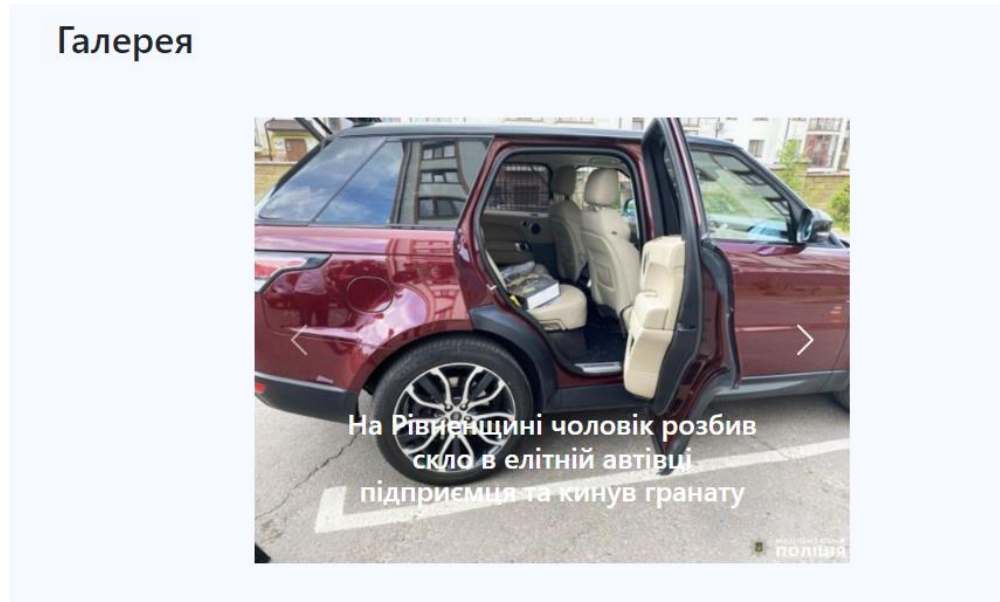


Рисунок 3.2 — Галерея

Галерея, що зображена на рисунку 3.2, побудована за принципом слайд-шоу, вона формується з фото доданих до новин.

Натиснувши на «Новини» користувач переходить в розділ Новини — рисунок 3.3.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		45

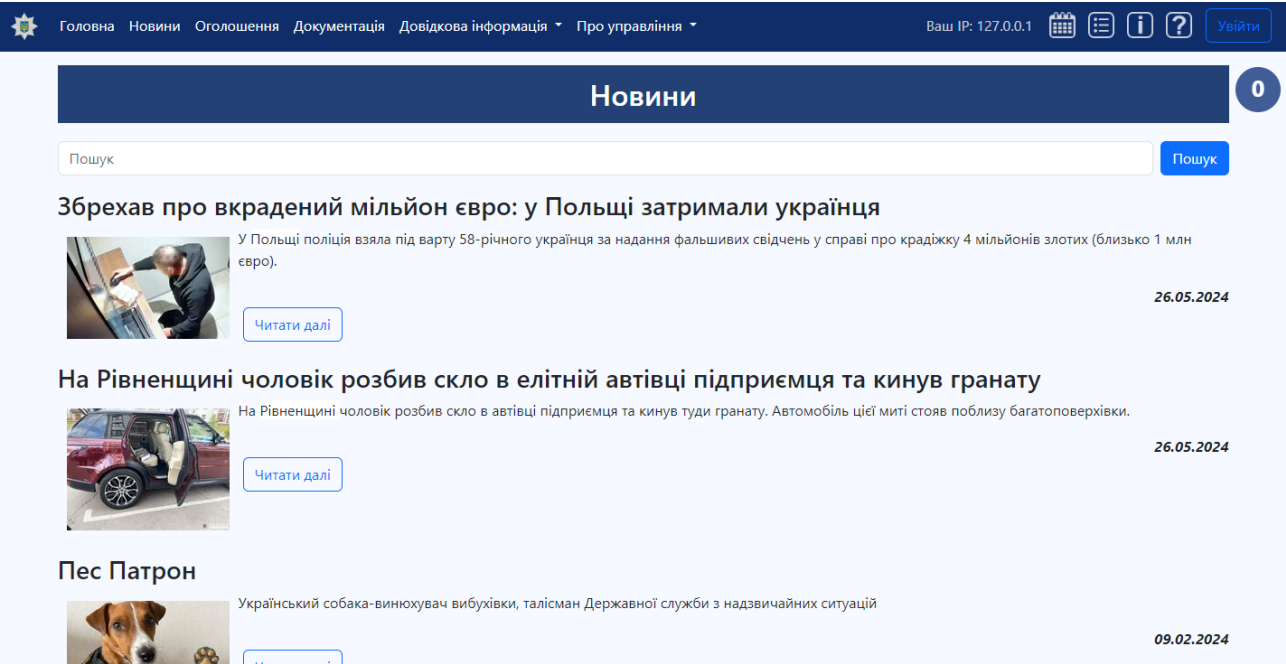


Рисунок 3.3 — Сторінка новин

Користувач має змогу здійснити пошук новин та переглянути новину детальніше — рисунок 3.4.



Рисунок 3.4 — Перегляд новини

Натиснувши на «Оголошення» користувач переходить в розділ Оголошення — рисунок 3.5. Тут користувач бачить оголошення, опис, тип оголошення та дату його публікації. Актуальні оголошення відображаються вище, далі відображаються не актуальні. Після втрати актуальності оголошення ще 10 днів зберігається в розділі «Неактуальні».

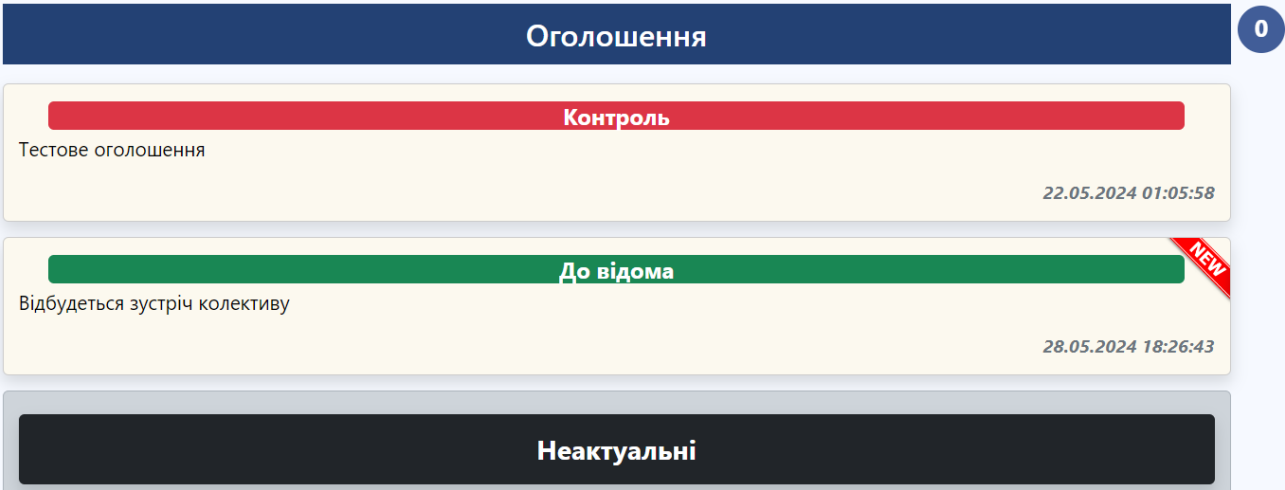


Рисунок 3.5 — Сторінка оголошень

Натиснувши на «Документація» користувач переходить в розділ Документація — рисунок 3.6. Можливості користувача тут залежать від дозволів до файлової системи, які надаються адміністратором. Даний користувач має змогу завантажувати та вивантажувати файли, додати, видалити та редагувати каталоги також є можливість завантажити весь каталог у вигляді архіву.

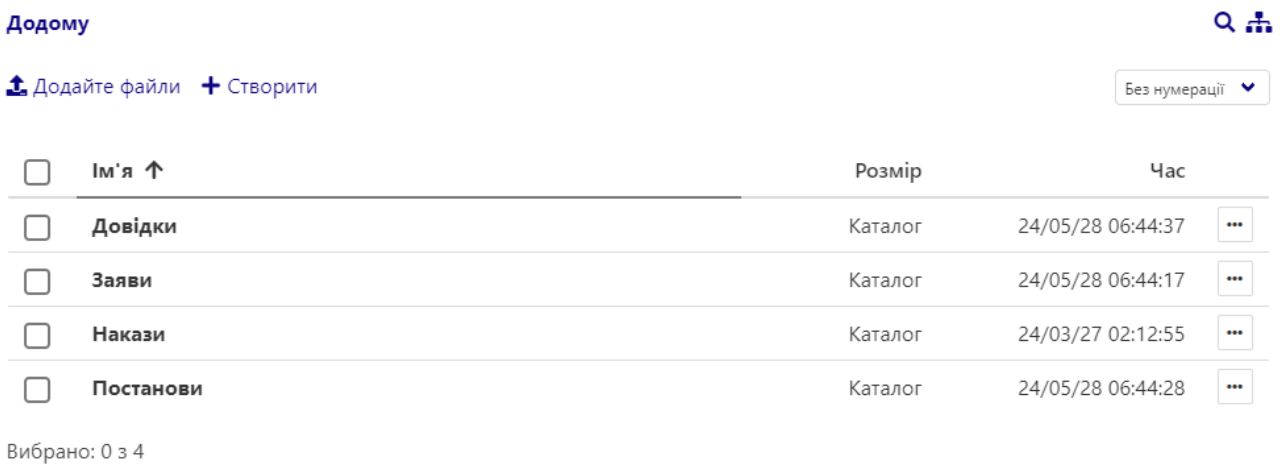


Рисунок 3.6 — Файлова система

Натиснувши на «Довідкова інформація» користувач бачить випадаюче меню як на рисунок 3.7.

