

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 проекту (роботи)

бакалавра

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ РЕЗЕРВУВАННЯ

(назва теми)

ПОСЛУГ З БЛАГОУСТРОЮ ТЕРИТОРІЇ. SOFTWARE FOR

RESERVATION OF LANDSCAPING SERVICES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-220

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ЖУЧЕНКО М.С.

(ПРІЗВИЩЕ та ініціали)

Керівник ЛЬОВКІН В.М.

(ПРІЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРІЗВИЩЕ та ініціали)

2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет _____ КНТ _____
Кафедра _____ програмних засобів _____
Ступінь вищої освіти _____ бакалавр _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і найменування)
Освітня програма (спеціалізація) _____ Комп'ютерні науки _____
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф. _____
Сергій СУББОТІН
« _____ » _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЖУЧЕНКА Микити Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення резервування послуг з благоустрою території. Software for Reservation of Landscaping Services

керівник проєкту (роботи) к.т.н., доцент ЛЬОВКІН Валерій Миколайович _____,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 24 » квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програми. 3. Розробка програми. 4. Експлуатація та тестування програми.

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЛЬОВКІН В.М., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання « 15 » квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка)

_____ Микита ЖУЧЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Валерій ЛЬОВКІН
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 95 с., 30 рис., 3 табл., 3 дод., 10 джерел.

ВЕБДОДАТОК, РЕЗЕРВУВАННЯ ПОСЛУГ, БЛАГОУСТРІЙ ТЕРИТОРІЇ, НАДАВАЧ ПОСЛУГ, БАЗА ДАНИХ.

Об'єкт роботи – процес резервування послуг з благоустрою території. Предмет роботи – програмне забезпечення резервування послуг з благоустрою території. Мета роботи – розробити програмне забезпечення резервування послуг з благоустрою території для забезпечення керування взаємодією між надавачами послуг і клієнтами.

Матеріали, методи та технічні засоби: мова Python, система PostgreSQL, шаблон розмітки сторінок, бібліотека Bootstrap, фреймворк Django.

Результати. Виконана розробка програмного забезпечення є результатом не тільки проектування і програмної реалізації, але і аналізу програм, які в тій чи іншій мірі можуть задовольнити розв'язання проблеми, що розглядається.

Висновки. Програма признається для забезпечення підтримки послуг з благоустрою території через виконання резервування цих послуг у робітників (компаній) клієнтами. Останніми вона може використовуватися для пошуку потрібних послуг, отримання інформації про такі послуги та резервування виконання цих послуг для забезпечення благоустрою власної території, а компаніями – для визначення послуг, що надаються, параметрів резервування цих послуг.

Галузь використання. Підтримка послуг з благоустрою території, що надаються невеликими компаніями, окремими робітниками.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 95 pages, 30 figures, 3 tables, 3 appendixes, 10 sources.

WEB APPLICATION, RESERVATION OF SERVICES, LANDSCAPING SERVICES, SERVICE PROVIDER, DATABASE.

The object of the work is the process of reserving landscaping services. The subject of the work is the software for reservation of landscaping services. The goal of the work is to develop landscaping service reservation software to manage interaction between service providers and customers.

Materials, methods and technical means: Python language, PostgreSQL system, page markup template, Bootstrap library, Django framework.

The results. The completed software development is the result not only of design and software implementation, but also of analysis of software, which to one degree or another can satisfy the solution of the problem under consideration.

Conclusions. The program is recognized to provide support for landscaping services through the reservation of these services from workers (companies) by clients. It can be used by the latter ones to search for the necessary services, obtain information about such services and reserve the performance of these services to ensure the improvement of their own territory, and by companies to determine the services provided and the parameters of the reservation of these services.

Field of use. Support for landscaping services provided by small companies and individual workers.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Огляд програм резервування послуг з благоустрою території	10
1.2 Порівняння аналогів для визначення характеристик власної розробки	16
1.3 Висновки за розділом 1	18
2 Проєктування програми.....	19
2.1 Вибір засобів проєктування і розробки	19
2.2 Моделювання взаємодії користувача з програмою	20
2.3 Визначення структури бази даних	25
2.4 Висновки за розділом 2	33
3 Розробка програми	35
3.1 Визначення послідовності роботи з програмою	35
3.2 Особливості програмної реалізації.....	37
3.3 Висновки за розділом 3	50
4 Експлуатація та тестування програми	51
4.1 Призначення програми	51
4.2 Умови виконання програми	51
4.3 Експлуатація програми.....	51
4.4 Тестування програми	55
4.5 Висновки за розділом 4	58
Висновки	59
Перелік джерел посилання	60
Додаток А Технічне завдання	61
Додаток Б Текст програми	66
Додаток В Слайди презентації.....	88

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

АП – адміністративна панель;

ПЗ – програмне забезпечення.

ВСТУП

Послуги з благоустрою території є доволі специфічними, оскільки вони з одного боку можуть бути пов'язані з виконанням проєктів ландшафтного дизайну, що є фактично індивідуальними, а з іншого боку з конкретними роботами, що є більш типовими. Наприклад, до останніх належать прибирання території, підрізання дерев, підстригання газону тощо. Тому найчастіше є певний розподіл між цими послугами.

У одних випадках друга група послуг розглядається як додаткова та не дуже афішується компаніями. В інших випадках навпаки надається інформація про типові послуги, а послуги з ландшафтного дизайну фактично є певною додатковою позицією, на якій не акцентується увага.

У першому випадку тоді кожна компанія створює свій власний вебсайт, який більшою мірою містить портфоліо і є деяким сайтом-візитівкою. Таким чином інформація про послуги з благоустрою території є фрагментованою, а у одному додатку зручно знайти потрібну інформацію стає важко. На подолання даної проблеми має бути в першу чергу націлена дана розробка. Інформація про вартість послуг, надання можливості резервування послуг є тими можливостями, які особливо уважно необхідно врахувати в розробці. Врахування необхідності виконання бронювання послуг типово не є реалізованим. Але ж газон потрібно підстригати з певною періодичністю. Тому кожен раз через певний час потрібно заново витрачати час на реалізацію тих же самих дій стосовно тієї ж самої послуги. Тож дана проблема не є повністю розв'язаною у існуючих програмах, ґрунтуючись на даних попереднього аналізу.

Завдання, на розв'язання яких направлена виконувана дипломна кваліфікаційна робота:

– проаналізувати проблему розробки програми резервування послуг з благоустрою території;

- спроектувати програмне забезпечення (ПЗ) резервування послуг з благоустрою території;
- реалізувати ПЗ резервування послуг з благоустрою території та виконати його тестування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд програм резервування послуг з благоустрою території

Сервіс і відповідні додатки Yelp [1] (рис. 1.1) не є прямим інструментом, ціль створення якого є резервування послуг з благоустрою території, однак це інструмент, який фактично може використовуватися для прийняття обґрунтованих рішень щодо вибору надавача послуг з благоустрою території та щодо резервування таких послуг також. Фактично ж Yelp є сервісом, який накопичує відгуки від клієнтів щодо сервісів у великій кількості областей. Однією з таких областей послуг є благоустрій території. Тож за рахунок можливості шукати такі компанії, отримувати інформацію про досвід роботи з компаніями інших клієнтів цей додаток на практиці можна використовувати для ознайомлення з існуючими послугами.

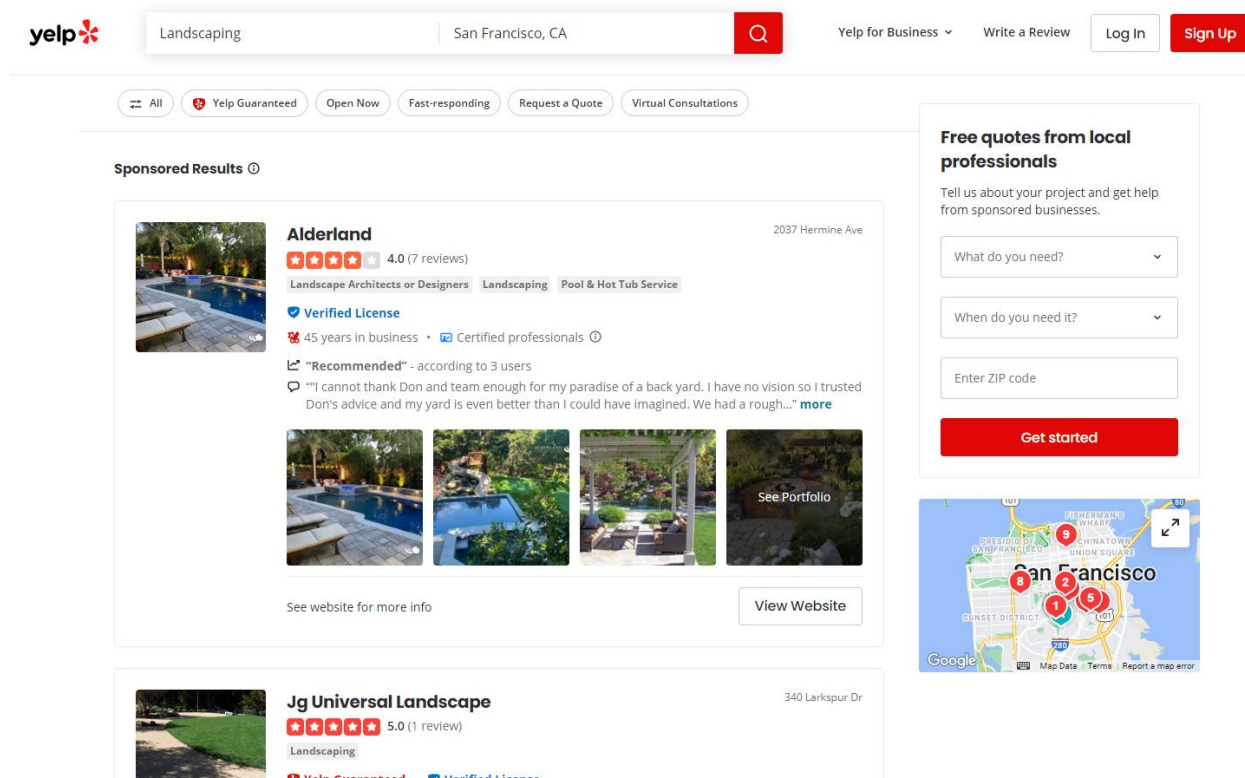


Рисунок 1.1 – Пошук послуг з благоустрою території у додатку Yelp

Безпосередньо сама сторінка компанії, яка надає послуги з благоустрою території, містить портфолію компанії, можливість написати відгук, зберегти компанію, щоб швидко повернутися до її сторінки в подальшому, підписатися на її оновлення, є достатньо інформативною сама по собі (рис. 1.2).

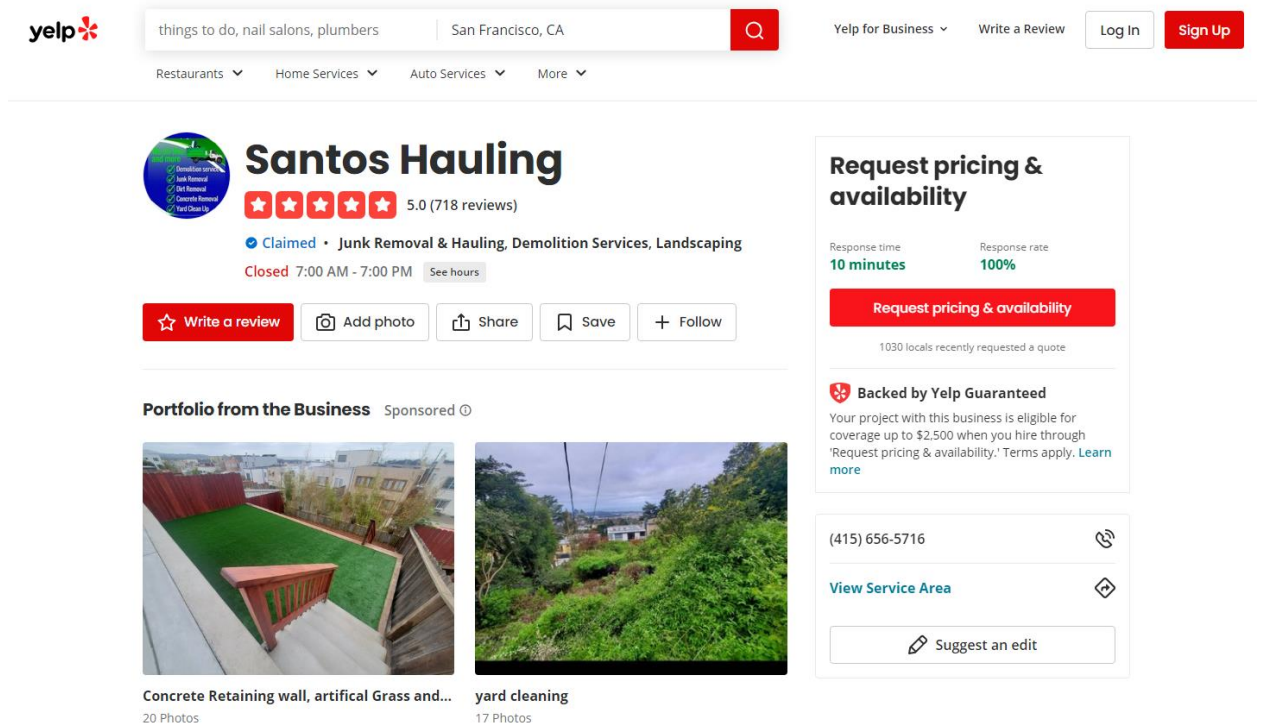
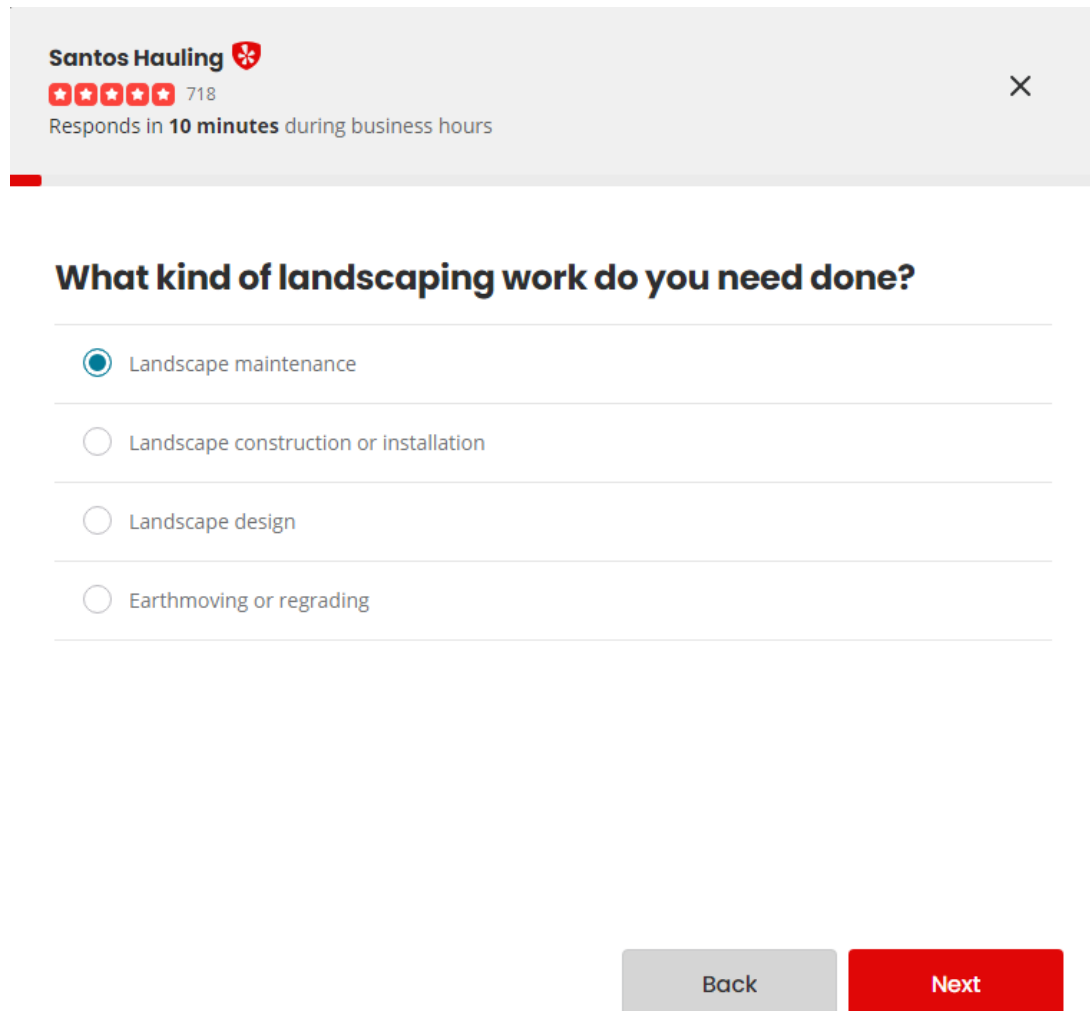


Рисунок 1.2 – Відображення даних сервісу з благоустрою території у додатку Yelp

Однак окрім такої інформації сторінка також містить і узагальнений з оцінок клієнтів рейтинг, а також і самі оцінки (відгуки) конкретних клієнтів. Це дозволяє зрозуміти, на скільки такому постачальнику послуг варто довіряти, що дуже важливо в цій сфері. А враховуючи, що саме на це спрямована взагалі логіка роботи сервісу Yelp, то це робить його важливим практичним інструментом для даної сфери.

Проте також важливим є те, що можна не тільки знайти потрібну компанію, отримати про неї інформацію, але і сформувати замовлення (рис. 1.3). Для цього використовується типова форма, яка направляє взаємодію з користувачем на послідовність етапів, на яких потрібно визначати параметри

замовлення. Для цього потрібно спочатку визначити тип робіт з благоустрою, а далі задати тип території, до якої має застосовуватися ця послуга (в загальному вигляді процедура така, але в залежності від обраного типу робіт на першому етапі наступні етапи можуть дещо відрізнятись).



Santos Hauling 

★★★★★ 718

Responds in **10 minutes** during business hours

What kind of landscaping work do you need done?

☒ Landscape maintenance

☐ Landscape construction or installation

☐ Landscape design

☐ Earthmoving or regrading

Back Next

Рисунок 1.3 – Замовлення послуги з благоустрою території у додатку Yelp

Проте варто уточнити, що дана форма є фактично формою зворотного зв'язку. Тобто це форма для уточнення вартості та доступності послуг за заданими параметрами. У підсумку представник компанії має зв'язатися з клієнтом. Період отримання відповіді всередньому зазначається для кожної компанії.

В Україні відомим сервісом замовлення послуг є сервіс, який має назву Кабанчик [2] (рис. 1.4). Цей сервіс дозволяє знаходити працівників, які

виконують ті чи інші роботи. Серед таких робіт є і роботи з категорії «Сад і огорода».

