

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ ДЕМОНТАЖНИХ РОБІТ
DESIGN AND DEVELOPMENT OF SOFTWARE TO SUPPORT
DISMANTLING WORK

Виконав(ла): студент(ка) 4 курсу, групи КНТ-111

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ВАСИЛЬЧЕНКО Я.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ШИТІКОВА О.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (КОД І НАЙМЕНУВАННЯ)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ВАСИЛЬЧЕНКА Ярослава Васильовича
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Проектування та розробка програмного забезпечення підтримки демонтажних робіт. Design and Development of Software to Support Dismantling Work

керівник проєкту (роботи) к.т.н. ШИТИКОВА Олена Вікторівна,
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 02 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Проектування програми. 3. Розробка програми. 4. Експлуатація та тестування програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ШИТІКОВА О.В., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання «14» квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Ярослав ВАСИЛЬЧЕНКО
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Олена ШИТІКОВА
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 108 с., 3 табл., 26 рис., 3 дод., 7 джерел.

ДЕМОНТАЖ, ДЕМОНТАЖНА БРИГАДА, ДЕМОНТАЖНА РОБОТА, ЗАМОВЛЕННЯ, ЗАПИТ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт роботи – процеси пошуку і замовлення демонтажних робіт. Предмет роботи – програмне забезпечення підтримки демонтажних робіт. Мета роботи – розробити програмне забезпечення підтримки демонтажних робіт для організації контактів між бригадами, що виконують демонтаж об'єктів, та замовниками таких послуг.

Матеріали, методи та технічні засоби: мова Python, система MySQL, фреймворк веброзробки Django.

Результати. Результати включають аналіз існуючих програм підтримки демонтажних робіт, проєктування, реалізацію і тестування власного програмного забезпечення для підтримки демонтажних робіт. Розроблені функції направлені на реалізацію даного рішення у складі цілісного застосунку для фреймворку Django.

Висновки. Створена програма надає можливості пошуку, ознайомлення з можливостями демонтажних бригад, роботами, які вони здатні виконати, замовленням демонтажних робіт у них, розміщенням власних оголошень про необхідність демонтажу об'єктів, пошуку серед таких оголошень представниками демонтажних бригад, створення запитів на демонтаж від них, керування такими запитами.

Галузь використання. Надання демонтажних послуг окремими працівниками або спеціалізованими бригадами з та без використання спеціалізованої техніки. Пошук, початкове замовлення таких послуг з демонтажу об'єктів замовниками.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 108 pages, 3 tables, 26 figures, 3 appendixes, 7 sources.

DISMANTLING, DISMANTLING CREW, DISMANTLING WORK, ORDER, REQUEST, SOFTWARE.

The object of the work is the processes of searching and ordering dismantling works. The subject of the work is the software for supporting dismantling works. The purpose of the work is to develop the software for supporting dismantling works to organize contacts between the crews that perform the dismantling of objects and the customers of such services.

Materials, methods and technical tools: Python language, MySQL system, Django web development framework.

Results. The results include the analysis of existing programs for supporting dismantling works, the design, implementation and testing of own software for supporting dismantling works. The developed functions are aimed at implementing this solution as part of a holistic application for the Django framework.

Conclusions. The created program provides the ability to search, familiarize yourself with the capabilities of dismantling crews, the work they are able to perform, order dismantling works from them, place own ads about the need to dismantle objects, search among such ads by representatives of dismantling crews, create requests for dismantling from them, manage such requests.

Field of use. Provision of dismantling services by individual workers or specialized crews with and without the use of specialized equipment. Search, initial order of such services for dismantling objects by customers.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз проблеми та постановка завдань дослідження	9
1.1 Стан розробки програм підтримки демонтажних робіт.....	9
1.2 Порівняння програм підтримки демонтажних робіт, що існують	12
1.3 Висновки за розділом 1	13
2 Проєктування програми.....	14
2.1 Обґрунтування вибору засобів розробки програми	14
2.2 Визначення вимог користувачів до програми підтримки демонтажних робіт.....	15
2.3 Проєктування структури бази даних.....	25
2.4 Висновки за розділом 2	38
3 Розробка програми	39
3.1 Керування станами основних сутностей	39
3.2 Реалізація програми підтримки демонтажних робіт	40
3.3 Висновки за розділом 3	58
4 Експлуатація та тестування програми	60
4.1 Призначення програми	60
4.2 Опис роботи з програмою підтримки демонтажних робіт	60
4.3 Тестування програми підтримки демонтажних робіт	64
4.4 Висновки за розділом 4	71
Висновки	72
Перелік джерел посилення	73
Додаток А Технічне завдання	74
Додаток Б Текст програми	80
Додаток В Слайди презентації.....	102

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

MVT – Model-View-Template;

ФВР – фреймворк веброзробки.

ВСТУП

Нерідко увага людей стосується будівництва нових об'єктів для житла, роботи, різних активностей, проте демонтаж існуючих об'єктів є важливою складовою даного процесу, якому приділяється недостатньо уваги. Потреби у демонтажі виникають у індивідуальних власників на їх ділянках перед тим, як наприклад, переорганізувати простір. У таких випадках необхідно прибрати господарчі будівлі, гаражі, залишки колишніх об'єктів, прибрати територію, прибрати будівельне сміття. Не завжди такі роботи власними можуть виконати самостійно. Зокрема це може траплятися тому, що такі роботи потребують спеціального обладнання або взагалі для виконання, або для скорочення часу, який в такому обсязі власник не має.

Супутніми роботами до демонтажних є перевезення певних матеріалів, обладнання на інше місце. Усе це потребує вирішення, а в той же час існує багато компаній, окремих бригад, працівників, які такі послуги можуть надавати. Найчастіше всі вони мають свої окремі вебсайти, що ускладнює порівняння і пошук того варіанту, який задовольнить користувача як замовника демонтажних послуг. Це може бути вирішено через створення програмного рішення, яке б об'єднало всі ці можливості разом.

Тож в цій роботі необхідно розробити програмне забезпечення підтримки демонтажних робіт для організації контактів між бригадами, що виконують демонтаж об'єктів, та замовниками таких послуг.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Стан розробки програм підтримки демонтажних робіт

Поточний стан підтримки демонтажних робіт вказує на наявність великої кількості окремих рішень для кожної такої компанії, демонтажної бригади.

Вебсайт demontag.zp.ua [1] (рис. 1.1) надає базову інформацію про демонтажні роботи у даної компанії. Окрім інформацію про компанію представлено також перелік послуг з демонтажу, які надаються компанією, вартість цих послуг, умови гарантії, галерея проведених робіт, відгуки. Відгук можна залишити через стандартну форму, реєстрація при цьому взагалі не потребується. Тобто перевірити достовірність відгуків навіть на рівні ідентифікації користувачів неможливо. Також присутня форма для створення замовлення демонтажних робіт.

 ДЕМОНТАЖНА КОМПАНІЯ №1 demontag.zp.ua		Україна, м. Запоріжжя ☎ +38(066) 012-19-40 ☎ +38(067) 614-55-84 RU UA	
ПРО КОМПАНІЮ		ПОСЛУГИ	ГАРАНТІЇ ГАЛЕРЕЯ РОБІТ ВІДГУКИ КОРИСНЕ КОНТАКТИ
АЛМАЗНА РІЗКА БЕТОНУ ЦІНА ЦІНИ НА 31.05.2025			
№ з.п.	Найменування робіт	Вартість	
1.	Алмазна різка отворів	від 243 до 3150 грн/шт.	
2.	Алмазна різка стін	від 85 до 416 грн/м.пог.	
3.	Алмазна різка залізобетону	від 155 грн/м.пог.	
4.	Алмазна різка цегляної стіни	від 83 до 375 грн/м.пог.	
5.	Алмазна різка бетону без пилу	від 306 до 985 грн/м.пог.	
6.	Алмазна різка дверних отворів	від 650 грн/шт.	
7.	Посилення проїомів	від 145 грн/м.пог.	
8.	Алмазна різка штроб	від 79 до 337 грн/м.пог.	
9.	Алмазна різка фундаменту	від 130 грн/м.пог.	

Рисунок 1.1 – Сторінка вартості демонтажних робіт на вебсайті demontag.zp.ua

Подібним прикладом є вебсайт demontazh-kyiv.com.ua [2] (рис. 1.2), де можна отримати інформацію про послуги з демонтажу, надається загальна форма для замовлення, галерея виконаних робіт. При цьому відгуків та даних про вартість демонтажних робіт не надається.

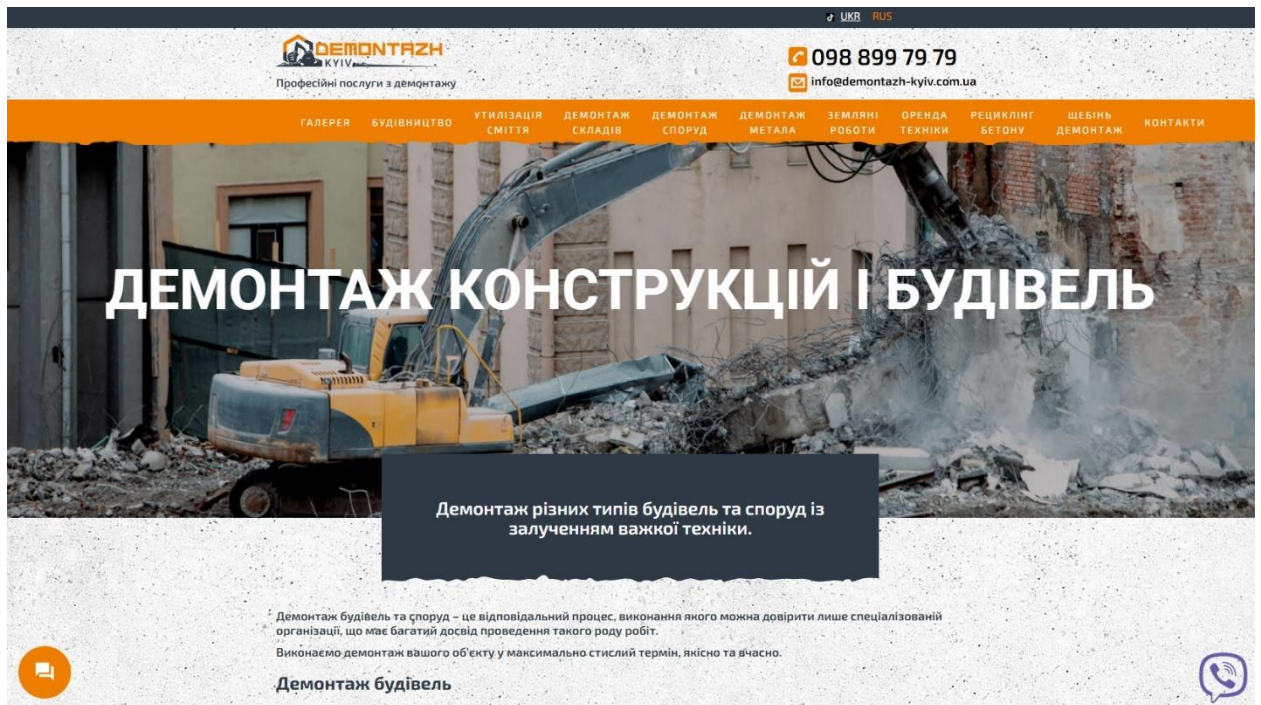


Рисунок 1.2 – Сторінка демонтажної роботи на вебсайті demontazh-kyiv.com.ua

Вебсайт demontazh.net.ua [3] (рис. 1.3), на відміну від попереднього, містить детальну інформацію про вартість демонтажних робіт. Також надається інформація про розділені за категоріями демонтажні роботи. Існує форма для створення замовлення на розрахунок вартості. При цьому замовлення на конкретний час сформувати не можна. Також є посилання на відгуки про компанію в Google.

Платформа Olx [4] (рис. 1.4) дозволяє під час пошуку обрати серед фільтрів демонтажні роботи. Відповідно будь-які користувачі можуть додавати свої оголошення, таким чином присутній майданчик для взаємодії сторін. При цьому вартість, наявні послуги задаються довільно в оголошенні

користувачем. Тому найчастіше дані про вартість відсутні. Під час пошуку задати відповідні фільтри неможливо.

ГОЛОВНАПОСЛУГИЦІНИВІДГУКИПРО КОМПАНІЮКОНТАКТИЗворотний зв'язок

Ціни на демонтажні роботи у Києві та області

Види робіт	Вартість
Знесення та демонтаж стін	
Знесення старих будівель заввишки до 9 метрів	від 40 гривень за м ³
Демонтаж дачного будинку	від 450 гривень за м ³
Демонтаж згорілих будівель	від 510 гривень за м ³
Знесення цегляних та бетонних труб	від 230 гривень за м ³
Руйнування бетонних споруд	від 145 гривень за м ³
Знесення господарських будівель	
Демонтаж туалетів	від 50 гривень за шт.
Демонтаж господарських будівель	від 200 гривень за шт.
Демонтаж стін	
Демонтаж стін із газосилекатного блоку	від 12 гривень за м ²
Демонтаж стін із цегли	від 13 гривень за м ²
Демонтаж стін із залізобетону 8 см.	від 25 гривень за м ²
Демонтаж стін із залізобетону 16 см.	від 35 гривень за м ²
Демонтаж стін із залізобетону 21 см.	від 55 гривень за м ²
Демонтаж ЗБВ	від 170 гривень за м ³
Демонтаж ФБС	від 260 гривень за м ³

Рисунок 1.3 – Сторінка вартості демонтажних робіт на вебсайті demontazh.net.ua

olx

Чат

Укр | Рус

Ваш профіль

Додати оголошення

🔍 демонтаж

✕

📍 Київ, Київська область

✕

+0 km

▼

Пошук

☐ Тільки з фото

☐ Доставка по всій країні

Додати в обране

Фільтри

Категорія

Підкатегорія

Будівництво та ремо...

Всі оголошення

Всі оголошення

Бізнес

Приватні

Скинути фільтри

Послуги з будівництва та ремонту Київ - демонтаж

Сторінка / Послуги з будівництва та ремонту / Будівництво та ремонт - Київська область / Будівництво та ремонт - Київ

Сортувати за: Рекомендоване вам

Валюта: грн. \$ €

Будівництво під ключ 12

Ремонт під ключ 58

Проектування та дизайн 57

Будівельні послуги 228

Сантехніка та комунікації 81

Оздоблення, ремонт 105

Електромонтажні послуги 13

Вентиляція та кондиціонування 20

Покажи все

Реклама

Empower Ukrainian Children's Future

Help War-Impacted Kids Access Education

Ukraine Friends

Donate

Ми знайшли 903 оголошення



Демонтаж. Демонтажні роботи. Демонтаж стін, стяжки, штукатурки.

Київ, Солом'янський - Сьогодні о 11:17

Рисунок 1.4 – Результати пошуку демонтажних робіт на вебсайті Olx

1.2 Порівняння програм підтримки демонтажних робіт, що існують

Для детального порівняння програм підтримки демонтажних робіт виділено вебсайт demontag.zp.ua, вебзастосунок Olx, які було порівняно з розробкою (табл. 1.1).

Таблиця 1.1 – Порівняння програм підтримки демонтажних робіт, що існують, з розробкою

Критерій	demontag.zp.ua	Olx	Розробка
Оголошення про демонтажні роботи, що надаються як послуги	Так	Так	Так
Доступна бригада для демонтажу	Одна	Змінюється за кількістю пропозицій	Змінюється за кількістю пропозицій
Оголошення про необхідність демонтажу об'єктів	Ні	Фактично не застосовується	Так
Відгуки про результати демонтажу	Довільні користувачі	Немає	Користувачі, які виконали замовлення
Вартість послуг з демонтажу	Так	Так	Так
Пошук за параметрами демонтажу	Ні	Ні	Так

Розробка повинна дозволити надавати інформацію різним демонтажним бригадам про свої послуги, а окрім того і самим замовникам демонтажних робіт створювати оголошення про необхідність демонтажу їх об'єктів за такої необхідності, таким чином виступаючи ініціаторами

подальшої роботи. Також програма повинна надавати можливість залишити відгуки, але за результатами тих замовлень, які були ініційовані в програмі, тобто не будь-який користувач мережі зможе це зробити. Окрім того, на відміну від Olx, пошук повинен за параметрами враховувати демонтажні роботи, а не бути фактично пошуком за назвою з можливістю обмеження робіт загалом до демонтажних.

1.3 Висновки за розділом 1

Виконаний огляд програм підтримки демонтажних робіт визначив, що існуючі програми або представляють окремі компанії (бригади), або є загальними майданчиками для публікації оголошень, які ніяким чином у пошуку не враховують особливості виконання демонтажних робіт. Цей недолік має бути усунутим у роботі шляхом розробки програми підтримки демонтажних робіт, що заплановано.

Завдання роботи у підсумку:

- аналіз стану проблеми розробки програм підтримки демонтажних робіт, що виконано в цьому розділі;
- проєктування програми підтримки демонтажних робіт;
- реалізація, тестування програми підтримки демонтажних робіт.

2 ПРОЄКТУВАННЯ ПРОГРАМИ

2.1 Обґрунтування вибору засобів розробки програми

Фреймворки веброзробки (ФВР) є ключовим засобом забезпечення веброзробки разом з мовою програмування. Python ефективно підтримує веброзробку, надає широке коло засобів, має простий синтаксис, надає різні парадигми. ФВР для Python зокрема включають Django [5] і Pyramid [6] (табл. 2.1).

Таблиця 2.1 – Порівняння ФВР для розробки програми підтримки демонтажних робіт

Критерій	Django	Pyramid
Модульність	Монолітна, всі модулі вбудовані, можуть бути розширені	Стандартна, модулі можуть підключатися
Ідентифікація користувачів	Вбудована	Вбудованої немає, можна застосувати підключене рішення
Механізм шаблонів	Django Template Language	Jinja2, Chameleon, Mako можливо використати
Робота з базами даних	Об'єктно-реляційне відображення є і одразу придатне для різних систем (MySQL, PostgreSQL, NoSQL, Oracle)	Можна встановити і використати SQLAlchemy

ФВР Django надає монолітний засіб розробки, який містить всі необхідні засоби, які зарекомендували себе на практиці в тому числі під навантаженням. З іншого боку комбінація таких модулів у Pyramid потребує тестування в тому числі під навантаженням, а вже тоді застосування. Тому Django є логічним вибором для розробки роботи.

2.2 Визначення вимог користувачів до програми підтримки демонтажних робіт

Сценарії в програмі підтримки демонтажних робіт для неавторизованого користувача (рис. 2.1):

- пошук бригад, які виконують демонтажні роботи;
- перегляд інформації про демонтажну бригаду;
- перегляд вартості робіт демонтажної бригади;
- перегляд рейтингу демонтажних бригад;
- пошук за оголошеннями про демонтаж об'єктів;
- перегляд оголошення про демонтаж об'єктів;
- перегляд оголошень про демонтаж об'єктів замовника;
- перегляд оцінок демонтажних робіт від користувача;
- перегляд оцінок демонтажної бригади;
- перегляд оцінок замовника демонтажних робіт;
- перегляд даних замовника демонтажних робіт;
- авторизація.

Альтернативний сценарій в програмі підтримки демонтажних робіт для неавторизованого користувача – реєстрація, що є альтернативою авторизації для випадків, коли користувач бажає працювати як представник демонтажної бригади або замовник демонтажних робіт, не маючи такого акаунту в програмі.

Сценарії в програмі підтримки демонтажних робіт для замовника демонтажних робіт:

- утворення закладки на сторінці демонтажної бригади;
- перегляд закладених сторінок демонтажних бригад;
- замовлення демонтажних робіт;
- підтвердження виконання замовлення демонтажних робіт;
- перегляд своїх нових замовлень демонтажних робіт;
- перегляд своїх виконаних замовлень демонтажних робіт.

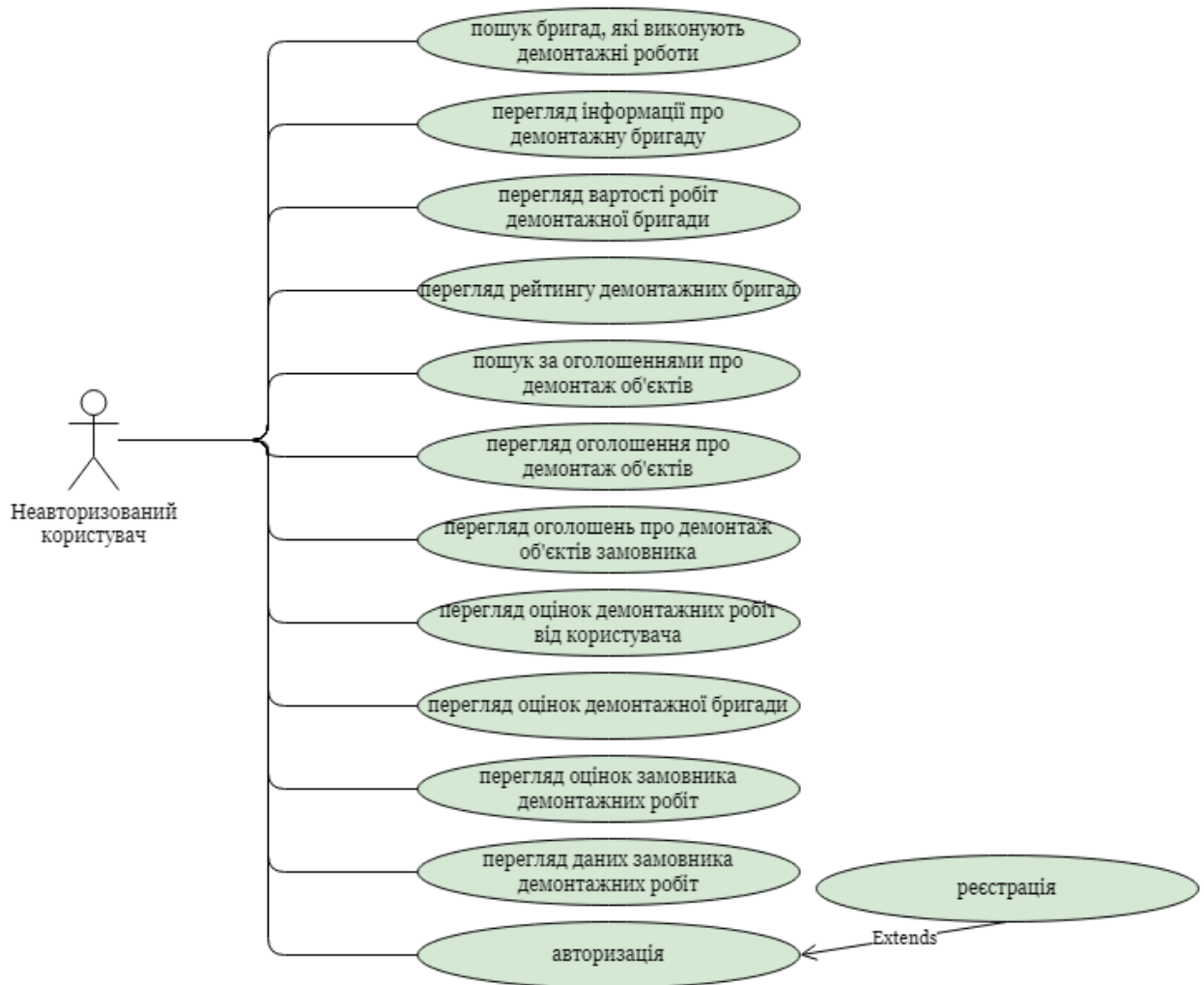


Рисунок 2.1 – Діаграма варіантів використання програми підтримки демонтажних робіт для неавторизованого користувача

Також сценарії в програмі підтримки демонтажних робіт для замовника демонтажних робіт:

- перегляд своїх запланованих замовлень демонтажних робіт;
- перегляд своїх проблемних замовлень демонтажних робіт;
- перегляд своїх скасованих замовлень демонтажних робіт;
- створення оголошення про демонтаж об'єктів;
- перегляд своїх оголошень про демонтаж об'єктів;
- утворення закладки на оголошенні про демонтаж об'єктів;
- перегляд закладених оголошень про демонтаж об'єктів;
- перегляд адресованих заявок про демонтаж об'єктів;
- підтвердження заявки про демонтаж об'єкту;

- підтвердження виконання заявки про демонтаж об'єкту;
- оцінювання виконання демонтажних робіт;
- редагування даних замовника демонтажних робіт (рис. 2.2).