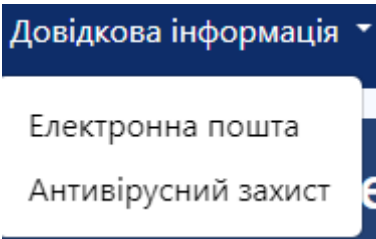


Рисунок 3.7 — Випадаюче меню модуля «Довідкова інформація»

Натиснувши «Електронна пошта» користувач бачить інформацію по пошті та посилання на неї — рисунок 3.8.

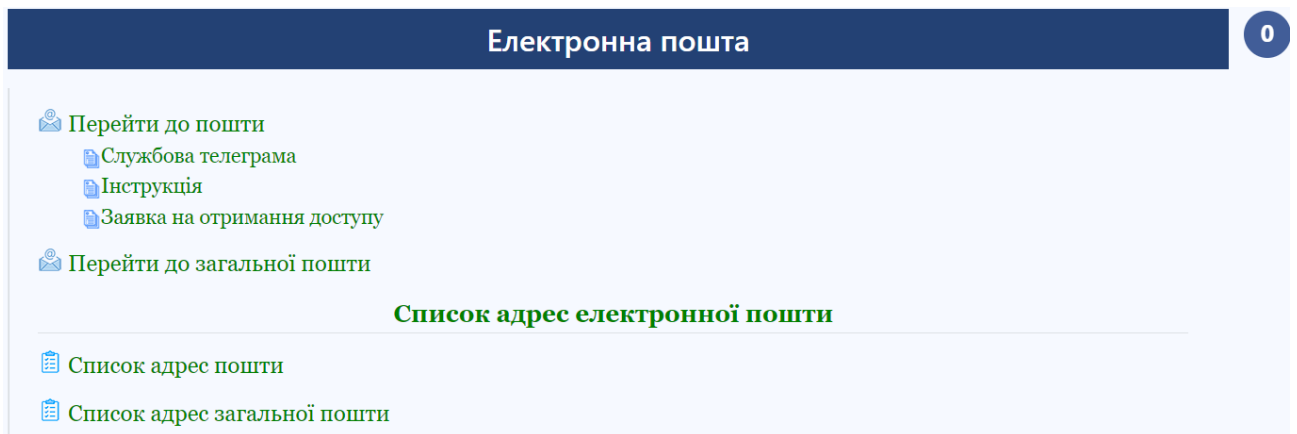


Рисунок 3.8 — Довідкова інформація по пошті

Натиснувши «Антивірусний захист» користувач бачить інформацію по антивірусному програмному забезпеченню та посилання на нього — рисунок 3.9.

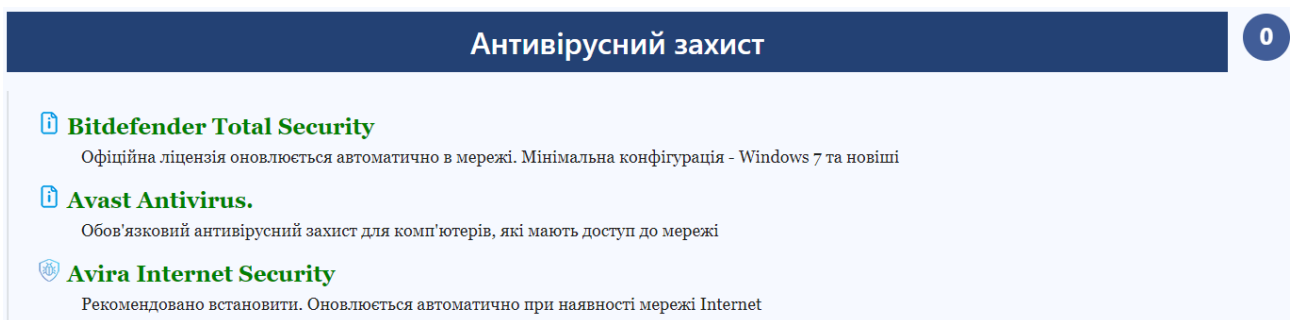


Рисунок 3.9 — Довідкова інформація по антивірусному програмному забезпеченню

Натиснувши на «Довідкова інформація» користувач бачить випадające меню як на рисунок 3.10.

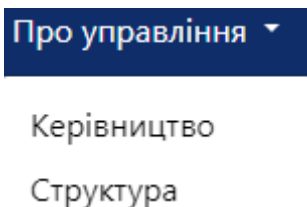


Рисунок 3.10 — Випадающее меню модуля «Про управління»

Натиснувши «Керівництво» користувач бачить інформацію про керівництво управління — рисунок 3.11.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		48

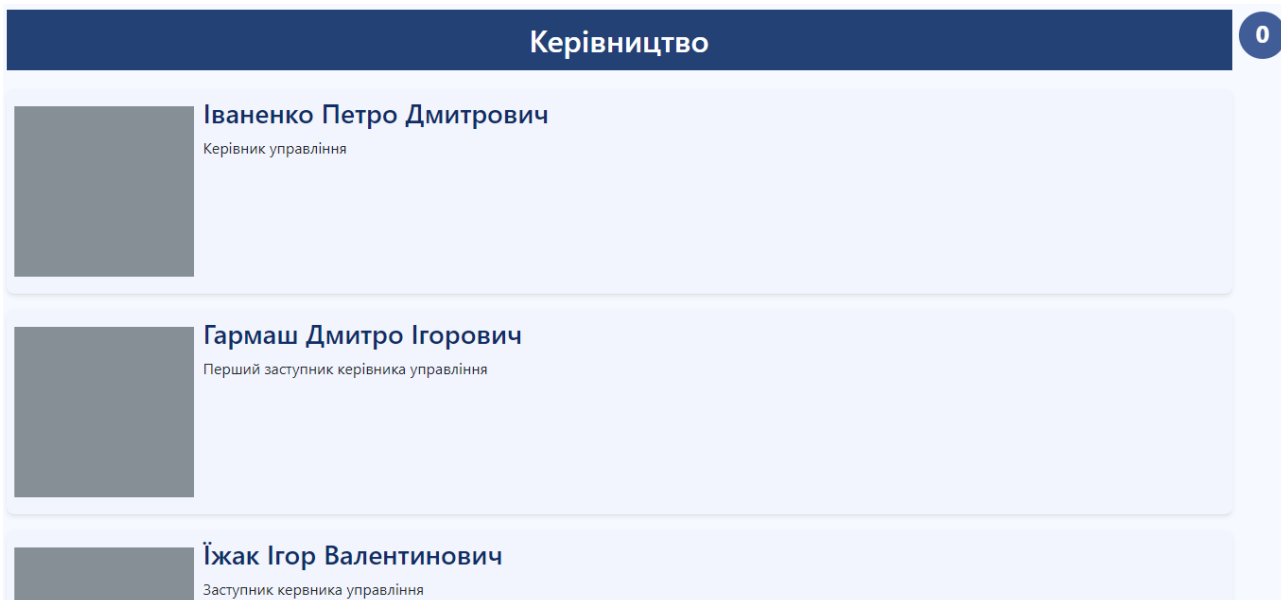


Рисунок 3.11 — Інформація про керівництво

Натиснувши «Структура» користувач бачить інформацію про структуру управління — рисунок 3.12.