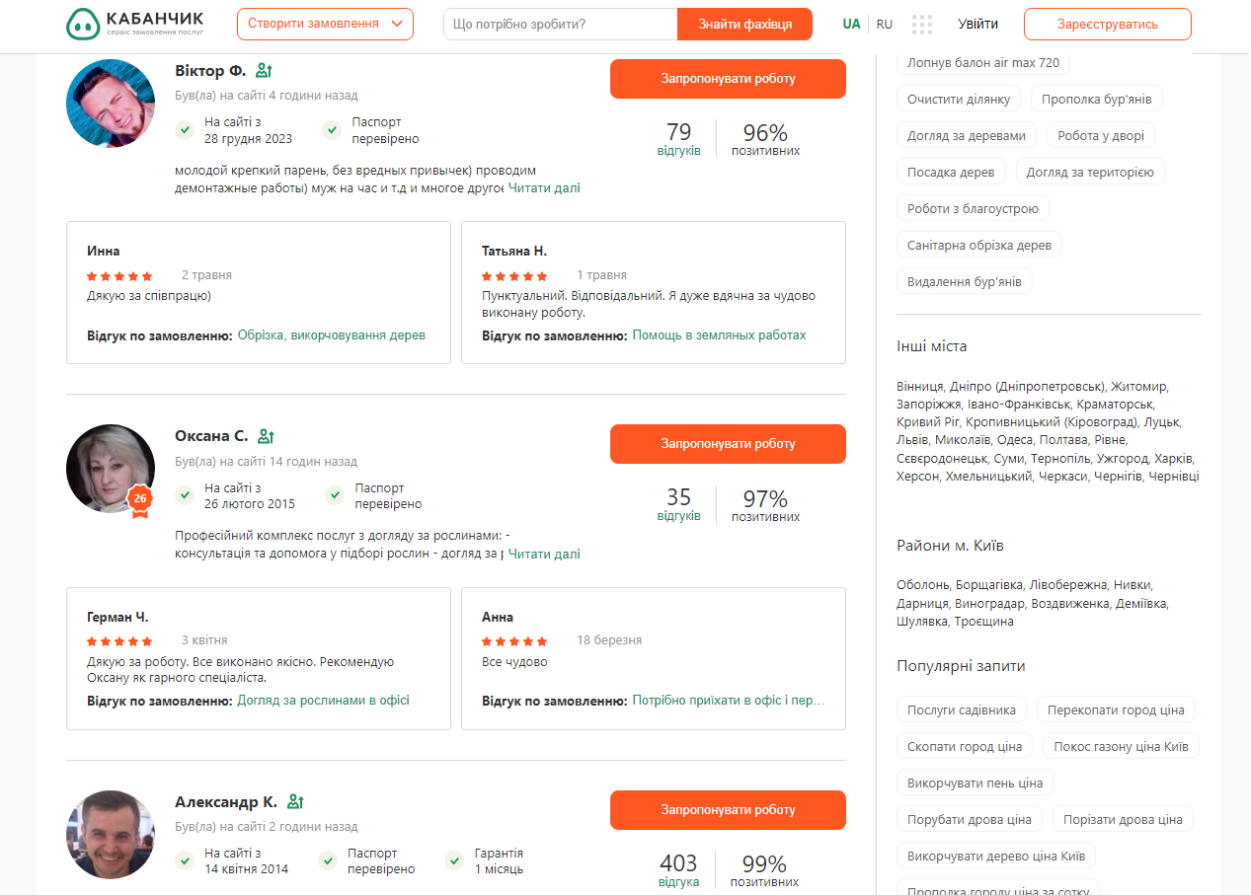


Рисунок 1.4 – Пошук індивідуальних послуг з благоустрою території у сервісі замовлення послуг Кабанчик

Вебдодаток дозволяє шукати оголошення від робітників з обраної категорії, при цьому надається перелік найкращих за рейтингом. За кожним робітником надається загальна інформація з основними відгуками одразу в результатах пошуку (хоча цей спосіб відображення швидше схожий на формування рейтингу). Повністю всі відгуки, дані про встановлену вартість послуг робітником на ті послуги з благоустрою території, які він для себе визначив, знаходяться на власній сторінці робітника (рис. 1.5).

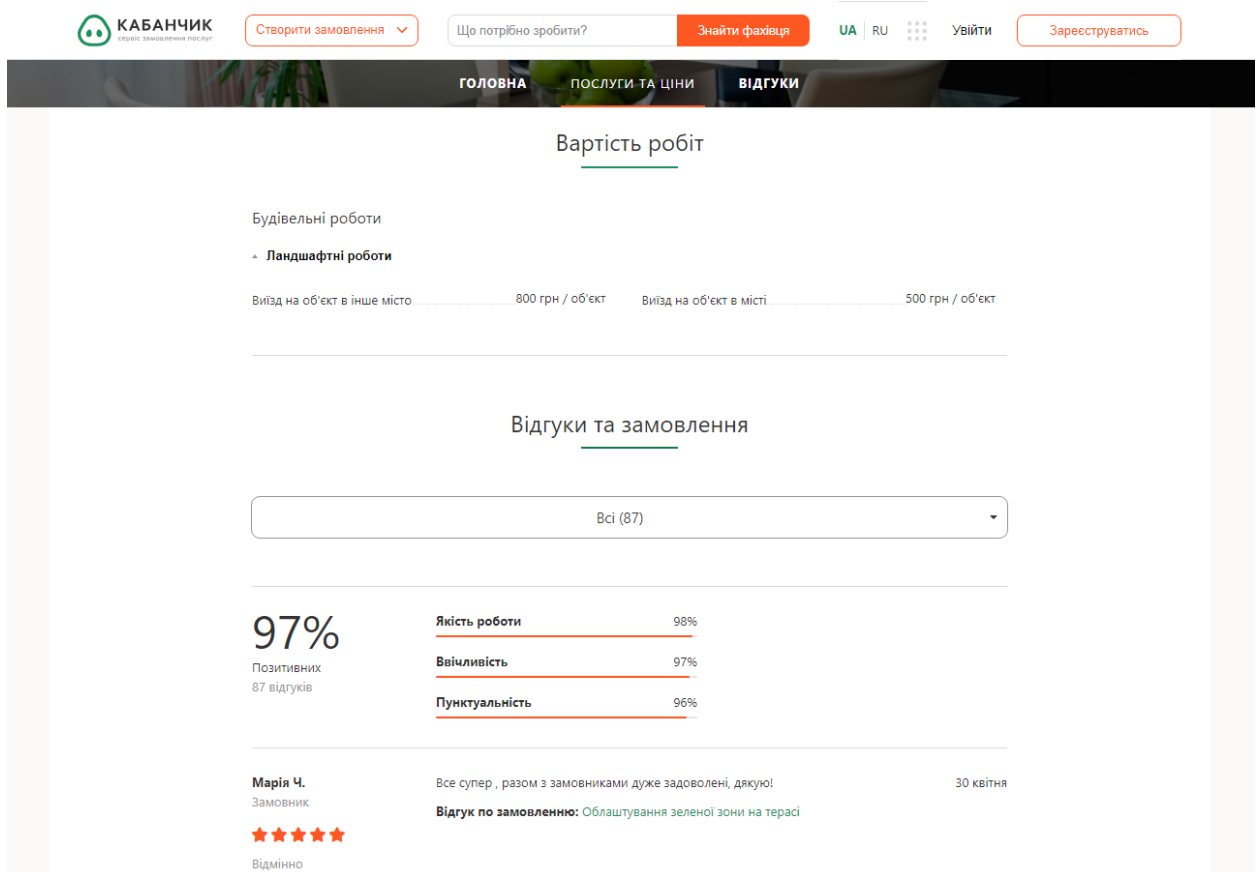


Рисунок 1.5 – Сторінка майстра з благоустрою території у сервісі замовлення послуг Кабанчик

Наявність таких програмних інструментів, які фактично є агрегаторами послуг, не відкидає наявності цілого ряду вебсайтів, які представляють окремі компанії, що надають послуги з благоустрою території (ландшафтного дизайну). Такі вебсайти не надають інформації про різні компанії, присвячені представленню інтересів тільки однієї конкретної. До прикладу за результатами огляду було виділено декілька. Це вебсайт компанії Стаффаж [3] (рис. 1.6) та Gal Sad [4] (рис. 1.7).

Такі вебсайти зазвичай містять типову інформацію. Загальною рисою є те, що вони фактично містять контактну форму для з'єднання з компанією, яка представляється як форма замовлення послуг. Не всі з них містять певний особистий кабінет, тобто передбачають авторизацію. Не всі також містять і опис вартості типових послуг (рис. 1.6). Проте зазвичай всі містять портфоліо з прикладами виконаних раніше робіт з благоустрою території (рис. 1.7).



З любов'ю до природи
Від серця до серця

ПРО НАС

ПОСЛУГИ

ПОРТФОЛІО

ЦІНИ

САДОВИЙ ЦЕНТР

БЛОГ

КОНТАКТИ

ЗВОРОТНИЙ ДЗВІНОК

ПОСАДКА ЛИСТЯНИХ ДЕРЕВ

№	Висота	Обхват стовбура	Од. вим.	Ціна
1	3-3,5 м	8-10-12-14 см	шт.	2 240 грн.
2	3,5-4,5 м	16-18 см	шт.	2 700 грн.
3	5-6 м	20-25 см	шт.	4 200 грн.
4	6-7 м	25-30 см	шт.	7 200 грн.
5	7 м	35-40 см	шт.	15 500 грн.

При посадці листяних дерев в зимовий час застосовується коефіцієнт 1,4

ПОСАДКА ХВОЙНИХ ДЕРЕВ

№	Висота	Од. вим.	Ціна
1	1 м	шт.	950 грн.
2	1-1,5 м	шт.	1 490 грн.
3	1,5-2 м	шт.	3 280 грн.

Рисунок 1.6 – Вартість послуг, представлена на сторінці вебсайту компанії Стаффаж



Пошук

+380 68 46 777 59
Замовити дзвінок

Увійти
Мій кабінет

1% від кожної Вашої покупки в інтернеті

Каталог

Акції

СТОК %

Наші роботи

Як купити?

Новини

Про компанію

Контакти

Статті

...

Ландшафтний дизайн

Головна - Наші роботи

Всі проектиЛандшафтний дизайнЗелені стіниФітодизайн



Ресторан 'Park. Art of Rest'



Український Католицький Університет



Готель 'Цитадель'

Рисунок 1.7 – Сторінка портфоліо компанії Gal Sad

1.2 Порівняння аналогів для визначення характеристик власної розробки

Зважаючи на те, що додатки Yelp, Кабанчик об'єднують різних надавачів послуг з благоустрою території, то вони були обрані для порівняння, оскільки розроблювана програма також націлена на роботу не з однією конкретною компанією, а з певною множиною компаній (табл. 1.1).

Додаток сервісу замовлення послуг Кабанчик не дозволяє встановлювати критерії впорядкування результатів пошуку послуг, що робить його фактично інструментом для перегляду рейтингу робітників за обраною послугою. Відповідно пошук менш гнучкий ніж у Yelp, тому саме підхід останнього варто використовувати.

Розроблювана програма повинна виконувати саме бронювання послуг з благоустрою території. Це означає, що компанія може пов'язувати конкретне замовлення з позицією в робочому часі, наприклад, після консультації з клієнтом. Відповідно під час пошуку можна враховувати, чи може ця компанія або цей робітник виконувати потрібні роботи у потрібний для клієнта час, а не тільки виконувати загальний пошук. Виходячи з цього, робітники, компанії можуть керувати своїм робочим часом, визначаючи, коли вони виконують певне замовлення, а коли мають можливість виконання робіт для інших клієнтів.

Додаток Yelp не надає інформацію про стандартну вартість послуг. Для того щоб дізнатися вартість, обов'язково потрібно оформити замовлення, яке зважаючи на відсутність бронювання є фактично замовленням консультації. При цьому важливим для клієнтів є не тільки рейтинг компанії, але і інформація про вартість, адже це дозволяє зрозуміти, чи варто витратити час на взаємодію з такою компанією. Якщо вартість послуг зовелика для клієнта, то він може одразу шукати іншу компанію. Окрім того корисною є запропонована у роботі можливість врахування інформації від клієнта про

вартість послуг. Це дозволяє зрозуміти, на скільки надана інформація про вартість відповідає дійсності.

Таблиця 1.1 – Порівняння існуючих аналогів та розроблюваного ПЗ резервування послуг з благоустрою території

Характеристика	Yelp	Кабанчик	LandServe
Пошук послуг з благоустрою території	+	+	+
Визначення критеріїв впорядкування результатів пошуку послуг	+	—	+
Замовлення/бронювання послуг з благоустрою території	Замовлення	Замовлення	Бронювання
Керування робочим часом компанії	—	—	+
Оцінювання наданих послуг з благоустрою території	+	+	+
Надання інформації про стандартну вартість послуг	—	+	+
Врахування інформації від клієнта про вартість послуг	—	—	+
Відображення фінансових результатів конкретної послуги з благоустрою території	—	—	+
Визначення повторюваного замовлення послуг з благоустрою території	—	—	+

1.3 Висновки за розділом 1

Огляд програм резервування послуг з благоустрою території, проведений у першому розділі, охоплює цілий ряд додатків, включаючи сервіс надання відгуків Yelp, сервіс замовлення послуг Кабанчик, а також вебсайти окремих компаній з благоустрою території. За результатами аналізу можливостей цих рішень виділено особливості розробки, які окрім найважливіших функцій включають також врахування інформації від клієнта про вартість послуг, керування робочим часом компанії, що пов'язане з виконанням саме бронювання послуг, а не тільки формування замовлень як у програмах-аналогах.

2 ПРОЄКТУВАННЯ ПРОГРАМИ

2.1 Вибір засобів проєктування і розробки

Основні засоби проєктування і розробки охоплювали рішення стосовно цілого ряду складових розробки:

- основна мова програмування, використана для створення ПЗ резервування послуг з благоустрою території, – Python [5], причина чому є можливість створення ПЗ різного типу, в тому числі підсумкове забезпечення вебдодатків, широкі можливості використання готового коду на основі пакетів;

- вебфреймворк – Django [6]-[7], альтернативним якому детально було розглянуто Pyramid [8]-[9] (табл. 2.1): Django має більшу швидкість створення програмних прототипів, тому і перша, і наступні версії ПЗ під час дипломування створити швидше і простіше, до того ж і заповнення даних вже першої версії ПЗ виконати швидше, використовуючи адміністративну панель (АП), також наявний зручний механізм роботи з формами, який включає автоматичну валідацію даних;

- реалізацію бази даних та роботи з базою даних вирішено забезпечувати через створення бази даних на основі системи PostgreSQL, яка є проєктом з відкритим вихідним кодом, надійною, має розширену підтримку захисту даних, відзначається масштабованістю;

- шаблон розмітки сторінок Gardener [10], що є безкоштовним, включає скрипти мовою JavaScript, бібліотеку Bootstrap та дозволяє створювати вебсторінки для реалізації ПЗ на основі готової розмітки, зменшуючи в підсумку потрібний час для фактично створення фронтенд-частини вебдодатку.

Таблиця 2.1 – Порівняння фреймворків для реалізації ПЗ резервування послуг з благоустрою території

Характеристика	Pyramid	Django
Швидкість створення програмних прототипів	Нижча	Вища за рахунок кількості вбудованих інструментів
Початкове заповнення даними	Необхідно створити сторінки або працювати через систему керування базами даних	Можна розпочати одразу через АП
Механізм валідації форм	Ні	Так
Створення і забезпечення представлень	Функції, класи	Функції, класи

2.2 Моделювання взаємодії користувача з програмою

ПЗ резервування послуг з благоустрою території використовується такими користувачами як надавач послуг з благоустрою території, клієнт та менеджер програми.

Клієнт може взаємодіяти з ПЗ наступним чином:

- пошук надаваних послуг з благоустрою території;
- резервування послуг з благоустрою території;
- визначення повторюваного замовлення послуг з благоустрою території;
- відображення загальної інформації про надавача послуг з благоустрою території;
- відображення всіх послуг з благоустрою території заданого надавача послуг;

- відображення всіх виставлених оцінок надавачу послуг;
 - відображення портфолію надавача послуг;
 - оцінювання наданих послуг з благоустрою території;
 - відображення оцінювань клієнта;
 - відображення інформації про клієнта;
 - відображення власних замовлених резервувань послуг з благоустрою території;
 - авторизація;
 - уподобання надавача послуг з благоустрою території;
 - відображення уподобаних надавачів послуг з благоустрою території
- (рис. 2.1).

Альтернативним сценарієм для резервування послуг з благоустрою території є скасування власного резервування послуг з благоустрою території, що відбувається в тому випадку, коли виконується одразу 3 передумови:

- час початку реалізації послуг не менше ніж за 24 години від поточного моменту;
- саме резервування вже створене;
- хоча би 1 з 2 останніх резервувань клієнта було завершено без скасування клієнтом.

Ще одним альтернативним сценарієм є оскарження резервування послуг з благоустрою території.

Альтернативним сценарієм для уподобання надавача послуг з благоустрою території є скасування уподобання надавача послуг з благоустрою території, що відбувається в тому випадку, коли виконується передумова, що уподобання цим клієнтом цього надавача послуг вже виконано.

Альтернативним сценарієм для представленої авторизації забезпечується реєстрації за передумови наявності доступу до програми, але відсутності власного акаунту.

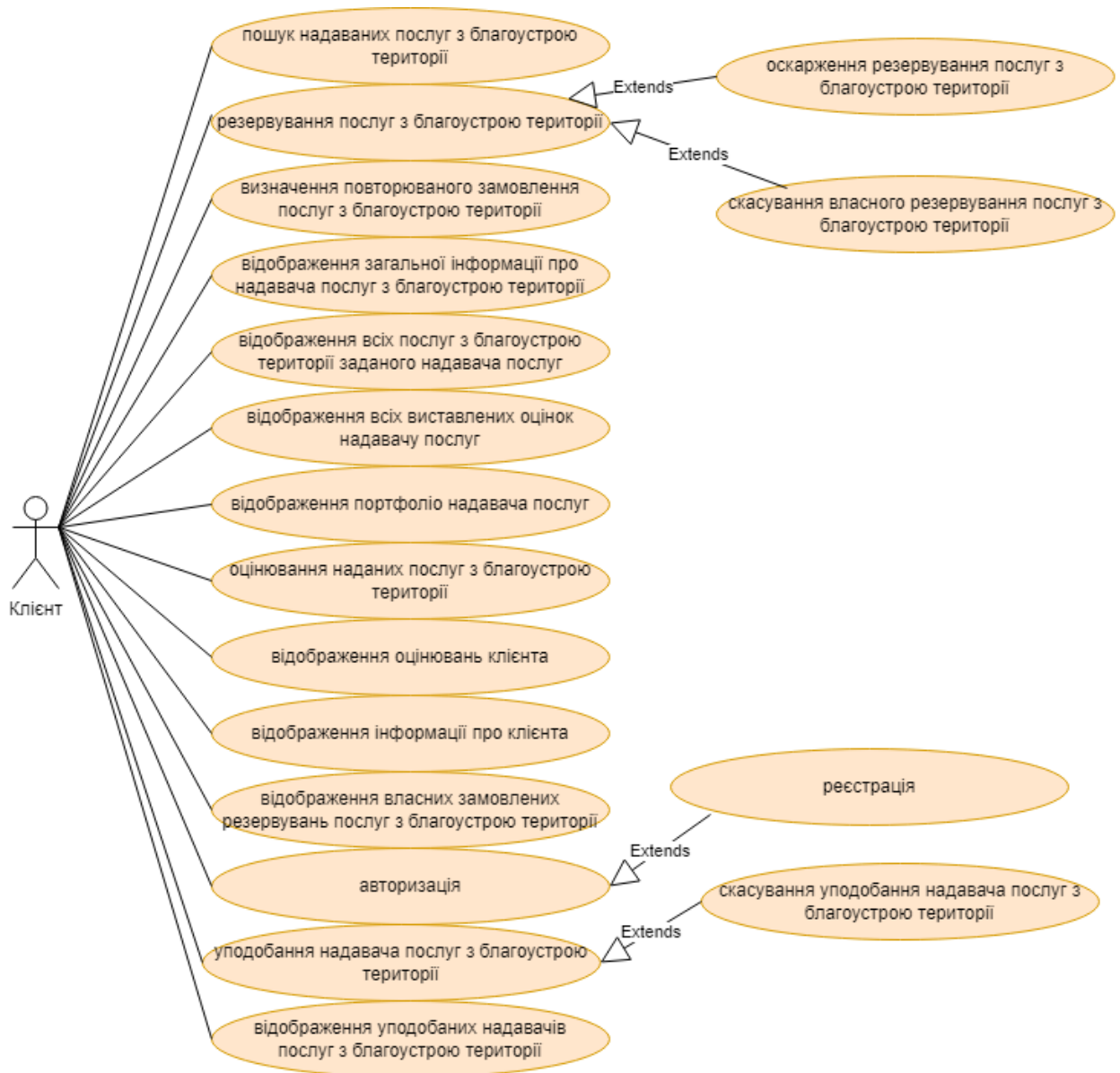


Рисунок 2.1 – Діаграма прецедентів клієнта для ПЗ резервування послуг з благоустрою території

Надавач послуг з благоустрою території може взаємодіяти з ПЗ на основі прецедентів клієнта та додатково наступним чином:

- підтвердження резервування послуг з благоустрою території;
- додавання інформації про послугу з благоустрою території;
- визначення часу, доступного для виконання робіт, надавачем послуг;
- відображення резервувань послуг з благоустрою території за заданим станом;
- створення нового надавача послуг з благоустрою території;

- визначення завершеного обслуговування за резервуванням;
- визначення розпочатого обслуговування за резервуванням;
- відображення робочого часу надавача послуг за календарем;
- відображення фінансових результатів конкретної послуги з благоустрою території надавача послуг (рис. 2.2).