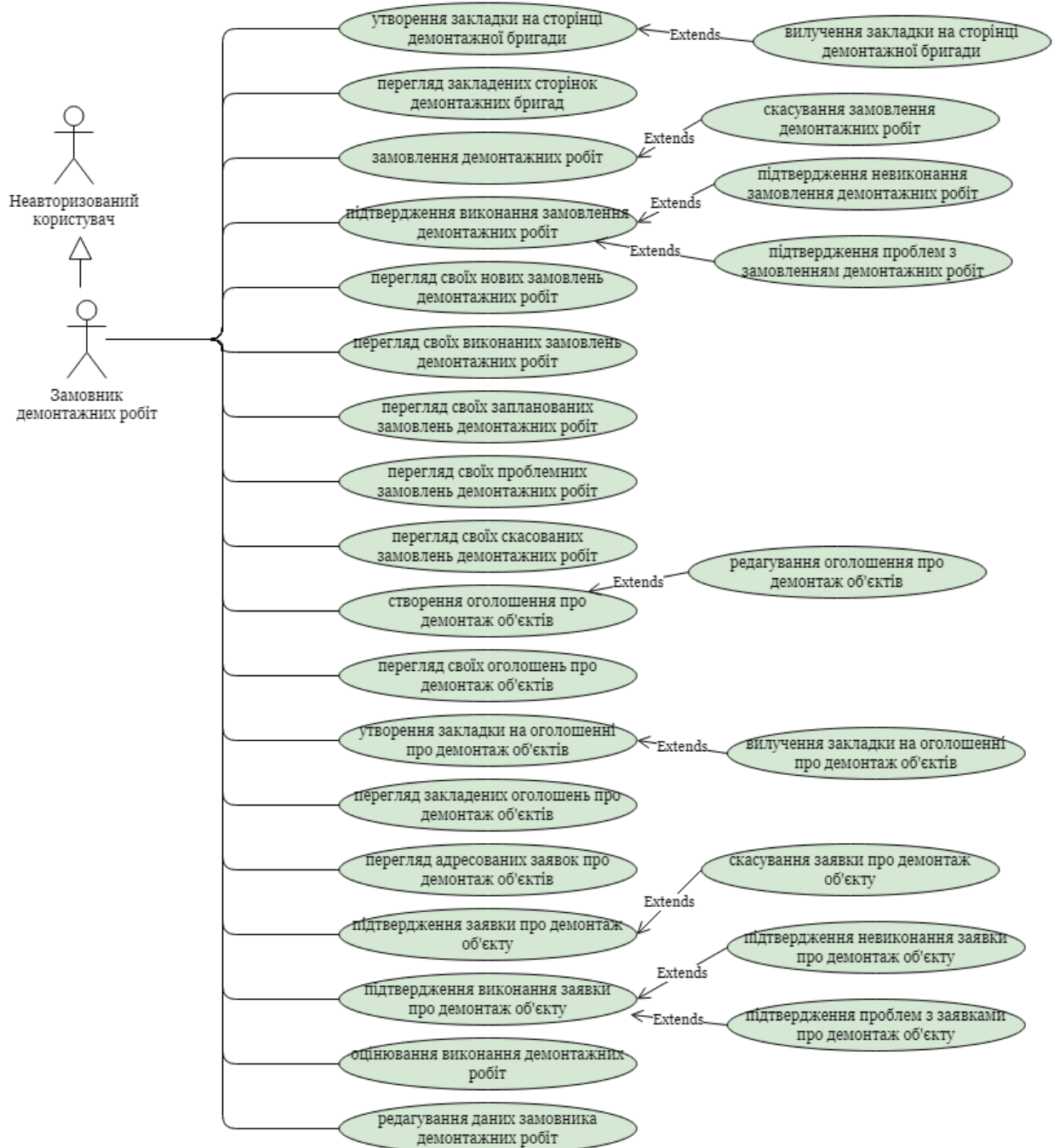


Рисунок 2.2 – Діаграма варіантів використання програми підтримки демонтажних робіт для замовника демонтажних робіт

Альтернативні сценарії в програмі підтримки демонтажних робіт для замовника демонтажних робіт:

- вилучення закладки на сторінці демонтажної бригади є альтернативою для утворення закладки на сторінці демонтажної бригади, коли закладка на цій сторінці демонтажної бригади цим замовником демонтажних робіт утворена була раніше;

- підтвердження невиконання заявки про демонтаж об'єкту є альтернативою для підтвердження виконання заявки про демонтаж об'єкту, коли за заявкою про демонтаж об'єкту не було виконано в обумовлений час роботи;

- підтвердження проблем з заявками про демонтаж об'єкту є альтернативою для підтвердження виконання заявки про демонтаж об'єкту, коли відбулись певні проблеми взаємодії стосовно демонтажу об'єкту;

- скасування заявки про демонтаж об'єкту є альтернативою для підтвердження заявки про демонтаж об'єкту, коли демонтажна бригада, що створила заявку, не відповідає вимогам замовника демонтажних робіт;

- вилучення закладки на оголошенні про демонтаж об'єктів є альтернативою для утворення закладки на оголошенні про демонтаж об'єктів, коли закладка на цьому оголошенні про демонтаж об'єктів цим замовником демонтажних робіт утворена була раніше;

- редагування оголошення про демонтаж об'єктів є альтернативою для створення оголошення про демонтаж об'єктів, коли за поточним станом виконання демонтажу об'єкту відбулись зміни порівняно зі збереженими раніше;

- підтвердження невиконання замовлення демонтажних робіт є альтернативою для підтвердження виконання замовлення демонтажних робіт, коли за замовленням демонтажних робіт не було виконано в обумовлений час роботи;

– підтвердження проблем з замовленням демонтажних робіт є альтернативою для підтвердження виконання замовлення демонтажних робіт, коли відбулись певні проблеми взаємодії стосовно демонтажних робіт;

– скасування замовлення демонтажних робіт є альтернативою для замовлення демонтажних робіт, коли замовлення демонтажних робіт ще не було підтверджено демонтажною бригадою, але замовник демонтажних робіт вже не хоче його виконувати.

Сценарії в програмі підтримки демонтажних робіт для представника демонтажною бригади:

- утворення закладки на сторінці демонтажною бригади;
- перегляд закладених сторінок демонтажних бригад;
- заповнення інформації про демонтажну бригаду;
- додавання техніки, використовуваною демонтажною бригадою;
- додавання демонтажної роботи до послуг бригади;
- підтвердження замовлення демонтажних робіт;
- підтвердження виконання замовлення демонтажних робіт;
- перегляд адресованих демонтажній бригаді нових замовлень;
- перегляд адресованих демонтажній бригаді виконаних замовлень;
- перегляд адресованих демонтажній бригаді запланованих замовлень;
- перегляд адресованих демонтажній бригаді проблемних замовлень;
- перегляд адресованих демонтажній бригаді скасованих замовлень;
- утворення закладки на оголошенні про демонтаж об'єктів;
- перегляд закладених оголошень про демонтаж об'єктів;
- відбір оголошень про демонтаж об'єктів за можливостями бригади;
- подання заявки на демонтаж об'єкту;
- перегляд своїх нових заявок про демонтаж об'єктів;
- перегляд своїх виконаних заявок про демонтаж об'єктів;
- перегляд своїх запланованих заявок про демонтаж об'єктів;
- перегляд своїх проблемних заявок про демонтаж об'єктів;
- перегляд своїх скасованих заявок про демонтаж об'єктів;

- підтвердження виконання заявки про демонтаж об'єкту;
- оцінювання замовника демонтажних робіт;
- визначення параметрів графіку роботи демонтажної бригади;
- відкриття доступного часу роботи демонтажної бригади;
- відображення графіку роботи демонтажної бригади на дату;
- перегляд усіх виконаних демонтажних робіт бригади.

Альтернативні сценарії в програмі підтримки демонтажних робіт для представника демонтажної бригади:

- вилучення закладки на сторінці демонтажної бригади є альтернативою для утворення закладки на сторінці демонтажної бригади, коли закладка на цій сторінці демонтажної бригади цим представником демонтажної бригади утворена була раніше;

- редагування інформації про демонтажну бригаду є альтернативою для заповнення інформації про демонтажну бригаду, коли поточна інформація про демонтажну бригаду не відповідає збереженій;

- вилучення техніки, використовуваної демонтажною бригадою, є альтернативою для додавання техніки, використовуваної демонтажною бригадою, коли ця техніка цією демонтажною бригадою більше не використовується;

- редагування параметрів демонтажної роботи у послугах бригади є альтернативою для додавання демонтажної роботи до послуг бригади, коли поточні значення параметрів демонтажної роботи у послугах бригади не відповідають збереженим;

- вилучення демонтажної роботи з послуг бригади є альтернативою для додавання демонтажної роботи до послуг бригади, коли такі демонтажні роботи цією бригадою більше не надаються;

- скасування замовлення демонтажних робіт є альтернативою для підтвердження замовлення демонтажних робіт, коли замовлення викликає сумніви у демонтажній бригаді або не може бути виконано з тих чи інших причин;

– підтвердження невиконання замовлення демонтажних робіт є альтернативою для підтвердження виконання замовлення демонтажних робіт, коли за замовленням демонтажних робіт не було виконано в обумовлений час роботи з вини замовника або самої демонтажної бригади;

– підтвердження проблем з замовленням демонтажних робіт є альтернативою для підтвердження виконання замовлення демонтажних робіт, коли відбулись певні проблеми взаємодії стосовно демонтажних робіт;

– вилучення закладки на оголошенні про демонтаж об'єктів є альтернативою для утворення закладки на оголошенні про демонтаж об'єктів, коли закладка на цьому оголошенні про демонтаж об'єктів цим представником демонтажної бригади утворена була раніше;

– скасування заявки про демонтаж об'єкту є альтернативою для подання заявки на демонтаж об'єкту, коли заявка про демонтаж об'єкту ще не була підтверджена замовником демонтажних робіт, але демонтажна бригада вже не хоче її виконувати;

– підтвердження невиконання заявки про демонтаж об'єкту є альтернативою для підтвердження виконання заявки про демонтаж об'єкту, коли за заявкою про демонтаж об'єкту не було виконано в обумовлений час роботи;

– підтвердження проблем з заявками про демонтаж об'єкту є альтернативою для підтвердження виконання заявки про демонтаж об'єкту, коли відбулись певні проблеми взаємодії стосовно демонтажу об'єкту;

– редагування параметрів графіку роботи демонтажної бригади є альтернативою для визначення параметрів графіку роботи демонтажної бригади, коли поточні параметри графіку роботи демонтажної бригади не відповідають збереженим;

– вилучення доступного часу роботи демонтажної бригади є альтернативою для відкриття доступного часу роботи демонтажної бригади, коли доступний час роботи демонтажної бригади був збережений, але більше не є таким (рис. 2.3).

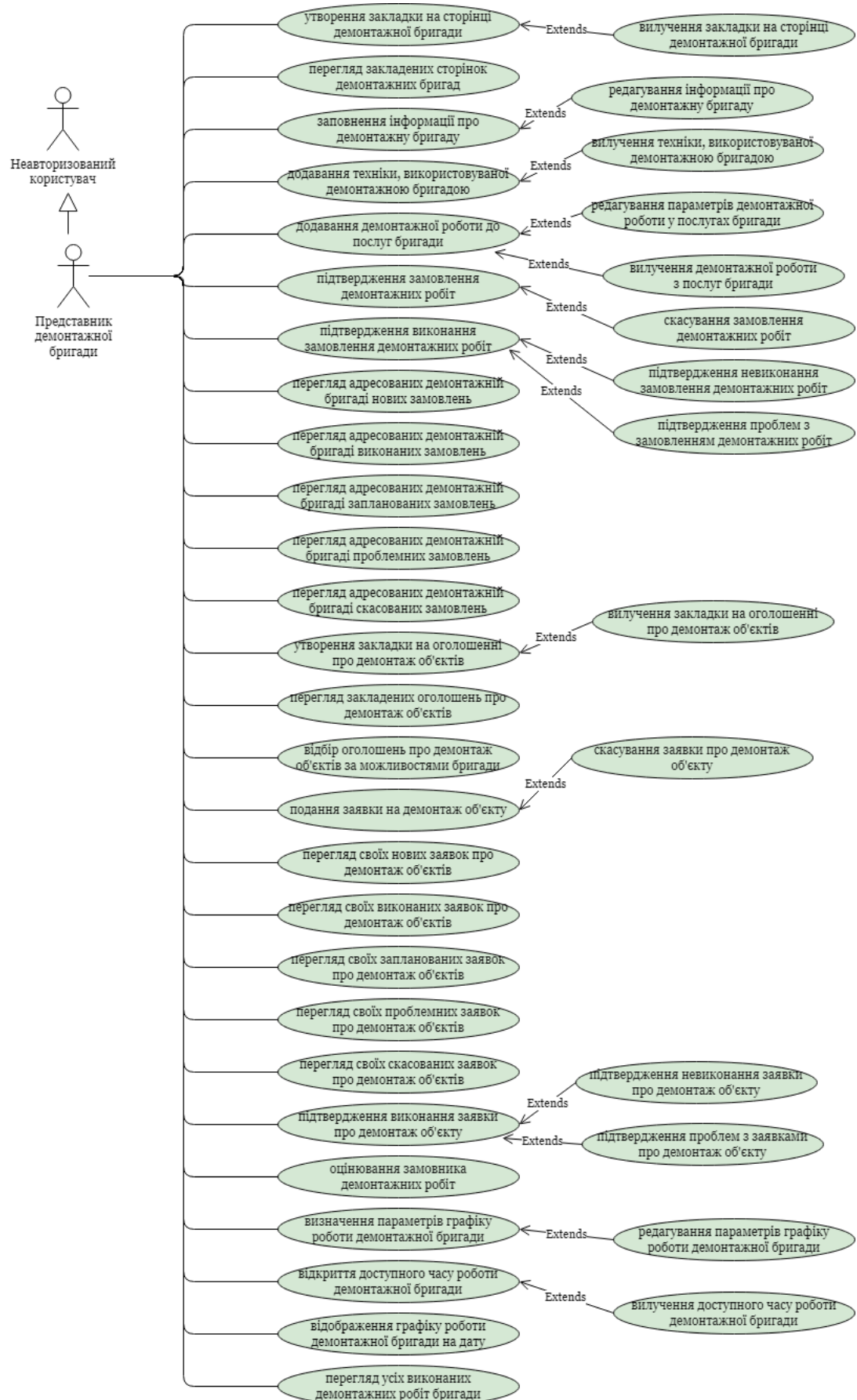


Рисунок 2.3 – Діаграма варіантів використання програми підтримки демонтажних робіт представником демонтажної бригади

Сценарії в програмі підтримки демонтажних робіт для менеджера програми:

- утворення закладки на сторінці демонтажної бригади;
- перегляд закладених сторінок демонтажних бригад;
- редагування інформації про демонтажну бригаду;
- вилучення техніки, використовуваної демонтажною бригадою;
- редагування параметрів демонтажної роботи у послугах бригади;
- вилучення демонтажної роботи з послуг бригади;
- підтвердження замовлення демонтажних робіт;
- скасування замовлення демонтажних робіт;
- підтвердження виконання замовлення демонтажних робіт;
- підтвердження невиконання замовлення демонтажних робіт;
- підтвердження проблем з замовленням демонтажних робіт;
- перегляд проблемних замовлень демонтажних робіт;
- пошук серед проблемних замовлень демонтажних робіт;
- блокування доступу замовника демонтажних робіт до програми;
- блокування обслуговування демонтажної бригади програмою;
- пошук серед заблокованих для обслуговування програмою демонтажних бригад;
- пошук серед заблокованих замовників демонтажних робіт;
- редагування оголошення про демонтаж об'єктів;
- утворення закладки на оголошенні про демонтаж об'єктів;
- перегляд закладених оголошень про демонтаж об'єктів;
- підтвердження заявки про демонтаж об'єкту;
- скасування заявки про демонтаж об'єкту;
- підтвердження виконання заявки про демонтаж об'єкту;
- підтвердження невиконання заявки про демонтаж об'єкту;
- підтвердження проблем з заявками про демонтаж об'єкту;
- перегляд проблемних заявок про демонтаж об'єктів;
- пошук серед проблемних заявок про демонтаж об'єктів;

– редагування даних замовника демонтажних робіт (рис. 2.4).

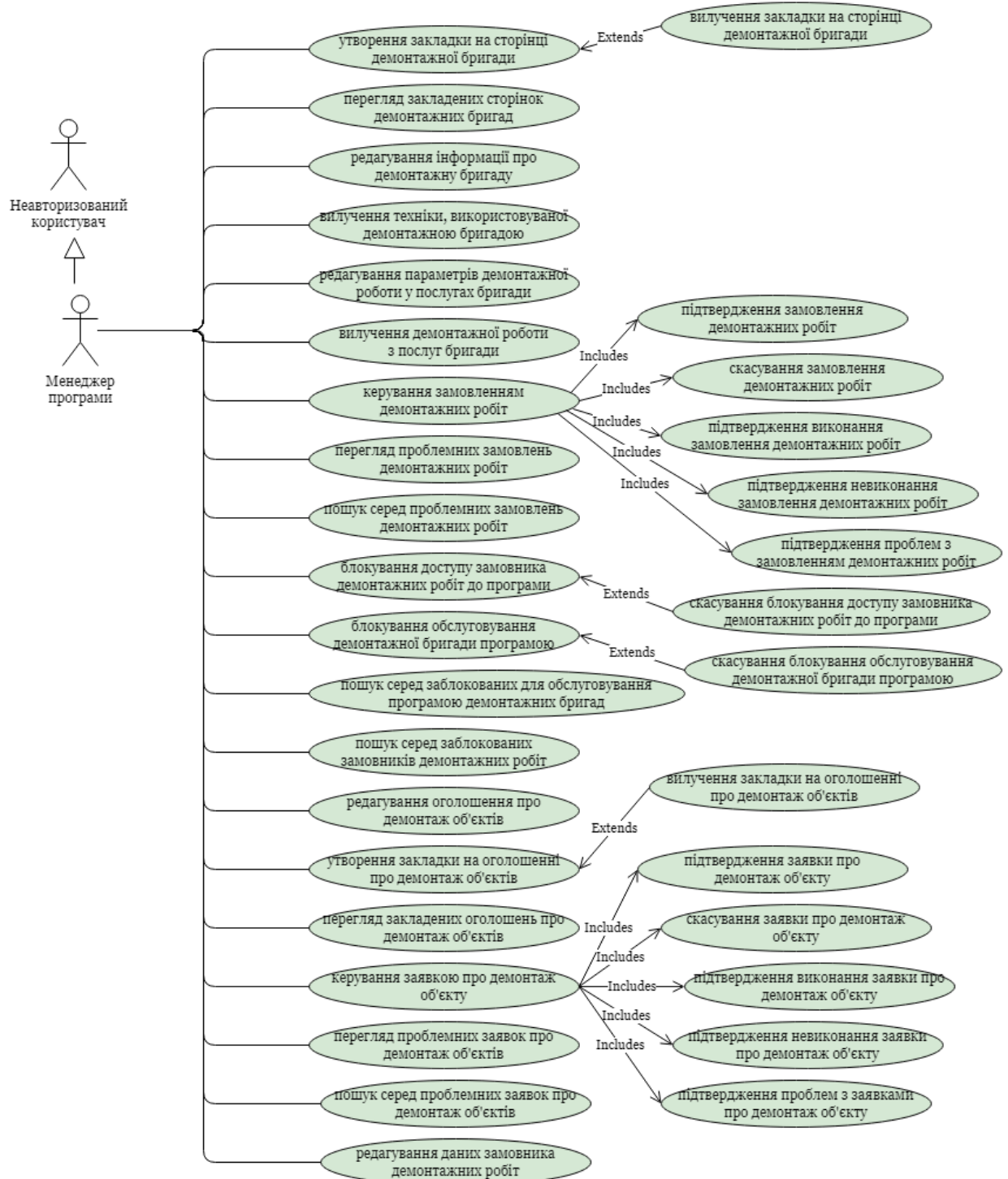


Рисунок 2.4 – Діаграма варіантів використання програми підтримки демонтажних робіт менеджером програми

Альтернативні сценарії в програмі підтримки демонтажних робіт для менеджера програми:

- вилучення закладки на оголошенні про демонтаж об'єктів є альтернативою для утворення закладки на оголошенні про демонтаж об'єктів, коли закладка на цьому оголошенні про демонтаж об'єктів цим менеджером програми утворена була раніше;

- скасування блокування доступу замовника демонтажних робіт до програми є альтернативою для блокування доступу замовника демонтажних робіт до програми, коли таке блокування більше не має тривати;

- скасування блокування обслуговування демонтажної бригади програмою є альтернативою для блокування обслуговування демонтажної бригади програмою, коли таке блокування більше не має тривати;

- вилучення закладки на сторінці демонтажної бригади є альтернативою для утворення закладки на сторінці демонтажної бригади, коли закладка на цій сторінці демонтажної бригади цим менеджером програми утворена була раніше.