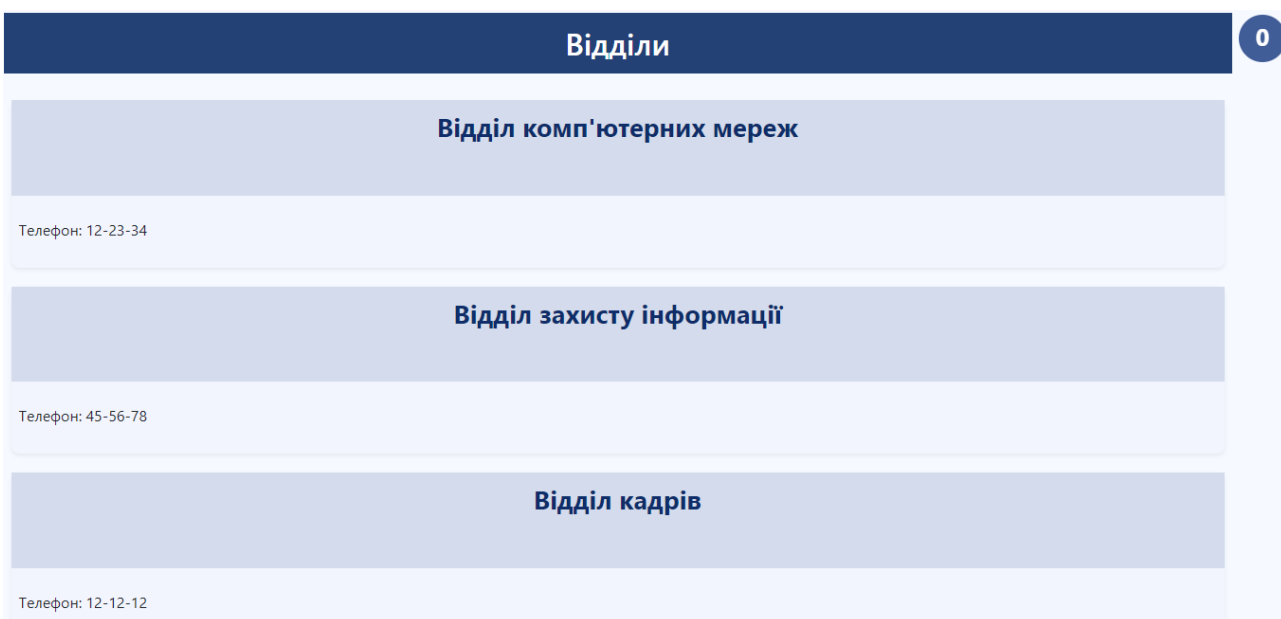


Рисунок 3.12 — Інформація про структуру управління

Натиснувши на користувач переходить в розділ Кадендар — рисунок 3.13. Тут користувач має змогу створювати календарі різних типів (загальнодоступні, приватні, доступні обраним користувачам) — рисунок 3.14, переглянути календар поточного місяця — рисунок 3.15. Користувач може додавати, видаляти редагувати записи в календарі — рисунок 3.16 перегляд календарів доступних користувачеві — рисунок 3.17. Є можливість перегляду

всіх записів календаря — рисунок 3.18, користувач може перемикатися між записами окремих календарів, обирати поточні та майбутні події.

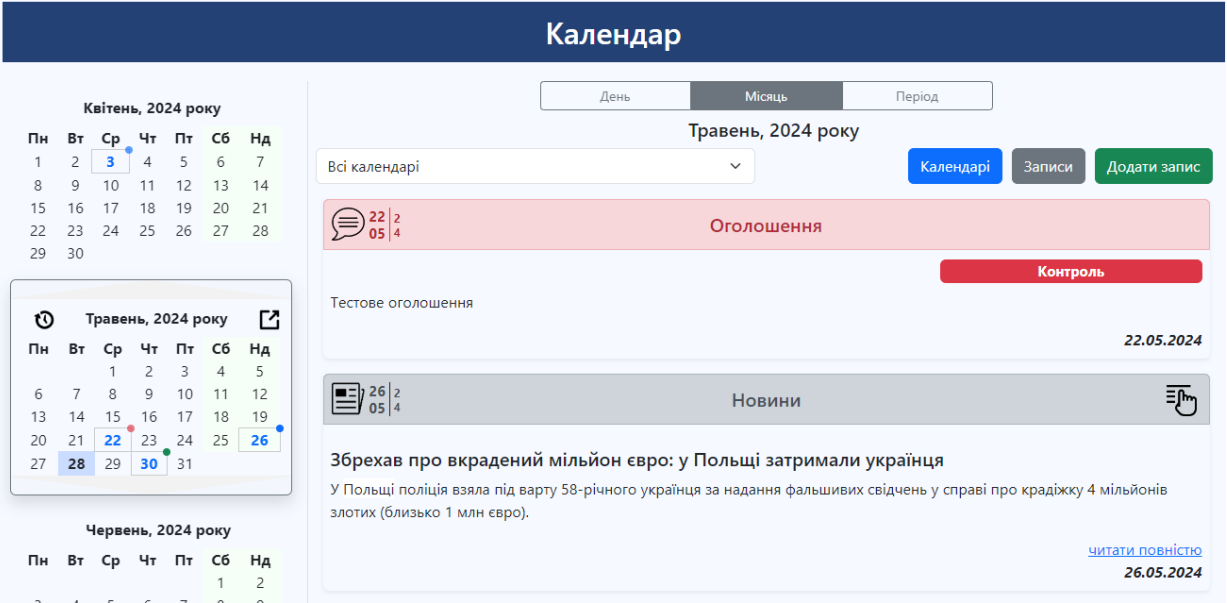


Рисунок 3.13 — Основна сторінка модуля «Календар»

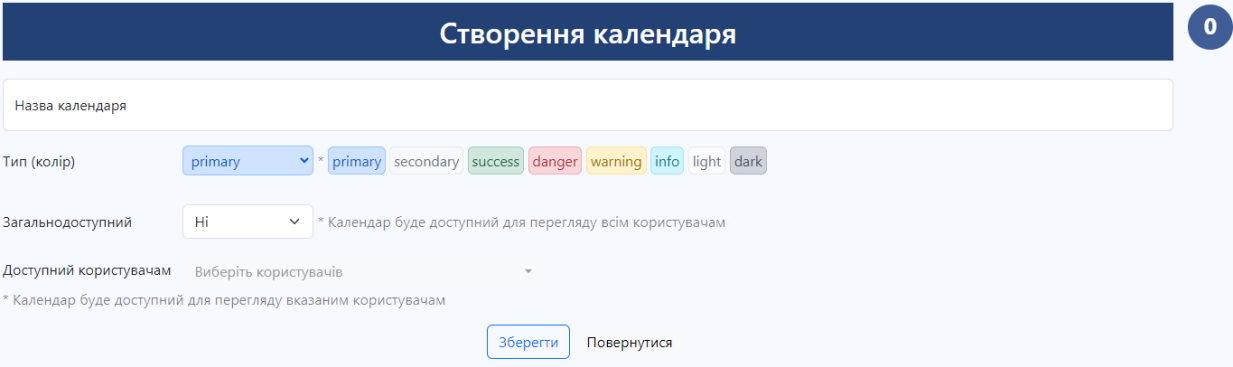


Рисунок 3.14 — Створення календаря

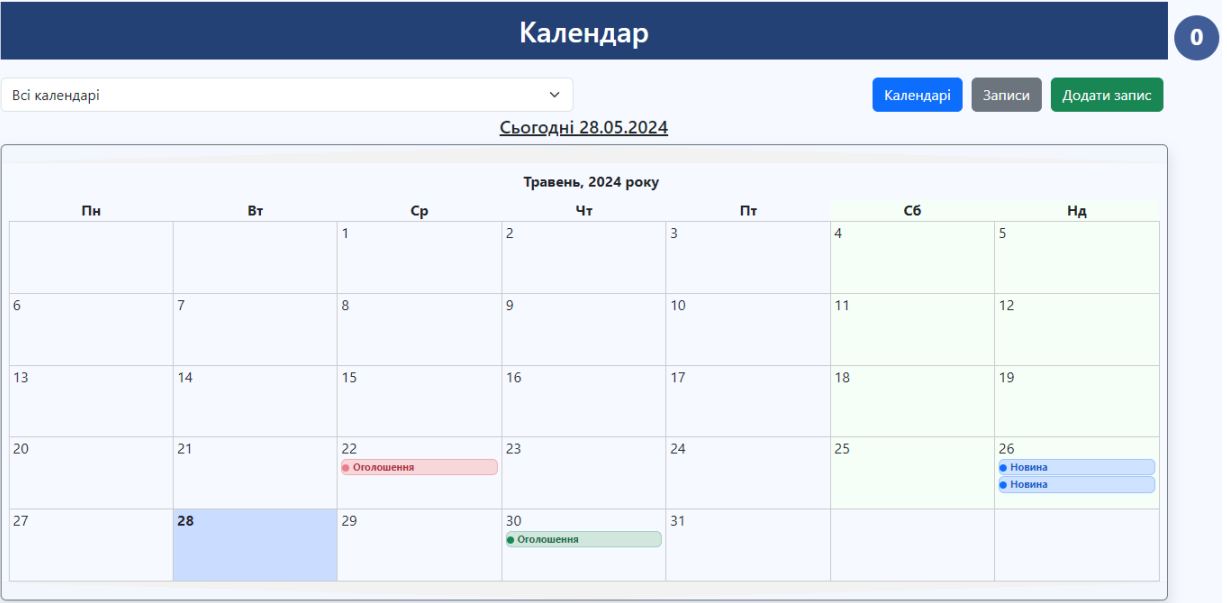


Рисунок 3.15 — Перегляд поточного календаря місяця

Додавання запису календаря

0

Календар

Виберіть календар

* Вибейть календар, до якого додати запис

Заголовок

Текст

ФайлРедагуватиВиглядВставитиФорматТаблицяДовідка

↶↷BBIU↺Системний ш...12ptАбзац

≡≡≡≡≡≡≡≡⋮⋮⋮⋮

A✎IΩ☺🔍🖼️▶️🔗📌{t}🔧🔑

Список календарів						
					Повернутися	Створити календар
#	Назва	Тип (колір)	Загальнодоступний	Доступний вибраним користувачам	Користувач	Не відображати
1	MyCalendar	light	Hi	Так		☐
2	Календар Влад	primary	Hi	Так	vlad	☐

Записи календаря

0

Майбутні події

Всі події

Всі календарі

Список календарів

Додати запис

09 2

01 4

Конференція

Підключитися до онлайн наради

09.01.2024

09 2

01 4

Перевірити звіти

Перевірити звіти

09.01.2024

03 2

04 4

Документи

Зробити звіти

03.04.2024

Рисунок 3.18 — Записи календаря

Натиснувши на  користувач переходить в розділ Задачі — рисунок 3.19.