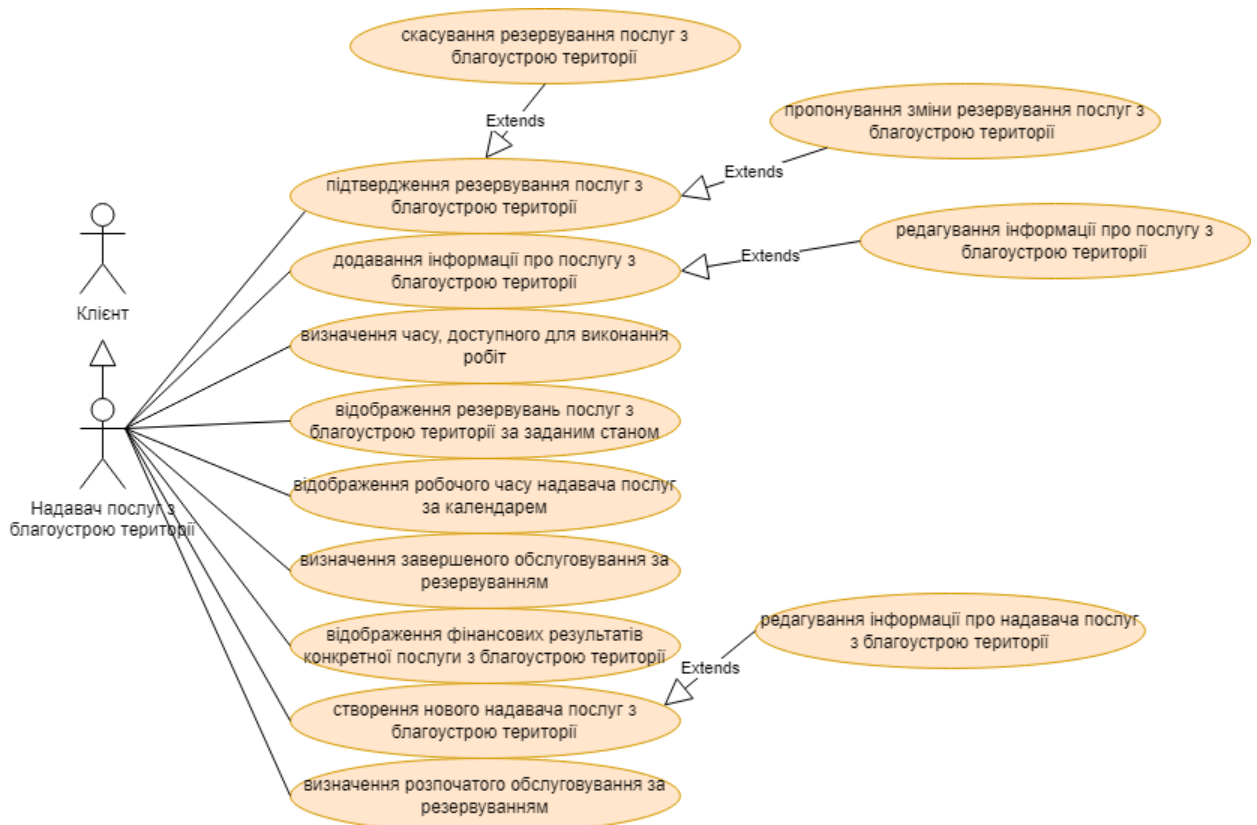


Рисунок 2.2 – Діаграма прецедентів надавача послуг з благоустрою території

Підтвердження резервування послуг з благоустрою території має одразу 2 альтернативні сценарії:

- скасування резервування послуг з благоустрою території, що реалізується надавачем послуг в тому випадку, коли резервування створене клієнтом, але дані клієнта не виглядають надійними або резервування з такими параметрами неможливо здійснити;
- пропонування зміни резервування послуг з благоустрою території надавачем послуг реалізується в тому випадку, коли резервування створене клієнтом, але виконати ці послуги в заданий час неможливо, а надавач послуг

пропонує виконати обслуговування в інший час.

Альтернативним сценарієм для додавання інформації про послугу з благоустрою території є редагування інформації про послугу з благоустрою території надавачем послуг для випадку, коли послуга визначена, але інформація має бути оновлена.

Альтернативним сценарієм для створення нового надавача послуг з благоустрою території є редагування інформації про надавача послуг з благоустрою території.

Менеджер програми може взаємодіяти з ПЗ на основі ряду прецедентів (рис. 2.3).



Рисунок 2.3 – Діаграма прецедентів менеджера програми резервування послуг з благоустрою території

Основний потік прецедентів включає:

- пошук резервувань послуг з благоустрою території;
- перегляд резервування послуг з благоустрою території;
- редагування резервування послуг з благоустрою території, включаючи зміну стану резервування;
- призначення представника надавача послуг;
- авторизація;
- пошук надаваних послуг з благоустрою території;
- відображення загальної інформації про надавача послуг з благоустрою території;
- відображення всіх послуг з благоустрою території заданого надавача послуг;
- відображення всіх виставлених оцінок надавачу послуг;
- відображення портфолію надавача послуг;
- відображення оцінювань клієнта;
- відображення інформації про клієнта.

Альтернативною для авторизації є реєстрація. Вона необхідна, коли менеджер не має акаунта, при цьому акаунт не був для нього створений і адміністратором через АП. Реєстрація виконується разом з надавачами послуг з благоустрою території. Але далі адміністратор має тоді визначити для такого користувача менеджерську роль.

2.3 Визначення структури бази даних

Детально обрану структуру бази даних описано нижче.

Категорії послуг з благоустрою територій містяться в таблиці Landscapcategory, що має такі поля:

- id – ідентифікатор категорії послуг з благоустрою території (первинний ключ);
- category_name – назва категорії послуг з благоустрою території.

Послуги з благоустрою територій містяться в таблиці Landscapeservice, що має такі поля:

- id – ідентифікатор послуги з благоустрою територій (первинний ключ);
- landscape_category – зовнішній ключ, який задає категорію послуги з благоустрою території;
- service_name – назва послуги з благоустрою територій;
- measurement – базова одиниця визначення кількості застосування послуги з благоустрою території (штуки, квадратні метри, кубічні метри).

Пропозиції послуги з благоустрою територій відрізняються від попередньої сутності тим, що вони вже стосуються конкретних надавачів послуг, та містяться в таблиці Landscaperproposition, що має такі поля:

- id – ідентифікатор пропозиції послуги з благоустрою територій (первинний ключ);
- landscape_service – зовнішній ключ, який задає послугу з благоустрою території, за якою сформована пропозиція надавачем послуг;
- service_provider – зовнішній ключ, який задає надавача послуг, що визначив цю пропозицію;
- photo – фотографія, на якій показано і яка буде ілюструвати пропозицію надавача послуг;
- minimal_price – мінімальна вартість, за яку цю послугу може виконувати цей надавач послуг (або стандартна вартість, якщо для неї не визначається певний діапазон);
- maximum_price – максимальна вартість, за яку цю послугу може виконувати цей надавач послуг (може бути пропущена, якщо вартість послуги є стандартною);
- measurement_unit – кількість одиниць виміру, за яку встановлюється ця вартість (якщо надавач послуги визначає фактично інші одиниці виміру – наприклад, 10 штук, а не 1 штука, то цей коефіцієнт дозволяє визначити це за стандартної базової одиниці виміру для такої послуги);

Дані надавачів послуг містяться в таблиці `Serviceprovider`, що має такі поля:

- `id` – ідентифікатор надавача послуг з благоустрою території (первинний ключ);
- `business_name` – назва компанії, яка надає ці послуги з благоустрою території (якщо це індивідуальний робітник, то назва може бути пропущена);
- `presenter_surname` – прізвище основного представника компанії або робітника;
- `presenter_name` – ім'я основного представника компанії або робітника;
- `business_city` – місто, в якому надаються послуги компанією або робітником;
- `profile_photo` – фотографія профілю надавача послуг з благоустрою території;
- `details` – опис можливостей або особливостей (або представлення) надавача послуг з благоустрою території;
- `client_score` – оцінка, на яку всередньому оцінили роботу надавача послуг з благоустрою території клієнти.

Мови, якими вміють спілкуватися представники надавача послуг, містяться в таблиці `Serviceproviderlanguage`, що має такі поля:

- `id` – ідентифікатор мови, якою вміють спілкуватися представники надавача послуг (первинний ключ);
- `service_provider` – зовнішній ключ, який задає надавача послуг, якого стосується ця інформація;
- `language` – мова, якою вміють спілкуватися представники надавача послуг.

Проекти з благоустрою території (ландшафтного дизайну), якими визначається портфоліо надавача послуг, містяться в таблиці `Portfolioproject`, що має такі поля:

– id – ідентифікатор проєкту з портфоліо надавача послуг з благоустрою території (первинний ключ);

– serviceprovider – зовнішній ключ, що задає надавача послуг, який виконав цей проєкт з благоустрою території;

– project_name – назва проєкту з благоустрою території;

– details – опис особливостей проєкту з благоустрою території.

Фотографії проєктів з благоустрою території містяться в таблиці Portfoliophoto, що має такі поля:

– id – ідентифікатор фотографії проєкту з благоустрою території (первинний ключ);

– portfolioproject – зовнішній ключ, який задає проєкт з благоустрою території, який представляє ця фотографія;

– photo – фотографія, яка зображує результати виконання проєкту з благоустрою території;

– details – опис особливостей проєкту з благоустрою території, які показані на фотографії.

Дані клієнтів сервісу містяться в таблиці Client, що має такі поля:

– id – ідентифікатор клієнта сервісу (первинний ключ);

– profile_photo – фото профіля клієнта сервісу;

– username – ім'я клієнта сервісу як логін;

– client_name – ім'я клієнта сервісу як його фактичне ім'я;

– password – пароль клієнта сервісу;

– details – опис для профілю клієнта сервісу;

– phone – номер телефону, за яким здійснюється обслуговування клієнта сервісу.

Резервування послуг з благоустрою території містяться в таблиці Reservation, що має такі поля:

– id – ідентифікатор резервування послуг з благоустрою території (первинний ключ);

- client – зовнішній ключ, який задає клієнта, який зарезервував ці послуги з благоустрою території;
- serviceprovider – зовнішній ключ, який задає надавача послуг з благоустрою території;
- reserve_date – час, коли було створене резервування послуг з благоустрою території;
- assessed_price – оцінювана початково вартість даних послуг з благоустрою території;
- actual_price – фактична підсумкова вартість даних послуг з благоустрою території;
- approved_price – вказує на те, чи було підтверджено вказану фактичну підсумкову вартість другою стороною, чи дані визначені тільки за свідченням однієї зі сторін;
- status – стан резервування послуг з благоустрою території;
- address – адреса, за якою мають проводитися роботи з благоустрою території;
- proposed_start – час, коли пропонується розпочати виконання послуг з благоустрою території (початковий час, який запропоновано клієнтом, або відкоригований час, який запропоновано надавачем послуг, якщо він не може виконати послуги у попередньо узгоджений час);
- proposed_duration – час, коли пропонується закінчити виконання послуг з благоустрою території;
- service_start – узгоджений час початку виконання робіт з благоустрою території;
- service_end – узгоджений час завершення виконання робіт з благоустрою території.

Склад резервування послуг з благоустрою території міститься в таблиці Landscaperpropositionprice, що має такі поля:

- id – ідентифікатор позиції у складі резервування послуг з благоустрою території (первинний ключ);

- reservation – зовнішній ключ, який задає резервування послуг з благоустрою території, склад якого визначається цим рядком таблиці;
- landscapereproposition – зовнішній ключ, який задає пропозицію послуги з благоустрою територій, яка входить до складу резервування;
- quantity – кількість замовлених одиниць відповідної пропозиції послуги з благоустрою територій;
- unit_price – вартість одиниці цієї послуги з благоустрою територій, що використана в цьому резервуванні;
- total_price – сумарна вартість цієї пропозиції послуги з благоустрою територій у складі резервування;
- details – опис особливостей застосування цієї пропозиції послуги з благоустрою територій.

Дані щодо періоду роботи надавача послуг з благоустрою території містяться в таблиці Servicetime, що має такі поля:

- id – ідентифікатор періоду роботи надавача послуг з благоустрою території (первинний ключ);
- service_provider – зовнішній ключ, що задає надавача послуг з благоустрою території, для якого визначено цей період роботи;
- start_service – час початку роботи на відповідну дату даного надавача послуг з благоустрою території;
- end_service – час завершення роботи на відповідну дату даного надавача послуг з благоустрою території.

Дані щодо необхідності повторення резервування (створення шаблону резервування) послуг з благоустрою території містяться в таблиці Reservationrepeat, що має такі поля:

- id – ідентифікатор повторення резервування послуг з благоустрою території (первинний ключ);
- reservation – зовнішній ключ, який задає резервування послуг з благоустрою території, яке має використовуватися для повторення резервування;

– repeat_period_time – період повторення резервування (наприклад, 1, якщо в одиницях вимірювання задано місяць);

– repeat_period_measurement – одиниця вимірювання періоду повторення резервування (наприклад, місяць);

– period_end – час, до якого має відбуватися повторення резервування.

Уподобання надавачів послуг з благоустрою територій містяться в таблиці Serviceproviderstar, що має такі поля:

– id – ідентифікатор уподобання надавача послуг з благоустрою територій клієнтом (первинний ключ);

– serviceprovider – зовнішній ключ, що задає надавача послуг з благоустрою територій, якого уподобано;

– client – зовнішній ключ, який задає клієнта, який уподобав надавача послуг з благоустрою територій;

– star_time – час, коли було виконано уподобання надавача послуг з благоустрою територій клієнтом.

Оцінювання надавачів послуг з благоустрою територій клієнтами містяться в таблиці Serviceproviderscore, що має такі поля:

– id – ідентифікатор оцінювання надавачів послуг з благоустрою територій клієнтом (первинний ключ);

– service_provider – зовнішній ключ, що задає надавача послуг з благоустрою територій, якого було оцінено;

– client – зовнішній ключ, який задає клієнта, який оцінив надавача послуг з благоустрою територій;

– score_time – час, коли виконано оцінювання надавача послуг з благоустрою територій;

– score_integer – числова оцінка, виставлена надавачу послуг з благоустрою територій;

– score_text – текст оцінювання надавача послуг з благоустрою територій;

– fair_price – визначає, чи фактична підсумкова вартість виконання робіт відповідала інформації про вартість робіт з благоустрою територій, яку для себе визначив цей надавач послуг (рис. 2.4).

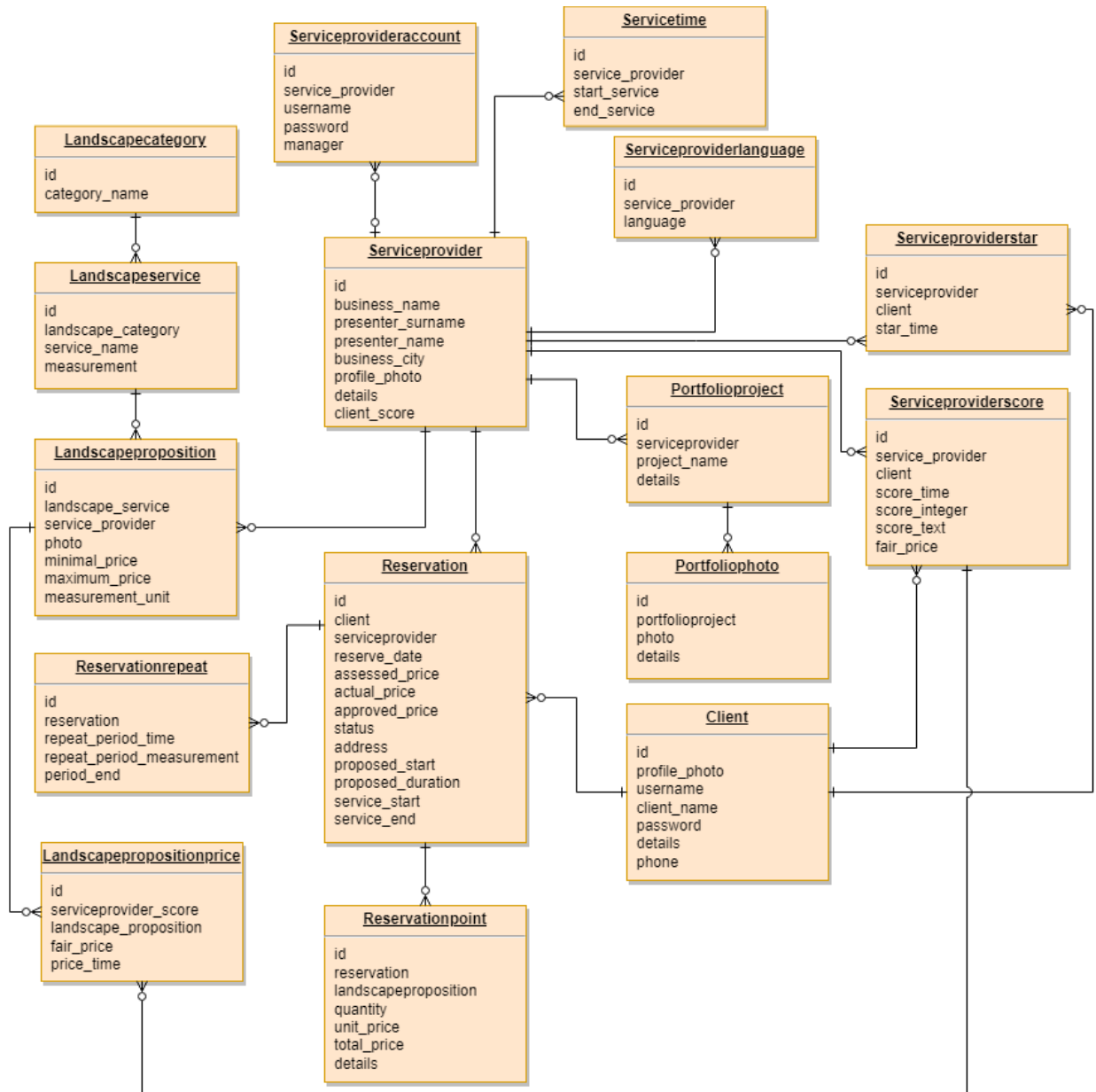


Рисунок 2.4 – Схема бази даних ПЗ резервування послуг з благоустрою території

Дані акаунтів сервісів (або менеджерів програми) містяться в таблиці Serviceprovideraccount, що має такі поля:

– id – ідентифікатор клієнта сервісу (первинний ключ);

- username – ім'я акаунта сервісу як логін;
- password – пароль акаунта сервісу;
- manager – акаунт сервісу або акаунт менеджера програми;
- service_provider – зовнішній ключ, що задає надавача послуг з благоустрою територій, до якого відноситься акаунт.

Деталізація відповідності вартості окремих послуг з благоустрою території даним, які стосовно себе наводив у програмі надавач послуг, міститься в таблиці Landscaperepropositionprice бази даних, що має наступні поля:

- id – ідентифікатор деталізації відповідності вартості окремих послуг з благоустрою території даним, які стосовно себе наводив у програмі надавач послуг (первинний ключ);

- serviceprovider_score – зовнішній ключ, який задає оцінювання надавачів послуг з благоустрою територій клієнтом, до якого відноситься деталізація;

- landscape_proposition – зовнішній ключ, який задає пропозицію послуг з благоустрою території, стосовно якої було визначено відповідність вартості даним, які стосовно цієї пропозиції наводив у програмі надавач послуг з благоустрою території;

- fair_price – визначає, чи вартість цієї послуги з благоустрою території, як вона була виконана надавачем послуг, відповідає обіцяному проміжку цін, визначеному в програмі;

- price_time – час, коли була визначена відповідність вартості окремих послуг з благоустрою території даним, які стосовно себе наводив у програмі надавач послуг.

2.4 Висновки за розділом 2

Проектування програми виконано за результатами вибору мови Python та фреймворку Django.

Діаграми прецедентів створені для надавача послуг, менеджера програми та клієнта для визначення відповідних функціональних вимог до програми.

Розроблену структуру бази даних ПЗ резервування послуг з благоустрою території презентовано у розділі. Реалізацію бази даних та роботи з базою даних забезпечено через створення бази даних на основі системи PostgreSQL, яка є проєктом з відкритим вихідним кодом, надійною, має розширену підтримку захисту даних, відзначається підтримкою масштабованості.

З РОЗРОБКА ПРОГРАМИ

3.1 Визначення послідовності роботи з програмою

Виходячи з предметної області, функціональних вимог до програми, саме резервування послуг з благоустрою території як клас є зв'язуючою сутністю для роботи додатку. Саме з резервуванням підсумково контактують всі ролі користувачів. Саме у резервуваннях визначаються поточні та підсумкові результати роботи. Тому важливо одразу визначити, які стани резервування послуг з благоустрою території можуть бути та як вони змінюються (рис. 3.1).