2.3 Проектування структури бази даних

Реалізація роботи з даними виконана за допомогою системи керування базами даних MySQL, що є вільною, безкоштовною, надійною. При реалізації враховано, що ФВР Django передбачає визначення структури таблиць на основі коду моделей.

Користувач-представник демонтажної бригади представлений даними в таблиці Dismantler зі стовпцями:

- id – ідентифікатор користувача-представника демонтажної бригади;
- dismantlingdjangouser – акаунт користувача в Django, що застосовується представником демонтажної бригади, виражений зв'язком з таблицею User через зовнішній ключ (тип один-до-одного);
- searchphone – номер телефону представника демонтажної бригади;

– dismantlingcrew – демонтажна бригада, в якій працює і яку представляє цей користувач, виражений зв'язком з таблицею Dismantlingcrew через зовнішній ключ (тип один-до-багатьох).

Користувач-замовник демонтажних робіт представлений даними в таблиці Dismantlecustomer зі стовпцями:

- id – ідентифікатор користувача-замовника демонтажних робіт;
- dismantlingdjangouser – акаунт користувача в Django, що застосовується замовником демонтажних робіт, виражений зв'язком з таблицею User через зовнішній ключ (тип один-до-одного);
- searchphone – номер телефону замовника демонтажних робіт;
- dismantlingestimate – усереднена оцінка замовника демонтажних робіт;
- dismantlingblock – позначка про блокування доступу замовника демонтажних робіт для роботи з програмою;
- dismantlingphoto – фотографія замовника демонтажних робіт.

Оцінки замовників демонтажних робіт від демонтажних бригад представлений даними в таблиці Dismantlecustomerestimation зі стовпцями:

- id – ідентифікатор оцінки замовника демонтажних робіт від демонтажної бригади;
- dismantlecustomer – замовник демонтажних робіт, якому виставлено демонтажної бригадою у обличчі її представника оцінку, виражений зв'язком з таблицею Dismantlecustomer через зовнішній ключ (тип один-до-багатьох);
- dismantler – користувач-представник демонтажної бригади, який виконав оцінювання, виражений зв'язком з таблицею Dismantler через зовнішній ключ (тип один-до-багатьох);
- dismantlingmark – сама числова оцінка, поставлена замовнику демонтажних робіт демонтажною бригадою;
- dismantlingtime – час оцінювання демонтажною бригадою замовника демонтажних робіт;

– dismantlingcontent – оцінювання демонтажною бригадою замовника демонтажних робіт у текстовому представленні.

Типи об'єктів для демонтажу представлені даними в таблиці Dismantleobjecttype зі стовпцями:

- id – ідентифікатор типу об'єктів для демонтажу;
- dismantleobjecttypecaption – назва типу об'єктів для демонтажу.

Оголошення про демонтаж об'єктів представлені даними в таблиці Dismantlead зі стовпцями:

- id – ідентифікатор оголошення про демонтаж об'єкта;
- dismantleobjecttype – тип об'єкта, демонтаж якого необхідно виконати за оголошенням замовника демонтажних робіт;
- dismantlecustomer – замовник демонтажних робіт, який створив це оголошення про демонтаж об'єкту, виражений зв'язком з таблицею Dismantlecustomer через зовнішній ключ (тип один-до-багатьох);
- dismantleadcaption – назва оголошення про демонтаж об'єкту;
- dismantlingworkspec – опис особливостей демонтажних робіт, які за оголошенням необхідно виконати для демонтажу об'єкту;
- dismantleobjectphoto – фотографія об'єкту, на якому за оголошенням потрібно виконати демонтаж;
- dismantlingworkbasecity – місто, в якому знаходиться об'єкт, демонтаж якого за оголошенням потрібно виконати, виражений зв'язком з таблицею Dismantlingworkbasecity через зовнішній ключ (тип один-до-багатьох);
- dismantlingobjectaddress – адреса об'єкту, на якому за оголошенням потрібно виконати демонтаж;
- dismantlingminstart – дата, починаючи з якої потрібно виконати демонтаж об'єкту за оголошенням (найбільш рання дата);
- dismantlingmaxstart – дата, до якої потрібно почати виконання демонтаж об'єкту за оголошенням (найбільш пізня дата початку);

– `dismantlingmaxend` – дата, до якої потрібно закінчити демонтаж об'єкту за оголошенням;

– `dismantlingmaxprice` – максимальна вартість, яку за виконання цих робіт для демонтажу об'єкту за оголошенням готовий заплатити замовник демонтажних робіт.

Демонтажні роботи, які мають бути на думку замовника демонтажних робіт виконані для демонтажу його об'єкту за оголошенням, представлений даними в таблиці `Dismantleadwork` зі стовпцями:

– `id` – ідентифікатор демонтажної роботи, яку має бути на думку замовника демонтажних робіт виконано для демонтажу його об'єкту за оголошенням;

– `dismantlead` – оголошення про демонтаж об'єкту замовника, за яким на його думку має бути виконано цю демонтажну роботу для досягнення результату, виражене зв'язком з таблицею `Dismantlead` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingwork` – демонтажна робота, яка на думку замовника таких робіт має бути виконана за його оголошенням про демонтаж об'єкту для досягнення результату, виражена зв'язком з таблицею `Dismantlingwork` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingquantity` – кількість одиниць обчислення цієї демонтажної роботи (наприклад, квадратних кілометрів площі), яка на думку замовника демонтажних робіт має бути виконана за його оголошенням про демонтаж об'єкту для досягнення результату.

Закладки за оголошеннями про демонтаж об'єктів представлені даними в таблиці `Dismantleadbook` зі стовпцями:

– `id` – ідентифікатор закладки за оголошенням про демонтаж об'єкта;

– `dismantlead` – оголошення про демонтаж об'єкта, за яким користувач залишив закладку, виражене зв'язком з таблицею `Dismantlead` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingdjangouser` – користувач Django, який утворив закладку за оголошенням про демонтаж об'єкта, виражений зв'язком з таблицею `User` через зовнішній ключ (тип один-до-багатьох): зв'язок саме з цією таблицею встановлюється, тому що користувач може бути представником демонтажної бригади, замовником демонтажних робіт, менеджером програми.

Категорії техніки, що може використовуватися під час демонтажних робіт, представлені даними в таблиці `Dismantlingequipmentcategory` зі стовпцями:

– `id` – ідентифікатор категорії техніки, що може використовуватися під час демонтажних робіт;

– `dismantlingcategorycaption` – назва категорії техніки, що може використовуватися під час демонтажних робіт.

Виробники техніки, що можуть використовуватися під час демонтажних робіт, представлені даними в таблиці `Dismantlingequipmentbrand` зі стовпцями:

– `id` – ідентифікатор виробника техніки, що може використовуватися під час демонтажних робіт;

– `dismantlingbrandcaption` – назва виробника техніки, що може використовуватися під час демонтажних робіт.

Техніка, що може використовуватися для демонтажних робіт, представлена даними в таблиці `Dismantlingequipment` зі стовпцями:

– `id` – ідентифікатор техніки, що може використовуватися для демонтажних робіт;

– `dismantlingequipmentcategory` – категорія техніки, що може використовуватися для демонтажних робіт, виражена зв'язком з таблицею `Dismantlingequipmentcategory` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingcaption` – назва техніки або моделі техніки, що може використовуватися для демонтажних робіт;

– `dismantlingequipmentbrand` – виробник цієї моделі техніки, що може використовуватися для демонтажних робіт, виражений зв'язком з таблицею `Dismantlingequipmentbrand` через зовнішній ключ (тип один-до-багатьох).

Міста, в яких відбувається підтримка демонтажних робіт, представлені даними в таблиці `Dismantlingworkbasecity` зі стовпцями:

– `id` – ідентифікатор міста, в якому відбувається підтримка демонтажних робіт;

– `basecityname` – назва міста, в якому відбувається підтримка демонтажних робіт;

– `citycountry` – країна, де знаходиться місто, в якому відбувається підтримка демонтажних робіт.

Демонтажні бригади представлені даними в таблиці `Dismantlingcrew` зі стовпцями:

– `id` – ідентифікатор демонтажної бригади;

– `dismantlingcrewnumber` – кількість працівників у демонтажній бригаді;

– `dismantlingcrewregistration` – дата реєстрації демонтажної бригади у програмі;

– `dismantlingcrewestimate` – усереднена оцінка демонтажної бригади, отримана від замовників демонтажних робіт;

– `dismantlingcrewspes` – опис деталей, які представила про себе демонтажна бригада;

– `dismantlingcrewphoto` – фотографія демонтажної бригади;

– `dismantlingcrewblock` – позначка про блокування доступу демонтажної бригади для роботи з програмою;

– `dismantlingworkbasecity` – основне місто, де працює ця демонтажна бригада, виражене зв'язком з таблицею `Dismantlingworkbasecity` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingworkinregion` – позначка про готовність демонтажної бригади виконувати роботи не тільки у місті, де вона знаходиться, але і в області;

– `dismantlingworkincountry` – позначка про готовність демонтажної бригади виконувати роботи не тільки у місті, де вона знаходиться, але і в країні.

Техніка, що використовується демонтажною бригадою для виконання демонтажних робіт, представлена даними в таблиці `Dismantlingcrewequipment` зі стовпцями:

– `id` – ідентифікатор техніки, що використовується цією демонтажною бригадою для виконання демонтажних робіт;

– `dismantlingcrew` – демонtajна бригада, яка використовує цю техніку для демонтажних робіт, виражена зв'язком з таблицею `Dismantlingcrew` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingequipment` – модель техніки, що використовується в роботі цієї демонтажної бригади, виражена зв'язком з таблицею `Dismantlingequipment` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingquantity` – кількість об'єктів цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади;

– `dismantlingoldestproduction` – рік виготовлення найстарішого об'єкту цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади;

– `dismantlingnewestproduction` – рік виготовлення найновішого об'єкту цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади;

– `dismantlingphoto` – фотографія одного з об'єктів цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади.

Стандартний час роботи демонтажної бригади на день представлений даними в таблиці `Dismantlingcrewstdtime` зі стовпцями:

– id – ідентифікатор стандартного часу роботи демонтажної бригади на день;

– dismantlingcrew – демонtajна бригада, для якої визначено цей стандартний час роботи на день, виражена зв'язком з таблицею Dismantlingcrew через зовнішній ключ (тип один-до-багатьох);

– dismantlingdayoftheweek – день тижня або святковий день, у який застосовується такий графік роботи демонтажної бригади;

– dismantlingdaystart – час початку роботи демонтажної бригади у цей день тижня;

– dismantlingdayend – час закінчення роботи демонтажної бригади у цей день тижня;

– dismantlingreststart – час початку стандартної перерви у роботі демонтажної бригади у цей день тижня;

– dismantlingrestend – час закінчення стандартної перерви у роботі демонтажної бригади у цей день тижня.

Доступний час роботи демонтажної бригади на конкретні дати представлений даними в таблиці Dismantlingcrewtime зі стовпцями:

– id – ідентифікатор доступного часу роботи демонтажної бригади на конкретну дату;

– dismantlingcrew – демонtajна бригада, у якої доступний цей проміжок часу роботи на конкретну дату;

– dismantlingblocktimestart – дата і час початку доступного вікна для роботи у демонтажної бригади;

– dismantlingblocktimend – дата і час закінчення доступного вікна для роботи у демонтажної бригади.

Закладки за демонтажними бригадами представлені даними в таблиці Dismantlingcrewbook зі стовпцями:

– id – ідентифікатор закладки за демонтажною бригадою;

– dismantlingcrew – демонтажна бригада, за якою було утворено закладку, виражена зв'язком з таблицею Dismantlingcrew через зовнішній ключ (тип один-до-багатьох);

– dismantlingdjangouser – користувач Django, який утворив цю закладку за демонтажною бригадою.

Оцінки демонтажних бригад замовниками демонтажних послуг представлені даними в таблиці Dismantlingcrewestimation зі стовпцями:

– id – ідентифікатор оцінки демонтажної бригади замовником демонтажних послуг;

– dismantlecustomer – замовник демонтажних послуг, який оцінив таким чином демонтажну бригаду, виражений зв'язком з таблицею Dismantlecustomer через зовнішній ключ (тип один-до-багатьох);

– dismantlingcrew – демонтажна бригада, яку оцінив замовник демонтажних послуг, виражена зв'язком з таблицею Dismantlingcrew через зовнішній ключ (тип один-до-багатьох);

– dismantlingresultphoto – фотографія з результатом виконання демонтажних робіт бригадою, яку додав замовник демонтажних послуг;

– dismantlingmark – сама числова оцінка, поставлена замовником демонтажних робіт демонтажній бригаді;

– dismantlingtime – час оцінювання демонтажної бригади замовником демонтажних робіт;

– dismantlingcontent – оцінювання демонтажної бригади замовником демонтажних робіт у текстовому представленні.

Категорії демонтажних робіт представлені даними в таблиці Dismantlingcategory зі стовпцями:

– id – ідентифікатор категорії демонтажних робіт;

– dismantlingcategorycaption – назва категорії демонтажних робіт.

Безпосередньо демонтажні роботи представлені даними в таблиці Dismantlingwork зі стовпцями:

– id – ідентифікатор демонтажної роботи;

– `dismantlingcategory` – категорія, до якої належить демонтажна робота, виражена зв'язком з таблицею `Dismantlingcategory` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingworkcaption` – назва демонтажної роботи.

Демонтажні роботи, які вміють виконувати демонтажні бригади, представлені даними в таблиці `Dismantlingcrewwork` зі стовпцями:

– `id` – ідентифікатор демонтажної роботи, яку вміє виконувати демонтажна бригада;

– `dismantlingwork` – демонтажна робота, яку вміє виконувати демонтажна бригада, виражена зв'язком з таблицею `Dismantlingwork` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingcrew` – демонтажна бригада, яка здатна виконувати цю роботу, виражена зв'язком з таблицею `Dismantlingcrew` через зовнішній ключ (тип один-до-багатьох);

– `dismantlingspec` – опис особливостей виконання цієї демонтажної роботи цією бригадою;

– `dismantlingphoto` – фотографія виконання цієї демонтажної роботи цією бригадою;

– `dismantlingunit` – одиниця виконання цієї демонтажної роботи цією бригадою (наприклад, квадратний метр);

– `dismantlingprice` – вартість виконання одиниці цієї демонтажної роботи цією бригадою.

Запити за оголошеннями про демонтаж об'єктів представлені даними в таблиці `Dismantleadorder` зі стовпцями:

– `id` – ідентифікатор запиту за оголошенням про демонтаж об'єкта;

– `dismantlead` – оголошення про демонтаж об'єкта, за яким сформовано запит демонтажною бригадою, виражене зв'язком з таблицею `Dismantlead` через зовнішній ключ (тип один-до-багатьох);

– dismantlingcrew – демонтажна бригада, яка створила запит на виконання робіт за оголошенням про демонтаж об'єкта, виражена зв'язком з таблицею Dismantlingcrew через зовнішній ключ (тип один-до-багатьох);

– dismantlingstatus – стан запиту за оголошенням про демонтаж об'єктів;

– dismantlingprice – вартість робіт за демонтажем об'єкта з оголошення;

– dismantlingstart – час початку демонтажних робіт об'єкта з оголошення, зазначений у запиті;

– dismantlingend – час закінчення демонтажних робіт об'єкта з оголошення, зазначений у запиті;

– dismantlingspec – опис особливостей демонтажних робіт об'єкта з оголошення, визначених демонтажною бригадою (також може бути деталізацією виявленої проблеми за запитом).

Заплановані демонтажні роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта представлені даними в таблиці Dismantleadorderwork зі стовпцями:

– id – ідентифікатор запланованої демонтажної роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта;

– dismantleadorder – запит за оголошенням про демонтаж об'єкта, за яким запланована демонтажною бригадою ця робота, виражений зв'язком з таблицею Dismantleadorder через зовнішній ключ (тип один-до-багатьох);

– dismantlingcrewwork – демонтажна робота, яка запланована за запитом демонтажної бригади для виконання для демонтажу об'єкта за оголошенням, виражена зв'язком з таблицею Dismantlingcrewwork через зовнішній ключ (тип один-до-багатьох);

– dismantlingquantity – кількість одиниць виконання цієї демонтажної роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта;

– dismantlingprice – вартість виконання цієї демонтажної роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта;

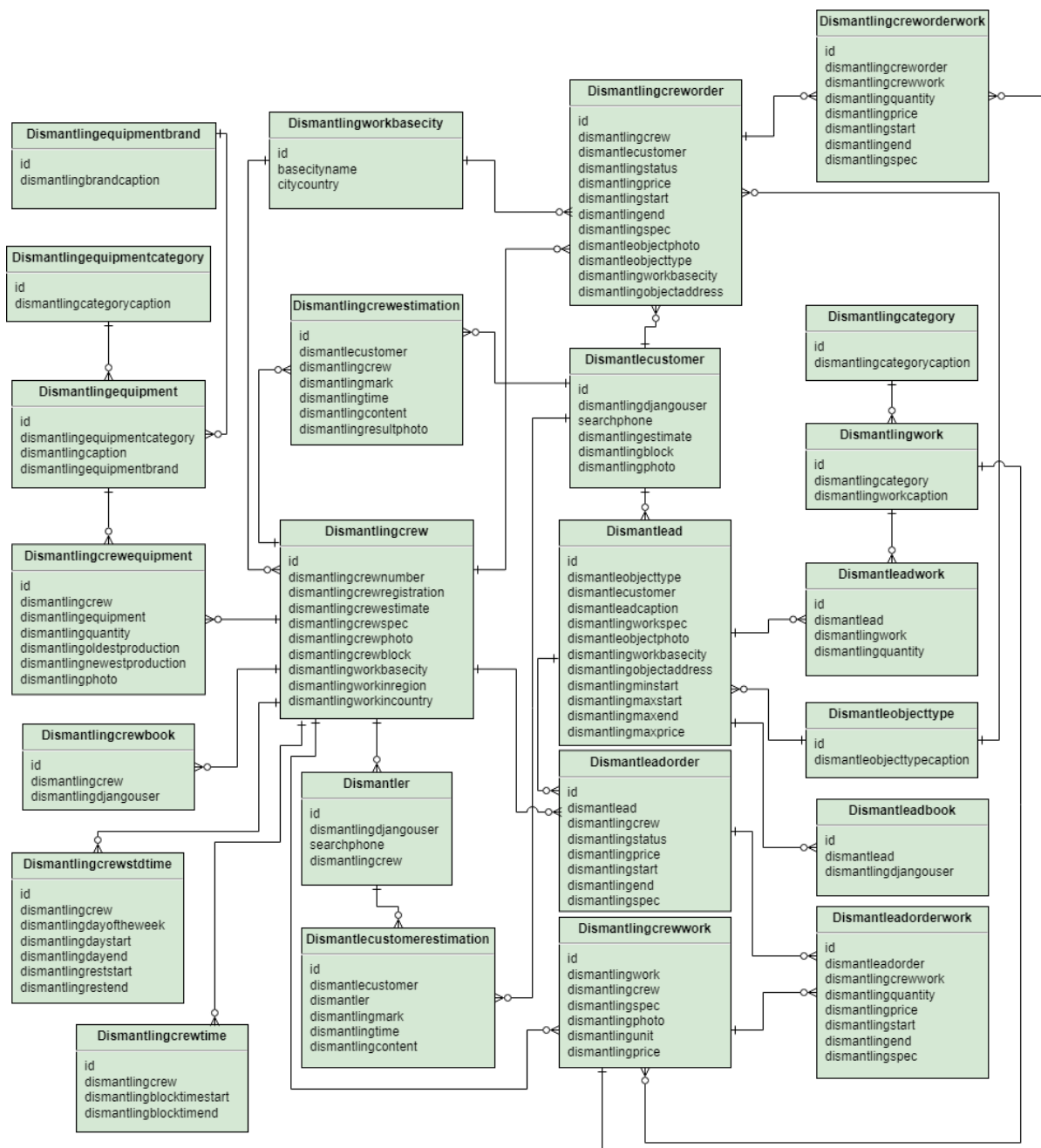


Рисунок 2.5 – Схема бази даних програми підтримки демонтажних робіт

Замовлення демонтажних робіт представлені даними в таблиці Dismantlingcreworder зі стовпцями:

- id – ідентифікатор замовлення демонтажних робіт;
- dismantlingcrew – демонтажна бригада, у якій замовлено ці демонтажні роботи замовником, виражена зв'язком з таблицею Dismantlingcrew через зовнішній ключ (тип один-до-багатьох);
- dismantlecustomer – замовник демонтажних робіт, який замовив ці демонтажні роботи у демонтажній бригаді, виражений зв'язком з таблицею Dismantlecustomer через зовнішній ключ (тип один-до-багатьох);
- dismantlingstatus – стан замовлення демонтажних робіт;
- dismantlingprice – вартість демонтажних робіт за замовленням;
- dismantlingstart – час початку виконання замовлених демонтажних робіт;
- dismantlingend – час закінчення виконання замовлених демонтажних робіт;
- dismantlingspec – опис особливостей виконання замовлених демонтажних робіт (також може бути деталізацією виявленої проблеми за замовленням);
- dismantleobjectphoto – фотографія об'єкта, стосовно якого замовлено виконання демонтажних робіт у бригаді;
- dismantleobjecttype – тип об'єкта, стосовно якого замовлено виконання демонтажних робіт у бригаді, виражений зв'язком з таблицею Dismantleobjecttype через зовнішній ключ (тип один-до-багатьох);
- dismantlingworkbasecity – місто, де знаходиться об'єкт, для якого замовлено демонтажні роботи;
- dismantlingobjectaddress – адреса об'єкта, для якого замовлено демонтажні роботи.

Склад демонтажних робіт, які замовлено для виконання у демонтажній бригаді, представлений даними в таблиці Dismantlingcreworderwork зі стовпцями:

- `id` – ідентифікатор демонтажної роботи у складі замовлення для виконання у демонтажній бригаді;
- `dismantlingcreworder` – замовлення демонтажних робіт, для якого визначено ці роботи, виражене зв'язком з таблицею `Dismantlingcreworder` через зовнішній ключ (тип один-до-багатьох);
- `dismantlingcrewwork` – демонтажна робота, яка входить до складу замовлення, виражена зв'язком з таблицею `Dismantlingcrewwork` через зовнішній ключ (тип один-до-багатьох);
- `dismantlingquantity` – кількість одиниць виконання демонтажних робіт, яку замовлено;
- `dismantlingprice` – вартість виконання цих демонтажних робіт з замовлення;
- `dismantlingstart` – час початку виконання цих демонтажних робіт з замовлення;
- `dismantlingend` – час закінчення виконання цих демонтажних робіт з замовлення;
- `dismantlingspec` – опис особливостей виконання цих демонтажних робіт з замовлення.