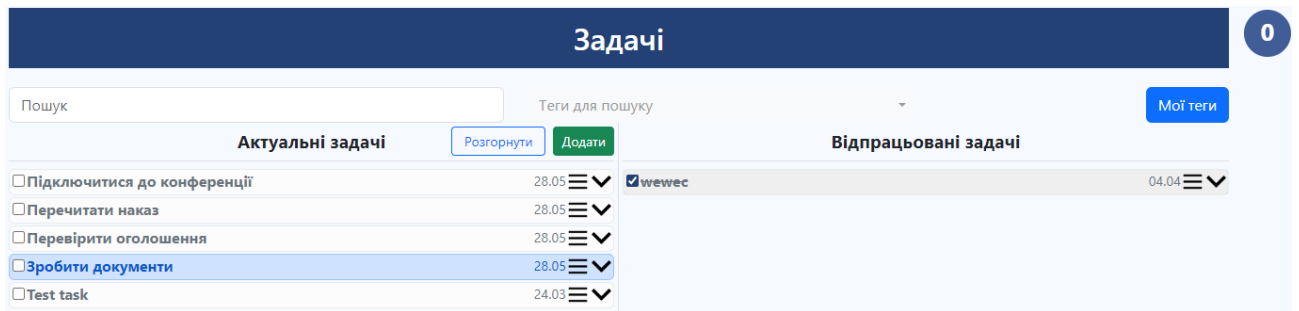


Рисунок 3.19 — Модуль задач

В даному модулі користувач має змогу створення, видалення та редагування задач, пошук, додавання тегів, пошук за тегами, переміщення задач до відпрацьованих, поновлення відпрацьованих задач в списку актуальних, додавання, видалення та редагування тегів.

Натиснувши на  користувач переходить в розділ Пам'ятки — рисунок 3.20.

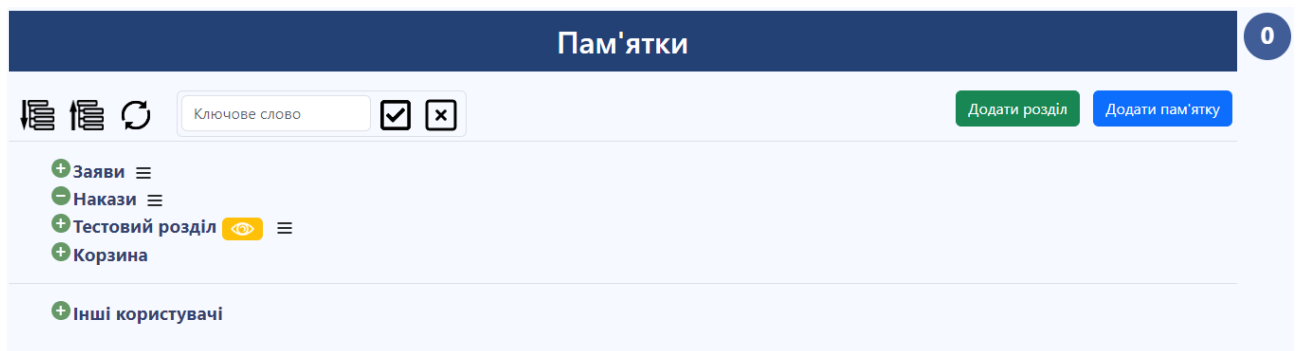


Рисунок 3.20 — Пам'ятки

В даному модулі користувач має змогу створення, видалення та редагування пам'яток, пошук за ключовими словами, створення та редагування розділів для пам'яток.

Натиснувши на логін користувача, в правому верхньому куті екрана, користувач переходить до власного профілю — рисунок 3.21. Тут користувач має змогу переглянути особисту інформацію та змінити пароль.

Інформація про користувача

0

Основні дані користувача

Логін

vlad

Рівень доступу *

Користувач

Прізвище

Райковський

Ім'я

Владислав

По батькові

Андрійович

Регіон

Вкажіть регіон

Служба

Вкажіть службу

Посада

спеціаліст з захисту інформації

Телефон

0671111111

Пароль ***

Залиште пустим що б не змінювати

Повтор паролю

Має співпадати з полем 'Пароль'

*** Пароль повинен містити від 8 до 20 символів, хоча б одну цифру та одну літеру

Рисунок 3.21 — Інформація про користувача

3.2 Тестування системи

Тестування є важливим на завершальному етапі перед введенням системи в експлуатацію. Воно допомагає уникнути проблем та непередбачуваних наслідків, які можуть виникнути як наслідок впровадження системи, яка має помилки, чи працює не коректно.

Було проведено функціональне тестування інформаційної системи. Тестування здійснювалось помодульно, тобто брався окремий модуль системи, здійснювалось його тестування [17]. Розглянемо декілька тест-кейсів модуля задачі.

TestCase 1

Test Case ID	001	Test Case Title	Add a task		
Tester's Name	Vladyslav Raikovskiyi	Date Tested	16-Mar-2024	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Open site: http://uiap/
2	
3	
4	

S #	Test Data
1	Title = "Test task"
2	Type (color) = "light"
3	Description = "Description of the test task"
4	Tags = "Documents"

Test Description		Create a task, display it in the task list			
Step #	Step Details	Expected Results	Actual Results		

				Pass / Fail / Not executed / Suspended
1	Go to the task module (click on the Tasks icon)			Pass
2	Click "Add" button			Pass
3	Enter the Title, type, description, search tags			Pass
4	Click "Save" button	The task is saved and displayed in the general list of tasks	As Expected	Pass

TestCase 2

Test Case ID	002	Test Case Title	Edit the task		
Tester's Name	Vladyslav Raikovskiy	Date Tested	16-Mar-2024	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Open site: http://uiap/
2	Created task
3	
4	

S #	Test Data
1	Title = "New test task"
2	Type (color) = "light"
3	Description = "New description of the test task"
4	Tags = "Documents"

Test Description	Edit the task, display it in the task list			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Go to the task module (click on the Tasks icon)			Pass
2	Click "More" button on the task			Pass
3	Click "Edit" button			Pass
4	Edit the Title, type, description, search tags			Pass
5	Click "Save" button	The update task will be saved and displayed in the general task list	As Expected	Pass

TestCase 3

Test Case ID	003	Test Case Title	Complete the task		
Tester's Name	Vladyslav Raikovskiy	Date Tested	16-Mar-2024	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Open site: http://uiap/
2	Created task

S #	Test Data
1	
2	

Test Description	Create a task, display it in the task list			
Step #	Step Details	Expected Results	Actual Results	

				Pass / Fail / Not executed / Suspended
1	Go to the task module (click on the Tasks icon)			Pass
2	Click on the checkbox on the task	The task has disappeared from the list of active tasks, it is displayed in the list of completed tasks	As Expected	Pass

TestCase 4

Test Case ID	004	Test Case Title	Make the task relevant		
Tester's Name	Vladyslav Raikovskiy	Date Tested	16-Mar-2024	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:	S #	Test Data
1	Open site: http://uiap/	1	
2	Completed task	2	

Test Description	Resume a completed task, move it from the list of completed tasks to the list of current tasks			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Go to the task module (click on the Tasks icon)			Pass
2	Click on the checkbox on the completed task			Pass
3	Click the "Restore" button, confirm the recovery	The task has disappeared from the list of completed tasks, it is displayed in the list of active tasks	As Expected	Pass

TestCase 5

Test Case ID	005	Test Case Title			
Tester's Name	Vladyslav Raikovskiy	Date Tested	16-Mar-2024	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:	S #	Test Data
1	Open site: http://uiap/	1	
2	Created task	2	

Test Description	Remove a task from the list of current tasks			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Go to the task module (click on the Tasks icon)			Pass

2	Click "More" button on the task			Pass
3	Click "Delete" button			Pass
4	Confirm the deletion of a task	The task has been deleted and is not displayed	As Expected	Pass

В результаті виконання тестування помилок не було виявлено, система функціонує так як передбачалося.