Спершу, коли об'єкт резервування взагалі з'являється, він має стан створеного. Якщо створене резервування підтвердить надавач послуг, то воно стає підтвердженим. Якщо він скасує його, то резервування стане скасованим, а також до такого стану його перевести може клієнт. Окрім того надавач послуг може запропонувати інші параметри резервування для клієнта, якщо наприклад, він не може виконати послуги у заданий час, але може в інший. Тоді клієнт може або відмовитися від змін, перевіривши резервування до стану скасованого, або підтвердити зміни, перевіривши резервування до стану підтвердженого. Перевести резервування до стану «запропоновано зміни» надавач послуг може і в той момент, коли резервування вже є підтвердженим, але ще не розпочатим. Зі стану підтвердженого клієнт може також перевести резервування до стану скасованого, якщо час початку реалізації послуг не менше ніж за 24 години від поточного моменту, хоча би 1 з 2 останніх резервувань клієнта було завершено без скасування клієнтом.

Резервування переходить зі стану підтвердженого до розпочатого, якщо надавач послуг підтвердив початок виконання послуг або якщо пройшла доба з початку виконання і жодної дії не відбулось.

Розпочате резервування після підтвердження закінчення робіт надавачем послуг переводиться до закінченого резервування. Якщо за розпочатого резервування надавач послуг або клієнт визначив невиконання

надавачем послуг обов'язків, то резервування переходить до стану “не виконано виконавцем”. Якщо за розпочатого резервування надавач послуг або клієнт визначив невиконання клієнтом обов'язків, то резервування переходить до стану “не виконано клієнтом”.

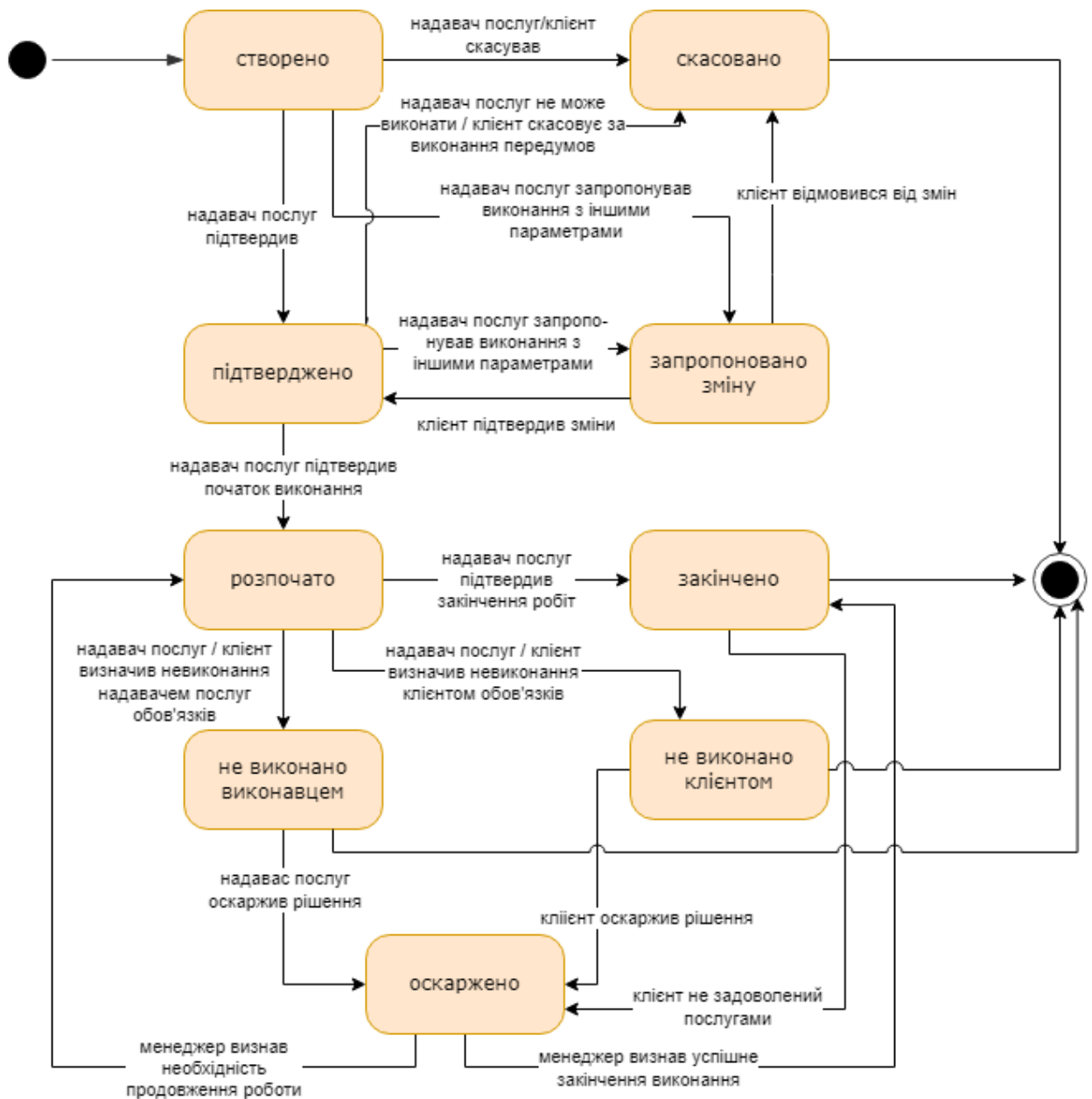


Рисунок 3.1 – Діаграма станів резервування послуг з благоустрою території

Зі стану “не виконано виконавцем” резервування може бути надавачем послуг, якщо він з цим незгодний, переведено до стану “оскаржено”. Зі стану

“не виконано клієнтом” резервування може бути клієнтом, якщо він з цим незгодний, переведено до стану “оскаржено”.

Оскаржене резервування може бути менеджером переведено до стану розпочатого, якщо менеджер визнав необхідність продовження роботи за результатами розгляду оскарження, або до стану закінченого, якщо менеджер визнав успішне закінчення виконання за результатами розгляду оскарження.

3.2 Особливості програмної реалізації

Фреймворк Django чітко визначає структуру, яку повинен мати додаток, тому на відповідний варіант розділення структури і було зорієнтовано створений додаток. Детально це описано нижче.

АП забезпечено через реєстрацію моделей у файлі `admin.py`. Самі ж моделі описані нижче.

Модель категорії послуг з благоустрою територій є класом `Landscapcategory`. Програмно визначено (ідентифікатор у класу моделі створюється за замовчуванням) поле `category_name` класу є назвою категорії послуг з благоустрою території типу `CharField` з довжиною 50 символів максимально.

Модель послуги з благоустрою територій є класом `Landscapeservice`.

Поле `landscape_category` класу є категорією послуги з благоустрою території типу `ForeignKey`.

Поле `service_name` класу є назвою послуги з благоустрою територій типу `CharField` з довжиною 50 символів максимально.

Поле `measurement` класу є базовою одиницею визначення кількості застосування послуги з благоустрою території (штуки, квадратні метри, кубічні метри) типу `CharField`, що відповідає значенню з відповідного набору кортежів, якими встановлюється відповідність символу позначення фактичному значенню.

Модель пропозиції послуги з благоустрою територій є класом `Landscapeproposition`.

Поле `landscape_service` класу є послугою з благоустрою території, за якою сформована пропозиція надавачем послуг, типу `ForeignKey`.

Поле `service_provider` класу є надавачем послуг, що визначив цю пропозицію, типу `ForeignKey`.

Поле `photo` класу є фотографією, на якій показано і яка буде ілюструвати пропозицію надавача послуг, має тип `ImageField`.

Поле `manager` класу є визначенням того, чи це акаунт сервісу або акаунт менеджера програми, має тип `BooleanField`.

Поле `minimal_price` класу є мінімальною вартістю, за яку цю послугу може виконувати цей надавач послуг, типу `DecimalField`.

Поле `maximum_price` класу є максимальною вартістю, за яку цю послугу може виконувати цей надавач послуг, типу `DecimalField`.

Поле `measurement_unit` класу є кількістю одиниць виміру, за яку встановлюється ця вартість, типу `SmallIntegerField`.

Модель надавачів послуг є класом `Serviceprovider`.

Поле `business_name` класу є назвою компанії, яка надає ці послуги з благоустрою території, типу `CharField` з довжиною 50 символів максимально.

Поле `presenter_surname` класу є прізвищем основного представника компанії або робітника типу `CharField` з довжиною 30 символів максимально.

Поле `presenter_name` класу є іменем основного представника компанії або робітника типу `CharField` з довжиною 50 символів максимально.

Поле `business_city` класу є містом, в якому надаються послуги компанією або робітником, типу `CharField` з довжиною 30 символів максимально.

Поле `profile_photo` класу є фотографією профілю надавача послуг з благоустрою території типу `ImageField`.

Поле `details` класу є описом можливостей або особливостей (або представлення) надавача послуг з благоустрою території типу `TextField`.

Поле `client_score` класу є оцінкою, на яку всередньому оцінили роботу надавача послуг з благоустрою території клієнти, типу `DecimalField`.

Модель мов, якими вміють спілкуватися представники надавача послуг, є класом `Serviceproviderlanguage`.

Поле `service_provider` класу є надавачем послуг, якого стосується ця інформація, типу `ForeignKey`.

Поле `language` класу є мовою, якою вміють спілкуватися представники надавача послуг, типу `CharField`.

Модель проєктів з благоустрою території (ландшафтного дизайну), якими визначається портфоліо надавача послуг, є класом `Portfolioproject`.

Поле `serviceprovider` класу є надавачем послуг, який виконав цей проєкт з благоустрою території, типу `ForeignKey`.

Поле `project_name` класу є назвою проєкту з благоустрою території типу `CharField` з довжиною 50 символів максимально.

Поле `details` класу є описом особливостей проєкту з благоустрою території типу `TextField`.

Модель фотографії проєктів з благоустрою території є класом `Portfoliophoto`.

Поле `portfolioproject` класу є проєктом з благоустрою території, який представляє ця фотографія типу `ForeignKey`.

Поле `photo` класу є фотографією, яка зображує результати виконання проєкту з благоустрою території, типу `ImageField`.

Поле `details` класу є описом особливостей проєкту з благоустрою території, які показані на фотографії, типу `CharField` з довжиною 100 символів максимально.

Модель клієнтів сервісу є класом `Client`.

Поле `profile_photo` класу є фотографією профілю клієнта сервісу типу `ImageField`.

Поле `username` класу є ім'ям клієнта сервісу як логіном типу `CharField` з довжиною 20 символів максимально.

Поле `client_name` класу є ім'ям клієнта сервісу як його фактичним ім'ям типу `CharField` з довжиною 100 символів максимально.

Поле `password` класу є паролем клієнта сервісу типу `CharField` з довжиною 20 символів максимально.

Поле `details` класу є описом для профілю клієнта сервісу типу `TextField`.

Поле `phone` класу є номером телефону, за яким здійснюється обслуговування клієнта сервісу, типу `CharField`.

Модель резервування послуг з благоустрою території є класом `Reservation`.

Поле `client` класу є клієнтом, який зарезервував ці послуги з благоустрою території, типу `ForeignKey`.

Поле `serviceprovider` класу є надавачем послуг з благоустрою території типу `ForeignKey`.

Поле `reserve_date` класу є часом, коли було створене резервування послуг з благоустрою території, типу `DateTimeField`.

Поле `assessed_price` класу є оцінюваною початковою вартістю даних послуг з благоустрою території типу `DecimalField`.

Поле `actual_price` класу є фактичною підсумковою вартістю даних послуг з благоустрою території типу `DecimalField`.

Поле `approved_price` класу є визначенням того, чи було підтверджено вказану фактичну підсумкову вартість другою стороною, чи дані визначені тільки за свідченням однієї зі сторін. Має тип `BooleanField`.

Поле `status` класу є станом резервування послуг з благоустрою території типу `SmallIntegerField`.

Поле `address` класу є адресою, за якою мають проводитися роботи з благоустрою території, типу `CharField` з довжиною 150 символів максимально.

Поле `proposed_start` класу є часом, коли пропонується розпочати виконання послуг з благоустрою території, типу `DateTimeField`.

Поле `proposed_duration` класу є часом, коли пропонується закінчити виконання послуг з благоустрою території, типу `DateTimeField`.

Поле `service_start` класу є узгодженим часом початку виконання робіт з благоустрою території типу `DateTimeField`.

Поле `service_end` класу є узгодженим часом завершення виконання робіт з благоустрою території типу `DateTimeField`.

Модель складу резервування послуг з благоустрою території є класом `Landscapepropositionprice`.

Поле `reservation` класу є резервуванням послуг з благоустрою території, склад якого визначається цією моделлю, має тип `ForeignKey`.

Поле `landscapeproposition` класу є пропозицією послуги з благоустрою територій, яка входить до складу резервування, має тип `ForeignKey`.

Поле `quantity` класу є кількістю замовлених одиниць відповідної пропозиції послуги з благоустрою територій типу `IntegerField`.

Поле `unit_price` класу є вартістю одиниці цієї послуги з благоустрою територій, що використана в цьому резервуванні, має тип `DecimalField`.

Поле `total_price` класу є сумарною вартістю цієї пропозиції послуги з благоустрою територій у складі резервування типу `DecimalField`.

Поле `details` класу є описом особливостей застосування цієї пропозиції послуги з благоустрою територій типу `CharField` з довжиною 100 символів максимально.

Модель часу роботи надавача послуг з благоустрою території є класом `Servicetime`.

Поле `service_provider` класу є надавачем послуг з благоустрою території, для якого визначено цей період роботи, має тип `ForeignKey`.

Поле `start_service` класу є часом початку роботи на відповідну дату даного надавача послуг з благоустрою території, має тип `DateTimeField`.

Поле `end_service` класу є часом завершення роботи на відповідну дату даного надавача послуг з благоустрою території, має тип `DateTimeField`.

Модель повторення резервування послуг з благоустрою території є класом `Reservationrepeat`.

Поле reservation класу є резервуванням послуг з благоустрою території, яке має використовуватися для повторення резервування, має тип ForeignKey.

Поле repeat_period_time класу є періодом повторення резервування типу SmallIntegerField.

Поле repeat_period_measurement класу є одиницею вимірювання періоду повторення резервування типу CharField.

Поле period_end класу є часом, до якого має відбуватися повторення резервування, має тип DateTimeField.

Модель уподобання надавачів послуг з благоустрою територій є класом Serviceproviderstar.

Поле serviceprovider класу є надавачем послуг з благоустрою територій, якого уподобано, має тип ForeignKey.

Поле client класу є клієнтом, який уподобав надавача послуг з благоустрою територій, має тип ForeignKey.

Поле star_time класу є часом, коли було виконано уподобання надавача послуг з благоустрою територій клієнтом, має тип DateTimeField.

Модель оцінювання надавачів послуг з благоустрою територій клієнтами є класом Serviceproviderscore.

Поле service_provider класу є надавачем послуг з благоустрою територій, якого було оцінено, має тип ForeignKey.

Поле client класу є клієнтом, який оцінив надавача послуг з благоустрою територій, має тип ForeignKey.

Поле score_time класу є часом, коли виконано оцінювання надавача послуг з благоустрою територій, має тип DateTimeField.

Поле score_integer класу є числовою оцінкою, виставленою надавачу послуг з благоустрою територій, має тип SmallIntegerField.

Поле score_text класу є текстом оцінювання надавача послуг з благоустрою територій типу TextField.

Поле fair_price класу є визначенням того, чи фактична підсумкова вартість виконання робіт відповідала інформації про вартість робіт з

благоустрою територій, яку для себе визначив цей надавач послуг, має тип BooleanField.

Модель акаунтів сервісів є класом Serviceprovideraccount.

Поле username класу є ім'ям акаунта сервісу як логіном типу CharField з довжиною 20 символів максимально.

Поле password класу є паролем акаунта сервісу типу CharField з довжиною 20 символів максимально.

Поле manager класу є визначенням того, чи цей акаунт представляє надавача послуг, чи менеджера програми, має тип BooleanField.

Поле service_provider класу є надавачем послуг з благоустрою територій, до якого відноситься акаунт, має тип ForeignKey.

Модель деталізації відповідності вартості окремих послуг з благоустрою території даним, які стосовно себе наводив у програмі надавач послуг, є класом Landscapepropositionprice.

Поле serviceprovider_score класу є оцінюванням надавачів послуг з благоустрою територій клієнтом, до якого відноситься деталізація, має тип ForeignKey.

Поле landscape_proposition класу є пропозицією послуг з благоустрою території, стосовно якої було визначено відповідність вартості даним, які стосовно цієї пропозиції наводив у програмі надавач послуг з благоустрою території, має тип ForeignKey.

Поле fair_price класу є визначенням того, чи вартість цієї послуги з благоустрою території, як вона була виконана надавачем послуг, відповідає обіцяному проміжку цін, визначеному в програмі, має тип BooleanField.

Поле price_time класу є часом, коли була визначена відповідність вартості окремих послуг з благоустрою території даним, які стосовно себе наводив у програмі надавач послуг, має тип DateTimeField.

Представлення пошуку надаваних послуг з благоустрою території реалізовано функцією land_service_search. Усі параметри в даному випадку передаються через сам запит, тобто адресу звернення, в параметрі land_city

якої може міститися місто, де працює надавач послуг, в параметрі `score_left` – найменша підходяща оцінка, виставлена надавачу послуг клієнтами, `score_right` – найбільша підходяща оцінка, виставлена надавачу послуг клієнтами, `price_left` – найменша підходяща вартість за одиницю виконання цієї послуги, `price_right` – найбільша підходяща вартість за одиницю виконання цієї послуги, `land_service` – ідентифікатор потрібної послуги, `lang` – мова, якою має комунікувати надавач послуг, `service_date` – потрібний час виконання обслуговування.

Представлення резервування послуг з благоустрою території реалізовано функцією `reservation_create`.

Представлення визначення повторюваного замовлення послуг з благоустрою території реалізовано функцією `reservation_repeat`. Ідентифікатор резервування визначається параметром `reservationId`. Саме повторення визначається формою. Перевіряється, чи є авторизований клієнт тим, який створив це резервування.

Представлення відображення загальної інформації про надавача послуг з благоустрою території реалізовано функцією `service_provider_info`, де ідентифікатор самого надавача послуг задається параметром функції `providerId`. У функції для заповнення даними шаблону вебсторінки витягаються дані об'єкта моделі надавача послуг за заданим ідентифікатором, визначаються мови, за якими комунікує надавач послуг, витягаються 5 останніх проєктів з портфолію, 10 останніх пропозицій послуг з благоустрою території, інформація про те (якщо клієнт авторизувався), чи він уподобав цього надавача послуг та чи він оцінював його роботу, чи може оцінювати його тепер. Оцінювання може відбуватися тоді, коли остання оцінка цим клієнтом цього надавача послуг відбулась раніше ніж останнє резервування цього клієнта у цього надавача послуг.

Представлення відображення всіх послуг з благоустрою території заданого надавача послуг реалізовано функцією `service_provider_propositions`, де ідентифікатор самого надавача послуг задається параметром функції

providerId. Формується перелік всіх пропозицій послуг з благоустрою території заданого надавача послуг з розподілом за категоріями та за послугами в категоріях, після чого за кожною послугою визначаються всі пропозиції цього надавача послуг. Визначається також інформація про те (якщо клієнт авторизувався), чи він уподобав цього надавача послуг та чи він оцінював його роботу, чи може оцінювати його тепер. Оцінювання може відбуватися тоді, коли остання оцінка цим клієнтом цього надавача послуг відбулась раніше ніж останнє резервування цього клієнта у цього надавача послуг.

Представлення відображення всіх виставлених оцінок надавачу послуг реалізовано функцією service_provider_info, де ідентифікатор самого надавача послуг задається параметром функції providerId. Усі оцінки впорядковуються за часом виставлення, формується перелік всіх пропозицій послуг з благоустрою території з розподілом за категоріями та послугами, визначаючи для кожної всі оцінювання коректності визначення вартості послуг.

Представлення відображення портфолію надавача послуг реалізовано функцією service_provider_portfolios, де ідентифікатор самого надавача послуг задається параметром функції providerId. Окрім формування переліку проєктів з фотографіями за портфолію також визначається інформація про те (якщо клієнт авторизувався), чи він уподобав цього надавача послуг та чи він оцінював його роботу і чи може оцінювати її тепер. Оцінювання може відбуватися тоді, коли остання оцінка цим клієнтом цього надавача послуг відбулась раніше ніж останнє резервування цього клієнта у цього надавача послуг.

Представлення оцінювання наданих послуг з благоустрою території реалізовано функцією service_provider_score. Оцінювання допускається тільки за наявності в клієнта резервування, з часу завершення якого не було виконано оцінювання того самого надавача послуг. До оцінювання також входить обробка форм з перевіркою відповідності вартості виконання послуг початково зазначеним на своїй сторінці надавачем послуг значенням.