2.4 Висновки за розділом 2

Підсумком розділу 2 є результати виконання проектування програми підтримки демонтажних робіт, включаючи обґрунтування вибору зв'язки Python і Django для розробки, визначення сценаріїв роботи представників демонтажних бригад, замовників демонтажних робіт, неавторизованих користувачів, менеджерів програми, визначення структури бази даних.

3 РОЗРОБКА ПРОГРАМИ

3.1 Керування станами основних сутностей

Зміна станів замовлень демонтажних робіт (рис. 3.1) та запитів за оголошеннями про демонтаж об'єктів відбувається за однаковою схемою, проте сторони учасники змінюються місцями.

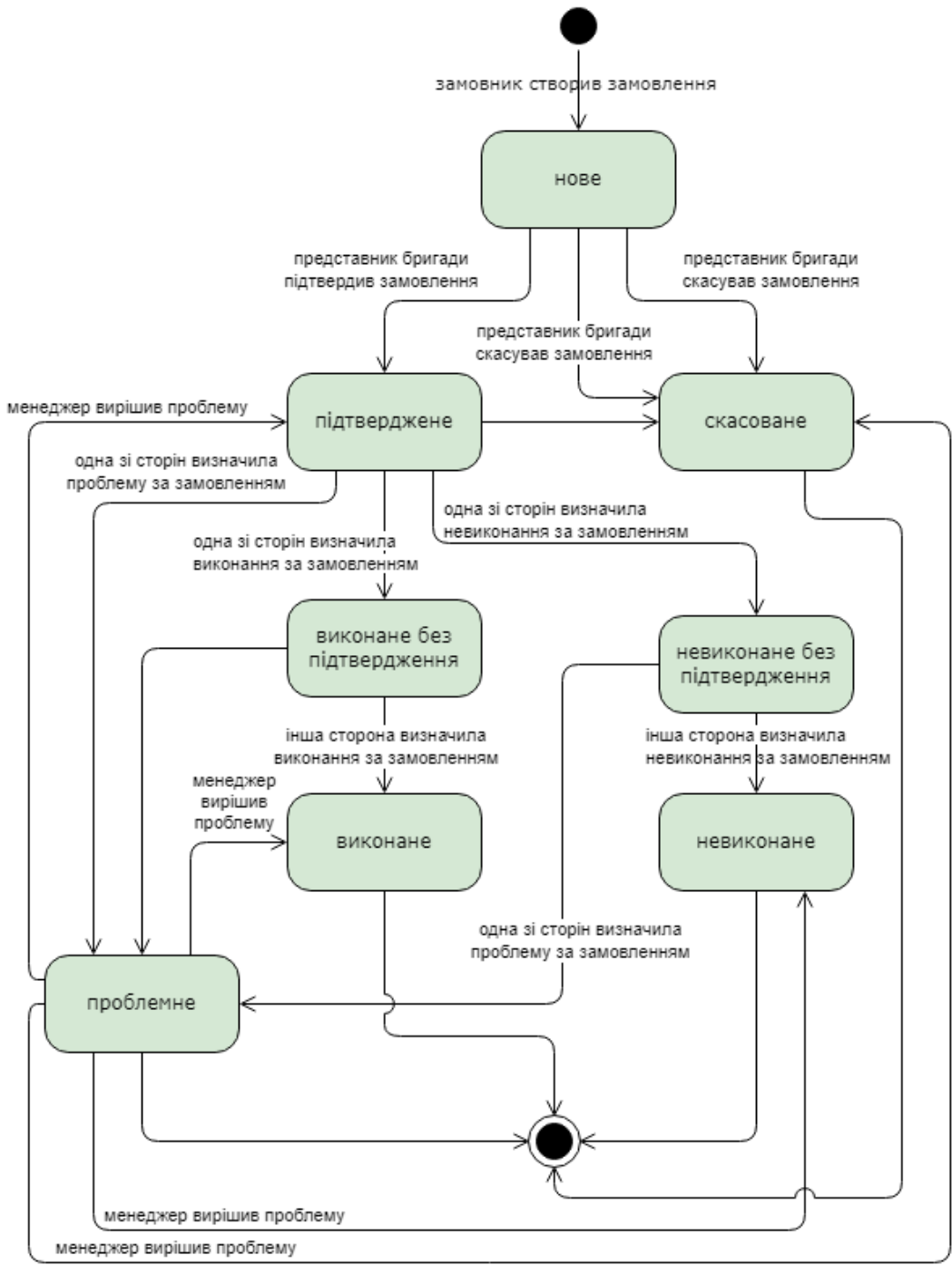


Рисунок 3.1 – Діаграма станів замовлення демонтажних робіт

3.2 Реалізація програми підтримки демонтажних робіт

Реалізацію програми підтримки демонтажних робіт забезпечено ФВР Django, організовуючи структуру за шаблоном Model-View-Template (MVT).

Нижче моделі зі структури MVT описані.

Модель представника демонтажної бригади реалізована класом Dismantler.

Поле `dismantlingdjangouser` в класі Dismantler з типом `ForeignKey` є акаунтом користувача в Django, що застосовується представником демонтажної бригади.

Поле `searchphone` в класі Dismantler з типом `CharField` є номером телефону представника демонтажної бригади.

Поле `dismantlingcrew` в класі Dismantler з типом `ForeignKey` є демонтажною бригадою, в якій працює і яку представляє цей користувач.

Модель замовника демонтажних робіт реалізована класом Dismantlecustomer.

Поле `dismantlingdjangouser` в класі Dismantlecustomer з типом `ForeignKey` є акаунтом користувача в Django, що застосовується замовником демонтажних робіт.

Поле `searchphone` в класі Dismantlecustomer з типом `CharField` є номером телефону замовника демонтажних робіт.

Поле `dismantlingestimate` в класі Dismantlecustomer з типом `DecimalField` є усередненою оцінкою замовника демонтажних робіт.

Поле `dismantlingblock` в класі Dismantlecustomer з типом `BooleanField` є позначкою про блокування доступу замовника демонтажних робіт для роботи з програмою.

Поле `dismantlingphoto` в класі Dismantlecustomer з типом `ImageField` є фотографією замовника демонтажних робіт.

Модель оцінки замовників демонтажних робіт від демонтажних бригад реалізована класом Dismantlecustomerestimation.

Поле `dismantlecustomer` в класі `Dismantlecustomerestimation` з типом `ForeignKey` є замовником демонтажних робіт, якому виставлено демонтажної бригадою у обличчі її представника оцінку.

Поле `dismantler` в класі `Dismantlecustomerestimation` з типом `ForeignKey` є користувачем-представником демонтажної бригади, який виконав оцінювання.

Поле `dismantlingmark` в класі `Dismantlecustomerestimation` з типом `PositiveSmallIntegerField` є самою числовою оцінкою, поставленою замовнику демонтажних робіт демонтажною бригадою.

Поле `dismantlingtime` в класі `Dismantlecustomerestimation` з типом `DateTimeField` є часом оцінювання демонтажною бригадою замовника демонтажних робіт.

Поле `dismantlingcontent` в класі `Dismantlecustomerestimation` з типом `CharField` є оцінюванням демонтажною бригадою замовника демонтажних робіт у текстовому представленні.

Модель типів об'єктів для демонтажу реалізована класом `Dismantleobjecttype`.

Поле `dismantleobjecttypecaption` в класі `Dismantleobjecttype` з типом `CharField` є назвою типу об'єктів для демонтажу.

Модель оголошення про демонтаж об'єктів пр реалізована класом `Dismantlead`.

Поле `dismantleobjecttype` в класі `Dismantlead` з типом `ForeignKey` є типом об'єкта, демонтаж якого необхідно виконати за оголошенням замовника демонтажних робіт.

Поле `dismantlecustomer` в класі `Dismantlead` з типом `ForeignKey` є замовником демонтажних робіт, який створив це оголошення про демонтаж об'єкту.

Поле `dismantleadcaption` в класі `Dismantlead` з типом `CharField` є назвою оголошення про демонтаж об'єкту.

Поле `dismantlingworkspec` в класі `Dismantlead` з типом `CharField` є описом особливостей демонтажних робіт, які за оголошенням необхідно виконати для демонтажу об'єкту.

Поле `dismantleobjectphoto` в класі `Dismantlead` з типом `ImageField` є фотографією об'єкту, на якому за оголошенням потрібно виконати демонтаж.

Поле `dismantlingworkbasecity` в класі `Dismantlead` з типом `ForeignKey` є містом, в якому знаходиться об'єкт, демонтаж якого за оголошенням потрібно виконати.

Поле `dismantlingobjectaddress` в класі `Dismantlead` з типом `CharField` є адресою об'єкту, на якому за оголошенням потрібно виконати демонтаж.

Поле `dismantlingminstart` в класі `Dismantlead` з типом `DateField` є датою, починаючи з якої потрібно виконати демонтаж об'єкту за оголошенням (найбільш рання дата).

Поле `dismantlingmaxstart` в класі `Dismantlead` з типом `DateField` є датою, до якої потрібно почати виконання демонтаж об'єкту за оголошенням (найбільш пізня дата початку).

Поле `dismantlingmaxend` в класі `Dismantlead` з типом `DateField` є датою, до якої потрібно закінчити демонтаж об'єкту за оголошенням.

Поле `dismantlingmaxprice` в класі `Dismantlead` з типом `DecimalField` є максимальною вартістю, яку за виконання цих робіт для демонтажу об'єкту за оголошенням готовий заплатити замовник демонтажних робіт.

Поле `bookedbyusers` в класі `Dismantlead` з типом `ManyToManyField` є закладками за цим оголошенням про демонтаж об'єктів, зробленими іншими користувачами, на основі моделі `User` і таким чином типу багато-до-багатьох.

Модель демонтажних робіт, які мають бути на думку замовника демонтажних робіт виконані для демонтажу його об'єкту за оголошенням, реалізована класом `Dismantleadwork`.

Поле `dismantlead` в класі `Dismantlead` з типом `ForeignKey` є оголошенням про демонтаж об'єкту замовника, за яким на його думку має бути виконано цю демонтажну роботу для досягнення результату.

Поле `dismantlingwork` в класі `Dismantlead` з типом `ForeignKey` є демонтажною роботою, яка на думку замовника таких робіт має бути виконана за його оголошенням про демонтаж об'єкту для досягнення результату.

Поле `dismantlingquantity` в класі `Dismantlead` з типом `DecimalField` є кількістю одиниць обчислення цієї демонтажної роботи (наприклад, квадратних кілометрів площі), яка на думку замовника демонтажних робіт має бути виконана за його оголошенням про демонтаж об'єкту для досягнення результату.

Модель категорії техніки, що може використовуватися під час демонтажних робіт, реалізована класом `Dismantlingequipmentcategory`.

Поле `dismantlingcategorycaption` в класі `Dismantlead` з типом `CharField` є назвою категорії техніки, що може використовуватися під час демонтажних робіт.

Модель виробників техніки, що можуть використовуватися під час демонтажних робіт, реалізована класом `Dismantlingequipmentbrand`.

Поле `dismantlingbrandcaption` в класі `Dismantlingequipmentbrand` з типом `CharField` є назвою виробника техніки, що може використовуватися під час демонтажних робіт.

Модель техніки, що може використовуватися для демонтажних робіт, реалізована класом `Dismantlingequipment`.

Поле `dismantlingequipmentcategory` в класі `Dismantlingequipment` з типом `ForeignKey` є категорією техніки, що може використовуватися для демонтажних робіт.

Поле `dismantlingcaption` в класі `Dismantlingequipment` з типом `CharField` є назвою техніки або моделі техніки, що може використовуватися для демонтажних робіт.

Поле `dismantlingequipmentbrand` в класі `Dismantlingequipment` з типом `ForeignKey` є виробником цієї моделі техніки, що може використовуватися для демонтажних робіт.

Модель міст, в яких відбувається підтримка демонтажних робіт, реалізована класом `Dismantlingworkbasecity`.

Поле `basecityname` в класі `Dismantlingequipment` з типом `CharField` є назвою міста, в якому відбувається підтримка демонтажних робіт.

Поле `citycountry` в класі `Dismantlingequipment` з типом `CharField` є країною, де знаходиться місто, в якому відбувається підтримка демонтажних робіт.

Модель демонтажних бригад реалізована класом `Dismantlingcrew`.

Поле `dismantlingcrewnumber` в класі `Dismantlingcrew` з типом `PositiveSmallIntegerField` є кількістю працівників у демонтажній бригаді.

Поле `dismantlingcrewregistration` в класі `Dismantlingcrew` з типом `DateField` є датою реєстрації демонтажної бригади у програмі.

Поле `dismantlingcrewestimate` в класі `Dismantlingcrew` з типом `DecimalField` є усередненою оцінкою демонтажної бригади, отримана від замовників демонтажних робіт.

Поле `dismantlingcrewspec` в класі `Dismantlingcrew` з типом `CharField` є описом деталей, які представила про себе демонтажна бригада.

Поле `dismantlingcrewphoto` в класі `Dismantlingcrew` з типом `ImageField` є фотографією демонтажної бригади.

Поле `dismantlingcrewblock` в класі `Dismantlingcrew` з типом `BooleanField` є позначкою про блокування доступу демонтажної бригади для роботи з програмою.

Поле `dismantlingworkbasecity` в класі `Dismantlingcrew` з типом `ForeignKey` є основним містом, де працює ця демонтажна бригада.

Поле `dismantlingworkinregion` в класі `Dismantlingcrew` з типом `BooleanField` є позначкою про готовність демонтажної бригади виконувати роботи не тільки у місті, де вона знаходиться, але і в області.

Поле `dismantlingworkincountry` в класі `Dismantlingcrew` з типом `CharField` є позначкою про готовність демонтажної бригади виконувати роботи не тільки у місті, де вона знаходиться, але і в країні.

Поле `bookedbyusers` в класі `Dismantlingcrew` з типом `ManyToManyField` є закладками за цією демонтажною бригадою, зробленими іншими користувачами, на основі моделі `User` і таким чином типу багато-до-багатьох.

Модель техніки, що використовується демонтажною бригадою для виконання демонтажних робіт, реалізована класом `Dismantlingcrewequipment`.

Поле `dismantlingcrew` в класі `Dismantlingcrewequipment` з типом `ForeignKey` є демонтажною бригадою, яка використовує цю техніку для демонтажних робіт.

Поле `dismantlingequipment` в класі `Dismantlingcrewequipment` з типом `ForeignKey` є моделлю техніки, що використовується в роботі цієї демонтажної бригади.

Поле `dismantlingquantity` в класі `Dismantlingcrewequipment` з типом `PositiveSmallIntegerField` є кількістю об'єктів цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади.

Поле `dismantlingoldestproduction` в класі `Dismantlingcrewequipment` з типом `PositiveSmallIntegerField` є роком виготовлення найстарішого об'єкту цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади.

Поле `dismantlingnewestproduction` в класі `Dismantlingcrewequipment` з типом `PositiveSmallIntegerField` є роком виготовлення найновішого об'єкту цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади.

Поле `dismantlingphoto` в класі `Dismantlingcrewequipment` з типом `ImageField` є фотографією одного з об'єктів цієї моделі цієї техніки, що використовується в роботі цієї демонтажної бригади.

Модель стандартного часу роботи демонтажної бригади на день реалізована класом `Dismantlingcrewstdtime`.

Поле `dismantlingcrew` в класі `Dismantlingcrewstdtime` з типом `ForeignKey` є демонтажною бригадою, для якої визначено цей стандартний час роботи на день.

Поле `dismantlingdayoftheweek` в класі `Dismantlingcrewstdtime` з типом `PositiveSmallIntegerField` є днем тижня або святковим днем, у який застосовується такий графік роботи демонтажної бригади.

Поле `dismantlingdaystart` в класі `Dismantlingcrewstdtime` з типом `TimeField` є часом початку роботи демонтажної бригади у цей день тижня.

Поле `dismantlingdayend` в класі `Dismantlingcrewstdtime` з типом `TimeField` є часом закінчення роботи демонтажної бригади у цей день тижня.

Поле `dismantlingreststart` в класі `Dismantlingcrewstdtime` з типом `TimeField` є часом початку стандартної перерви у роботі демонтажної бригади у цей день тижня.

Поле `dismantlingrestend` в класі `Dismantlingcrewstdtime` з типом `TimeField` є часом закінчення стандартної перерви у роботі демонтажної бригади у цей день тижня.

Модель доступного часу роботи демонтажної бригади на конкретні дати реалізована класом `Dismantlingcrewtime`.

Поле `dismantlingcrew` в класі `Dismantlingcrewtime` з типом `ForeignKey` є демонтажною бригадою, у якої доступний цей проміжок часу роботи на конкретну дату.

Поле `dismantlingblocktimestart` в класі `Dismantlingcrewtime` з типом `DateTimeField` є датою і часом початку доступного вікна для роботи у демонтажної бригади.

Поле `dismantlingblocktimend` в класі `Dismantlingcrewtime` з типом `DateTimeField` є датою і часом закінчення доступного вікна для роботи у демонтажної бригади.

Модель оцінки демонтажних бригад замовниками демонтажних послуг реалізована класом `Dismantlingcrewestimation`.

Поле `dismantlecustomer` в класі `Dismantlingcrewestimation` з типом `ForeignKey` є замовником демонтажних послуг, який оцінив таким чином демонтажну бригаду.

Поле `dismantlingcrew` в класі `Dismantlingcrewestimation` з типом `ForeignKey` є демонтажною бригадою, яку оцінив замовник демонтажних послуг.

Поле `dismantlingresultphoto` в класі `Dismantlingcrewestimation` з типом `ImageField` є фотографією з результатом виконання демонтажних робіт бригадою, яку додав замовник демонтажних послуг.

Поле `dismantlingmark` в класі `Dismantlingcrewestimation` з типом `DecimalField` є самою числовою оцінкою, поставленою замовником демонтажних робіт демонтажній бригаді.

Поле `dismantlingtime` в класі `Dismantlingcrewestimation` з типом `DateTimeField` є часом оцінювання демонтажної бригади замовником демонтажних робіт.

Поле `dismantlingcontent` в класі `Dismantlingcrewestimation` з типом `CharField` є оцінюванням демонтажної бригади замовником демонтажних робіт у текстовому представленні.

Модель категорії демонтажних робіт реалізована класом `Dismantlingcategory`.

Поле `dismantlingcategorycaption` в класі `Dismantlingcategory` з типом `CharField` є назвою категорії демонтажних робіт.

Модель демонтажних робіт реалізована класом `Dismantlingwork`.

Поле `dismantlingcategory` в класі `Dismantlingwork` з типом `ForeignKey` є категорією, до якої належить демонтажна робота.

Поле `dismantlingworkcaption` в класі `Dismantlingwork` з типом `CharField` є назвою демонтажної роботи.

Модель демонтажних робіт, які вміють виконувати демонтажні бригади, реалізована класом `Dismantlingcrewwork`.

Поле `dismantlingwork` в класі `Dismantlingcrewwork` з типом `ForeignKey` є демонтажною роботою, яку вміє виконувати демонтажна бригада.

Поле `dismantlingcrew` в класі `Dismantlingcrewwork` з типом `ForeignKey` є демонтажною бригадою, яка здатна виконувати цю роботу.

Поле `dismantlingspec` в класі `Dismantlingcrewwork` з типом `CharField` є описом особливостей виконання цієї демонтажної роботи цією бригадою.

Поле `dismantlingphoto` в класі `Dismantlingcrewwork` з типом `ImageField` є фотографією виконання цієї демонтажної роботи цією бригадою.

Поле `dismantlingunit` в класі `Dismantlingcrewwork` з типом `CharField` є одиницею виконання цієї демонтажної роботи цією бригадою.

Поле `dismantlingprice` в класі `Dismantlingcrewwork` з типом `DecimalField` є вартістю виконання одиниці цієї демонтажної роботи цією бригадою.

Модель запитів за оголошеннями про демонтаж об'єктів реалізована класом `Dismantleadorder`.

Поле `dismantlead` в класі `Dismantleadorder` з типом `ForeignKey` є оголошенням про демонтаж об'єкта, за яким сформовано запит демонтажною бригадою.

Поле `dismantlingcrew` в класі `Dismantleadorder` з типом `ForeignKey` є демонтажною бригадою, яка створила запит на виконання робіт за оголошенням про демонтаж об'єкта.

Поле `dismantlingstatus` в класі `Dismantleadorder` з типом `CharField` є станом запиту за оголошенням про демонтаж об'єктів.

Поле `dismantlingprice` в класі `Dismantleadorder` з типом `DecimalField` є вартістю робіт за демонтажем об'єкта з оголошення.

Поле `dismantlingstart` в класі `Dismantleadorder` з типом `DateTimeField` є часом початку демонтажних робіт об'єкта з оголошення, зазначений у запиті.

Поле `dismantlingend` в класі `Dismantleadorder` з типом `DateTimeField` є часом закінчення демонтажних робіт об'єкта з оголошення, зазначеним у запиті.

Поле `dismantlingspec` в класі `Dismantleadorder` з типом `CharField` є описом особливостей демонтажних робіт об'єкта з оголошення, визначених демонтажною бригадою.

Модель запланованих демонтажних робіт за запитом демонтажної бригади за оголошенням про демонтаж об'єкта реалізована класом `Dismantleadorderwork`.

Поле `dismantleadorder` в класі `Dismantleadorderwork` з типом `ForeignKey` є запитом за оголошенням про демонтаж об'єкта, за яким запланована демонтажною бригадою ця робота.

Поле `dismantlingcrewwork` в класі `Dismantleadorderwork` з типом `ForeignKey` є демонтажною роботою, яка запланована за запитом демонтажної бригади для виконання для демонтажу об'єкта за оголошенням.

Поле `dismantlingquantity` в класі `Dismantleadorderwork` з типом `DecimalField` є кількістю одиниць виконання цієї демонтажної роботи за запитом демонтажної бригади за оголошенням.

Поле `dismantlingprice` в класі `Dismantleadorderwork` з типом `DecimalField` є вартістю виконання цієї демонтажної роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта.

Поле `dismantlingstart` в класі `Dismantleadorderwork` з типом `DateTimeField` є часом початку виконання цієї демонтажної роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта.

Поле `dismantlingend` в класі `Dismantleadorderwork` з типом `DateTimeField` є часом закінчення виконання цієї демонтажної роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта.

Поле `dismantlingspec` в класі `Dismantleadorderwork` з типом `CharField` є описом особливостей виконання цієї демонтажної роботи за запитом демонтажної бригади за оголошенням про демонтаж об'єкта.

Модель замовлення демонтажних робіт реалізована класом `Dismantlingcreworder`.

Поле `dismantlingcrew` в класі `Dismantlingcreworder` з типом `ForeignKey` є демонтажною бригадою, у якої замовлено ці демонтажні роботи замовником.

Поле `dismantlecustomer` в класі `Dismantlingcreworder` з типом `ForeignKey` є замовником демонтажних робіт, який замовив ці демонтажні роботи у демонтажної бригади.