Висновки до третього розділу

В результаті виконання третього розділу було детально оглянуто особливості інтерфейсу розробленої інформаційної системи та проаналізовано всі його можливості. У вигляді скриншотів показано роботу та функціонал кожного з модулів системи.

Виконано функціональне тестування шляхом розбиття системи на модулі, та окреме тестування функціоналу кожного модуля. При тестуванні промилків та дефектів системи не було виявлено, всі бізнес процеси відбуваються по запланованому сценарію.

Спираючись на отримані результати тестування можна сказати що система відповідає зазначеним вимогам та може бути введена в експлуатацію на підприємстві.

ВИСНОВОК

В результаті виконання кваліфікаційної роботи було запропоновано вирішення проблеми, що пов'язана з оптимізацією та цифровізацією внутрішніх процесів Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області шляхом створення інформаційної системи.

Було проведено аналіз внутрішніх процесів управління, розроблено алгоритми та методи оптимізації цих процесів. Спираючись на результати аналізу сформовано вимоги до інформаційної системи. Виконавши всі попередні кроки було розроблено інформаційну систему внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області, що включає в себе 12 модулів: календар, корисні джерела, користувачі, моніторинг, новини, сповіщення, люди, головна сторінка, задачі, довідка, працівники, відділи. Було виконано тестування розробленого програмного продукту, в результаті можна зробити висновок, що система повністю відповідає заявленим вимогам, не містить помилок та дефектів що впливають на коректність роботи системи.

Розроблена інформаційна система має високу практичну цінність. Вона підвищує продуктивність та організованість роботи управління, оптимізовує внутрішні процеси та спрощує доступ до необхідної користувачам інформації. Впровадження системи надасть можливість вийти на новий рівень роботи управління.

Розроблена інформаційна система була успішно розгорнута та впроваджена в роботу управління ГУНП в Житомирській області. В результаті впровадження до внутрішньої локальної мережі зазначеного управління, дана інформаційна система продемонструвала стабільну роботу та повністю задовільняє потреби користувачів.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		57

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Рейтинг мов програмування 2023 [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/language-rating-2023/>.
- [2] Рейтинг мов програмування 2024 [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/language-rating-2024/>.
- [3] Exploring the Advantages and Disadvantages of PHP [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://wpwebinfotech.com/blog/advantages-and-disadvantages-of-php/>.
- [4] Граф М.С., Райковський В.А. Використання JavaScript в розробці сучасного програмного забезпечення. Тези III Всеукраїнської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих учених "Інформаційно-комп'ютерні технології: стан, досягнення та перспективи розвитку", 25-26 листопада 2021 року. Житомир : «Житомирська політехніка», 2021. С.111-113.
- [5] JavaScript - краща мова для програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://apeps.kpi.ua/javascript-krashcha-mova-dlia-programuvania>.
- [6] PhpStorm Empowering PHP Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/phpstorm/>.
- [7] Apache HTTP Server Project [Електронний ресурс] – Режим доступу до ресурсу: <https://httpd.apache.org/dev/devnotes.html>.
- [8] Sergey Z. PHP OOP Way / Zhuk Sergey., 2021. – 249 с.
- [9] 10 Rules of Good UI Design to Follow On Every Web Design Project [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.elegantthemes.com/blog/resources/10-rules-of-good-ui-design-to-follow-on-every-web-design-project>.
- [10] Граф М.С., Райковський В.А. Розробка структури інформаційної системи внутрішньої мережі Управління інформаційно-аналітичної підтримки ГУНП в Житомирській області. Тези XIV Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології –

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				58
Змн.	Арк.	№ докум.	Підпис	Дата		

2024», 28 – 29 березня 2024 року. – Житомир: «Житомирська політехніка», 2024. С.119-121.

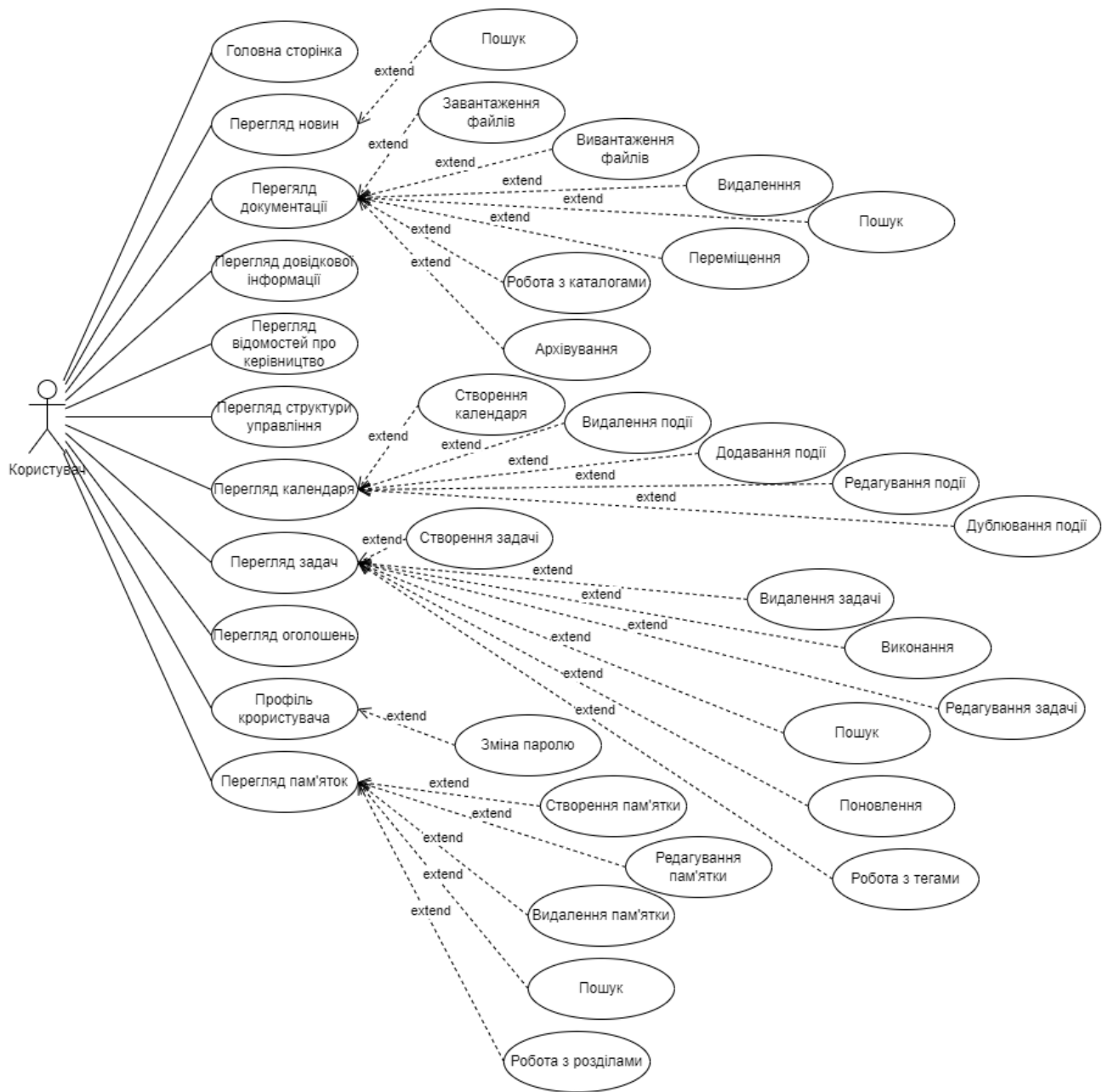
- [11] Райковський В.А., Вольський Р.А. Використання сучасних архітектурних підходів для розробки інформаційної системи для управління інформаційно-аналітичної підтримки. Тези Інтегровані інтелектуальні робототехнічні комплекси (ІРТК-2024). Сімнадцята міжнародна науково-практична конференція, 12-22 травня 2024 року. — Київ: НАУ, 2024. С.419-421.
- [12] Мурза С.П., Азарова А.О. Використання рольової моделі управління доступом для запобігання витоку інформації через співробітників підприємства. Тези конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2019), 11-30 травня 2019 року. — Вінниця: ВНТУ, 2019.
- [13] Information Systems for Business and Beyond / D.BOURGEOIS, J. MORTATI, S. WANG, J. SMITH., 2019. – 324 с. – (Biola University).
- [14] Граф М.С., Райковський В.А. Алгоритми оптимізації даних. Комп'ютерні технології: інновації, проблеми, рішення: тези VI Всеукраїнської науково-технічної конференції, 29-30 листопада 2023 року. Житомир : «Житомирська політехніка», 2023. С.143-144.
- [15] Романько Р.О., Кожем'яко А.В. Актуальність реляційних баз даних. Тези LII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії, 21-23 червня 2023 року. — Вінниця: ВНТУ, 2023.
- [16] MySQL 8.4 Reference Manual [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/8.4/en/>.
- [17] Модульне тестування [Електронний ресурс] – Режим доступу до ресурсу: <https://qalight.ua/baza-znaniy/modulne-testuvannya/>.