Представлення відображення оцінювань клієнта реалізовано функцією `client_scores`. Логін клієнта задається параметром `userName`, однак це значення може бути і пропущене: у такому випадку використовується логін авторизованого акаунта. Тобто представлення опрацьовує перегляд і власних оцінювань, і заданого клієнта.

Представлення відображення інформації про клієнта реалізовано функцією `client_profile`. Логін клієнта задається параметром `userName`, однак це значення може бути і пропущене: у такому випадку використовується логін авторизованого акаунта. Тобто представлення опрацьовує перегляд і власного профілю, і заданого клієнта. Окрім самого профілю сторінка також заповнюється даними оцінювань клієнта.

Представлення відображення власних замовлених резервувань послуг з благоустрою території реалізовано функцією `my_reservations`. Параметр `sort` визначає спосіб роботи з власними замовленнями резервувань. Значення 0 – всі резервування, але впорядковані від останніх за часом завершення роботи, 1 – всі резервування, але впорядковані від останніх за часом резервування, 2 – всі резервування, але впорядковані за збільшенням вартості, 3 – всі резервування, але впорядковані за зменшенням вартості, 4 – архівні резервування, впорядковані від останнього, 5 – незавершені резервування від найближчого до поточної миті, 6 – тільки створені резервування, впорядковані від останнього за часом завершення, 7 – скасовані резервування або резервування з запропонованими змінами, 8 – підтверджені резервування, 9 – розпочаті резервування, 10 – закінчені резервування, 11 – провалені або оскаржені резервування.

Представлення уподобання надавача послуг з благоустрою території разом зі скасуванням уподобання надавача послуг з благоустрою території реалізовано функцією `star_service_provider`. Параметр `serviceproviderId` має задати ідентифікатор надавача послуг з благоустрою території. Якщо уподобання вже було виставлено і збережено в базі даних, то воно

видаляється, а якщо ж в базі даних саме такого уподобання (цим клієнтом цього надавача послуг) ще не було, то створюється.

Представлення відображення уподобаних надавачів послуг з благоустрою території реалізовано функцією `star_providers`. Якщо користувач не авторизувався, то даних не буде надано. Впорядкування за номером від останнього реалізується.

Представлення скасування власного резервування послуг з благоустрою території реалізовано функцією `cancel_landscape_reservation`. Ідентифікатор резервування визначається параметром `reservationId`. Скасування відбувається тільки в тому випадку, якщо резервування було створене клієнтом, який авторизувався, на даний момент воно є створеним, запропонованим до змін або підтвердженим, але тоді хоча би 1 з 2 останніх резервувань цього клієнта не було скасовано.

Представлення підтвердження резервування послуг з благоустрою території реалізовано функцією `approve_landscape_reservation`. Параметр `reservationId` задає ідентифікатор резервування, а за цим ідентифікатором перевіряється, що авторизований акаунт стосується надавача послуг, до якого адресовано це резервування, а також що резервування перебуває в придатному для цього стані, тобто створеного, попередньо зміненого.

Представлення додавання інформації про послугу з благоустрою території реалізовано функцією `landscapereproposition_create`.

Представлення визначення часу, доступного для виконання робіт, надавачем послуг реалізовано функціями `service_time_slot_create` для створення проміжку часу та `service_time_slot_change` для зміни проміжку часу.

Представлення відображення резервувань послуг з благоустрою території за заданим станом у надавача послуг реалізовано функцією `provider_reservations`. Надавач послуг визначається тим акаунтом, який авторизувався: для цього перевіряється, до якого надавача послуг належить цей акаунт. Параметр `status` функції задає стан необхідних резервувань. Параметр `sort` функції визначає порядок сортування.

Представлення відображення робочого часу надавача послуг за календарем реалізовано функцією `provider_calendar`. Надавач послуг визначається тим акаунтом, який авторизувався: для цього перевіряється, до якого надавача послуг належить цей акаунт. Проміжок часу має бути обмежений: початок – параметром `service_start`, кінець – `service_end`. Виділяються тільки підтверджені, розпочаті, закінчені, провалені та оскаржені резервування.

Представлення відображення фінансових результатів конкретної послуги з благоустрою території надавача послуг реалізовано функцією `service_provider_statistics`. Працювати з функцією може тільки представник надавача послуг. Статистика визначається на всіма пропозиціями послуг цього надавача послуг, об'єднуючи їх за категоріями та в категоріях за відповідними послугами. За кожною пропозицією визначається статистика за кожен місяць. У місяць входить сума за резервуваннями, які були закінченими і які виконувались у цей місяць. Статистика підводиться за останній рік (12 останніх місяців).

Представлення скасування резервування послуг з благоустрою території надавачем послуг реалізовано функцією `cancel_landscape_reservation_provider`. Ідентифікатор резервування визначається параметром `reservationId`. Скасування відбувається тільки в тому випадку, якщо резервування адресовано надавачу послуг, представник якого авторизувався.

Представлення пропонування зміни резервування послуг з благоустрою території надавачем послуг реалізовано функцією `reservation_change_proposition`. Ідентифікатор резервування визначається параметром `reservationId`.

Представлення редагування інформації про послугу з благоустрою території реалізовано функцією `landscapereproposition_change`.

Представлення пошуку резервувань послуг з благоустрою території реалізовано функцією `land_reservation_search`. Виконання забезпечується

тільки для менеджерів програми. Параметри пошуку включають логін клієнта, номер телефону клієнта, ім'я клієнта, назву надавача послуг, номер резервування або його частину, дату резервування, дату виконання обслуговування, стан.

Представлення перегляду резервування послуг з благоустрою території реалізовано функцією `land_reservation_view`. Ідентифікатор резервування визначається параметром `reservationId`.

Представлення редагування резервування послуг з благоустрою території реалізовано функцією `land_reservation_manager`. Параметр `reservationId` задає ідентифікатор резервування. Виконується тільки для менеджера.

Представлення визначення розпочатого обслуговування за резервуванням реалізовано функцією `start_landscape_reservation`. Параметр `reservationId` задає ідентифікатор резервування, а за цим ідентифікатором перевіряється, що авторизований акаунт стосується надавача послуг, до якого адресовано це резервування, а також що резервування перебуває в придатному для цього стані, тобто є підтвердженням.

Представлення визначення завершеного обслуговування за резервуванням реалізовано функцією `stop_landscape_reservation`. Параметр `reservationId` задає ідентифікатор резервування, а за цим ідентифікатором перевіряється, що авторизований акаунт стосується надавача послуг, до якого адресовано це резервування, а також що резервування перебуває в придатному для цього стані, тобто є розпочатим.

Представлення оскарження резервування послуг з благоустрою території реалізовано функцією `claim_landscape_reservation`. Параметр `reservationId` задає ідентифікатор резервування, а параметр `claimstatus` – очікуваний стан оскарження (або безпосередньо оскарження, або визначення того, що обслуговування не виконано з вини клієнта або надавача послуг. Відповідно за ідентифікатором резервування перевіряється, що авторизований акаунт стосується надавача послуг, до якого адресовано це резервування, або

є клієнтом, який створив це резервування. Виконується перевірка станів для відповідності послідовності, заданої діаграмою станів на рис. 3.1.

Створення нового надавача послуг з благоустрою території реалізовано функцією `service_provider_add`.

Редагування інформації про надавача послуг з благоустрою території реалізовано функцією `service_provider_change`.

Призначення представника надавача послуг реалізовано функцією `service_provider_account_add`. Параметр `sercideproviderId` задає ідентифікатор надавача послуг з благоустрою території, `spaccountName` – логін користувача, який буде працювати за цим надавачем послуг. Перед виконанням збереження реалізується валідація права користувача працювати менеджером програми. Менеджери мають ті самі акаунти, що і акаунти надавачів послуг, при цьому вони не пов'язані з конкретним надавачем послуг. При цьому менеджери мають відповідну позначку про таку роль. Встановити цю позначку може адміністратор, працюючи через АП додатку.

Підтримка виконання дій за датою (зокрема створення резервувань як повторень) відбувається на основі роботи скрипта, який запускається за розкладом кожен день.

3.3 Висновки за розділом 3

У розділі описано ті результати реалізації, які було отримано та які дозволили повністю реалізувати заплановані функції, включаючи як функції для надавачів послуг з благоустрою території, так і для клієнтів та менеджерів програми на основі визначення логіки створених програмних функцій для реалізації представлень та створених класів моделей.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми

Програма призначена для забезпечення підтримки послуг з благоустрою території через виконання резервування цих послуг у робітників (компаній) клієнтами.

Програма може використовуватися компаніями або окремими робітниками, які надають власні послуги з благоустрою території, для визначення послуг, що надаються, параметрів резервування цих послуг.

Програма може використовуватися клієнтами для пошуку потрібних послуг, отримання інформації про такі послуги та резервування виконання цих послуг для забезпечення благоустрою власної території.

4.2 Умови виконання програми

Умовами виконання програми для підтримки 50 запитів на секунду є наступні технічні засоби:

- процесор не менше Intel Xeon E5440 з частотою 2.83 ГГц;
- оперативна пам'ять – 12 Гб;
- жорсткий диск – 40 Гб простору.

Виконання програми відносно програмних вимог можливе тоді, коли використовується сервер зі встановленими інтерпретатором Python версії 3, засобами створення вебдодатків на основі Django, забезпеченим доступом до системи керування базами даних PostgreSQL з можливістю програмного створення бази даних засобами Django.

4.3 Експлуатація програми

Звернення до програми має результатом відкриття вебдодатку в режимі роботи клієнта. Цей режим виділяється доступним меню. Таке меню включає

пункти “Послуги”, звідки можна виконувати пошук послуг з благоустрою території, “Резервування”, звідки можна переглянути вже створені власні резервування, “Оцінювання”, звідки можна переглянути виконані клієнтом оцінювання надавачів послуг з благоустрою території, “Уподобання”, звідки можна переглянути вподобаних клієнтом надавачів послуг. Якщо клієнт не авторизувався, то останні 3 пункти не будуть містити результатів для відображення. Тоді потрібно обрати крайню справа зелену кнопку для виконання авторизації. З цієї сторінки можна також перейти до реєстрації або до авторизації представника надавача послуг. Після виконаної авторизації також у меню є пункт “Акаунт”, де доступні дані профілю клієнта.

З результатів пошуку можна відкрити сторінку надавача послуги (рис. 4.1), що зверху включає фотографію, назву компанії, місто розташування, підтримувані мови комунікації та середню виставлену оцінку.

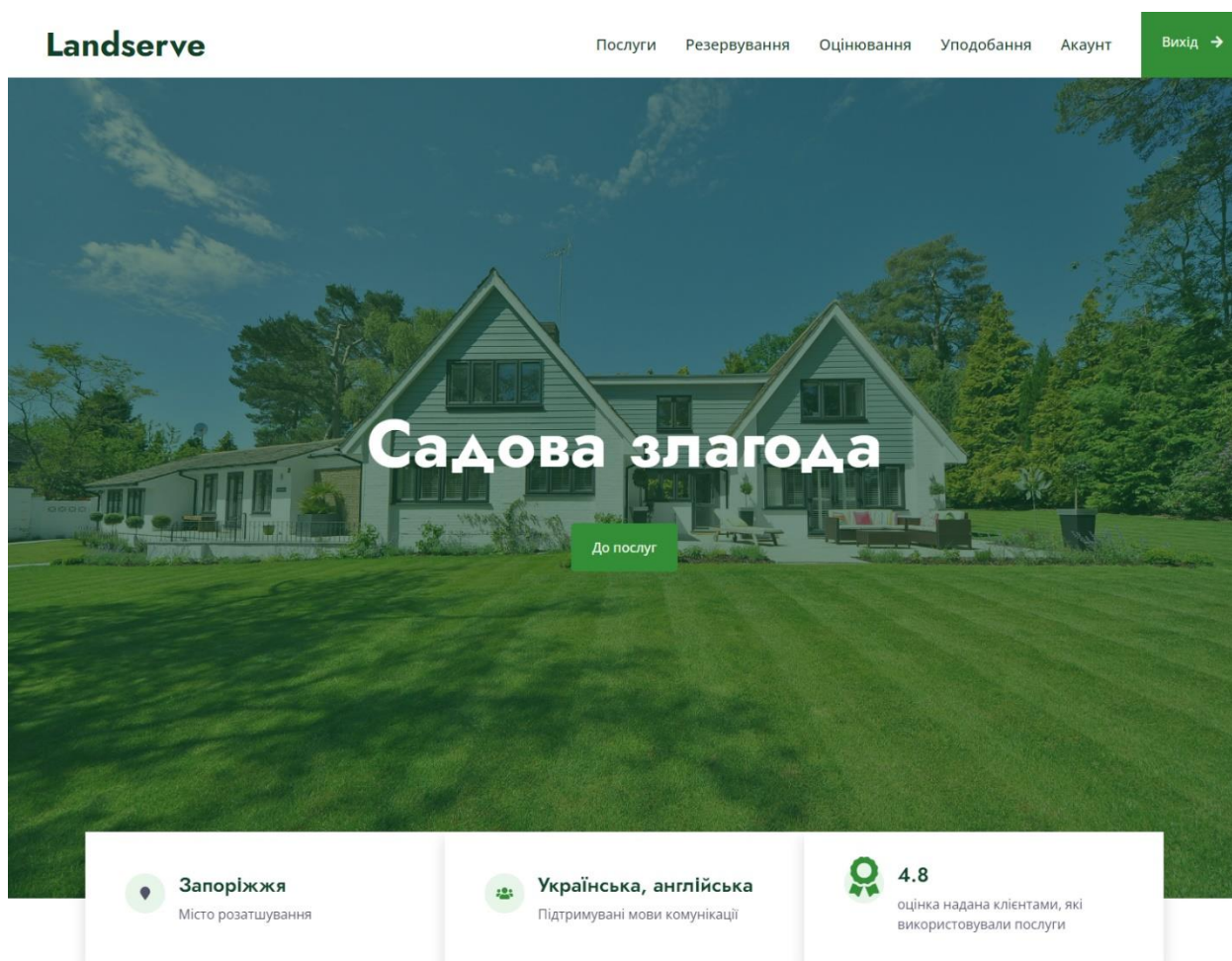


Рисунок 4.1 – Вебсторінка надавача послуг з благоустрою території

Також на цій сторінці знаходяться оцінювання, послуги компанії та портфоліо з виконаними проєктами. Відображаються тільки деякі, але можна переглянути повністю, натиснувши на відповідне посилання. Наприклад, сторінка зі всіма послугами з благоустрою території надавача послуг (рис. 4.2) дозволяє обрати категорію, потім конкретну послугу, а вже нижче фотографіями показані пропозиції. При наведенні на пропозицію демонструється її назва, очікування вартість та одиниця виміру. Також доступні кнопки для замовлення (одразу формується резервування з цієї послуги як тільки однієї або зі всіх, які раніше були додані до кошику) або для додавання до кошику.

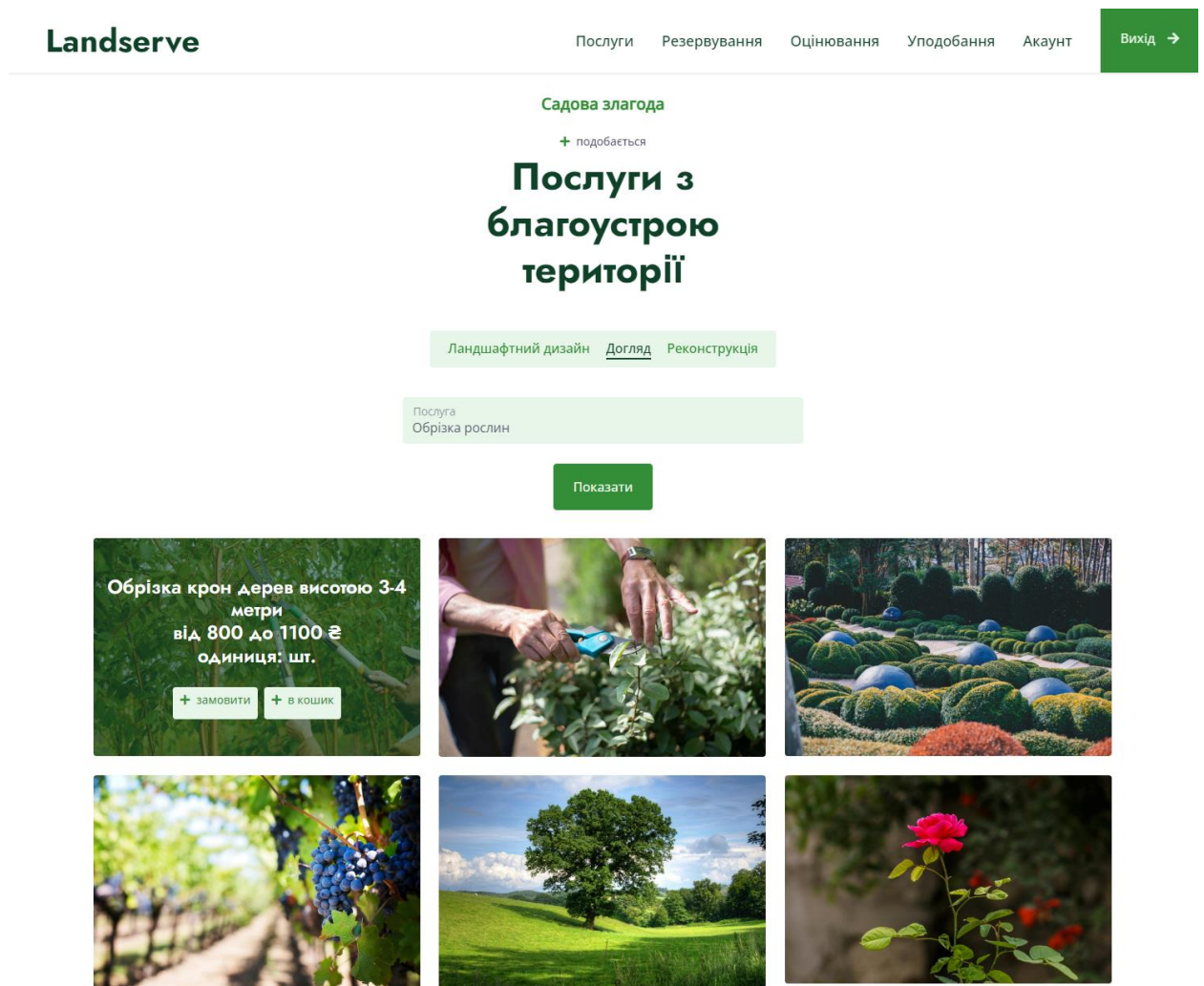


Рисунок 4.2 – Вебсторінка послуг надавача послуг з благоустрою території

Сторінка резервувань клієнта (рис. 4.3) містить можливість обрати тип відображення резервувань, тоді нижче виводяться всі відповідні резервування. За кожним резервуванням наводиться інформація та склад резервування. Між позиціями можна перемикатися стрілками. Для кожного резервування доступна дія. Конкретну дію потрібно обрати з випадаючого переліку, а тоді потрібно натиснути на кнопку “Підтвердити”.