Поле `dismantlingstatus` в класі `Dismantlingcreworder` з типом `CharField` є станом замовлення демонтажних робіт.

Поле `dismantlingprice` в класі `Dismantlingcreworder` з типом `DecimalField` є вартістю демонтажних робіт за замовленням.

Поле `dismantlingstart` в класі `Dismantlingcreworder` з типом `DateTimeField` є часом початку виконання замовлених демонтажних робіт.

Поле `dismantlingend` в класі `Dismantlingcreworder` з типом `DateTimeField` є часом закінчення виконання замовлених демонтажних робіт.

Поле `dismantlingspec` в класі `Dismantlingcreworder` з типом `CharField` є описом особливостей виконання замовлених демонтажних робіт.

Поле `dismantleobjectphoto` в класі `Dismantlingcreworder` з типом `ImageField` є фотографією об'єкта, стосовно якого замовлено виконання демонтажних робіт у бригади.

Поле `dismantleobjecttype` в класі `Dismantlingcreworder` з типом `ForeignKey` є типом об'єкта, стосовно якого замовлено виконання демонтажних робіт у бригади.

Поле `dismantlingworkbasecity` в класі `Dismantlingcreworder` з типом `ForeignKey` є містом, де знаходиться об'єкт, для якого замовлено демонтажні роботи.

Поле `dismantlingobjectaddress` в класі `Dismantlingcreworder` з типом `CharField` є адресою об'єкта, для якого замовлено демонтажні роботи.

Модель складу демонтажних робіт, які замовлено для виконання у демонтажної бригади, реалізована класом `Dismantlingcreworderwork`.

Поле `dismantlingcreworder` в класі `Dismantlingcreworderwork` з типом `ForeignKey` є замовленням демонтажних робіт, для якого визначено ці роботи.

Поле `dismantlingcrewwork` в класі `Dismantlingcreworderwork` з типом `ForeignKey` є демонтажною роботою, яка входить до складу замовлення.

Поле `dismantlingquantity` в класі `Dismantlingcreworderwork` з типом `DecimalField` є кількістю одиниць виконання демонтажних робіт, яку замовлено.

Поле `dismantlingprice` в класі `Dismantlingcreworderwork` з типом `DecimalField` є вартістю виконання цих демонтажних робіт з замовлення.

Поле `dismantlingstart` в класі `Dismantlingcreworderwork` з типом `DateTimeField` є часом початку виконання цих демонтажних робіт з замовлення.

Поле `dismantlingend` в класі `Dismantlingcreworderwork` з типом `DateTimeField` є часом закінчення виконання цих демонтажних робіт з замовлення.

Поле `dismantlingspec` в класі `Dismantlingcreworderwork` з типом `CharField` є описом особливостей виконання цих демонтажних робіт з замовлення.

Розроблені представлення та шаблони зі структури MVT описані основним чином нижче. Для кожного представлення створено шаблон вебсторінки. Для реалізації шаблонів використано вільний шаблон [7].

Перегляд своїх нових замовлень демонтажних робіт забезпечено функцією `own_new_dismantlingcrew_orders`, яка не потребує параметрів як і аналогічні наступні функції, тому що всі потрібні дані беруться з акаунту, яким авторизовано користувача, та відповідно з бази даних.

Перегляд своїх виконаних замовлень демонтажних робіт забезпечено функцією `own_executed_dismantlingcrew_orders`, яка має параметр `execute_level`, який визначає, що це тільки повністю виконані замовлення, тільки підтверджені однією стороною (без підтвердження), або і ті, і ті.

Перегляд своїх запланованих замовлень демонтажних робіт забезпечено функцією `own_planned_dismantlingcrew_orders`.

Перегляд своїх проблемних замовлень демонтажних робіт забезпечено функцією `own_problem_dismantlingcrew_orders`, а параметр `problem_level` визначає, чи це тільки проблемні, чи також проблемні і потенційно або

повністю невиконані замовлення, або відповідно невиконані без підтвердження або з підтвердженням.

Перегляд своїх скасованих замовлень демонтажних робіт забезпечено функцією `own_cancel_dismantlingcrew_orders`.

Перегляд інформації про демонтажну бригаду забезпечено функцією `dismantlingcrew_info`, а параметр `dismantlingcrewid` задає ідентифікатор сторінки демонтажної бригади.

Утворення закладки на сторінці демонтажної бригади забезпечено функцією `book_dismantlingcrew`, а параметр `dismantlingcrewid` задає ідентифікатор сторінки демонтажної бригади. Якщо така закладка вже в базі даних збережена для цього користувача, то реалізується вилучення закладки на сторінці демонтажної бригади.

Перегляд закладених сторінок демонтажних бригад забезпечено функцією `books_dismantlingcrew`.

Перегляд адресованих демонтажній бригаді нових замовлень забезпечено функцією `group_new_dismantlingcrew_orders`.

Перегляд адресованих демонтажній бригаді виконаних замовлень забезпечено функцією `group_own_executed_dismantlingcrew_orders`, яка має параметр `execute_level`, який визначає, що це тільки повністю виконані замовлення, тільки підтверджені однією стороною (без підтвердження), або і ті, і ті.

Перегляд адресованих демонтажній бригаді запланованих замовлень забезпечено функцією `group_own_planned_dismantlingcrew_orders`.

Перегляд адресованих демонтажній бригаді проблемних замовлень забезпечено функцією `group_own_problem_dismantlingcrew_orders`, а параметр `problem_level` визначає, чи це тільки проблемні, чи також проблемні і потенційно або повністю невиконані замовлення, або відповідно невиконані без підтвердження або з підтвердженням.

Перегляд адресованих демонтажній бригаді скасованих замовлень забезпечено функцією `group_own_cancel_dismantlingcrew_orders`.

Заповнення інформації про демонтажну бригаду забезпечено функцією `dismantlingcrew_fill` на основі класів форм `DismantlingcrewForm`, `DismantlingcrewequipmentForm`.

Редагування інформації про демонтажну бригаду забезпечено функцією `dismantlingcrew_edit`, задаючи ідентифікатор демонтажної бригади параметром `dismantlingcrewid`.

Додавання техніки, використовуваної демонтажною бригадою, забезпечено функцією `dismantlingcrew_equipment_add`, що працює на основі класу форми `DismantlingcrewequipmentForm`.

Вилучення техніки, використовуваної демонтажною бригадою, забезпечено функцією `dismantlingcrew_equipment_delete` з параметром `dismantlingcrewequipmentid`, що задає ідентифікатор такої техніки.

Додавання демонтажної роботи до послуг бригади забезпечено функцією `dismantlingcrew_work_add`, що працює на основі класу форми `DismantlingcrewworkForm`.

Редагування параметрів демонтажної роботи у послугах бригади забезпечено функцією `dismantlingcrew_work_edit`, що працює на основі класу форми `DismantlingcrewworkForm`, а параметр `dismantlingcrewworkid` задає ідентифікатор самої демонтажної роботи у бригади.

Вилучення демонтажної роботи з послуг бригади забезпечено функцією `dismantlingcrew_work_delete`, а параметр `dismantlingcrewworkid` задає ідентифікатор самої демонтажної роботи у бригади.

Замовлення демонтажних робіт забезпечено функцією `dismantlingcreworder_new`, яка працює на основі класів форм `DismantlingcreworderForm` та `DismantlingcreworderworkForm`.

Підтвердження замовлення демонтажних робіт забезпечено функцією `dismantlingcreworder_approve`, параметр якої `dismantlingcreworderid` задає ідентифікатор замовлення демонтажних робіт.

Скасування замовлення демонтажних робіт забезпечено функцією `dismantlingcreworder_cancel`, параметр якої `dismantlingcreworderid` задає ідентифікатор замовлення демонтажних робіт.

Підтвердження виконання замовлення демонтажних робіт забезпечено функцією `dismantlingcreworder_execute`, параметр якої `dismantlingcreworderid` задає ідентифікатор замовлення демонтажних робіт.

Підтвердження невиконання замовлення демонтажних робіт забезпечено функцією `dismantlingcreworder_unexecute`, параметр якої `dismantlingcreworderid` задає ідентифікатор замовлення демонтажних робіт.

Підтвердження проблем з замовленням демонтажних робіт забезпечено функцією `dismantlingcreworder_problem`, параметр якої `dismantlingcreworderid` задає ідентифікатор замовлення демонтажних робіт.

Утворення закладки на оголошенні про демонтаж об'єктів забезпечено функцією `book_dismantlead`, а параметр `dismantleadid` задає ідентифікатор оголошення про демонтаж об'єкта. Якщо така закладка вже в базі даних збережена для цього користувача, то реалізується вилучення закладки на оголошенні про демонтаж об'єктів.

Перегляд закладених оголошень про демонтаж об'єктів забезпечено функцією `books_dismantlead`.

Перегляд адресованих заявок про демонтаж об'єктів забезпечено функцією `own_dismantlead_orders`.

Перегляд своїх нових заявок про демонтаж об'єктів забезпечено функцією `own_new_dismantlead_orders`. Заявки обираються за демонтажною бригадою, представником якої є авторизований користувач.

Перегляд своїх виконаних заявок про демонтаж об'єктів забезпечено функцією `own_executed_dismantlead_orders`, яка має параметр `execute_level`, який визначає, що це тільки повністю виконані замовлення, тільки підтверджені однією стороною (без підтвердження), або і ті, і ті.

Перегляд своїх запланованих заявок про демонтаж об'єктів забезпечено функцією `own_planned_dismantlead_orders`.

Перегляд своїх проблемних заявок про демонтаж об'єктів забезпечено функцією `own_problem_dismantlead_orders`, а параметр `problem_level` визначає, чи це тільки проблемні, чи також проблемні і потенційно або повністю невиконані замовлення, або відповідно невиконані без підтвердження або з підтвердженням.

Перегляд своїх скасованих заявок про демонтаж об'єктів забезпечено функцією `own_cancel_dismantlead_orders`.

Перегляд оцінок демонтажних робіт від користувача забезпечено функцією `user_dismantle_estimations`, а логін користувача задається параметром `userlogin`, параметр `estimtype` встановлює, чи це відгуки за замовленнями демонтажних робіт, чи це відгуки за запитами за оголошеннями про демонтаж об'єкту.

Перегляд оцінок демонтажної бригади забезпечено функцією `dismantlingcrew_estimations`, а параметр `dismantlingcrewid` задає ідентифікатор демонтажної бригади.

Перегляд оцінок замовника демонтажних робіт забезпечено функцією `dismantlecustomer_estimations`, а параметр `dismantlecustomerid` задає ідентифікатор замовника демонтажних робіт.

Відображення графіку роботи демонтажної бригади на дату забезпечено функцією `dismantlingcrew_schedule`, параметр якої `dismantlingdate` задає дату, а ідентифікатор демонтажної бригади встановлюється за тим, яку бригаду представляє авторизований користувач.

Перегляд усіх виконаних демонтажних робіт бригади забезпечено функцією `executed_dismantlead_orders`, яка має параметр `execute_level`, який визначає, що це тільки повністю виконані замовлення, тільки підтверджені однією стороною (без підтвердження), або і ті, і ті. У результати об'єднуються і замовлення демонтажних робіт у цієї демонтажної бригади, і запити самої демонтажної бригади на виконання демонтажу об'єктів.

Перегляд проблемних замовлень демонтажних робіт забезпечено функцією `problem_dismantlingcreworders`, при цьому параметр `dismantlingdate` задає дату, на яку такі замовлення мають бути оформлені, хоча це значення може бути і не задано.

Перегляд проблемних заявок про демонтаж об'єктів забезпечено функцією `problem_dismantleadorders`, при цьому параметр `dismantlingdate` задає дату, на яку такі заявки мають бути оформлені, хоча це значення може бути і не задано.

Пошук серед проблемних замовлень демонтажних робіт забезпечено функцією `dismantling_orders_with_problem_search`, що має параметри `dismantlingtype`, що задає, чи це замовлення демонтажних робіт, чи це запит за оголошенням про демонтаж об'єкта, `dismantlingdate`, що задає дату виконання робіт. Параметри пошуку також задані ключами GET-запиту, включаючи `dismantleobjecttypeid`, що задає ідентифікатор типу об'єкту для демонтажу, `dismantlingwork`, що задає ідентифікатор демонтажної роботи.

Перегляд вартості робіт демонтажної бригади забезпечено функцією `dismantlingcrew_works_price`, а параметр `dismantlingcrewid` задає ідентифікатор демонтажної бригади.

Перегляд рейтингу демонтажних бригад забезпечено функцією `dismantlingcrew_rating`.

Пошук бригад, які виконують демонтажні роботи, забезпечено функцією `dismantlingcrew_search`. Параметри пошуку вказані ключами GET-запиту: `dismantlingcrewestimate` – мінімальна потрібна оцінка демонтажної бригади, `dismantlingworkbasecity` – місто, в якому працює демонтажна бригада, `owncity` – місто, в якому необхідно виконати демонтаж, `dismantlingequipment` – ідентифікатор техніки, яка має бути в наявності у демонтажній бригаді, `dismantlingequipmentcategory` – категорія техніки, яка має бути в наявності у демонтажній бригаді, `productionmin` – найстраіший рік техніки у демонтажній бригаді, `dismantlingwork` – ідентифікатор демонтажної роботи, яку має виконувати бригада.

Блокування доступу замовника демонтажних робіт до програми забезпечено функцією `dismantlecustomer_block`, а параметр `dismantlecustomerlogin` задає логін такого замовника демонтажних робіт. Якщо доступ вже заблокований, то реалізується скасування блокування доступу замовника демонтажних робіт до програми.

Блокування обслуговування демонтажної бригади програмою забезпечено функцією `dismantlingcrew_block`, а параметр `dismantlingcrewid` задає ідентифікатор демонтажної бригади. Якщо доступ вже заблокований, то реалізується скасування блокування обслуговування демонтажної бригади.

Перегляд оголошення про демонтаж об'єктів забезпечено функцією `dismantlead_info` з параметром `dismantleadid`, який задає ідентифікатор оголошення про демонтаж об'єктів.

Перегляд своїх оголошень про демонтаж об'єктів забезпечено функцією `own_all_dismantleads`.

Перегляд оголошень про демонтаж об'єктів замовника забезпечено функцією `customer_all_dismantleads` з параметром `dismantlecustomerlogin`, Що задає логін замовника демонтажних робіт.

Підтвердження заявки про демонтаж об'єкту забезпечено функцією `dismantleadorder_approve`, а параметр `dismantleadorderid` задає ідентифікатор заявки про демонтаж об'єкту.

Скасування заявки про демонтаж об'єкту забезпечено функцією `dismantleadorder_cancel`, а параметр `dismantleadorderid` задає ідентифікатор заявки про демонтаж об'єкту.

Підтвердження виконання заявки про демонтаж об'єкту забезпечено функцією `dismantleadorder_execute`, а параметр `dismantleadorderid` задає ідентифікатор заявки про демонтаж об'єкту.

Підтвердження невиконання заявки про демонтаж об'єкту забезпечено функцією `dismantleadorder_unexecute`, а параметр `dismantleadorderid` задає ідентифікатор заявки про демонтаж об'єкту.

Підтвердження проблем з заявками про демонтаж об'єкту забезпечено функцією `dismantleadorder_problem`, а параметр `dismantleadorderid` задає ідентифікатор заявки про демонтаж об'єкту.

Створення оголошення про демонтаж об'єктів забезпечено функцією `dismantlead_new` на основі класу форми `DismantleadForm`.

Подання заявки на демонтаж об'єкту забезпечено функцією `dismantleadorder_new`, що має параметр `dismantleadid` з ідентифікатором оголошення про демонтаж об'єкту, працюючи на основі класів форм `DismantleadorderForm` та `DismantleadorderworkForm`.

Редагування оголошення про демонтаж об'єктів забезпечено функцією `dismantlead_edit` з параметром `dismantleadid`, який задає ідентифікатор оголошення про демонтаж об'єктів, на основі класу форми `DismantleadForm`.

Перегляд даних замовника демонтажних робіт забезпечено функцією `dismantlecustomer_info`, а параметр `dismantlecustomerlogin` задає логін замовника демонтажних робіт.

Редагування даних замовника демонтажних робіт забезпечено функцією `dismantlecustomer_edit` на основі класу форми `DismantlecustomerForm`.

Оцінювання виконання демонтажних робіт забезпечено функцією `dismantlingcrewestimation_new`, яка має параметр `dismantlingcrew`, що задає ідентифікатор демонтажної бригади, на основі класу форми `DismantlingcrewestimationForm`.

Оцінювання замовника демонтажних робіт забезпечено функцією `dismantlecustomerestimation_new`, яка має параметр `dismantlecustomerlogin`, що задає логін замовника демонтажних робіт, на основі класу форми `DismantlecustomerestimationForm`.

3.3 Висновки за розділом 3

ФВР Django продиктував визначення структури розробленого

застосунку підтримки демонтажних робіт за допомогою шаблону MVT з реалізацією всіх моделей, представлень застосунку для забезпечення всіх функцій програми, що і було в підсумку успішно завершено.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми

Представлена програма розроблена для підтримки демонтажних робіт, повинна надавати можливості пошуку, ознайомлення з можливостями демонтажних бригад, роботами, які вони здатні виконати, замовленням демонтажних робіт у них, розміщенням власних оголошень про необхідність демонтажу об'єктів, пошуку серед таких оголошень представниками демонтажних бригад, створення запитів на демонтаж від них, керування такими запитами.

4.2 Опис роботи з програмою підтримки демонтажних робіт

Для роботи з програмою як користувачу не має необхідності мати спеціалізовані знання, достатньо розуміння основних можливостей.

Якщо користувач є неавторизованим, то він працює, керуючи доступними засобами через розділ меню «Бригади», звідки можна перейти до пошуку демонтажних бригад, перегляду рейтингу бригад, пункту «Демонтаж», звідки виконується перехід до пошуку серед оголошень про демонтаж об'єктів. Також можна авторизуватись.

Зі сторінки самої демонтажної бригади (рис. 4.1) користувачу тоді доступний опис, фотографія, посилання для додавання в закладки, хоча для того, щоб виконати цю дію, потрібно авторизуватись. Тобто дія вже буде виконана тоді, коли користувач змінить свою роль. Зі сторінки доступна основна техніка демонтажної бригади, при чому можна відкрити окрему сторінку для вибору категорії техніки та відображення за нею всієї техніки у бригади, також доступні оцінки, звідки можна перейти до окремої сторінки зі всіма оцінками, перелік доступних демонтажних робіт що виконує бригада. Відповідно також можна відкрити окрему сторінку зі всіма доступними

демонтажними роботами та вартістю цих робіт, обираючи категорію на сторінці.

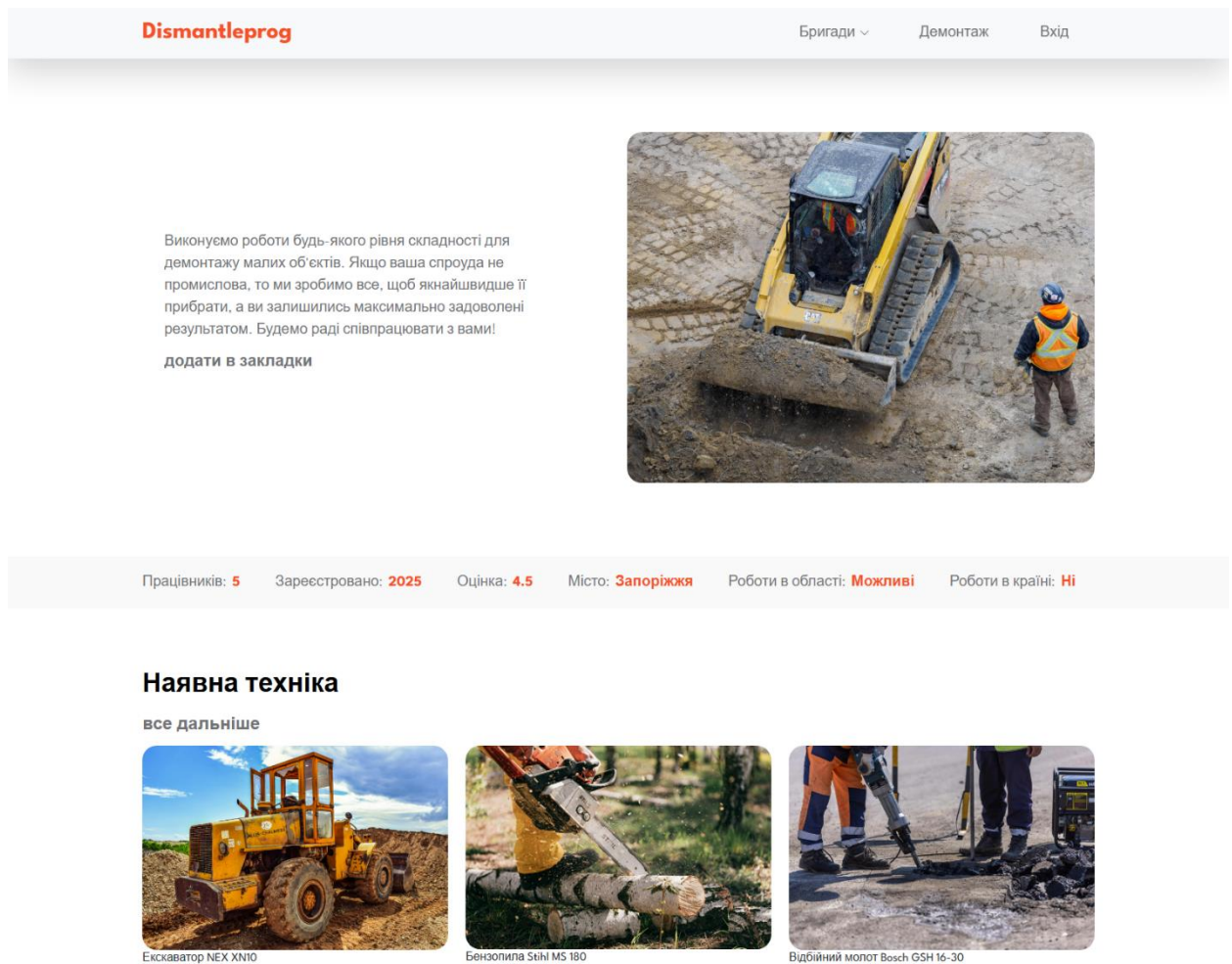


Рисунок 4.1 – Сторінка демонтажної бригади для неавторизованого користувача

Замовники демонтажних робіт мають розділи меню «Бригади» з закладеними бригадами, пошуком бригад, рейтингом, «Замовлення» з варіантами своїх замовлень демонтажних робіт за станами, «Заявки» для керування поданими від демонтажних бригад заявками за оголошеннями замовника, «Демонтаж» для пошуку серед оголошень про демонтаж об'єктів, відображення закладених оголошень, створення нового оголошення, «Замовник» для редагування і перегляду даних власної сторінки.