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				59
Змн.	Арк.	№ докум.	Підпис	Дата		

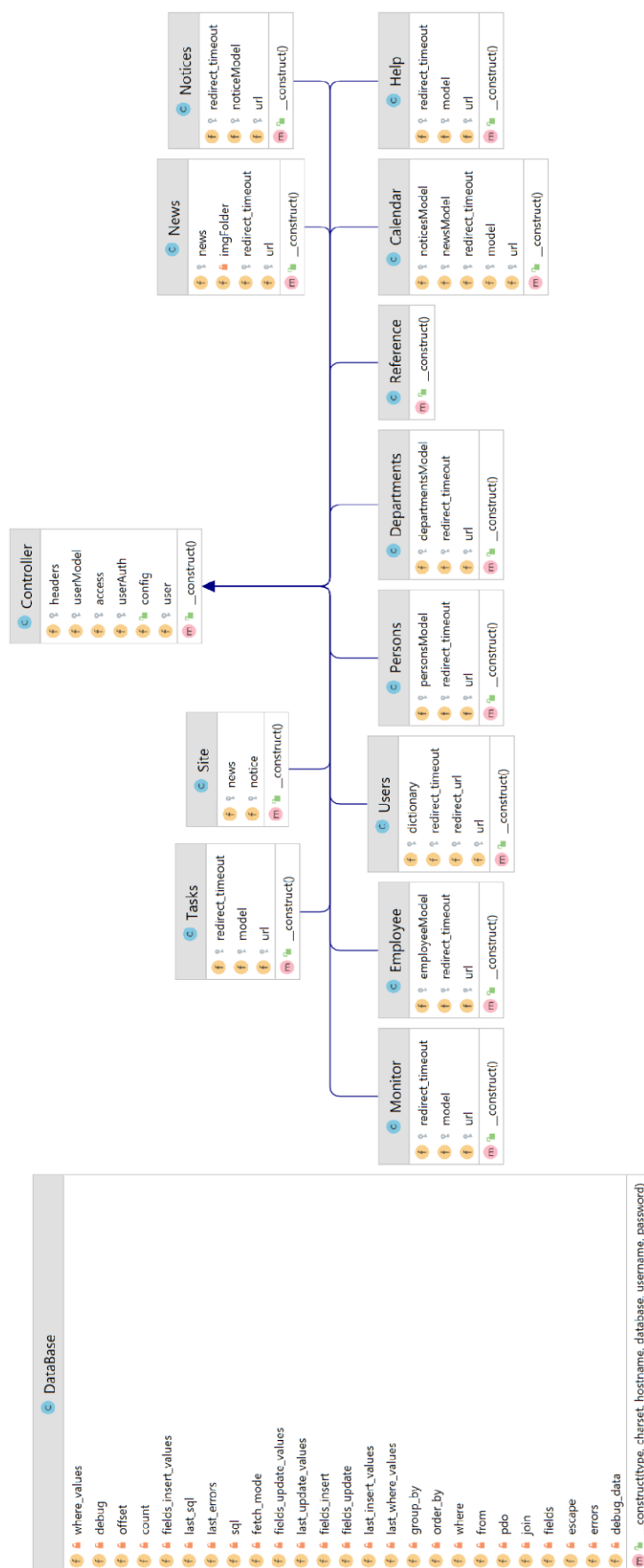
ДОДАТКИ

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		60

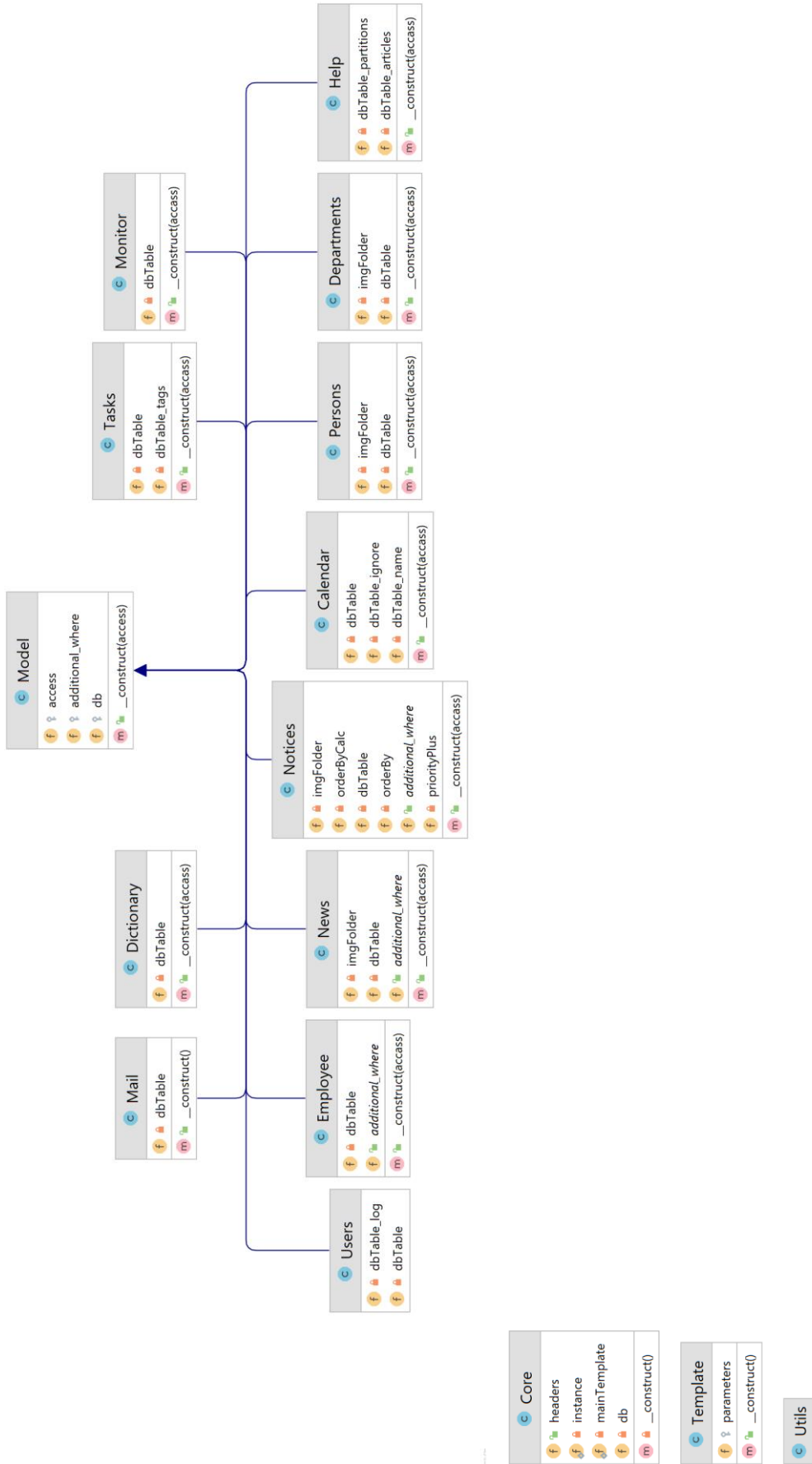
Діаграма Use case для ролі користувача



Діаграма класів системи (частина 1)



Діаграма класів системи (частина 2)



		В.А.РАЙКОВСЬКИЙ		
		М.С. ГРАФ		
Змн.	Арк.	№ докум.	Підпис	Дата

Контролер модуля для обліку кадрів — клас Employee.php

```

class Employee extends Controller
{
    protected $employeeModel;
    protected string $url = "/employee/";
    protected int $redirect_timeout;

    public function __construct()
    {
        parent::__construct();
        $this->employeeModel = new \models\Employee($this->access);
        $this->redirect_timeout = $this->config->news['redirectTimeout'];
    }

    public function actionIndex()
    {
        $page = (isset($_GET['page']) && is_numeric($_GET['page']) &&
is_int((int)$_GET['page']) && (int)$_GET['page'] > 0) ? (int)$_GET['page'] : 1;
        $q = isset($_GET['q']);

        $users = [];
        if ($this->access()) {
            $all_users = $this->userModel->GetAllUsers();
            foreach ($all_users as $user) {
                $users[$user["id"]] = $user["login"];
            }
        }

        if (($this->isPost() && isset($_POST['search']) &&
!empty($_POST['search'])) || ($q && isset($_SESSION['search']))) {
            if (!$q) {
                $_SESSION['search'] = $_POST['search'];
            }

            if ($this->employeeModel->countSearched($_SESSION['search'])) {
                $employee = $this->employeeModel->Search($_SESSION['search']);
                return $this->render('index',
                    [
                        'rows' => $employee,
                        'page' => $page,
                        'user' => $this->user,
                        'users' => $users,
                        'search' => true,
                        'search_text' => $_SESSION['search'],
                        'access' => $this->access()
                    ],
                    [
                        'MainTitle' => 'Пошук',
                        'PageTitle' => 'Пошук у новинах'
                    ]
                );
            } else {