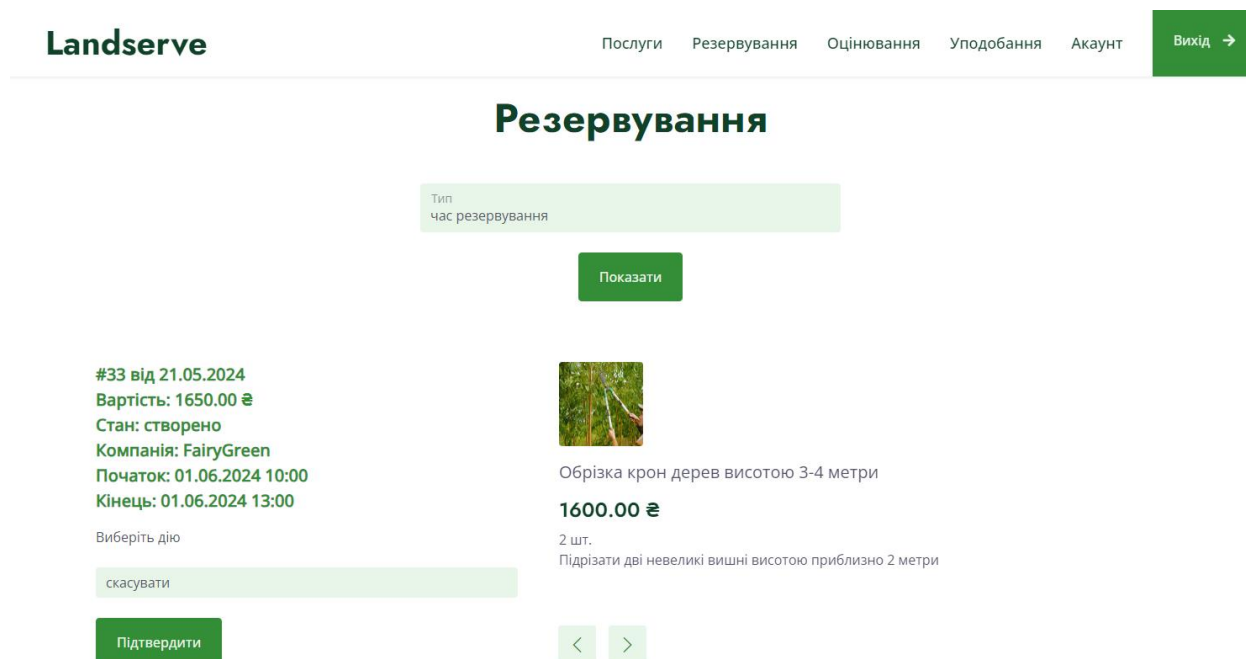


Рисунок 4.3 – Вебсторінка резервувань клієнта

Авторизація представника надавача послуги тягне за собою зміну меню на таку, що включає також перехід до пошуку послуг, а резервування вже стосуються резервувань не клієнта, а конкретного надавача послуг (рис. 4.4). За резервуваннями вже відображається інформація про клієнтів. Також доступний розділ меню “Компанія”, звідки можна перейти до інформації про надавача послуг, змінити у даному випадку за потреби ці дані, внести дані проєктів портфолію таким чином тощо. Також з цього розділу доступні уподобані надавачі послуг цим користувачем (адже йому доступні всі функції, що доступні і клієнту).

Якщо авторизувався менеджер, то його меню містить пункти для пошуку резервувань, пошуку послуг, призначення представників надавачів послуг.

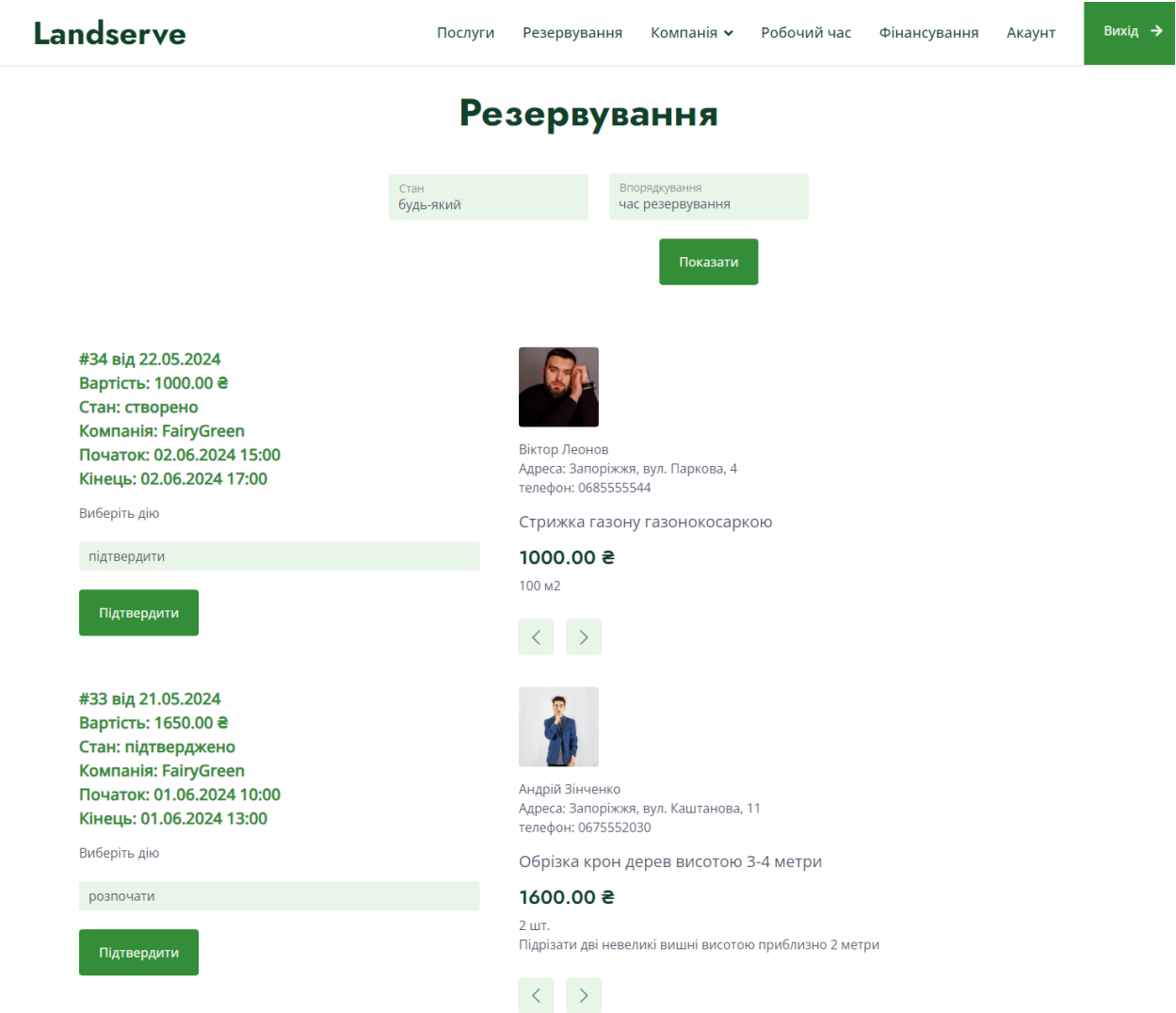


Рисунок 4.4 – Вебсторінка керування резервувань надавачем послуг з благоустрою територій

4.4 Тестування програми

Проведення тестування забезпечувалось тими сценаріями, які були визначені у розділі 2. Кожен сценарій опрацьовувався з перевіркою результатів. У разі коректних результатів у таблицю результатів тестування (табл. 4.1) вносились позначка про успішне закінчення.

Таблиця 4.1 – Результати тестування ПЗ резервування послуг з благоустрою території

Функціональна можливість	Результат
1	2
Пошук надаваних послуг з благоустрою території	Успішно
Резервування послуг з благоустрою території	Успішно
Визначення повторюваного замовлення послуг з благоустрою території	Успішно
Відображення загальної інформації про надавача послуг з благоустрою території	Успішно
Відображення всіх послуг з благоустрою території заданого надавача послуг	Успішно
Відображення всіх виставлених оцінок надавачу послуг	Успішно
Відображення портфолію надавача послуг	Успішно
Оцінювання наданих послуг з благоустрою території	Успішно
Відображення оцінювань клієнта	Успішно
Відображення інформації про клієнта	Успішно
Відображення власних замовлених резервувань послуг з благоустрою території	Успішно
Авторизація клієнта, надавача послуг, менеджера	Успішно
Уподобання надавача послуг з благоустрою території	Успішно
Відображення уподобаних надавачів послуг з благоустрою території	Успішно
Скасування власного резервування послуг з благоустрою території	Успішно
Скасування уподобання надавача послуг з благоустрою території	Успішно
Реєстрація клієнта, надавача послуг	Успішно

Продовження таблиці 4.1

1	2
Підтвердження резервування послуг з благоустрою території	Успішно
Додавання інформації про послугу з благоустрою території	Успішно
Визначення часу, доступного для виконання робіт, надавачем послуг	Успішно
Відображення резервувань послуг з благоустрою території за заданим станом	Успішно
Відображення робочого часу надавача послуг за календарем	Успішно
Відображення фінансових результатів конкретної послуги з благоустрою території надавача послуг	Успішно
Скасування резервування послуг з благоустрою території надавачем послуг	Успішно
Пропонування зміни резервування послуг з благоустрою території надавачем послуг	Успішно
Редагування інформації про послугу з благоустрою території	Успішно
Пошук резервувань послуг з благоустрою території	Успішно
Перегляд резервування послуг з благоустрою території	Успішно
Редагування резервування послуг з благоустрою території	Успішно
Створення нового надавача послуг з благоустрою території	Успішно
Редагування інформації про надавача послуг з благоустрою території	Успішно
Призначення представника надавача послуг	Успішно
Визначення завершеного обслуговування за резервуванням	Успішно
Визначення розпочатого обслуговування за резервуванням	Успішно
Оскарження резервування послуг з благоустрою території	Успішно

4.5 Висновки за розділом 4

ПЗ резервування послуг з благоустрою території було успішно протестовано стосовно виконання тих функціональних можливостей, які визначалися під час моделювання, та описано стосовно головних принципів роботи.

ВИСНОВКИ

У роботі розроблено ПЗ резервування послуг з благоустрою території для забезпечення керування взаємодією між надавачами послуг і клієнтами.

Виконаний огляд програм, які можуть використовуватися для резервування послуг з благоустрою території, охоплює цілий ряд додатків, включаючи сервіс надання відгуків Yelp, сервіс замовлення послуг Кабанчик, а також вебсайти окремих компаній з благоустрою території. За результатами аналізу можливостей цих рішень виділено особливості розробки, які окрім найважливіших функцій включають також врахування інформації від клієнта про вартість послуг, керування робочим часом компанії, що пов'язане з виконанням саме бронювання послуг, а не тільки формування замовлень як у програмах-аналогах.

Проектування програми виконано за результатами вибору мови Python та фреймворку Django. Діаграми прецедентів створені для надавача послуг, менеджера програми та клієнта для визначення відповідних функціональних вимог до програми. Реалізацію бази даних та роботи з базою даних забезпечено через створення бази даних на основі системи PostgreSQL, яка є проектом з відкритим вихідним кодом, надійною, має розширену підтримку захисту даних, відзначається підтримкою масштабованості.

Заплановані функції, включаючи як функції для надавачів послуг з благоустрою території, так і для клієнтів та менеджерів програми, повністю реалізовано і описано на основі визначення логіки створених програмних функцій для реалізації представлень та створених класів моделей.

ПЗ резервування послуг з благоустрою території було успішно протестовано стосовно виконання тих функціональних можливостей, які визначалися під час моделювання, та описано стосовно головних принципів роботи. Програма признається для забезпечення підтримки послуг з благоустрою території через виконання резервування цих послуг у робітників (компаній) клієнтами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Restaurants, Dentists, Bars, Beauty Salons, Doctors – Yelp [Electronic resource]. – Access mode : <https://www.yelp.com/>.
2. Kabanchik.ua – онлайн-сервіс замовлення послуг [Електрон. ресурс]. – Режим доступу : <https://kabanchik.ua/>.
3. Ландшафтний дизайн і авторські сади, обслуговування ділянок. Київ [Електрон. ресурс]. – Режим доступу : <https://staffage.ua/>.
4. Садовий центр Львів, інтернет магазин рослин ГалСад [Електрон. ресурс]. – Режим доступу : <https://galsad.ua/>.
5. What are the possibilities with Python? [Electronic resource]. – Access mode : https://medium.com/@k_a_r_a/what-are-the-possibilities-with-python-0ad56a675be6.
6. Dinder, M. Becoming an Enterprise Django Developer [Text] : Discover best practices, tooling, and solutions for writing and organizing Django applications in production / M. Dinder. – Birmingham : Packt Publishing, 2022. – 526 p.
7. Django Web Framework (Python) [Electronic resource]. – Access mode : <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>.
8. Pyramid Documentation [Electronic resource]. – Access mode : <https://trypyramid.com/documentation.html>.
9. The Pyramid Web Framework [Electronic resource]. – Access mode : <https://docs.pylonsproject.org/projects/pyramid/en/latest/>.
10. Gardener – Free Responsive Bootstrap 5 Gardening Website Template [Electronic resource]. – Access mode : <https://themewagon.com/themes/free-responsive-bootstrap-5-gardening-website-template-gardener/>.

ДОДАТОК А
Технічне завдання

Вступ

Послуги з благоустрою території є доволі специфічними, оскільки вони з одного боку можуть бути пов'язані з виконанням проєктів ландшафтного дизайну, що є фактично індивідуальними, а з іншого боку з конкретними роботами, що є більш типовими (наприклад, прибирання території, підрізання дерев тощо). Тому найчастіше є певний розподіл між цими послугами. У одних випадках друга група послуг розглядається як додаткова та не дуже афішується компаніями. В інших випадках навпаки надається інформація про типові послуги, а послуги з ландшафтного дизайну фактично є певною додатковою позицією, на якій не акцентується увага. У першому випадку тоді кожна компанія створює свій власний вебсайт, який більшою мірою містить портфоліо і є деяким сайтом-візитівкою. Таким чином інформація про послуги з благоустрою території є фрагментованою, а у одному додатку зручно знайти потрібну інформацію стає важко. На подолання даної проблеми має бути в першу чергу націлена дана розробка. Інформація про вартість послуг, надання можливості резервування послуг є тими можливостями, які особливо уважно необхідно врахувати в розробці.

A.1 Підстави для розробки

Наказом № 168 від 24 квітня 2024 р. за Національним університетом «Запорізька політехніка» визначено тему «Програмне забезпечення резервування послуг з благоустрою території» для виконання дипломної кваліфікаційної роботи.

A.2 Призначення розробки

Розробка програми призначена для забезпечення підтримки послуг з благоустрою території через виконання резервування цих послуг у робітників

(компаній) клієнтами.

Програма може використовуватися компаніями або окремими робітниками, які надають власні послуги з благоустрою території, для визначення послуг, що надаються, параметрів резервування цих послуг.

Програма може використовуватися клієнтами для пошуку потрібних послуг, отримання інформації про такі послуги та резервування виконання цих послуг для забезпечення благоустрою власної території.

А.3 Основні вимоги до програми

А.3.1 Вимоги до функціональних характеристик

Функціональні характеристики програми резервування послуг з благоустрою території повинні включати:

- пошук надаваних послуг з благоустрою території;
- резервування послуг з благоустрою території;
- підтвердження або скасування резервування послуг з благоустрою території;
- пропонування зміни резервування послуг з благоустрою території надавачем послуг;
- визначення повторюваного замовлення послуг з благоустрою території;
- додавання і редагування інформації про послугу з благоустрою території надавачем послуг;
- відображення загальної інформації про надавача послуг з благоустрою території;
- відображення всіх послуг з благоустрою території заданого надавача послуг;
- відображення всіх виставлених оцінок надавачу послуг;
- відображення портфолію надавача послуг;
- визначення часу, доступного для виконання робіт, надавачем послуг;

- оцінювання наданих послуг з благоустрою території клієнтом з визначенням точності дотримання вартості;
- відображення оцінювань клієнта;
- відображення інформації про клієнта;
- відображення резервувань послуг з благоустрою території надавача послуг за заданим станом;
- відображення робочого часу надавача послуг за календарем;
- відображення замовлених резервувань послуг з благоустрою території клієнтом;
- авторизація і реєстрація;
- відображення фінансових результатів конкретної послуги з благоустрою території надавача послуг;
- уподобання надавача послуг з благоустрою території;
- відображення уподобаних надавачів послуг з благоустрою території;
- скасування уподобання надавача послуг з благоустрою території.

A.3.2 Умови експлуатації

Експлуатація можлива тільки за використання сервера, на якому можливо встановити і в подальшому встановлено засоби, які включають інтерпретатор Python версії 3, засоби створення вебдодатків на основі Django, забезпечено доступ до системи керування базами даних PostgreSQL з можливістю програмного створення бази даних засобами Django.

A.3.3 Вимоги до складу та параметрів технічних засобів

Технічні засоби визначені на рівні, що має забезпечувати приблизно 50 запитів на секунду:

- процесор не менше Intel Xeon E5440 з частотою 2.83 ГГц;
- оперативна пам'ять – 12 Гб;

– жорсткий диск – 40 Гб простору.

A.4 Порядок контролю та приймання

Виконання дипломної кваліфікаційної роботи бакалавра має бути закінчено у восьми тижневий термін з розподіленням за етапами, що визначено завданням на роботу у відповідному календарному плані. Після виконання даного плану реалізується процедура захисту роботи. Захист відбувається перед екзаменаційною комісією.

ДОДАТОК Б
Текст програми

```

from django.shortcuts import render
from .models import Serviceprovider
from .models import Portfolioproject
from .models import Portfoliophoto
from .models import Landscapeproposition
from .models import Client
from .models import Serviceproviderstar
from .models import Serviceproviderscore
from .models import Landscapepropositionprice
from .models import Reservation
from .models import Reservationpoint
from .models import Serviceprovideraccount
from django.shortcuts import redirect

def service_provider_info(request, providerId):
    providerdictionary = {}
    service_provider = Serviceprovider.objects.get(id =
providerId)
    providerdictionary['provider'] = service_provider
    languages =
Serviceproviderlanguage.objects.filter(service_provider =
service_provider)
    providerdictionary['languages'] = languages
    last_portfolio_projects =
Portfolioproject.objects.filter(serviceprovider =
service_provider).order_by('-id')[:5]
    for portfolio in last_portfolio_projects:
        portfolio.photo =
Portfoliophoto.objects.filter(portfolioproject =
portfolio).order_by('-id')[0]
        providerdictionary['portfolio'] = last_portfolio_projects
        propositions =
Landscapeproposition.objects.filter(service_provider =
service_provider).select_related('landscape_service').select_related('
landscape_service__landscape_category').order_by('-id')[:10]
        providerdictionary['propositions'] = propositions
        stared = False
        scored = False
        if request.user.is_active:
            client_authorized = Client.objects.get(username =
request.user.username)
            star_result =
Serviceproviderstar.objects.filter(serviceprovider = service_provider,
client = client_authorized)
            if star_result:
                stared = True

```

```

import datetime
current_reserve = datetime.datetime.now()
last_reservation =
Reservation.objects.filter(serviceprovider = service_provider, client
= client_authorized, service_end__lte = current_reserve).order_by('-
service_end')[0]
last_score =
Serviceproviderscore.objects.filter(service_provider =
service_provider, client = client_authorized, score_time__gte =
last_reservation.service_end)
if not last_score:
    scored = True
    providerdictionary['stared'] = stared
    providerdictionary['scored'] = scored
    return render(request, "landserve/serviceprovider.html",
providerdictionary)

def service_provider_scores(request, providerId):
    providerdictionary = {}
    service_provider = Serviceprovider.objects.get(id =
providerId)
    providerdictionary['provider'] = service_provider
    scores =
Serviceproviderscore.objects.filter(service_provider =
service_provider).order_by('-score_time')
    providerdictionary['scores'] = scores
    propositions =
Landscapeproposition.objects.filter(service_provider =
service_provider).select_related('landscape_service').select_related('
landscape_service__landscape_category')
    landscape_categories_short = []
    landscape_categories_services = {}
    landscape_categories = []
    for propos in propositions:
        propos.scoreprices =
Landscapepropositionprice.objects.filter(landscape_proposition =
propos).order_by('-price_time')
        if propos.landscape_service.landscape_category not in
landscape_categories_short:
            add_category =
propos.landscape_service.landscape_category
            add_category.services = []
            add_service = propos.landscape_service
            add_service.propositions = []
            add_service.propositions.append(propos)
            add_category.services.append(add_service)
            landscape_categories.append(add_category)

```

```

landscape_categories_short.append(propos.landscape_service.landscape_c
ategory)

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id] = []

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id].append(propos.landscape_service)
    else:
        if propos.landscape_service not in
landscape_categories_services[propos.landscape_service.landscape_categ
ory.id]:
            add_service = propos.landscape_service
            add_service.propositions = []
            add_service.propositions.append(propos)

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id].append(propos.landscape_service)
    else:
        services =
landscape_categories_services[propos.landscape_service.landscape_categ
ory.id]
        for landscape_service in services:
            if landscape_service.id ==
propos.landscape_service.id:

landscape_service.propositions.append(propos)
        sort_categories = landscape_categories.sort(key = lambda x:
x.category_name)
        for category in sort_categories:
            category.services = category.services.sort(key = lambda
x: x.service_name)
        providerdictionary['categories'] = sort_categories
        return render(request, "landserve/providerscores.html",
providerdictionary)

def service_provider_propositions(request, providerId):
    providerdictionary = {}
    service_provider = Serviceprovider.objects.get(id =
providerId)
    providerdictionary['provider'] = service_provider
    propositions =
Landscapeproposition.objects.filter(service_provider =
service_provider).select_related('landscape_service').select_related('
landscape_service__landscape_category')
    landscape_categories_short = []