За результатами пошуку серед демонтажних бригад (рис. 4.2) відображається фотографія за кожною бригадою, кількість працівників у ній, середня оцінка. При наведенні на фотографію з'явиться напис «Більше» для відкриття сторінки деталізації. Одразу зі сторінки пошуку можна перейти до формування замовлення або утворити закладку.

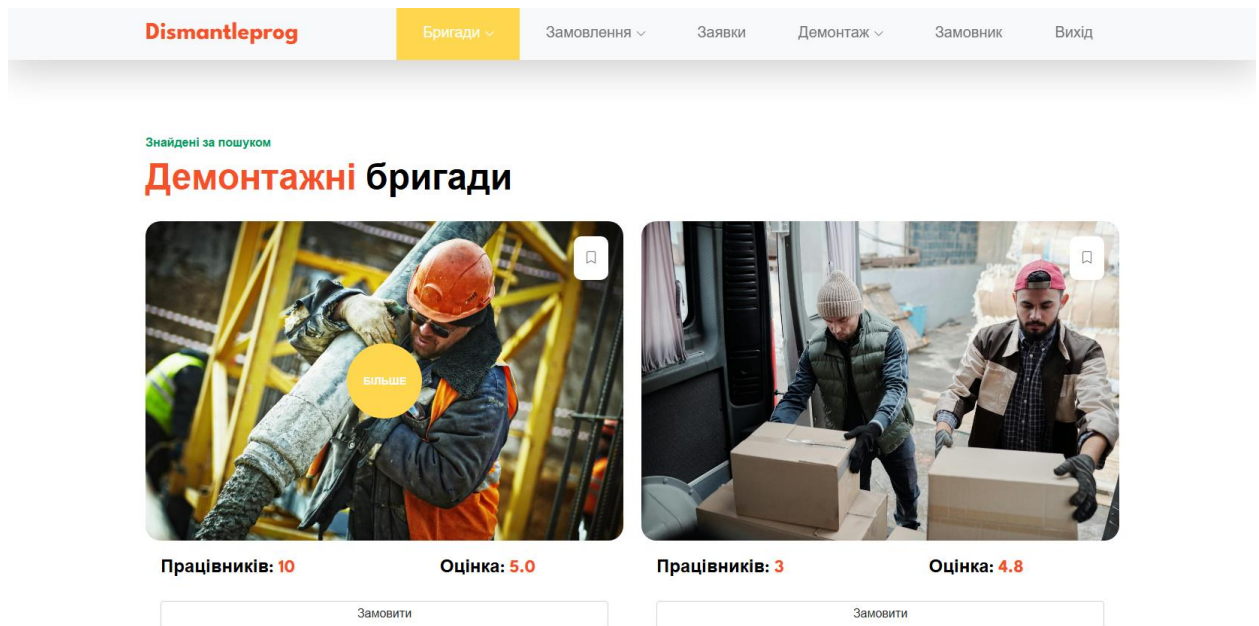


Рисунок 4.2 – Результати пошуку демонтажних бригад для замовника демонтажних робіт

Для представника демонтажної бригади доступні розділи «Бригади» з пошуком серед бригад, переглядом закладених, рейтингом, «Замовлення» для керування замовленнями з обранням стану з переліку розділу, «Заявки» для перегляду власних заявок за оголошеннями про демонтаж об'єктів замовників у відповідному стані, «Демонтаж» для виконання пошуку серед оголошень про необхідний демонтаж об'єктів, для виконання відбору за параметрами бригади таких оголошень, «Графік» для визначення графіку роботи бригади, «Бригада» для керування даними про саму бригаду, що відображаються на сторінці демонтажної бригади, керування внесеною технікою, вартістю і доступністю демонтажних робіт тощо.

Під час пошуку за оголошеннями про демонтаж об'єктів (рис. 4.3) представник бригади може утворювати закладки, переходити до детальної інформації, одразу формувати запит на виконання робіт.

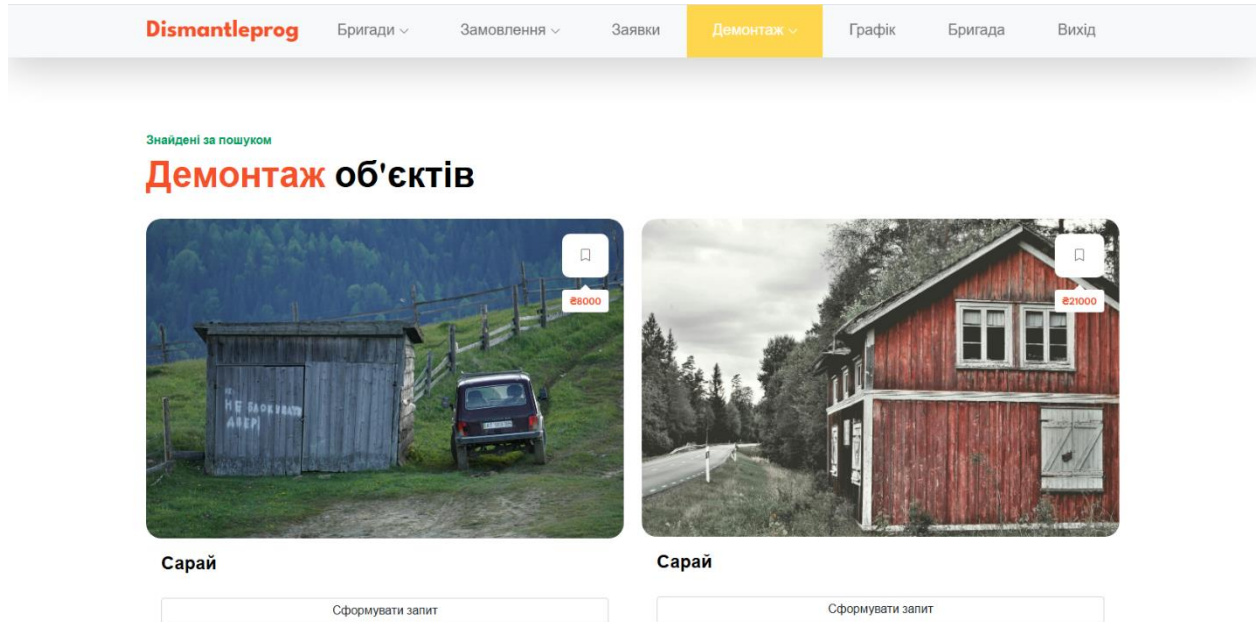


Рисунок 4.3 – Результати пошуку оголошень про демонтаж об'єктів для представника демонтажної бригади

Менеджер програми може працювати з розділом меню «Бригади» для виконання пошуку серед бригад, пошуку серед заблокованих бригад, перегляду рейтингу, перегляду закладених бригад. Аналогічно влаштовано розділ «Демонтаж», відповідно менеджер на відміну від попередніх ролей також звідси може виконувати пошук заблокованих замовників.

Менеджеру також доступний розділ меню «Замовлення» для роботи з проблемними замовленнями (рис. 4.4), при чому він може встановлювати дату відображення замовлень, а може переходити до розширеного пошуку серед таких проблемних замовлень, натискаючи на відповідне посилання. Аналогічно з розділу «Заявки» менеджер програми може працювати з проблемними заявками на виконання робіт за оголошеннями про демонтаж об'єктів.

Менеджер програми може переглядати дані замовлення, в тому числі склад робіт, може змінювати стан замовлення на необхідний. Тож він має широкий спектр оброблення замовлення.

Dismantleprog Бригади ▾ **Замовлення ▾** Заявки Демонтаж ▾ Вихід

Тип проблеми

Всі ▾

Дата

дд.мм.рррр ▾

Застосувати

Перейти до пошуку

Проблемні замовлення

В обумовлений час бригада не з'явилась на місці.

Ганна Агапова

Виконане ▾

ЗМІНИТИ СТАН

Номер замовлення: 14
Вартість: 811500
Виконання: 28.05.2025 13:00 - 28.05.2025 16:00
Тип об'єкта: Гараж
Місто: Запоріжжя
Адреса: Бул. Шевченка, 10
сторінка бригади
показати повністю

блокувати замовника
блокувати бригаду

Рисунок 4.4 – Перегляд проблемних замовлень демонтажних робіт для менеджера програми

4.3 Тестування програми підтримки демонтажних робіт

Програма підтримки демонтажних робіт для перевірки реалізації функціональних вимог, які мали бути забезпечені в роботі, потребувала виконання тестування за ними.

У підсумку таке тестування було реалізовано (табл. 4.1) за кожним пунктом за кожною роллю користувача з внесенням запису «недоступно», коли така функція такому користувачу доступна, «успіх», коли функція завершилась отриманням очікуваного результату, «проблема», коли функція завершилась отриманням результату, який не очікувався.

Слід зазначити, що тестування реалізовано з розташуванням застосунку на сервері та доступом до нього з персонального комп'ютеру та мобільного пристрою.

Таблиця 4.1 – Результати тестування програми підтримки демонтажних робіт

Функція	Неавто- ризова- ний користу- вач	Замов- ник де- мон- тажних робіт	Предс- тавник демон- тажної бригади	Менед- жер програ- ми
1	2	3	4	5
Пошук бригад, які виконують демонтажні роботи	Успіх	Успіх	Успіх	Успіх
Перегляд інформації про демонтажну бригаду	Успіх	Успіх	Успіх	Успіх
Утворення закладки на сторінці демонтажної бригади	Недос- тупно	Успіх	Успіх	Успіх
Вилучення закладки на сторінці демонтажної бригади	Недос- тупно	Успіх	Успіх	Успіх
Перегляд закладених сторінок демонтажних бригад	Недос- тупно	Успіх	Успіх	Успіх
Заповнення інформації про демонтажну бригаду	Недос- тупно	Недос- тупно	Успіх	Недос- тупно
Редагування інформації про демонтажну бригаду	Недос- тупно	Недос- тупно	Успіх	Успіх
Додавання техніки, використовуваної демонтажною бригадою	Недос- тупно	Недос- тупно	Успіх	Недос- тупно
Вилучення техніки, використовуваної демонтажною бригадою	Недос- тупно	Недос- тупно	Успіх	Успіх

Продовження таблиці 4.1

1	2	3	4	5
Додавання демонтажної роботи до послуг бригади	Недоступно	Недоступно	Успіх	Недоступно
Редагування параметрів демонтажної роботи у послугах бригади	Недоступно	Недоступно	Успіх	Успіх
Вилучення демонтажної роботи з послуг бригади	Недоступно	Недоступно	Успіх	Успіх
Замовлення демонтажних робіт	Недоступно	Успіх	Недоступно	Недоступно
Підтвердження замовлення демонтажних робіт	Недоступно	Недоступно	Успіх	Успіх
Скасування замовлення демонтажних робіт	Недоступно	Успіх	Успіх	Успіх
Підтвердження виконання замовлення демонтажних робіт	Недоступно	Успіх	Успіх	Успіх
Підтвердження невиконання замовлення демонтажних робіт	Недоступно	Успіх	Успіх	Успіх
Підтвердження проблем з замовленням демонтажних робіт	Недоступно	Успіх	Успіх	Успіх
Перегляд проблемних замовлень демонтажних робіт	Недоступно	Недоступно	Недоступно	Успіх
Пошук серед проблемних замовлень демонтажних робіт	Недоступно	Недоступно	Недоступно	Успіх
Перегляд своїх нових замовлень демонтажних робіт	Недоступно	Успіх	Недоступно	Недоступно
Перегляд своїх виконаних замовлень демонтажних робіт	Недоступно	Успіх	Недоступно	Недоступно

Продовження таблиці 4.1

1	2	3	4	5
Перегляд своїх запланованих замовлень демонтажних робіт	Недоступно	Успіх	Недоступно	Недоступно
Перегляд своїх проблемних замовлень демонтажних робіт	Недоступно	Успіх	Недоступно	Недоступно
Перегляд своїх скасованих замовлень демонтажних робіт	Недоступно	Успіх	Недоступно	Недоступно
Перегляд адресованих демонтажній бригаді нових замовлень	Недоступно	Недоступно	Успіх	Недоступно
Перегляд адресованих демонтажній бригаді виконаних замовлень	Недоступно	Недоступно	Успіх	Недоступно
Перегляд адресованих демонтажній бригаді запланованих замовлень	Недоступно	Недоступно	Успіх	Недоступно
Перегляд адресованих демонтажній бригаді проблемних замовлень	Недоступно	Недоступно	Успіх	Недоступно
Перегляд адресованих демонтажній бригаді скасованих замовлень	Недоступно	Недоступно	Успіх	Недоступно
Перегляд вартості робіт демонтажної бригади	Успіх	Успіх	Успіх	Успіх
Перегляд рейтингу демонтажних бригад	Успіх	Успіх	Успіх	Успіх
Блокування доступу замовника демонтажних робіт до програми	Недоступно	Недоступно	Недоступно	Успіх

Продовження таблиці 4.1

1	2	3	4	5
Скасування блокування доступу замовника демонтажних робіт до програми	Недоступно	Недоступно	Недоступно	Успіх
Блокування обслуговування демонтажної бригади програмою	Недоступно	Недоступно	Недоступно	Успіх
Скасування блокування обслуговування демонтажної бригади програмою	Недоступно	Недоступно	Недоступно	Успіх
Пошук серед заблокованих для обслуговування програмою демонтажних бригад	Недоступно	Недоступно	Недоступно	Успіх
Пошук серед заблокованих замовників демонтажних робіт	Недоступно	Недоступно	Недоступно	Успіх
Пошук за оголошеннями про демонтаж об'єктів	Успіх	Успіх	Успіх	Успіх
Перегляд оголошення про демонтаж об'єктів	Успіх	Успіх	Успіх	Успіх
Редагування оголошення про демонтаж об'єктів	Недоступно	Успіх	Недоступно	Успіх
Перегляд своїх оголошень про демонтаж об'єктів	Недоступно	Успіх	Недоступно	Недоступно
Перегляд оголошень про демонтаж об'єктів замовника	Успіх	Успіх	Успіх	Успіх
Утворення закладки на оголошенні про демонтаж об'єктів	Недоступно	Успіх	Успіх	Успіх
Вилучення закладки на оголошенні про демонтаж об'єктів	Недоступно	Успіх	Успіх	Успіх

Продовження таблиці 4.1

1	2	3	4	5
Перегляд закладених оголошень про демонтаж об'єктів	Недоступно	Успіх	Успіх	Успіх
Створення оголошення про демонтаж об'єктів	Недоступно	Успіх	Недоступно	Недоступно
Відбір оголошень про демонтаж об'єктів за можливостями бригади	Недоступно	Недоступно	Успіх	Недоступно
Подання заявки на демонтаж об'єкту	Недоступно	Недоступно	Успіх	Недоступно
Перегляд адресованих заявок про демонтаж об'єктів	Недоступно	Успіх	Недоступно	Недоступно
Перегляд своїх нових заявок про демонтаж об'єктів	Недоступно	Недоступно	Успіх	Недоступно
Перегляд своїх виконаних заявок про демонтаж об'єктів	Недоступно	Недоступно	Успіх	Недоступно
Перегляд своїх запланованих заявок про демонтаж об'єктів	Недоступно	Недоступно	Успіх	Недоступно
Перегляд своїх проблемних заявок про демонтаж об'єктів	Недоступно	Недоступно	Успіх	Недоступно
Перегляд своїх скасованих заявок про демонтаж об'єктів	Недоступно	Недоступно	Успіх	Недоступно
Підтвердження заявки про демонтаж об'єкту	Недоступно	Успіх	Недоступно	Успіх
Скасування заявки про демонтаж об'єкту	Недоступно	Успіх	Успіх	Успіх
Підтвердження виконання заявки про демонтаж об'єкту	Недоступно	Успіх	Успіх	Успіх

Продовження таблиці 4.1

1	2	3	4	5
Підтвердження невиконання заявки про демонтаж об'єкту	Недоступно	Успіх	Успіх	Успіх
Підтвердження проблем з заявками про демонтаж об'єкту	Недоступно	Успіх	Успіх	Успіх
Перегляд проблемних заявок про демонтаж об'єктів	Недоступно	Недоступно	Недоступно	Успіх
Пошук серед проблемних заявок про демонтаж об'єктів	Недоступно	Недоступно	Недоступно	Успіх
Оцінювання виконання демонтажних робіт	Недоступно	Успіх	Недоступно	Недоступно
Оцінювання замовника демонтажних робіт	Недоступно	Недоступно	Успіх	Недоступно
Перегляд оцінок демонтажних робіт від користувача	Успіх	Успіх	Успіх	Успіх
Перегляд оцінок демонтажної бригади	Успіх	Успіх	Успіх	Успіх
Перегляд оцінок замовника демонтажних робіт	Успіх	Успіх	Успіх	Успіх
Перегляд даних замовника демонтажних робіт	Успіх	Успіх	Успіх	Успіх
Редагування даних замовника демонтажних робіт	Недоступно	Успіх	Недоступно	Успіх
Авторизація	Недоступно	Успіх	Успіх	Успіх
Реєстрація	Недоступно	Успіх	Успіх	Недоступно

Кінець таблиці 4.1

1	2	3	4	5
Редагування параметрів графіку роботи демонтажної бригади	Недоступно	Недоступно	Успіх	Недоступно
Вилучення доступного часу роботи демонтажної бригади	Недоступно	Недоступно	Успіх	Недоступно
Відкриття доступного часу роботи демонтажної бригади	Недоступно	Недоступно	Успіх	Недоступно
Відображення графіку роботи демонтажної бригади на дату	Недоступно	Недоступно	Успіх	Недоступно
Перегляд усіх виконаних демонтажних робіт бригади	Недоступно	Недоступно	Успіх	Недоступно
Визначення параметрів графіку роботи демонтажної бригади	Недоступно	Недоступно	Успіх	Недоступно

4.4 Висновки за розділом 4

Усі функціональні можливості програми підтримки демонтажних робіт перевірені шляхом тестування з успішним завершенням роботи над кожною функцією, основні сценарії роботи описані.

ВИСНОВКИ

Основні отримані результати роботи включають аналіз існуючих програм підтримки демонтажних робіт, проєктування, реалізацію і тестування власного програмного забезпечення для підтримки демонтажних робіт. Існуючі програми підтримують взаємодію щодо демонтажних робіт, проте зазвичай представлені окремими вебсайтами окремих бригад. Цілісного ж майданчику для з'єднання пропозицій або не існує, або він не є спеціалізованим, тобто не враховує особливості саме демонтажних робіт. Розроблені функції програми направлені на реалізацію даного рішення у складі цілісного застосунку для фреймворку Django.

Для реалізації також використано мову Python для програмування і систему MySQL для створення і підтримки бази даних.

У роботі успішно розроблено програмне забезпечення підтримки демонтажних робіт для організації контактів між бригадами, що виконують демонтаж об'єктів, та замовниками таких послуг. Зі створеною програмою можуть працювати представники демонтажних бригад, замовників демонтажних робіт, неавторизовані користувачі, менеджери програми, що дозволяє охопити цілий ряд сценаріїв для кожної ролі користувачів, що і були початково визначені в роботі.

Створена програма надає можливості пошуку, ознайомлення з можливостями демонтажних бригад, роботами, які вони здатні виконати, замовленням демонтажних робіт у них, розміщенням власних оголошень про необхідність демонтажу об'єктів, пошуку серед таких оголошень представниками демонтажних бригад, створення запитів на демонтаж від них, керування такими запитами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Демонтажна компанія N1 у Запоріжжі [Електрон. ресурс]. – Режим доступу : <https://demontag.zp.ua/ua/home>.
2. Головна – Demontazh Kyiv Демонтаж будівель [Електрон. ресурс]. – Режим доступу : <https://demontazh-kyiv.com.ua/>.
3. Демонтажні роботи, вивіз будівельного сміття у Києві та Київській області. Демонтаж – низька ціна, професійно та швидко. Demontazh.net.ua [Електрон. ресурс]. – Режим доступу : <https://demontazh.net.ua/>.
4. Демонтаж – будівельні роботи та ремонт Київ – Ціна на OLX.ua [Електрон. ресурс]. – Режим доступу : <https://www.olx.ua/uk/uslugi/stroitelstvo-otdelka-remont/kiev/q-%D0%B4%D0%B5%D0%BC%D0%BE%D0%BD%D1%82%D0%B0%D0%B6/?currency=UAH>.
5. Django Introduction [Electronic resource]. – Access mode : https://www.w3schools.com/django/django_intro.php.
6. Quick Tutorial for Pyramid – The Pyramid Web Framework v2.0.2 [Electronic resource]. – Access mode : https://docs.pylonsproject.org/projects/pyramid/en/latest/quick_tutorial/index.html.
7. Moso Interior HTML Template by Tooplate [Electronic resource]. – Access mode : <https://www.tooplate.com/view/2133-moso-interior>.

ДОДАТОК А
Технічне завдання

Вступ

Нерідко увага людей стосується будівництва нових об'єктів для житла, роботи, різних активностей, проте демонтаж існуючих об'єктів є важливою складовою даного процесу, якому приділяється недостатньо уваги. Потреби у демонтажі виникають у індивідуальних власників на їх ділянках перед тим, як наприклад, переорганізувати простір. Супутніми роботами до демонтажних є перевезення певних матеріалів, обладнання на інше місце. Усе це потребує вирішення, а в той же час існує багато компаній, окремих бригад, працівників, які такі послуги можуть надавати. Найчастіше всі вони мають свої окремі вебсайти, що ускладнює порівняння і пошук того варіанту, який задовольнить користувача як замовника демонтажних послуг. Це може бути вирішено через створення програмного рішення, яке б об'єднало всі ці можливості разом.

A.1 Підстави для розробки

Наказ № 209 від 28 квітня 2025 року був виданий у Національному університеті «Запорізька політехніка», визначивши для виконання цієї дипломної кваліфікаційної роботи бакалавра тему «Проектування та розробка програмного забезпечення підтримки демонтажних робіт», яку було далі уточнено у вигляді завдання на дипломну кваліфікаційну роботу бакалавра.