```

```

        return $this->renderInformMessage('success', 'Нічого не
знайдено', null,
        [
            'MainTitle' => 'Пошук',
            'PageTitle' => 'Пошук у новинах'
        ]);
    }
}

$title = 'Працівники';
$offset = $this->config->news['perPage'] * ($page - 1);
$count = $this->employeeModel->getCount();
$add = (int)($count) % $this->config->news['perPage'] == 0 ? 0 : 1;
$countEmployee = intdiv((int)($count), $this->config->news['perPage'])
+ $add;
$employee = $this->employeeModel->getAll($this->config-
>news['perPage'], $offset);
return $this->render('index',
    [
        'rows' => $employee,
        'page' => $page,
        'count' => $countEmployee,
        'user' => $this->user,
        'users' => $users,
        'access' => $this->access()
    ],
    [
        'MainTitle' => $title,
        'PageTitle' => $title
    ]);
}

```

```

public function actionView($id)
{
    if (!$this->userAuth) return $this->forbidden();
    if (!$this->access()) return $this->forbidden();
    $title = "Історія роботи працівника";
    $employee_info = $this->employeeModel->getEmployeeInfoById($id);
    $employee = $this->employeeModel->getEmployeeById($id);

    if ($this->isPost()) {
        $_POST['people_id'] = $id;
        if (!empty($_POST['id'])) {
            $result = $this->employeeModel->updateHistoryRow($_POST);
        } else {
            $result = $this->employeeModel->add($_POST);
        }

        if ($result['error'] === false) {
            return $this->renderInformMessage('success', 'Зміни збережено',
            [
                'redirect_url' => $this->url,
                'redirect_timeout' => $this->redirect_timeout
            ]
        );
        }
    }
}

```

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		65

```

        ],
        [
            'MainTitle' => $title,
            'PageTitle' => $title
        ]));
    } else {
        $message = implode('<br/>', $result['messages']);
        return $this->render('form', ['model' => $_POST], [
            'MainTitle' => $title,
            'PageTitle' => $title,
            'InformText' => $message,
            'InformClass' => 'danger']);
    }
}

return $this->render('history',
    [
        'rows' => $employee_info,
        'employee' => $employee
    ],
    [
        'MainTitle' => $title,
        'PageTitle' => $title
    ]));
}

public function actionDeleteHistoryRow($id = 0, $confirm = null)
{
    if (!$this->access()) return $this->forbidden();

    $row = $this->employeeModel->getEmployeeInfoByIdRow($id);
    var_dump($row);
    $title = 'Видалення запису';
    if (!empty($confirm) && $confirm == 'yes') {
        if ($this->employeeModel->deleteHistoryRow($id)) {
            header('Location: /news/');
        } else
            return $this->renderInformMessage('error', 'Помилка видалення
запису', null,
                [
                    'MainTitle' => $title,
                    'PageTitle' => $title
                ]));
    }
    if (empty($row)) {
        return $this->renderInformMessage(
            "error",
            "Помилка. Записа не знайдено!",
            [
                'redirect_url' => $this->url,
                'redirect_timeout' => $this->redirect_timeout
            ],
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    }
}

```

```

        ));
    }
    return $this->render('delete', ['row' => $row], [
        'MainTitle' => $title,
        'PageTitle' => $title,
    ]);
}

public function actionAddPeople()
{
    if (!$this->access()) return $this->forbidden();

    $title = "Додавання нового працівника";

    if ($this->isPost()) {
        $result = $this->employeeModel->addEmployee($_POST);
        if (isset($result['error']) && $result['error'] === false) {
            return $this->renderInformMessage(
                'success',
                'Дані користувача успішно збережено',
                [
                    'redirect_url' => $this->url,
                    'redirect_timeout' => $this->redirect_timeout + 1000
                ],
                [
                    'MainTitle' => $title,
                    'PageTitle' => $title
                ]
            );
        } else {
            $message = implode('<br/>', $result);
            return $this->render('form', [
                'row' => $_POST
            ], [
                'MainTitle' => $title,
                'PageTitle' => $title,
                'InformText' => $message,
                'InformClass' => 'danger'
            ]);
        }
    } else {
        return $this->render('form', [],
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    }
}

public function actionUpdatePeople($id = 0)
{
    if (!$this->access()) return $this->forbidden();

    $title = "Редагування даних працівника";

    $employee = $this->employeeModel->getEmployeeById($id);
    if ($employee) {
        if ($this->isPost()) {
            $_POST['id'] = $id;

```

```

        $result = $this->employeeModel->updateEmployee($_POST);
        if (isset($result['error']) && $result['error'] === false) {
            return $this->renderInformMessage(
                'success',
                'Дані користувача успішно збережено',
                [
                    'redirect_url' => $this->url,
                    'redirect_timeout' => $this->redirect_timeout +
1000
                ],
                [
                    'MainTitle' => $title,
                    'PageTitle' => $title
                ]
            );
        } else {
            $message = implode('<br/>', $result);
            return $this->render('form', [
                'row' => $_POST
            ], [
                'MainTitle' => $title,
                'PageTitle' => $title,
                'InformText' => $message,
                'InformClass' => 'danger'
            ]);
        }
    } else {
        return $this->render('form', ['row' => $employee],
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    } else {
        return $this->renderInformMessage(
            'danger',
            'Працівника не існує',
            [
                'redirect_url' => $this->url,
                'redirect_timeout' => $this->redirect_timeout + 1000
            ],
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    }
}

public function actionDeletePeople($id = 0, $confirm = null)
{
    if (!$this->access()) return $this->forbidden();

    $row = $this->employeeModel->getEmployeeById($id);
    $title = 'Видалення працівника';
    if (!empty($confirm) && $confirm == 'yes') {
        if ($this->employeeModel->deleteEmployee($id)) {
            header('Location: /employee/');
        } else {
            return $this->renderInformMessage('error', 'Помилка видалення
працівника', null,

```

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        [
            'MainTitle' => $title,
            'PageTitle' => $title
        ]);
    }
    if (empty($row)) {
        return $this->renderInformMessage(
            "error",
            "Помилка. Записа не знайдено!",
            [
                'redirect_url' => $this->url,
                'redirect_timeout' => $this->redirect_timeout
            ],
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    }
    return $this->render('deletePeople', ['row' => $row], [
        'MainTitle' => $title,
        'PageTitle' => $title,
    ]);
}

public function actionFiredPeople($id = 0, $confirm = null)
{
    if (!$this->access()) return $this->forbidden();

    $row = $this->employeeModel->getEmployeeById($id);
    $title = 'Звільнення працівника';

    if (!empty($confirm) && $confirm == 'yes') {
        if ($this->employeeModel->firedEmployee($id)) {
            header('Location: /employee/');
        } else
            return $this->renderInformMessage('error', 'Помилка звільнення
працівника', null,
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    }
    if (empty($row)) {
        return $this->renderInformMessage(
            "error",
            "Помилка. Працівника не знайдено!",
            [
                'redirect_url' => $this->url,
                'redirect_timeout' => $this->redirect_timeout
            ],
            [
                'MainTitle' => $title,
                'PageTitle' => $title
            ]
        );
    }
}

```

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		69

```

        return $this->render('firedPeople', ['row' => $row], [
            'MainTitle' => $title,
            'PageTitle' => $title,
        ]);
    }
}

```

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				70
Змн.	Арк.	№ докум.	Підпис	Дата		

Модель модуля для обліку кадрів — клас Employee.php

```

class Employee extends \core\Model
{
    private string $dbTable = 'peple';

    public function __construct($access = -1)
    {
        parent::__construct($access);
        if ($this->access <= 0) {
            $this->additional_where['(public_date is not null AND
public_date<=DATE(NOW()))'] = null;
        }
    }

    public function getAll($count, $offset, $when = ""){
        if(!empty($count)){
            $this->db->count($count);
        }
        if(!empty($offset)){
            $this->db->offset($offset);
        }

        $result = $this->db->join_left('employee_info',
'employee_info.people_id = p.id')->get('peple p');