```

```

        landscape_categories_services = {}
        landscape_categories = []
        for propos in propositions:
            if propos.landscape_service.landscape_category not in
landscape_categories_short:
                add_category =
propos.landscape_service.landscape_category
                add_category.services = []
                add_service = propos.landscape_service
                add_service.propositions = []
                add_service.propositions.append(propos)
                add_category.services.append(add_service)
                landscape_categories.append(add_category)

landscape_categories_short.append(propos.landscape_service.landscape_c
ategory)

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id] = []

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id].append(propos.landscape_service)
            else:
                if propos.landscape_service not in
landscape_categories_services[propos.landscape_service.landscape_categ
ory.id]:
                    add_service = propos.landscape_service
                    add_service.propositions = []
                    add_service.propositions.append(propos)

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id].append(propos.landscape_service)
                else:
                    services =
landscape_categories_services[propos.landscape_service.landscape_categ
ory.id]

                    for landscape_service in services:
                        if landscape_service.id ==
propos.landscape_service.id:

landscape_service.propositions.append(propos)
                        sort_categories = landscape_categories.sort(key = lambda x:
x.category_name)
                        for category in sort_categories:
                            category.services = category.services.sort(key = lambda
x: x.service_name)
                        providerdictionary['categories'] = sort_categories

```

```

        stared = False
        scored = False
        if request.user.is_active:
            client_authorized = Client.objects.get(username =
request.user.username)
            star_result =
Serviceproviderstar.objects.filter(serviceprovider = service_provider,
client = client_authorized)
            if star_result:
                stared = True
                import datetime
                current_reserve = datetime.datetime.now()
                last_reservation =
Reservation.objects.filter(serviceprovider = service_provider, client
= client_authorized, service_end__lte = current_reserve).order_by('-
service_end')[0]
                last_score =
Serviceproviderscore.objects.filter(service_provider =
service_provider, client = client_authorized, score_time__gte =
last_reservation.service_end)
                if not last_score:
                    scored = True
                providerdictionary['stared'] = stared
                providerdictionary['scored'] = scored
                return render(request,
"landserve/providerpropositions.html", providerdictionary)

def service_provider_portfolios(request, providerId):
    providerdictionary = {}
    service_provider = Serviceprovider.objects.get(id =
providerId)
    providerdictionary['provider'] = service_provider
    portfolio_projects =
Portfoliopproject.objects.filter(serviceprovider =
service_provider).order_by('-id')
    for portfolio in portfolio_projects:
        portfolio.photo =
Portfoliophoto.objects.filter(portfoliopproject =
portfolio).order_by('-id')
    providerdictionary['portfolio'] = portfolio_projects
    stared = False
    scored = False
    if request.user.is_active:
        client_authorized = Client.objects.get(username =
request.user.username)

```

```

        star_result =
Serviceproviderstar.objects.filter(serviceprovider = service_provider,
client = client_authorized)
        if star_result:
            stared = True
            import datetime
            current_reserve = datetime.datetime.now()
            last_reservation =
Reservation.objects.filter(serviceprovider = service_provider, client
= client_authorized, service_end__lte = current_reserve).order_by('-
service_end')[0]
            last_score =
Serviceproviderscore.objects.filter(service_provider =
service_provider, client = client_authorized, score_time__gte =
last_reservation.service_end)
            if not last_score:
                scored = True
            providerdictionary['stared'] = stared
            providerdictionary['scored'] = scored
            return render(request, "landserve/portfolios.html",
providerdictionary)

def star_providers(request):
    providerdictionary = {}
    if request.user.is_active:
        client_authorized = Client.objects.get(username =
request.user.username)
        star_results =
Serviceproviderstar.objects.filter(serviceprovider = service_provider,
client = client_authorized).select_related('serviceprovider')
        providerdictionary['stars'] = star_results
        scored = False
        import datetime
        current_reserve = datetime.datetime.now()
        last_reservation =
Reservation.objects.filter(serviceprovider = service_provider, client
= client_authorized, service_end__lte = current_reserve).order_by('-
service_end')[0]
        last_score =
Serviceproviderscore.objects.filter(service_provider =
service_provider, client = client_authorized, score_time__gte =
last_reservation.service_end)
        if not last_score:
            scored = True
        providerdictionary['scored'] = scored
        return render(request, "landserve/starproviders.html",
providerdictionary)

```

```

def client_scores(request, userName = None):
    providerdictionary = {}
    client_authorized = None
    if userName:
        client = Client.objects.get(username =
request.user.username)
    elif not userName and request.user.is_active:
        client_authorized = Client.objects.get(username =
request.user.username)
    else:
        return render(request, "landserve/clientscores.html",
providerdictionary)
    if not client and client_authorized:
        client = client_authorized
    client_scores = Serviceproviderscore.objects.filter(client =
client).order_by('-score_time').select_related('service_provider')
    if client_authorized:
        for client_score in client_scores:
            client_score.ownscore =
Serviceproviderscore.objects.filter(service_provider =
client_score.service_provider, client = client_authorized)
        providerdictionary['scores'] = client_scores[:5]
        providerdictionary['client'] = client
        return render(request, "landserve/client_scores.html",
providerdictionary)

def client_profile(request, userName = None):
    providerdictionary = {}
    client_authorized = None
    if userName:
        client = Client.objects.get(username =
request.user.username)
    elif not userName and request.user.is_active:
        client_authorized = Client.objects.get(username =
request.user.username)
    else:
        return render(request, "landserve/clientscores.html",
providerdictionary)
    if not client and client_authorized:
        client = client_authorized
    client_scores = Serviceproviderscore.objects.filter(client =
client).order_by('-
score_time').select_related('service_provider').prefetch_related('serv
ice_provider')
    if client_authorized:
        for client_score in client_scores:

```

```

        client_score.ownscore =
Serviceproviderscore.objects.filter(service_provider =
client_score.service_provider, client = client_authorized)
        providerdictionary['scores'] = client_scores
        providerdictionary['client'] = client
        return render(request, "landserve/client.html",
providerdictionary)

def my_reservations(request, sort = 0):
    providerdictionary = {}
    client_authorized = None
    if request.user.is_active:
        client_authorized = Client.objects.get(username =
request.user.username)
        if sort == 0:
            client_reservations =
Reservation.objects.filter(client = client_authorized).order_by('-
service_end')
        elif sort == 1:
            client_reservations =
Reservation.objects.filter(client = client_authorized).order_by('-
reserve_date')
        elif sort == 2:
            client_reservations =
Reservation.objects.filter(client =
client_authorized).order_by('approved_price')
        elif sort == 3:
            client_reservations =
Reservation.objects.filter(client = client_authorized).order_by('-
approved_price')
        elif sort == 4:
            import datetime
            current_reserve = datetime.datetime.now()
            client_reservations =
Reservation.objects.filter(client = client_authorized,
service_end__lte = current_reserve).order_by('-service_end')
        elif sort == 5:
            import datetime
            current_reserve = datetime.datetime.now()
            client_reservations =
Reservation.objects.filter(client = client_authorized,
service_end__gte = current_reserve).order_by('service_end')
        elif sort == 6:
            client_reservations =
Reservation.objects.filter(client = client_authorized, status =
0).order_by('-service_end')
        elif sort == 7:

```

```

        client_reservations =
Reservation.objects.filter(client = client_authorized, status__in =
[1, 3]).order_by('-service_end')
        elif sort == 8:
            client_reservations =
Reservation.objects.filter(client = client_authorized, status =
2).order_by('-service_end')
        elif sort == 9:
            client_reservations =
Reservation.objects.filter(client = client_authorized, status =
4).order_by('-service_end')
        elif sort == 10:
            client_reservations =
Reservation.objects.filter(client = client_authorized, status =
5).order_by('-service_end')
        elif sort == 11:
            client_reservations =
Reservation.objects.filter(client = client_authorized, status__in =
[6,7,8]).order_by('-service_end')
            for client_reservation in client_reservations:
                client_reservation.squad =
Reservationpoint.objects.filter(reservation =
client_reservation).select_related('landscapeproposition').order_by('i
d')
                providerdictionary['reservations'] = client_reservations
            return render(request, "landserve/myreserves.html",
providerdictionary)

def provider_reservations(request, status = -1, sort = 0):
    providerdictionary = {}
    provider_authorized = None
    if request.user.is_active:
        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        if status == -1:
            provider_reservations =
Reservation.objects.filter(client = client_authorized)
        else:
            provider_reservations =
Reservation.objects.filter(client = client_authorized, status =
status)
        if sort == 0:
            provider_reservations =
provider_reservations.order_by('-service_end')
        elif sort == 1:

```

```

        provider_reservations =
provider_reservations.order_by('-reserve_date')
        elif sort == 2:
            provider_reservations =
provider_reservations.order_by('approved_price')
        elif sort == 3:
            provider_reservations =
provider_reservations.order_by('-approved_price')
        elif sort == 4:
            provider_reservations =
provider_reservations.order_by('-proposed_start')
            for provider_reservation in provider_reservations:
                provider_reservation.squad =
Reservationpoint.objects.filter(reservation =
client_reservation).select_related('landscapeproposition').order_by('i
d')
                providerdictionary['reservations'] =
provider_reservations
            return render(request, "landserve/providerreserves.html",
providerdictionary)

def provider_calendar(request, service_start = None, service_end
= None):
    providerdictionary = {}
    provider_authorized = None
    if request.user.is_active:
        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        import datetime
        if not service_start:
            service_start = datetime.datetime.now()
        if not service_end:
            service_end = service_start + datetime.timedelta(days =
30)
        provider_times = Servicetime.objects.filter(service_provider
= provider_authorized.service_provider, start_service__gte =
service_start, start_service__lte =
service_end).order_by('start_service')
        for provider_time in provider_times:
            provider_time_reservations =
Reservation.objects.filter(service_start__lte =
provider_time.start_service, service_end__gte =
provider_time.start_service, status__in = [2, 4, 5, 6, 7, 8])
            provider_time_reservations =
provider_time_reservations.union(Reservation.objects.filter(service_st
art__gte = provider_time.start_service, service_start__lte =

```

```

provider_time.end_service, service_end__gte =
provider_time.end_service, status__in = [2, 4, 5, 6, 7, 8]))
    provider_time_reservations =
provider_time_reservations.union(Reservation.objects.filter(service_start__gte = provider_time.start_service, service_end__lte =
provider_time.end_service, status__in = [2, 4, 5, 6, 7, 8]))
    provider_time_reservations =
provider_time_reservations.order_by('service_start')
    for provider_reservation in provider_time_reservations:
        provider_reservation.squad =
Reservationpoint.objects.filter(reservation =
client_reservation).select_related('landscapeproposition').order_by('id')
        provider_time.reservations = provider_time_reservations
    return render(request, "landserve/calendar.html",
providerdictionary)

```

```

def cancel_landscape_reservation(request, reservationId):
    client_authorized = None
    if request.user.is_active:
        client_authorized = Client.objects.filter(username =
request.user.username)
        if client_authorized:
            client_authorized = client_authorized[0]
            reservation_for_cancel =
Reservation.objects.filter(client = client_authorized, id =
reservationId)
            if reservation_for_cancel:
                reservation_for_cancel =
reservation_for_cancel[0]
                import datetime
                reserve_time = datetime.datetime.now() +
datetime.timedelta(days = 1)
                last_land_reservations =
Reservation.objects.filter(client = client_authorized).order_by('-
reserve_date')[:2]
                if reservation_for_cancel.status == 0 or
reservation_for_cancel.status == 3 or (reservation_for_cancel.status
== 2 and reservation_for_cancel.service_start >= reserve_time and
(last_land_reservations[0].status != 1 or
last_land_reservations[1].status != 1)):
                    reservation_for_cancel.status = 1
                    reservation_for_cancel.save()
                    return redirect('ownreservationsview')
            else:

```

```

        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        reservation_for_cancel =
Reservation.objects.filter(serviceprovider =
provider_authorized.service_provider, id = reservationId)
        if reservation_for_cancel:
            reservation_for_cancel =
reservation_for_cancel[0]
            if reservation_for_cancel.status in [0, 2, 3]:
                reservation_for_cancel.status = 1
                reservation_for_cancel.save()
                return redirect('providerreservationsview')

def approve_landscape_reservation(request, reservationId):
    if request.user.is_active:
        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        reservation_for_approve =
Reservation.objects.filter(serviceprovider =
provider_authorized.service_provider, id = reservationId)
        if reservation_for_approve:
            reservation_for_approve = reservation_for_approve[0]
            if reservation_for_approve.status in [0, 1, 3]:
                reservation_for_approve.status = 2
                reservation_for_approve.save()
                return redirect('providerreservationsview')

def start_landscape_reservation(request, reservationId):
    if request.user.is_active:
        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        reservation_for_start =
Reservation.objects.filter(serviceprovider =
provider_authorized.service_provider, id = reservationId)
        if reservation_for_start:
            reservation_for_start = reservation_for_start[0]
            if reservation_for_start.status == 2:
                reservation_for_start.status = 4
                reservation_for_start.save()
                return redirect('providerreservationsview')

def stop_landscape_reservation(request, reservationId):
    if request.user.is_active:

```

```

        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        reservation_for_stop =
Reservation.objects.filter(serviceprovider =
provider_authorized.service_provider, id = reservationId)
        if reservation_for_stop:
            reservation_for_stop = reservation_for_stop[0]
            if reservation_for_stop.status == 4:
                reservation_for_stop.status = 5
                reservation_for_stop.save()
            return redirect('providerreservationsview')

    def claim_landscape_reservation(request, reservationId,
claimstatus):
        if request.user.is_active and claimstatus in [6,7,8]:
            provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
            reservation_for_claim =
Reservation.objects.filter(serviceprovider =
provider_authorized.service_provider, id = reservationId)
            if reservation_for_claim:
                reservation_for_claim = reservation_for_claim[0]
                if (reservation_for_claim.status == 6 or
reservation_for_claim.status == 7) and (claimstatus <= 8 and
claimstatus >= 6):
                    reservation_for_claim.status = 8
                    reservation_for_claim.save()
                    return redirect('providerreservationsview')
                elif reservation_for_claim.status < 6 and
(claimstatus >= 6 and claimstatus <= 8):
                    reservation_for_claim.status = claimstatus
                    reservation_for_claim.save()
                    return redirect('providerreservationsview')
            client_authorized = Client.objects.filter(username =
request.user.username)
            if client_authorized:
                client_authorized = client_authorized[0]
                reservation_for_claim =
Reservation.objects.filter(client = client_authorized, id =
reservationId)
                if reservation_for_claim
                    reservation_for_claim = reservation_for_claim[0]
                    if (reservation_for_claim.status == 6 or
reservation_for_claim.status == 7) and (claimstatus >= 6 and
claimstatus <= 8):

```

```

        reservation_for_claim.status = 8
        reservation_for_claim.save()
        return redirect('ownreservationsview')
    elif reservation_for_claim.status < 6 and
    (claimstatus >= 6 and claimstatus <= 8):
        reservation_for_claim.status = claimstatus
        reservation_for_claim.save()
        return redirect('ownreservationsview')

    def star_service_provider(request, serviceproviderId):
        service_provider = Serviceprovider.objects.get(id =
serviceproviderId)
        if request.user.is_active:
            client_authorized = Client.objects.get(username =
request.user.username)
            stared =
Serviceproviderstar.objects.filter(serviceprovider = service_provider,
client = client_authorized)
            if stared:
                stared.delete()
                return redirect('starserviceproviders')
            else:
                import datetime
                now = datetime.datetime.now()
                stared = Serviceproviderstar(serviceprovider =
service_provider, client = client_authorized, star_time = now)
                stared.save()
                return redirect('starserviceproviders')

    def land_service_search(request):
        providerdictionary = {}
        land_services =
Landscapeproposition.objects.all().select_related('landscape_service')
.select_related('landscape_service__landscape_category').select_relate
d('service_provider')
        business_city = self.request.GET.get('land_city', None)
        if business_city:
            land_services =
land_services.filter(service_provider__business_city = business_city)
            client_score_left = self.request.GET.get('score_left', 0)
            if client_score_left > 0:
                land_services =
land_services.filter(service_provider__client_score__gte =
client_score_left)
            client_score_right = self.request.GET.get('score_right', 5)
            if client_score_right < 5:

```

```

        land_services =
land_services.filter(service_provider__client_score__lte =
client_score_right)
        service_price_left = self.request.GET.get('price_left',
None)
        if service_price_left:
            land_services = land_services.filter(minimal_price__gte
= service_price_left)
            service_price_right = self.request.GET.get('price_right',
None)
            if service_price_right:
                land_services = land_services.filter(minimal_price__lte
= service_price_right)
            land_categories = Landscapecategory.objects.all()
            providerdictionary['categories'] = land_categories
            possible_services =
Landscapeservice.objects.all().select_related('landscape_category')
            providerdictionary['services'] = possible_services
            land_category = self.request.GET.get('land_category', None)
            if land_category:
                land_services =
land_services.filter(landscape_service__landscape_category__category_n
ame = land_category)
                land_service = self.request.GET.get('land_service', None)
                if land_service:
                    land_services =
land_services.filter(landscape_service__service_name =
int(land_service))
                    service_language = self.request.GET.get('lang', None)
                    land_services_return = []
                    service_date = self.request.GET.get('service_date', None)
                    import datetime
                    service_date = datetime.datetime.strptime(service_date,
'%d.%m.%y')
                    for land_service in land_services:
                        ok = False
                        if service_language:
                            serv_lang =
Serviceproviderlanguage.objects.filter(service_provider =
land_service, language = service_language)
                            if serv_lang:
                                ok = True
                        else:
                            ok = True
                    if service_date:

```

```

        service_time =
Servicetime.objects.filter(service_provider = land_service,
start_service__lte = service_date, end_service__gte = service_date)
        if not service_time:
            ok = False
        else:
            service_time = service_time[0]
            reservation_service_time =
Reservation.objects.filter(serviceprovider = land_service,
proposed_start__lte = service_date, proposed_duration__gte =
service_date)

            if reservation_service_time:
                ok = False

        if ok:
            land_services_return.append(land_service)
            providerdictionary['result_services'] = land_services_return
            return render(request, "landserve/servicesearch.html",
providerdictionary)

from .forms import LandscapepropositionForm
from .forms import ServiceproviderForm
from .forms import ServicetimeForm

def landscapeproposition_create(request):
    if request.method == "POST":
        land_proposition_form =
LandscapepropositionForm(request.POST)
        if land_proposition_form.is_valid() and
request.user.is_active:
            land_proposition =
land_proposition_form.save(commit=False)
            provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
            land_proposition.landscape_service =
provider_authorized.service_provider
            land_proposition.save()
            return redirect(request.path_info)
        else:
            land_proposition_form = LandscapepropositionForm()
            return render(request,
"landserve/create_land_proposition.html", {"form":
land_proposition_form })

def landscapeproposition_change(request, landpropositionId):
    if request.user.is_active:

```

```

        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        try:
            landscape_proposition =
Landscapeproposition.objects.select_related("service_provider").get(id
= landpropositionId)
        except Landscapeproposition.DoesNotExist:
            landscape_proposition = None
            if landscape_proposition and provider_authorized and
landscape_proposition.service_provider ==
provider_authorized.service_provider:
                landscape_proposition_form =
LandscapepropositionForm(request.POST or None, instance =
landscape_proposition)
                if request.method == "POST" and
landscape_proposition_form.is_valid():
                    landscape_proposition_form.save()
                    return redirect('landpropositionview', id =
landscape_proposition.id)
                else:
                    landscape_proposition_form = None
                    return render(request,
"landserve/change_land_proposition.html", {"form":
landscape_proposition_form })

def service_provider_change(request, serviceproviderId):
    if request.user.is_active:
        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        try:
            service_provider = Serviceprovider.objects.get(id =
serviceproviderId)
        except Serviceprovider.DoesNotExist:
            service_provider = None
            if service_provider and provider_authorized and
service_provider == provider_authorized.service_provider:
                service_provider_form =
ServiceproviderForm(request.POST or None, instance = service_provider)
                if request.method == "POST" and
service_provider_form.is_valid():
                    service_provider_form.save()
                    return redirect('serviceproviderview', id =
serviceproviderId)
                else:
                    service_provider_form = None