A.2 Призначення розробки

Програма, яка розробляється для підтримки демонтажних робіт, повинна надавати можливості пошуку, ознайомлення з можливостями демонтажних бригад, роботами, які вони здатні виконати, замовленням демонтажних робіт у них, розміщенням власних оголошень про необхідність демонтажу об'єктів, пошуку серед таких оголошень представниками

демонтажних бригад, створення запитів на демонтаж від них, керування такими запитами.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Програма підтримки демонтажних робіт повинна надавати функції:

- пошук бригад, які виконують демонтажні роботи;
- перегляд інформації про демонтажну бригаду;
- утворення закладки на сторінці демонтажної бригади;
- вилучення закладки на сторінці демонтажної бригади;
- перегляд закладених сторінок демонтажних бригад;
- заповнення інформації про демонтажну бригаду;
- редагування інформації про демонтажну бригаду;
- додавання техніки, використовуваної демонтажною бригадою;
- вилучення техніки, використовуваної демонтажною бригадою;
- додавання демонтажної роботи до послуг бригади;
- редагування параметрів демонтажної роботи у послугах бригади;
- вилучення демонтажної роботи з послуг бригади;
- замовлення демонтажних робіт;
- підтвердження замовлення демонтажних робіт;
- скасування замовлення демонтажних робіт;
- підтвердження виконання замовлення демонтажних робіт;
- підтвердження невиконання замовлення демонтажних робіт;
- підтвердження проблем з замовленням демонтажних робіт;
- перегляд проблемних замовлень демонтажних робіт;
- пошук серед проблемних замовлень демонтажних робіт;
- перегляд своїх замовлень демонтажних робіт, у тому числі нових, виконаних, запланованих, проблемних, скасованих;
- перегляд адресованих демонтажній бригаді замовлень, у тому числі

нових, виконаних, запланованих, проблемних, скасованих;

- перегляд вартості робіт демонтажної бригади;
- перегляд рейтингу демонтажних бригад;
- блокування доступу замовника демонтажних робіт до програми;
- скасування блокування доступу замовника демонтажних робіт до

програми;

- блокування обслуговування демонтажної бригади програмою;
- скасування блокування обслуговування демонтажної бригади

програмою;

- пошук серед заблокованих для обслуговування програмою

демонтажних бригад;

- пошук серед заблокованих замовників демонтажних робіт;
- пошук за оголошеннями про демонтаж об'єктів;
- перегляд оголошення про демонтаж об'єктів;
- редагування оголошення про демонтаж об'єктів;
- перегляд своїх оголошень про демонтаж об'єктів;
- перегляд оголошень про демонтаж об'єктів замовника;
- утворення закладки на оголошенні про демонтаж об'єктів;
- вилучення закладки на оголошенні про демонтаж об'єктів;
- перегляд закладених оголошень про демонтаж об'єктів;
- створення оголошення про демонтаж об'єктів;
- відбір оголошень про демонтаж об'єктів за можливостями бригади;
- подання заявки на демонтаж об'єкту;
- перегляд адресованих заявок про демонтаж об'єктів;
- перегляд своїх заявок про демонтаж об'єктів, у тому числі нових,

виконаних, запланованих, проблемних, скасованих;

- підтвердження заявки про демонтаж об'єкту;
- скасування заявки про демонтаж об'єкту;
- підтвердження виконання заявки про демонтаж об'єкту;
- підтвердження невиконання заявки про демонтаж об'єкту;

- підтвердження проблем з заявками про демонтаж об'єкту;
- перегляд проблемних заявок про демонтаж об'єктів;
- пошук серед проблемних заявок про демонтаж об'єктів;
- оцінювання виконання демонтажних робіт;
- оцінювання замовника демонтажних робіт;
- перегляд оцінок демонтажних робіт від користувача;
- перегляд оцінок демонтажної бригади;
- перегляд оцінок замовника демонтажних робіт;
- перегляд даних замовника демонтажних робіт;
- редагування даних замовника демонтажних робіт;
- авторизація;
- реєстрація;
- визначення параметрів графіку роботи демонтажної бригади;
- редагування параметрів графіку роботи демонтажної бригади;
- вилучення доступного часу роботи демонтажної бригади;
- відкриття доступного часу роботи демонтажної бригади;
- відображення графіку роботи демонтажної бригади на дату;
- перегляд усіх виконаних демонтажних робіт бригади.

A.3.2 Умови експлуатації

Умовою експлуатації програми підтримки демонтажних робіт є наявність підготовлених для цього користувачів. Підготовка користувачів, що представляють бригади з виконання демонтажних робіт, та замовників демонтажних робіт реалізується на основі наявності базових вмінь користувача комп'ютера та розуміння наявного набору функцій у програмі. Те саме стосується менеджера програми. Адміністратор програми підтримки демонтажних робіт повинен забезпечити запуск та підтримку вебсервера, сервера бази даних, внесення основних даних через адміністративну панель. Відповідно це спеціалізовані вміння. Вміння працювати з такими

можливостями для адміністратора має бути забезпечено.

A.3.3 Вимоги до складу та параметрів технічних засобів

Технічно сервер повинен забезпечуватися обладнанням, що включає мінімум процесор з 2 ядрами, 4 Гб оперативної пам'яті та 10 Гб виділеного простору на жорсткому диску.

ДОДАТОК Б
Текст програми

```

from django.shortcuts import render
from .models import Dismantler
from .models import Dismantlecustomer
from .models import Dismantlecustomerestimation
from .models import Dismantleobjecttype
from .models import Dismantlead
from .models import Dismantleadwork
from .models import Dismantlingequipmentcategory
from .models import Dismantlingequipmentbrand
from .models import Dismantlingequipment
from .models import Dismantlingworkbasecity
from .models import Dismantlingcrew
from .models import Dismantlingcrewequipment
from .models import Dismantlingcrewstdtime
from .models import Dismantlingcrewtime
from .models import Dismantlingcrewestimation
from .models import Dismantlingcategory
from .models import Dismantlingwork
from .models import Dismantlingcrewwork
from django.db.models import Q
from .models import Dismantleadorder
from .models import Dismantleadorderwork
from .models import Dismantlingcreworder
from .models import Dismantlingcreworderwork
from django.core.paginator import Paginator

def own_new_dismantlingcrew_orders(request):
    if request.user.is_authenticated:
        try:
            dismantler_customer_object =
Dismantlecustomer.objects.get(dismantlingdjangouser = request.user)
        except Dismantlecustomer.DoesNotExist:
            dismantler_customer_object = None
    else:
        dismantler_customer_object = None
    if not dismantler_customer_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як замовник демонтажних робіт
для використання цієї функції."})
    dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",

```

```

"dismantlingworkbasecity").filter(dismantlingstatus = "new",
dismantlecustomer = dismantler_customer_object)
    dismantler_customer_orders =
dismantler_customer_orders.order_by("-dismantlingend")
    dismantler_customer_orders =
Paginator(dismantler_customer_orders, 10)
    dismantler_customer_orders_page = request.GET.get('page')
    dismantler_customer_orders =
dismantler_customer_orders.get_page(dismantler_customer_orders_page)
    for dismantler_customer_order in dismantler_customer_orders:
        dismantler_customer_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_customer_order)
        return render(request, "dismantleprog/newowndisorders.html",
{"orders": dismantler_customer_orders })

def own_executed_dismantlingcrew_orders(request, execute_level =
0):
    if request.user.is_authenticated:
        try:
            dismantler_customer_object =
Dismantlecustomer.objects.get(dismantlingdjangouser = request.user)
        except Dismantlecustomer.DoesNotExist:
            dismantler_customer_object = None
        else:
            dismantler_customer_object = None
        if not dismantler_customer_object:
            return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як замовник демонтажних робіт
для використання цієї функції."})
        if execute_level == 0:
            dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(Q(dismantlingstatus = "execute1") |
Q(dismantlingstatus = "execute2"))
            dismantler_customer_orders =
dismantler_customer_orders.filter(dismantlecustomer =
dismantler_customer_object)
        elif execute_level == 1:

```

```

        dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "execute1",
dismantlecustomer = dismantler_customer_object)
        elif execute_level == 2:
            dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "execute2",
dismantlecustomer = dismantler_customer_object)
            dismantler_customer_orders =
dismantler_customer_orders.order_by("-dismantlingend")
            dismantler_customer_orders =
Paginator(dismantler_customer_orders, 10)
            dismantler_customer_orders_page = request.GET.get('page')
            dismantler_customer_orders =
dismantler_customer_orders.get_page(dismantler_customer_orders_page)
            for dismantler_customer_order in dismantler_customer_orders:
                dismantler_customer_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_customer_order)
            return render(request, "dismantleprog/exowndisorders.html",
{"orders": dismantler_customer_orders })

```

```

def own_planned_dismantlingcrew_orders(request):
    if request.user.is_authenticated:
        try:
            dismantler_customer_object =
Dismantlecustomer.objects.get(dismantlingdjangouser = request.user)
            except Dismantlecustomer.DoesNotExist:
                dismantler_customer_object = None
        else:
            dismantler_customer_object = None
        if not dismantler_customer_object:
            return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як замовник демонтажних робіт
для використання цієї функції."})
        dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",

```

```

"dismantlingworkbasecity").filter(dismantlingstatus = "approved",
dismantlecustomer = dismantler_customer_object)
    dismantler_customer_orders =
dismantler_customer_orders.order_by("-dismantlingend")
    dismantler_customer_orders =
Paginator(dismantler_customer_orders, 10)
    dismantler_customer_orders_page = request.GET.get('page')
    dismantler_customer_orders =
dismantler_customer_orders.get_page(dismantler_customer_orders_page)
    for dismantler_customer_order in dismantler_customer_orders:
        dismantler_customer_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_customer_order)
    return render(request, "dismantleprog/appowndisorders.html",
{"orders": dismantler_customer_orders })

```

```

def own_problem_dismantlingcrew_orders(request, problem_level =
0):
    if request.user.is_authenticated:
        try:
            dismantler_customer_object =
Dismantlecustomer.objects.get(dismantlingdjangouser = request.user)
        except Dismantlecustomer.DoesNotExist:
            dismantler_customer_object = None
    else:
        dismantler_customer_object = None
    if not dismantler_customer_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як замовник демонтажних робіт
для використання цієї функції."})
    if problem_level == 0:
        dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(Q(dismantlingstatus = "problem") |
Q(dismantlingstatus = "undone1") | Q(dismantlingstatus = "undone2"))
        dismantler_customer_orders =
dismantler_customer_orders.filter(dismantlecustomer =
dismantler_customer_object)
    elif execute_level == 1:

```

```

        dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "problem")
        dismantler_customer_orders =
dismantler_customer_orders.filter(dismantlecustomer =
dismantler_customer_object)
        elif execute_level == 2:
            dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "undone1")
            dismantler_customer_orders =
dismantler_customer_orders.filter(dismantlecustomer =
dismantler_customer_object)
            elif execute_level == 3:
                dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "undone2")
                dismantler_customer_orders =
dismantler_customer_orders.filter(dismantlecustomer =
dismantler_customer_object)
                dismantler_customer_orders =
dismantler_customer_orders.order_by("-dismantlingend")
                dismantler_customer_orders =
Paginator(dismantler_customer_orders, 10)
                dismantler_customer_orders_page = request.GET.get('page')
                dismantler_customer_orders =
dismantler_customer_orders.get_page(dismantler_customer_orders_page)
                for dismantler_customer_order in dismantler_customer_orders:
                    dismantler_customer_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_customer_order)
                return render(request,
"dismantleprog/problemowndisorders.html", {"orders":
dismantler_customer_orders })

```

```

def own_cancel_dismantlingcrew_orders(request):
    if request.user.is_authenticated:
        try:

```

```

        dismantler_customer_object =
Dismantlecustomer.objects.get(dismantlingdjangouser = request.user)
        except Dismantlecustomer.DoesNotExist:
            dismantler_customer_object = None
        else:
            dismantler_customer_object = None
        if not dismantler_customer_object:
            return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як замовник демонтажних робіт
для використання цієї функції."})
            dismantler_customer_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "cancelled")
            dismantler_customer_orders =
dismantler_customer_orders.filter(dismantlecustomer =
dismantler_customer_object)
            dismantler_customer_orders =
dismantler_customer_orders.order_by("-dismantlingend")
            dismantler_customer_orders =
Paginator(dismantler_customer_orders, 10)
            dismantler_customer_orders_page = request.GET.get('page')
            dismantler_customer_orders =
dismantler_customer_orders.get_page(dismantler_customer_orders_page)
            for dismantler_customer_order in dismantler_customer_orders:
                dismantler_customer_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smanlingcreworder = dismantler_customer_order)
            return render(request,
"dismantleprog/cancelowndisorders.html", {"orders":
dismantler_customer_orders })

def group_new_dismantlingcrew_orders(request):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
        except Dismantler.DoesNotExist:
            dismantler_object = None

```

```

else:
    dismantler_object = None
    if not dismantler_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})
    dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "new")
    dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
    dismantler_orders = Paginator(dismantler_orders, 10)
    dismantler_orders_page = request.GET.get('page')
    dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
    for dismantler_order in dismantler_orders:
        dismantler_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_order)
        return render(request,
"dismantleprog/crewnewowndisorders.html", {"orders": dismantler_orders
})

def group_own_executed_dismantlingcrew_orders(request,
execute_level = 0):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
            except Dismantler.DoesNotExist:
                dismantler_object = None
        else:
            dismantler_object = None
            if not dismantler_object:
                return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})

```

```

        if execute_level == 0:
            dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "execute")
            elif execute_level == 1:
                dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "execute1")
            elif execute_level == 2:
                dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "execute2")
                dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
                dismantler_orders = Paginator(dismantler_orders, 10)
                dismantler_orders_page = request.GET.get('page')
                dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
                for dismantler_order in dismantler_orders:
                    dismantler_order_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smanlingcreworder = dismantler_order)
                    return render(request,
"dismantleprog/crewexowndisorders.html", {"orders": dismantler_orders
})

```

```

def group_own_planned_dismantlingcrew_orders(request):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
        except Dismantler.DoesNotExist:
            dismantler_object = None

```

```

else:
    dismantler_object = None
    if not dismantler_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})
    dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "approved")
    dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
    dismantler_orders = Paginator(dismantler_orders, 10)
    dismantler_orders_page = request.GET.get('page')
    dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
    for dismantler_order in dismantler_orders:
        dismantler_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_order)
        return render(request,
"dismantleprog/crewappowndisorders.html", {"orders": dismantler_orders
})

def group_own_problem_dismantlingcrew_orders(request,
problem_level = 0):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
        except Dismantler.DoesNotExist:
            dismantler_object = None
    else:
        dismantler_object = None
    if not dismantler_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})

```

```

        if problem_level == 0:
            dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(Q(dismantlingstatus = "problem") |
Q(dismantlingstatus = "undone1") | Q(dismantlingstatus = "undone2"))
            dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
            elif execute_level == 1:
                dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "problem")
                dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
                elif execute_level == 2:
                    dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "undone1")
                    dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
                    elif execute_level == 3:
                        dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "undone2")
                        dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
                        dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
                        dismantler_orders = Paginator(dismantler_orders, 10)
                        dismantler_orders_page = request.GET.get('page')
                        dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
                        for dismantler_order in dismantler_orders:
                            dismantler_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_order)

```

```

        return render(request,
            "dismantleprog/crewproblemowndisorders.html", {"orders":
dismantler_customer_orders })

    def group_own_cancel_dismantlingcrew_orders(request):
        if request.user.is_authenticated:
            try:
                dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
                dismantling_crew = dismantler_object.dismantlingcrew
            except Dismantler.DoesNotExist:
                dismantler_object = None
        else:
            dismantler_object = None
        if not dismantler_object:
            return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})
            dismantler_orders =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingstatus = "cancelled",
dismantlingcrew = dismantling_crew)
            dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
            dismantler_orders = Paginator(dismantler_orders, 10)
            dismantler_orders_page = request.GET.get('page')
            dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
            for dismantler_order in dismantler_orders:
                dismantler_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_order)
            return render(request,
            "dismantleprog/crewcancelowndisorders.html", {"orders":
dismantler_orders })

    def own_new_dismantlead_orders(request):
        if request.user.is_authenticated:
            try:

```

```

        dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
        dismantling_crew = dismantler_object.dismantlingcrew
except Dismantler.DoesNotExist:
        dismantler_object = None
    else:
        dismantler_object = None
    if not dismantler_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})
        dismantler_orders =
Dismantleorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"new", dismantlingcrew = dismantling_crew)
        dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
        dismantler_orders = Paginator(dismantler_orders, 10)
        dismantler_orders_page = request.GET.get('page')
        dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
        for dismantler_order in dismantler_orders:
            dismantler_order.works =
Dismantleorderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantleorder = dismantler_order)
            return render(request, "dismantleprog/newownadorders.html",
{"orders": dismantler_orders })

def own_executed_dismantlead_orders(request, execute_level = 0):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
except Dismantler.DoesNotExist:
            dismantler_object = None
        else:
            dismantler_object = None

```

```

        if not dismantler_object:
            return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної бригади для використання цієї функції."})
        if execute_level == 0:
            dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"execute", dismantlingcrew = dismantling_crew)
        elif execute_level == 1:
            dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"execute1", dismantlingcrew = dismantling_crew)
        elif execute_level == 2:
            dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"execute2", dismantlingcrew = dismantling_crew)
            dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
            for dismantler_order in dismantler_orders:
                dismantler_order.works =
Dismantleadorderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantleadorder = dismantler_order)
            return render(request, "dismantleprog/exownadorders.html",
{"orders": dismantler_orders })

def own_planned_dismantlead_orders(request):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
        except Dismantler.DoesNotExist:
            dismantler_object = None
    else:

```

```

        dismantler_object = None
    if not dismantler_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної бригади для використання цієї функції."})
    dismantler_orders =
Dismantleorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"approved", dismantlingcrew = dismantling_crew)
    dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
    dismantler_orders = Paginator(dismantler_orders, 10)
    dismantler_orders_page = request.GET.get('page')
    dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
    for dismantler_order in dismantler_orders:
        dismantler_order.works =
Dismantleorderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantleorder = dismantler_order)
        return render(request, "dismantleprog/appownadorders.html",
{"orders": dismantler_orders })

def own_problem_dismantlead_orders(request, problem_level = 0):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
        except Dismantler.DoesNotExist:
            dismantler_object = None
    else:
        dismantler_object = None
    if not dismantler_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної бригади для використання цієї функції."})
    if problem_level == 0:
        dismantler_orders =
Dismantleorder.objects.select_related("dismantlead",

```

```

"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(Q(dismantlingstatus =
"problem") | Q(dismantlingstatus = "undone1") | Q(dismantlingstatus =
"undone2"))

    dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
    elif execute_level == 1:
        dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"problem")

        dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
        elif execute_level == 2:
            dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"undone1")

            dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
            elif execute_level == 3:
                dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"undone2")

                dismantler_orders =
dismantler_orders.filter(dismantlingcrew = dismantling_crew)
                dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
                dismantler_orders = Paginator(dismantler_orders, 10)
                dismantler_orders_page = request.GET.get('page')
                dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
                for dismantler_order in dismantler_orders:
                    dismantler_order.works =
Dismantleadorderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantleadorder = dismantler_order)

```

```

        return render(request,
            "dismantleprog/problemownadorders.html", {"orders": dismantler_orders
        })

    def own_cancel_dismantlead_orders(request):
        if request.user.is_authenticated:
            try:
                dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
                dismantling_crew = dismantler_object.dismantlingcrew
            except Dismantler.DoesNotExist:
                dismantler_object = None
        else:
            dismantler_object = None
        if not dismantler_object:
            return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})
            dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"cancelled", dismantlingcrew = dismantling_crew)
            dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
            dismantler_orders = Paginator(dismantler_orders, 10)
            dismantler_orders_page = request.GET.get('page')
            dismantler_orders =
dismantler_orders.get_page(dismantler_orders_page)
            for dismantler_order in dismantler_orders:
                dismantler_order.works =
Dismantleadorderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantleadorder = dismantler_order)
            return render(request,
            "dismantleprog/cancelownadorders.html", {"orders": dismantler_orders
        })

    def own_dismantlead_orders(request):
        if request.user.is_authenticated:
            try:

```

```

        dismantler_customer_object =
Dismantlecustomer.objects.get(dismantlingdjangouser = request.user)
        except Dismantlecustomer.DoesNotExist:
            dismantler_customer_object = None
        else:
            dismantler_customer_object = None
        if not dismantler_customer_object:
            return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як замовник демонтажних робіт
для використання цієї функції."})
            dismantler_customer_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlead__dismantlec
ustomer dismantler_customer_object)
            dismantler_customer_orders =
dismantler_customer_orders.order_by("-dismantlingend")
            dismantler_customer_orders =
Paginator(dismantler_customer_orders, 10)
            dismantler_customer_orders_page = request.GET.get('page')
            dismantler_customer_orders =
dismantler_customer_orders.get_page(dismantler_customer_orders_page)
            for dismantler_customer_order in dismantler_customer_orders:
                dismantler_customer_order =
Dismantleadorderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantleadorder = dismantler_customer_order)
                return render(request, "dismantleprog/adorders.html",
{"orders": dismantler_customer_orders })

def dismantlingcrew_estimations(request, dismantlingcrewid):
    try:
        dismantling_crew = Dismantlingcrew.objects.get(id =
dismantlingcrewid)
    except Dismantlingcrew.DoesNotExist:
        dismantling_crew = None
    if not dismantling_crew:
        return render(request, "dismantleprog/error.html",
{"caption": "Така демонтажна бригада не визначена"})
    crew_estimations =
Dismantlingcrewestimation.objects.filter(dismantlingcrew =
dismantling_crew)

```

```

        crew_estimations = crew_estimations.order_by("-
dismantlingtime")
        crew_estimations = Paginator(crew_estimations, 10)
        crew_estimations_page = request.GET.get('page')
        crew_estimations =
crew_estimations.get_page(crew_estimations_page)
        return render(request, "dismantleprog/crewestims.html",
{"estimations": crew_estimations })

def dismantlecustomer_estimations(request, dismantlecustomerid):
    try:
        dismantler_customer_object =
Dismantlecustomer.objects.get(id = dismantlecustomerid)
    except Dismantlecustomer.DoesNotExist:
        dismantler_customer_object = None
    if not dismantler_customer_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Такий замовник демонтажних робіт не визначений"})
    dismantle_estimations =
Dismantlecustomerestimation.objects.filter(dismantlecustomer =
dismantler_customer_object)
    dismantle_estimations = dismantle_estimations.order_by("-
dismantlingtime")
    dismantle_estimations = Paginator(dismantle_estimations, 10)
    dismantle_estimations_page = request.GET.get('page')
    dismantle_estimations =
dismantle_estimations.get_page(dismantle_estimations_page)
    return render(request, "dismantleprog/custestims.html",
{"estimations": dismantle_estimations })

def user_dismantle_estimations(request, userlogin, estimtype =
0):
    if estimtype == 0:
        dismantle_estimations =
Dismantlecustomerestimation.objects.select_related("dismantler",
"dismantler__dismantlingdjangouser", "dismantler__dismantlingcrew",
"dismantler__dismantlingcrew__dismantlingworkbasecity").filter(dismant
ler__dismantlingdjangouser__username = userlogin)
        dismantle_estimations =
dismantle_estimations.order_by("-dismantlingtime")
        dismantle_estimations = Paginator(dismantle_estimations,
10)
        dismantle_estimations_page = request.GET.get('page')