        $sql = 'SELECT
em.id AS employee_id,
em.position,
em.role,
em.start_work,
em.end_work,
p.id AS peple_id,
p.first_name,
p.last_name,
p.patronymic,
p.birthdate,

p.region,
p.phone,
p.description
FROM
employee_info em
JOIN
peple p ON em.people_id = p.id';

        return $result; //$this->db->getCustom($sql);
    }

    public function getCount($where = []){
        $where = array_merge($this->additional_where, $where);
        return $this->db->getCount($this->dbTable, $where);
    }

    public function getEmployeeInfoById($id)

```

```

    {
        return $this->db->where(['people_id'=>$id])->order_by(['start_work' =>
'DESC'])->get("employee_info");
    }

    public function getEmployeeActualInfoByPeopleIdRow($id)
    {
        return $this->db->where(['people_id'=>$id, 'end_work IS NULL'=>>null])->
get("employee_info");
    }

    public function getEmployeeInfoByIdRow($id)
    {
        $result = $this->db->where(['id'=>$id])->get("employee_info");
        return $result[0];
    }

    public function getHistoryRowById($id)
    {
        return $this->db->where(['id'=>$id])->get("employee_info");
    }
    public function deleteHistoryRow($id)
    {
        return $this->db->delete("employee_info", ['id'=>$id]);
    }

    public function deleteEmployee($id)
    {
        if($this->db->delete("employee_info", ['people_id'=>$id])){
            return $this->db->delete("peple", ['id'=>$id]);
        }
        else
            return false;
    }

    public function firedEmployee($id)
    {
        return $this->db->update("employee_info", ['end_work'=>'CURDATE()'],
['people_id'=>$id, 'end_work IS NULL'=>>null]);
    }

    public function getEmployeeById($id){
        $rows = $this->db->where(['id'=>$id])->get($this->dbTable);
        return $rows[0]??false;
    }
    public function add($employeeHistoryRow){
        $validateResult = $this->validate($employeeHistoryRow);
        if(is_array($validateResult)) {
            return [
                'error' => true,
                'messages' => $validateResult
            ];
        }
        $fields = ['position', 'role', 'start_work', 'end_work', 'people_id'];

        $rowFiltered = Utils::ArrayFilter($employeeHistoryRow, $fields);

```

```

        $this->db->update("employee_info",
['end_work'=>$rowFiltered['start_work']],
['people_id'=>$rowFiltered['people_id'], 'end_work IS NULL'=>>null]);

        $id = $this->db->insert("employee_info", $rowFiltered);
        return [
            'error' => false,
            'id' => $id
        ];
    }

    public function addEmployee($employeeRow){
        $validateResult = $this->ValidateEmployee($employeeRow);
        if(is_array($validateResult)) {
            return [
                'error' => true,
                'messages' => $validateResult
            ];
        }
        $fieldsEmployee = ['first_name', 'last_name', 'patronymic',
'birthdate', 'region', 'phone'];

        $rowFilteredEmployee = Utils::ArrayFilter($employeeRow,
$fieldsEmployee);
        $id = $this->db->insert("peple", $rowFilteredEmployee);

        $fieldsEmployeeInfo = ['position'=>"-", 'role'=>"-", 'start_work'=>"-",
'end_work', 'people_id'=>$id];
        $this->db->insert("employee_info", $fieldsEmployeeInfo);
        return [
            'error' => false,
            'id' => $id
        ];
    }

    public function updateEmployee($employeeRow){
        $validateResult = $this->ValidateEmployee($employeeRow);
        if(is_array($validateResult)) {
            return [
                'error' => true,
                'messages' => $validateResult
            ];
        }
        $fieldsEmployee = ['id', 'first_name', 'last_name', 'patronymic',
'birthdate', 'region', 'phone', 'description'];

        $rowFilteredEmployee = Utils::ArrayFilter($employeeRow,
$fieldsEmployee);
        $id = $this->db->debug()->update("peple", $rowFilteredEmployee,
['id'=>$employeeRow['id']]);

        return [
            'error' => false,
            'id' => $id
        ];
    }
}

```

```

        public function countSearched($searchStr, $where = []){
            $where["(first_name like '%$searchStr%' or last_name like
            '%$searchStr%' or patronymic like '%$searchStr%')"] = null;
            //$result = \core\Core::getInstance()->getDB()->selectDB($this-
            >dbTable, 'title', $where);
            //$count = count($result);
            $count = $this->db->getCount($this->dbTable, $where);
            if($count>0)
                return $count;
            else
                return false;
        }

        public function Search($searchStr, $where = []){
            $where = array_merge($this->additional_where, $where);
            $where["(first_name like '%$searchStr%' or last_name like
            '%$searchStr%' or patronymic like '%$searchStr%')"] = null;
            if(!empty($count)){
                $this->db->count($count);
            }
            if(!empty($offset)){
                $this->db->offset($offset);
            }
            $this->db->order_by(['last_name' => 'ASC']);
            return $this->db->join_left('employee_info', 'employee_info.people_id =
            p.id')->where($where)->get('peple p');
            //return $this->db->get($this->dbTable,
            'id,title,short_text,date,photo,user_id,datetime_lastedit,public_date',
            $where);
        }

        public function updateHistoryRow($employeeHistoryRow){
            $validateResult = $this->Validate($employeeHistoryRow);
            if(is_array($validateResult)) {
                return [
                    'error' => true,
                    'messages' => $validateResult
                ];
            }
            $fields = ['position', 'role', 'start_work', 'end_work'];

            $rowFiltered = Utils::ArrayFilter($employeeHistoryRow, $fields);
            $this->db->update("employee_info", $rowFiltered,
            ['id'=>$employeeHistoryRow['id']]);
            return [
                'error' => false,
            ];
        }

        public function Validate($row){
            $errors = [];
            if(empty($row['position']))
                $errors [] = 'Поле "Посада" не може бути порожнім';
            if(empty($row['role']))
                $errors [] = 'Поле "Звання" не може бути порожнім';
        }
    
```

```

        if (count($errors)>0)
            return $errors;
        else
            return true;
    }
    public function ValidateEmployee($row){
        $errors = [];
        if(empty($row['last_name']))
            $errors [] = 'Поле "Ім\'я" не може бути порожнім';
        if(empty($row['first_name']))
            $errors [] = 'Поле "Прізвище" не може бути порожнім';
        if(empty($row['patronymic']))
            $errors [] = 'Поле "По батькові" не може бути порожнім';
        if(empty($row['birthdate']))
            $errors [] = 'Поле "Дата народження" не може бути порожнім';
        if(empty($row['region']))
            $errors [] = 'Поле "Регіон" не може бути порожнім';
        if(empty($row['phone']))
            $errors [] = 'Поле "Телефон" не може бути порожнім';
        if (count($errors)>0)
            return $errors;
        else
            return true;
    }
}

```

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				
Змн.	Арк.	№ докум.	Підпис	Дата		75

Представлення для головної сторінки модуля для обліку кадрів — файл
index.php

```

<div class="table-responsive">
  <form class="d-flex mb-3" method="post" action="/employee/">
    <input class="form-control me-2" type="text" placeholder="Пошук" aria-
label="Search" name="search" value="<?=$search_text?>" ?>">
    <button type="submit" class="btn btn-primary">Пошук</button>
    <a type="" class="btn btn-primary" href="/employee/">Відмінити</a>
  </form>
  <a href="/employee/AddPeople" class="btn btn-outline-primary float-
end">Додати працівника</a>
  <table class="table table-striped table-hover">
    <thead>
      <tr class="center middle">
        <th scope="col">#</th>
        <th scope="col">Прізвище</th>
        <th scope="col">Ім'я</th>
        <th scope="col">По-батькові</th>
        <th scope="col">Звання</th>
        <th scope="col">Посада</th>
      </tr>
    </thead>
    <tbody>
      <?php $counter = 0?>
      <?php foreach ($rows as $key=>$row): ?>
      <?php if ($row["end_work"]==null): ?>
        <?php $counter++ ?>
        <tr class="center middle" onclick="showUserDetails(<?=$key ?>)"
style='cursor: pointer;'>
          <td ><?=$counter ?></td>
          <td><?=$row['last_name'] ?></td>
          <td><?=$row['first_name'] ?></td>
          <td><?=$row['patronymic'] ?></td>
          <td><?=$row['role'] ?></td>
          <td><?=$row['position'] ?></td>
          <td><a href="/employee/UpdatePeople/<?=$row['people_id'] ?>"
onclick="event.stopPropagation();return true;" title="Редагувати" class="ps-
2"></a></td>
          <td><a href="/employee/DeletePeople/<?=$row['people_id'] ?>"
onclick="event.stopPropagation();return true;" title="Видалити" class="ps-
2"></a></td>
        </tr>
      </tbody>
    </table>
    <td colspan="10" class="px-0 pt-0">
      <div class="border-start border-end border-bottom p-2"
style="display: none;" id="user_details_<?=$key ?>">
        <div><b>Регіон:</b><?=$row['region'] ?></div>
        <div><b>Телефон:</b><?=$row['phone'] ?></div>
      </div>
    </td>
  </tr>
</div>

```

```

        <div><b>Опис: </b><?= $row['description'] ?></div>
        <div><b>Призначений: </b><?= $row['start_work']
?></div>
        <div><b><a href='/employee/view/<?= $row['people_id']
?>'</b>Історія роботи</a></div>
        <div class="border-bottom my-2"></div>
    </div>
</td>
</tr>
<?php endif;?>
<?php endforeach; ?>
</tbody>
</table>
</div>

```

		В.А.РАЙКОВСЬКИЙ			ІСТ.КР.Б – 126 – 08 – ПЗ	Арк.
		М.С. ГРАФ				77
Змн.	Арк.	№ докум.	Підпис	Дата		