```

```

        return render(request,
"landserve/change_service_provider.html", {"form":
service_provider_form })

def service_provider_statistics(request):
    providerdictionary = {}
    if request.user.is_active:
        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        service_provider = provider_authorized.service_provider
        providerdictionary['provider'] = service_provider
        propositions =
Landscapeproposition.objects.filter(service_provider =
service_provider).select_related('landscape_service').select_related('
landscape_service__landscape_category')
        landscape_categories_short = []
        landscape_categories_services = {}
        landscape_categories = []
        import datetime
        current_point = datetime.datetime.now()
        end_statistics_month = current_point.month
        end_statistics_year = current_point.year
        end_statistics_year = end_statistics_year + 1
        if end_statistics_year == 13:
            end_statistics_month = 1
            end_statistics_year = end_statistics_year + 1
        end_statistics_date =
datetime.datetime(end_statistics_year, end_statistics_month, 1, 0, 0)
        current_point = current_point -
datetime.timedelta(months=12)
        start_statistics_month = current_point.month
        start_statistics_year = current_point.year
        start_statistics_month = start_statistics_month + 1
        if start_statistics_month == 13:
            start_statistics_month = 1
            start_statistics_year = start_statistics_year + 1
        start_statistics_date =
datetime.datetime(start_statistics_year, start_statistics_month, 1, 0,
0)

        for propos in propositions:
            stat_months = []
            while start_statistics_date < end_statistics_date:
                year = start_statistics_date.year
                month = start_statistics_date.month
                month = month + 1
                if month == 13:

```

```

        month = 1
        year = year + 1
        start_statistics_date2 = datetime.datetime(year,
month, 1, 0, 0)

        done_reservations =
Reservation.objects.filter(serviceprovider = service_provider,
service_start__gte = start_statistics_date, service_end__lt =
start_statistics_date2, status = 5)
        fin_sum = 0
        for done_reservation in done_reservations:
            fin_sum = fin_sum +
done_reservation.actual_price
            stat_months.append(fin_sum)
            start_statistics_date = start_statistics_date2
        propos.statistics = stat_months
        if propos.landscape_service.landscape_category not
in landscape_categories_short:
            add_category =
propos.landscape_service.landscape_category
            add_category.services = []
            add_service = propos.landscape_service
            add_service.propositions = []
            add_service.propositions.append(propos)
            add_category.services.append(add_service)
            landscape_categories.append(add_category)

landscape_categories_short.append(propos.landscape_service.landscape_c
ategory)

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id] = []

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id].append(propos.landscape_service)
        else:
            if propos.landscape_service not in
landscape_categories_services[propos.landscape_service.landscape_categ
ory.id]:
                add_service = propos.landscape_service
                add_service.propositions = []
                add_service.propositions.append(propos)

landscape_categories_services[propos.landscape_service.landscape_categ
ory.id].append(propos.landscape_service)
        else:

```

```

        services =
landscape_categories_services[propos.landscape_service.landscape_category.id]

        for landscape_service in services:
            if landscape_service.id ==
propos.landscape_service.id:

landscape_service.propositions.append(propos)
            sort_categories = landscape_categories.sort(key = lambda
x: x.category_name)
            for category in sort_categories:
                category.services = category.services.sort(key =
lambda x: x.service_name)
            providerdictionary['categories'] = sort_categories
            return render(request, "landserve/statistics.html",
providerdictionary)

def service_time_slot_create(request):
    if request.method == "POST":
        service_time_slot_form = ServicetimeForm(request.POST)
        if service_time_slot_form.is_valid() and
request.user.is_active:
            service_time_slot =
service_time_slot_form.save(commit=False)
            provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
            service_time_slot.service_provider =
provider_authorized.service_provider
            service_time_slot.save()
            return redirect(request.path_info)
        else:
            service_time_slot_form = ServicetimeForm()
            return render(request, "landserve/create_service_time.html",
{"form": service_time_slot_form })

def service_time_slot_change(request, servicetimeslotId):
    if request.user.is_active:
        provider_authorized =
Serviceprovideraccount.objects.all().select_related('service_provider'
).get(username = request.user.username)
        try:
            service_time_slot =
Servicetime.objects.select_related("service_provider").get(id =
servicetimeslotId)
        except Servicetime.DoesNotExist:
            service_time_slot = None

```

```

        if service_time_slot and provider_authorized and
service_time_slot.service_provider ==
provider_authorized.service_provider:
            service_time_slot_form =
ServicetimeForm(request.POST or None, instance = service_time_slot)
            if request.method == "POST" and
service_time_slot_form.is_valid():
                service_time_slot_form.save()
                return redirect('calendarview')
            else:
                service_time_slot_form = None
                return render(request, "landserve/change_service_time.html",
{"form": service_time_slot_form })

```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Початковий слайд

- Об'єкт роботи – процес резервування послуг з благоустрою території.
- Предмет роботи – програмне забезпечення резервування послуг з благоустрою території.
- Мета роботи – розробити програмне забезпечення резервування послуг з благоустрою території для забезпечення керування взаємодією між надавачами послуг і клієнтами.

Рисунок В.2 – Об'єкт, предмет, мета

Завдання роботи

- проаналізувати проблему розробки програми резервування послуг з благоустрою території;
- спроектувати програмне забезпечення резервування послуг з благоустрою території;
- реалізувати програмне забезпечення резервування послуг з благоустрою території та виконати його тестування.

3

Рисунок В.3 – Завдання роботи

Порівняння існуючих аналогів та розроблюваного програмного забезпечення резервування послуг з благоустрою території

Характеристика	Yelp	Кабанчик	LandServe
Пошук послуг з благоустрою території	+	+	+
Визначення критеріїв впорядкування результатів пошуку послуг	+	–	+
Замовлення/бронювання послуг з благоустрою території	Замовлення	Замовлення	Бронювання
Керування робочим часом компанії	–	–	+
Оцінювання наданих послуг з благоустрою території	+	+	+
Надання інформації про стандартну вартість послуг	–	+	+
Врахування інформації від клієнта про вартість послуг	–	–	+
Відображення фінансових результатів конкретної послуги з благоустрою території	–	–	+
Визначення повторюваного замовлення послуг з благоустрою території	–	–	+

4

Рисунок В.4 – Порівняння існуючих аналогів та розроблюваного програмного забезпечення резервування послуг з благоустрою території

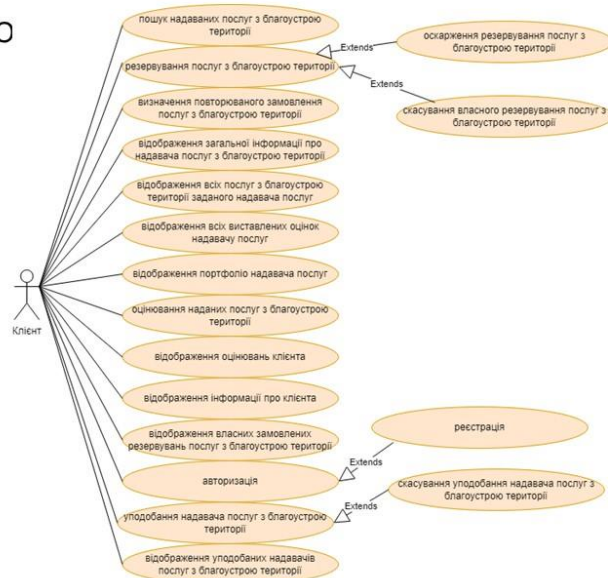
Порівняння фреймворків для реалізації програмного забезпечення резервування послуг з благоустрою території

Характеристика	Pyramid	Django
Швидкість створення програмних прототипів	Нижча	Вища за рахунок кількості вбудованих інструментів
Початкове заповнення даними	Необхідно створити сторінки або працювати через систему керування базами даних	Можна розпочати одразу через адміністративну панель
Механізм валідації форм	Ні	Так
Створення і забезпечення представлень	Функції, класи	Функції, класи

5

Рисунок В.5 – Порівняння фреймворків для реалізації програмного забезпечення резервування послуг з благоустрою території

Діаграма прецедентів клієнта для програмного забезпечення резервування послуг з благоустрою території



6

Рисунок В.6 – Діаграма прецедентів клієнта для програмного забезпечення резервування послуг з благоустрою території

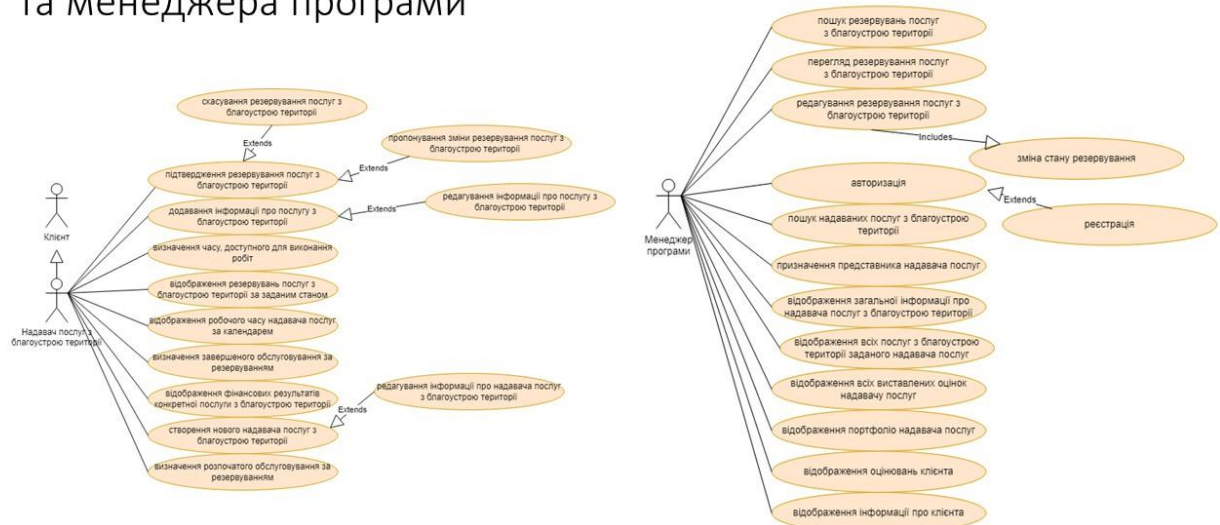


Рисунок В.7 – Діаграми прецедентів надавача послуг з благоустрою території та менеджера програми

Схема бази даних програмного забезпечення резервування послуг з благоустрою території

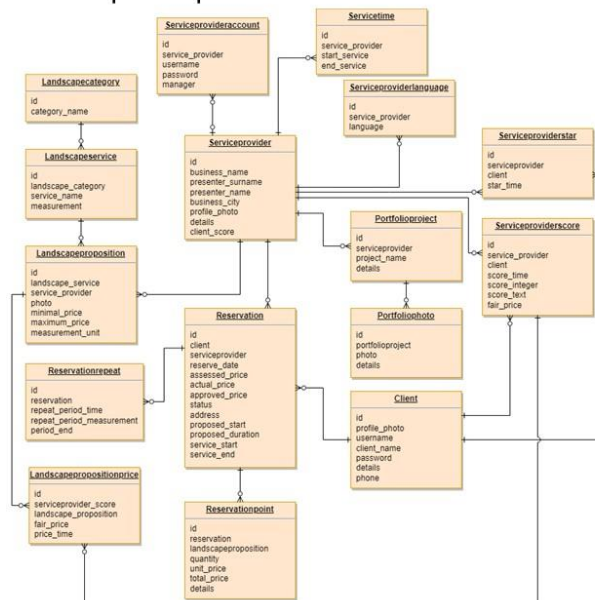
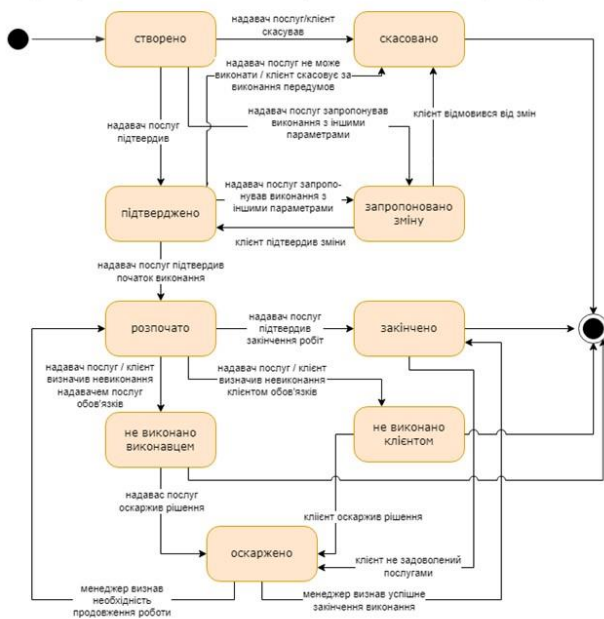


Рисунок В.8 – Схема бази даних програмного забезпечення резервування послуг з благоустрою території

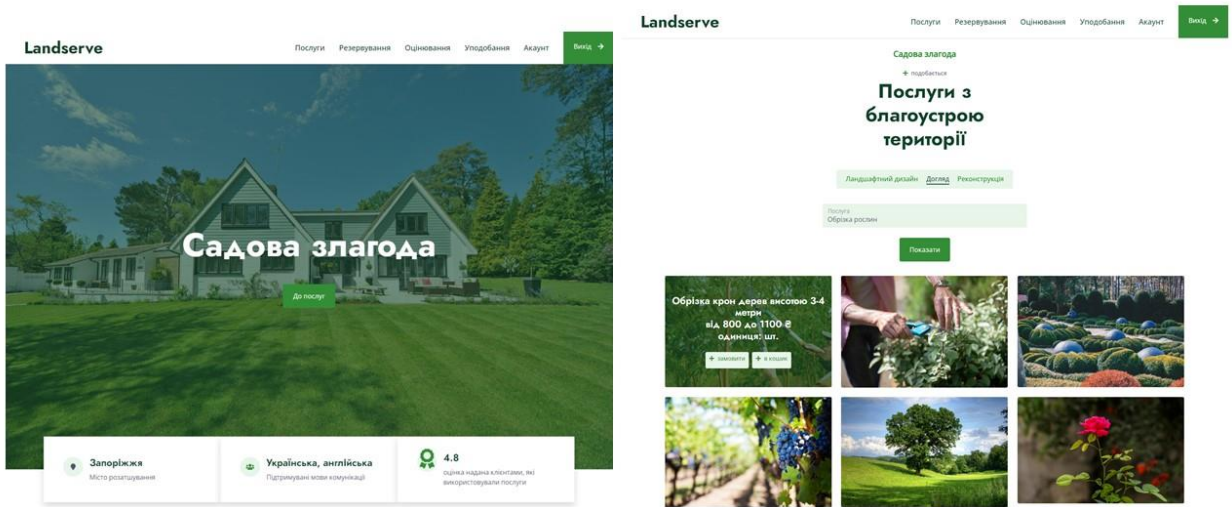
Діаграма станів резервування послуг з благоустрою території



9

Рисунок В.9 – Діаграма станів резервування послуг з благоустрою території

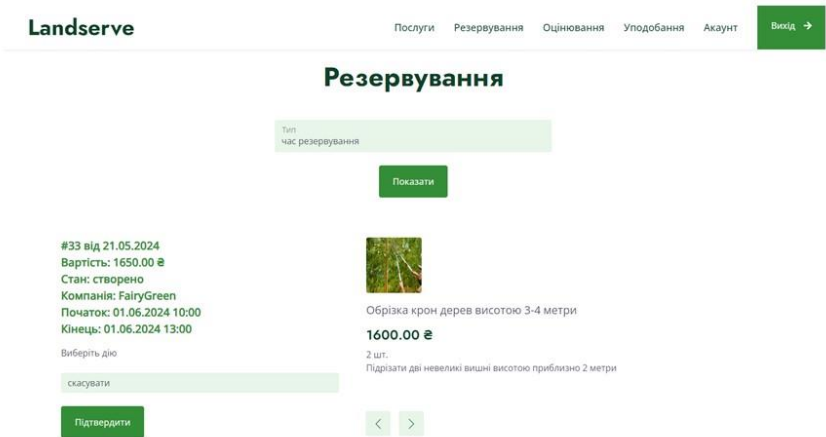
Вебсторінки надавача послуг з благоустрою території



10

Рисунок В.10 – Вебсторінки надавача послуг з благоустрою території

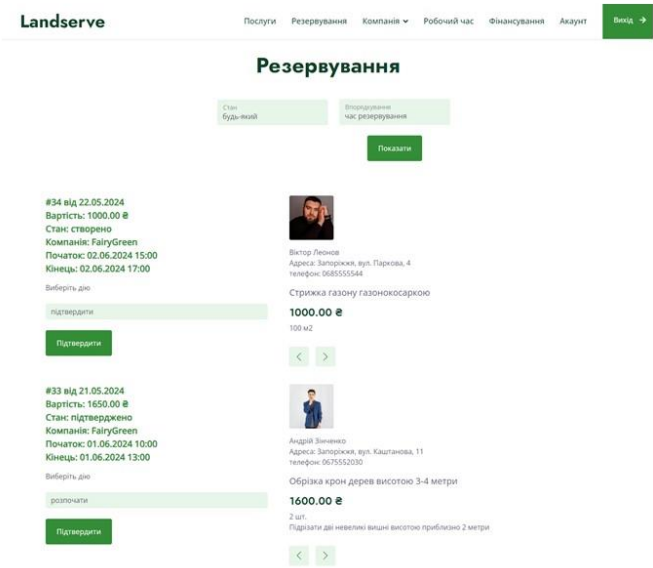
Вебсторінка резервувань клієнта



11

Рисунок В.11 – Вебсторінка резервувань клієнта

Вебсторінка керування резервувань надавачем послуг з благоустрою територій



12

Рисунок В.12 – Вебсторінка керування резервувань надавачем послуг з благоустрою територій

Тестування програми резервування послуг з благоустрою території

Функціональна можливість	Результат	1	2
Пошук надаваних послуг з благоустрою території	Успішно	Підтвердження резервування послуг з благоустрою території	Успішно
Резервування послуг з благоустрою території	Успішно	Додавання інформації про послугу з благоустрою території	Успішно
Визначення повторюваного замовлення послуг з благоустрою території	Успішно	Визначення часу, доступного для виконання робіт, надавачем послуг	Успішно
Відображення загальної інформації про надавача послуг з благоустрою території	Успішно	Відображення резервувань послуг з благоустрою території за заданим станом	Успішно
Відображення всіх послуг з благоустрою території заданого надавача послуг	Успішно	Відображення робочого часу надавача послуг за календарем	Успішно
Відображення всіх виставлених оцінок надавачу послуг	Успішно	Відображення фінансових результатів конкретної послуги з благоустрою території надавача послуг	Успішно
Відображення портфолію надавача послуг	Успішно	Скасування резервування послуг з благоустрою території надавачем послуг	Успішно
Оцінювання наданих послуг з благоустрою території	Успішно	Пропонування зміни резервування послуг з благоустрою території надавачем послуг	Успішно
Відображення оцінювань клієнта	Успішно	Редагування інформації про послугу з благоустрою території	Успішно
Відображення інформації про клієнта	Успішно	Пошук резервувань послуг з благоустрою території	Успішно
Відображення власних замовлених резервувань послуг з благоустрою території	Успішно	Перегляд резервування послуг з благоустрою території	Успішно
Авторизація клієнта, надавача послуг, менеджера	Успішно	Редагування резервування послуг з благоустрою території	Успішно
Уподобання надавача послуг з благоустрою території	Успішно	Створення нового надавача послуг з благоустрою території	Успішно
Відображення уподобаних надавачів послуг з благоустрою території	Успішно	Редагування інформації про надавача послуг з благоустрою території	Успішно
Скасування власного резервування послуг з благоустрою території	Успішно	Призначення представника надавача послуг	Успішно
Скасування уподобання надавача послуг з благоустрою території	Успішно	Визначення завершеного обслуговування за резервуванням	Успішно
Реєстрація клієнта, надавача послуг	Успішно	Визначення розпочатого обслуговування за резервуванням	Успішно
		Оскарження резервування послуг з благоустрою території	Успішно

13

Рисунок В.13 – Тестування програми резервування послуг з благоустрою території

Висновки

- за результатами аналізу можливостей програмних аналогів виділено особливості розробки, які окрім найважливіших функцій включають також врахування інформації від клієнта про вартість послуг, керування робочим часом компанії, що пов'язане з виконанням саме бронювання послуг, а не тільки формування замовлень як у програмах-аналогах;
- проєктування програми виконано за результатами вибору мови Python та фреймворку Django, діаграми прецедентів створені для надавача послуг, менеджера програми та клієнта для визначення відповідних функціональних вимог до програми, реалізацію бази даних та роботи з базою даних забезпечено через створення бази даних на основі системи PostgreSQL;
- заплановані функції, включаючи як функції для надавачів послуг з благоустрою території, так і для клієнтів та менеджерів програми, повністю реалізовано і описано на основі визначення логіки створених програмних функцій для реалізації представлень та створених класів моделей;
- програму резервування послуг з благоустрою території було успішно протестовано стосовно виконання тих функціональних можливостей, які визначалися під час моделювання, та описано стосовно головних принципів роботи.

14

Рисунок В.14 – Висновки