```

```

        dismantle_estimations =
dismantle_estimations.get_page(dismantle_estimations_page)
        return render(request, "dismantleprog/estims.html",
{"estimations": dismantle_estimations })
    else:
        dismantle_estimations =
Dismantlingcrewestimation.objects.select_related("dismantlingcrew",
"dismantlingcrew__dismantlingworkbasecity").filter(dismantlecustomer__
dismantlingdjangouser__username = userlogin)
        dismantle_estimations =
dismantle_estimations.order_by("-dismantlingtime")
        dismantle_estimations = Paginator(dismantle_estimations,
10)

        dismantle_estimations_page = request.GET.get('page')
        dismantle_estimations =
dismantle_estimations.get_page(dismantle_estimations_page)
        return render(request, "dismantleprog/estims.html",
{"estimations": dismantle_estimations })

def executed_dismantlead_orders(request, execute_level = 0):
    if request.user.is_authenticated:
        try:
            dismantler_object =
Dismantler.objects.select_related("dismantlingcrew").get(dismantlingdj
angouser = request.user)
            dismantling_crew = dismantler_object.dismantlingcrew
        except Dismantler.DoesNotExist:
            dismantler_object = None
    else:
        dismantler_object = None
    if not dismantler_object:
        return render(request, "dismantleprog/error.html",
{"caption": "Ви маєте бути авторизовані як представник демонтажної
бригади для використання цієї функції."})
    if execute_level == 0:
        dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"execute", dismantlingcrew = dismantling_crew)
    elif execute_level == 1:
        dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",

```

```

"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"execute1", dismantlingcrew = dismantling_crew)
    elif execute_level == 2:
        dismantler_orders =
Dismantleadorder.objects.select_related("dismantlead",
"dismantlead__dismantleobjecttype", "dismantlead__dismantlecustomer",
"dismantlead__dismantlingworkbasecity").filter(dismantlingstatus =
"execute2", dismantlingcrew = dismantling_crew)
        dismantler_orders = dismantler_orders.order_by("-
dismantlingend")
        if execute_level == 0:
            dismantler_orders2 =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "execute")
            elif execute_level == 1:
                dismantler_orders2 =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "execute1")
                elif execute_level == 2:
                    dismantler_orders2 =
Dismantlingcreworder.objects.select_related("dismantlingcrew",
"dismantleobjecttype",
"dismantlingworkbasecity").filter(dismantlingcrew = dismantling_crew,
dismantlingstatus = "execute2")
                    dismantler_orders2 = dismantler_orders2.order_by("-
dismantlingend")
                    dismantler_orders_all = []
                    i = 0
                    imax = dismantler_orders.count()
                    j = 0
                    jmax = dismantler_orders2.count()
                    while (i < imax) and (j < jmax):
                        if dismantler_orders[i].dismantlingend <
dismantler_orders2[j].dismantlingend:
                            dismantler_orders[i].type = "1"
                            dismantler_orders_all.append(dismantler_orders[i])
                            i = i + 1
                        else:

```

```

        dismantler_orders2[j].type = "2"
        dismantler_orders_all.append(dismantler_orders2[j])
        j = j + 1
    if i < imax:
        while i < imax:
            dismantler_orders[i].type = "1"
            dismantler_orders_all.append(dismantler_orders[i])
            i = i + 1
    if j < jmax:
        while j < jmax:
            dismantler_orders2[j].type = "2"
            dismantler_orders_all.append(dismantler_orders2[j])
            j = j + 1
    dismantler_orders_all = Paginator(dismantler_orders_all, 10)
    dismantler_orders_all_page = request.GET.get('page')
    dismantler_orders_all =
dismantler_orders_all.get_page(dismantler_orders_all_page)
    for dismantler_order in dismantler_orders_all:
        if dismantler_order.type == "1":
            dismantler_order.works =
Dismantleorderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantleadorder = dismantler_order)
        else:
            dismantler_order.works =
Dismantlingcreworderwork.objects.select_related("dismantlingcrewwork",
"dismantlingcrewwork__dismantlingwork",
"dismantlingcrewwork__dismantlingwork__dismantlingcategory").filter(di
smantlingcreworder = dismantler_order)
    return render(request, "dismantleprog/exallorders.html",
{"orders": dismantler_orders_all })

```

ДОДАТОК В
Слайди презентації

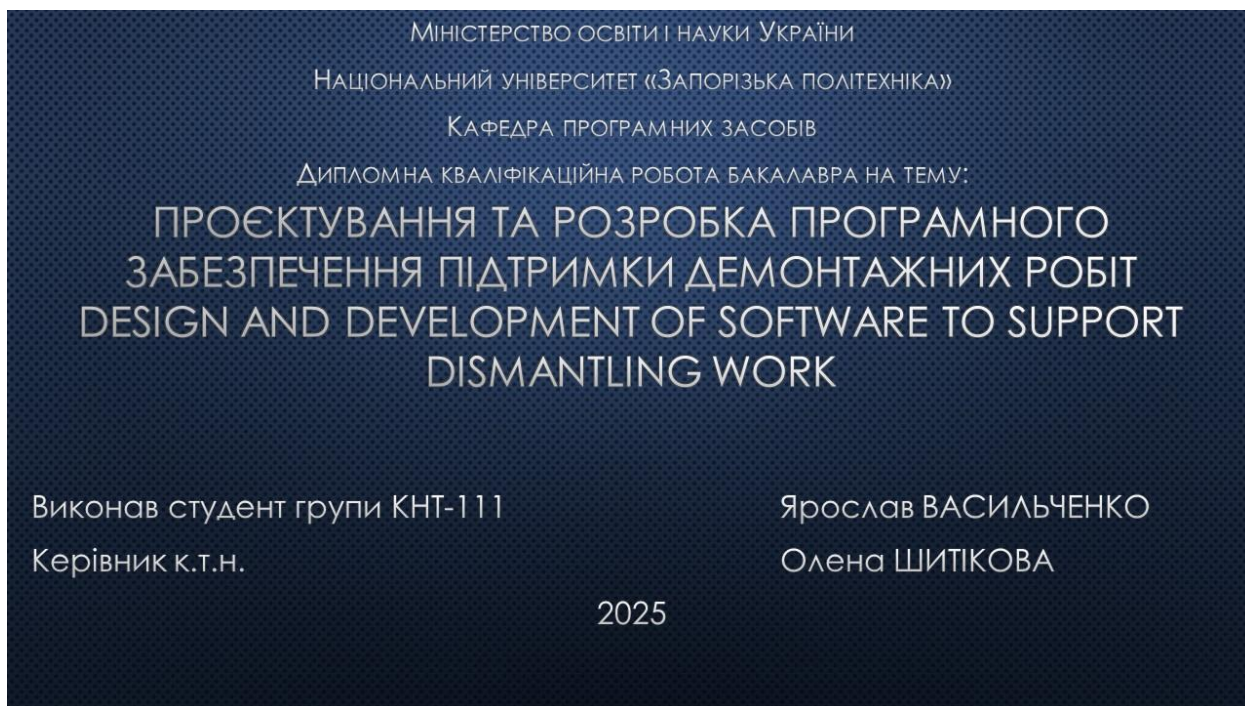


Рисунок В.1 – Слайд перший

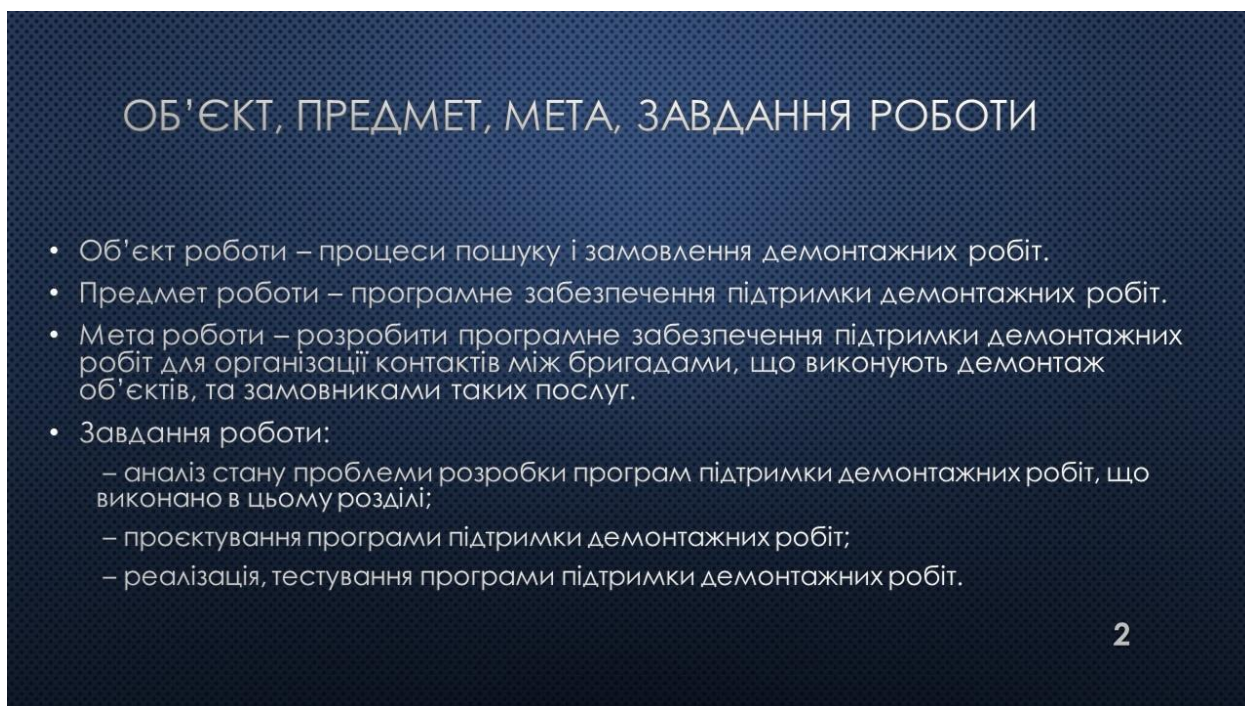


Рисунок В.2 – Об'єкт, предмет, мета, завдання роботи

ПОРІВНЯННЯ ПРОГРАМ ПІДТРИМКИ ДЕМОНТАЖНИХ РОБІТ, ЩО ІСНУЮТЬ, З РОЗРОБКОЮ

Критерій	demontag.zp.ua	Olx	Розробка
Оголошення про демонтажні роботи, що надаються як послуги	Так	Так	Так
Доступна бригада для демонтажу	Одна	Змінюється за кількістю пропозицій	Змінюється за кількістю пропозицій
Оголошення про необхідність демонтажу об'єктів	Ні	Фактично не застосовується	Так
Відгуки про результати демонтажу	Довільні користувачі	Немає	Користувачі, які виконали замовлення
Вартість послуг з демонтажу	Так	Так	Так
Пошук за параметрами демонтажу	Ні	Ні	Так

3

Рисунок В.3 – Порівняння програм підтримки демонтажних робіт, що існують, з розробкою

ПОРІВНЯННЯ ФРЕЙМВОРКІВ ВЕБРОЗРОБКИ ДЛЯ РОЗРОБКИ ПРОГРАМИ ПІДТРИМКИ ДЕМОНТАЖНИХ РОБІТ

Критерій	Django	Pyramid
Модульність	Монолітна, всі модулі вбудовані, можуть бути розширені	Стандартна, модулі можуть підключатися
Ідентифікація користувачів	Вбудована	Вбудованої немає, можна застосувати підключене рішення
Механізм шаблонів	Django Template Language	Jinja2, Chameleon, Mako можливо використати
Робота з базами даних	Об'єктно-реляційне відображення є і одразу придатне для різних систем (MySQL, PostgreSQL, NoSQL, Oracle)	Можна встановити і використати SQLAlchemy

4

Рисунок В.4 – Порівняння фреймворків веброзробки для розробки програми підтримки демонтажних робіт

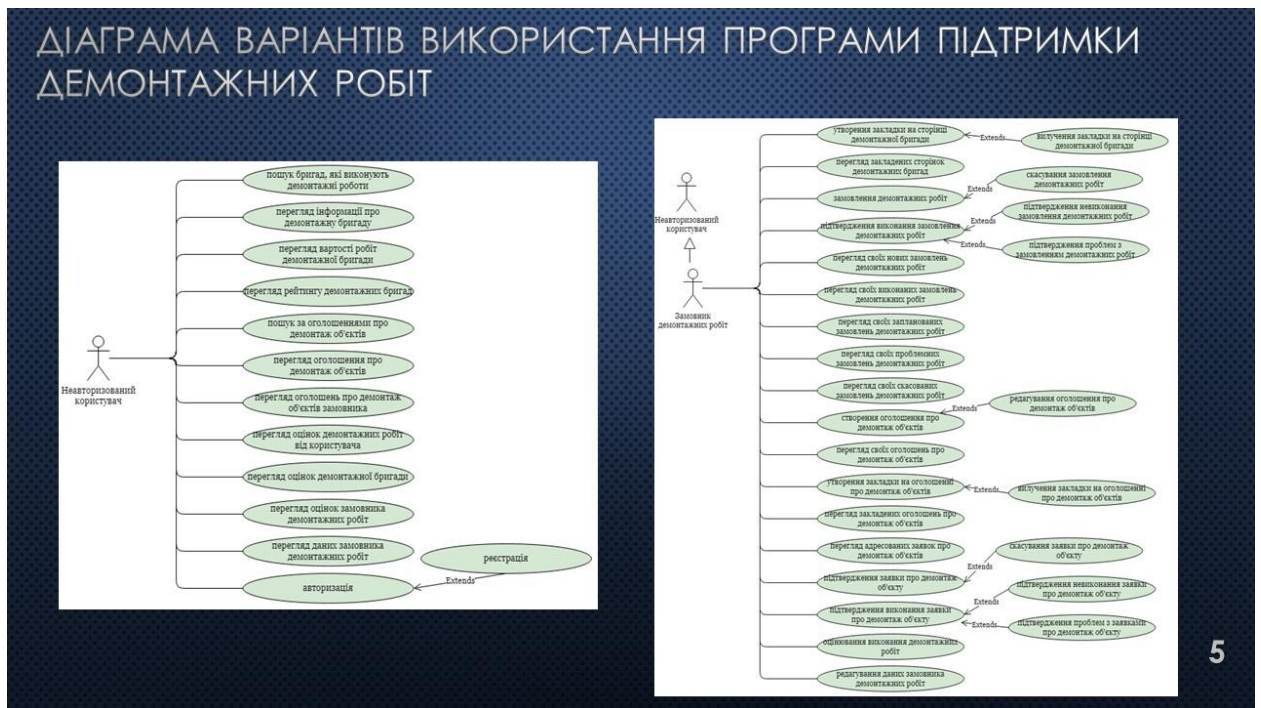


Рисунок В.5 – Діаграма варіантів використання програми підтримки
демонтажних робіт



Рисунок В.6 – Діаграма варіантів використання програми підтримки
демонтажних робіт

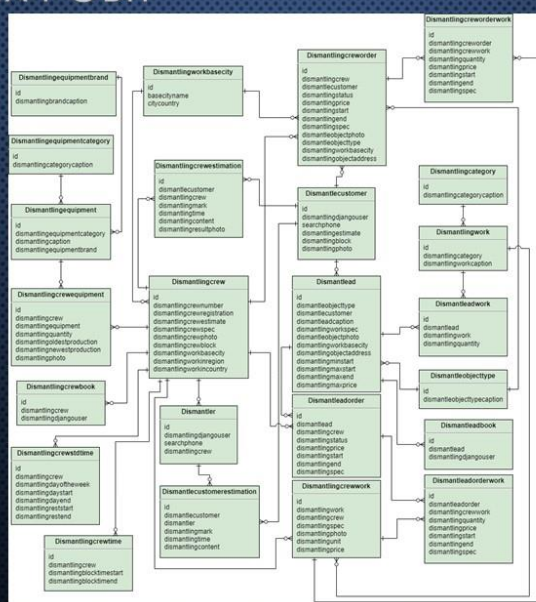


Рисунок В.7 – Схема бази даних програми підтримки демонтажних робіт

ДІАГРАМА СТАНІВ ЗАМОВЛЕННЯ ДЕМОНТАЖНИХ РОБІТ

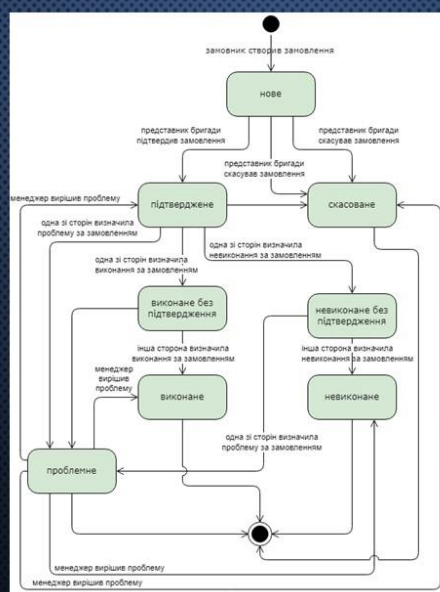
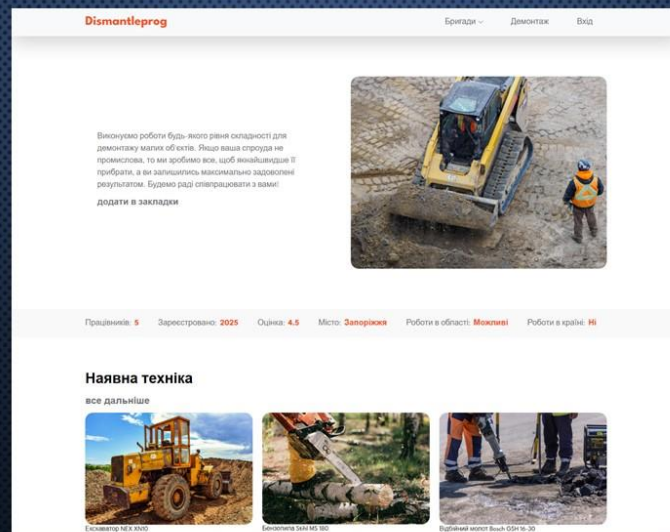


Рисунок В.8 – Діаграма станів замовлення демонтажних робіт

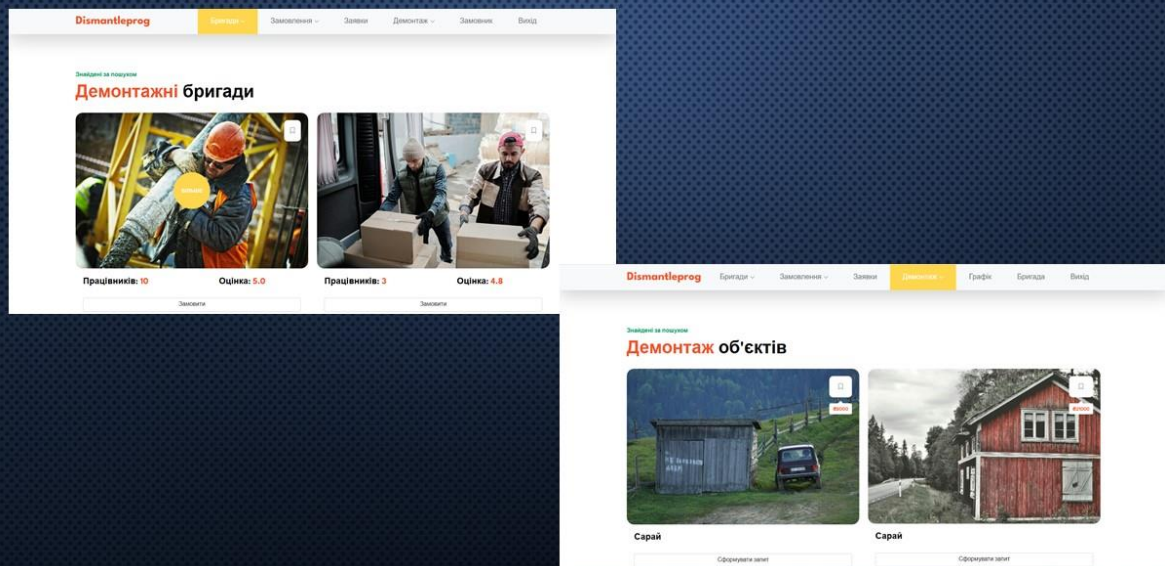
РЕЗУЛЬТАТИ ТЕСТУВАННЯ ПРОГРАМИ ПІДТРИМКИ ДЕМОНТАЖНИХ РОБІТ



9

Рисунок В.9 – Результати тестування програми підтримки демонтажних робіт

РЕЗУЛЬТАТИ ТЕСТУВАННЯ ПРОГРАМИ ПІДТРИМКИ ДЕМОНТАЖНИХ РОБІТ



10

Рисунок В.10 – Результати тестування програми підтримки демонтажних
робіт

РЕЗУЛЬТАТИ ТЕСТУВАННЯ ПРОГРАМИ ПІДТРИМКИ ДЕМОНТАЖНИХ РОБІТ

The screenshot displays the Dismantleprog web application. The header includes the logo and navigation links: Бригади, Замовлення, Заявки, Демонтаж, and Вихід. The main content area is divided into two sections. On the left, there is a search form with fields for 'Тип проблеми' (set to 'Всі'), 'Дата' (set to 'ДД.ММ.РРРР'), and buttons for 'Застосувати' and 'Перейти до пошуку'. On the right, the 'Проблемні замовлення' section shows a message: 'В обумовлений час бригада не з'явилась на місці.' Below this, a user profile for 'Ганна Агапова' is shown with a dropdown menu set to 'Виконати' and a 'Закрити сторінку' button. Further down, order details are listed: 'Номер замовлення: 14', 'Вартість: \$1000', 'Виконано: 28.05.2025 13:00 - 28.05.2025 16:00', 'Тип об'єкта: Гаряк', 'Місто: Запоріжжя', 'Адреса: бул. Шевченка, 10', 'сторінка бригади', 'показати повністю', 'блокувати замовлення', and 'блокувати бригаду'.

11

Рисунок В.11 – Результати тестування програми підтримки демонтажних робіт

ВИСНОВКИ

- Основні отримані результати роботи включають аналіз існуючих програм підтримки демонтажних робіт, проектування, реалізацію і тестування власного програмного забезпечення для підтримки демонтажних робіт.
- Існуючі програми підтримують взаємодію щодо демонтажних робіт, проте зазвичай представлені окремими вебсайтами окремих бригад. Цілісного ж майданчику для з'єднання пропозицій або не існує, або він не є спеціалізованим, тобто не враховує особливості саме демонтажних робіт.
- Розроблені функції програми направлені на реалізацію даного рішення у складі цілісного застосунку для фреймворку Django.

12

Рисунок В.12 – Висновки