

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)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ
НАДАННЯ ІНДИВІДУАЛЬНИХ РЕАБІЛІТАЦІЙНИХ
ПОСЛУГ ВДОМА
SOFTWARE TO SUPPORT PERSONALIZED REHABILITATION
SERVICES AT HOME

Виконав(ла): студент(ка) 4 курсу, групи КНТ-221

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ГАВРИЛЕНКО Р.Р.

(ПРІЗВИЩЕ та ініціали)

Керівник ШИТІКОВА О.В.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРІЗВИЩЕ та ініціали)

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

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 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
 “ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГАВРИЛЕНКА Рустама Руслановича
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення підтримки надання індивідуальних реабілітаційних послуг вдома. Software to Support Personalized Rehabilitation Services at Home

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

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

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

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

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

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Студент(ка)

(підпис) Рустам ГАВРИЛЕНКО
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Олена ШИТІКОВА
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 104 с., 3 табл., 26 рис., 3 дод., 8 джерел.

РЕАБІЛІТАЦІЙНА ПОСЛУГА, ЛІКАР, ПАЦІЄНТ, ОБЛАДНАННЯ, ВЕБЗАСТОСУНОК, ФРЕЙМВОРК.

Об'єкт роботи – процес надання індивідуальних реабілітаційних послуг вдома. Предмет роботи – програмне забезпечення підтримки надання індивідуальних реабілітаційних послуг вдома. Мета роботи – розробити програмне забезпечення підтримки надання індивідуальних реабілітаційних послуг для взаємодії між лікарями з реабілітації та пацієнтами на умовах реалізації реабілітаційних послуг у пацієнтів вдома.

Матеріали, методи та технічні засоби: мова Python і фреймворк Django для програмування вебзастосунку, система MySQL.

Результати. Розроблена програма є результатом виконаного аналізу стану розробки програмного забезпечення підтримки надання індивідуальних реабілітаційних послуг вдома та визначених сценаріїв роботи користувачів.

Висновки. Розроблена програма призначена для підтримки надання індивідуальних реабілітаційних послуг з врахуванням особливостей реалізації послуг вдома у пацієнта. Для цього програма забезпечує пацієнтів засобами пошуку лікарів з реабілітації, послуг з реабілітації, запису на прийом до лікарів з врахуванням графіку роботи з виїздами додому до пацієнтів, лікарів – засобами контролю свого графіку роботи, керування записами до них на прийом та відслідковування плану роботи, визначення рекомендованих процедур для подальшого лікування.

Галузь використання. Послуги з реабілітації, що надаються окремими лікарями або лікарями, що працюють в об'єднаних медичних центрах, з реалізацією у пацієнтів вдома.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 104 pages, 3 tables, 26 figures, 3 appendixes, 8 sources.

REHABILITATION SERVICE, DOCTOR, PATIENT, EQUIPMENT, WEB APPLICATION, FRAMEWORK.

The object of the work is the process of providing individual rehabilitation services at home. The subject of the work is software to support the provision of individual rehabilitation services at home. The purpose of the work is to develop software to support the provision of individual rehabilitation services for interaction between rehabilitation doctors and patients under the conditions of implementing rehabilitation services at patients' homes.

Materials, methods and technical tools: Python language and Django framework for web application programming, MySQL system.

Results. The developed program is the result of an analysis of the state of development of software to support the provision of individual rehabilitation services at home and specific user work scenarios.

Conclusions. The developed program is intended to support the provision of individual rehabilitation services, taking into account the peculiarities of implementing services at the patient's home. To this end, the program provides patients with the means to search for rehabilitation doctors, rehabilitation services, make appointments with doctors taking into account the work schedule with home visits to patients, doctors with the means to control their work schedule, manage their appointments and track the work plan, determine recommended procedures for further treatment.

Scope of use. Rehabilitation services provided by individual doctors or doctors working in combined medical centers, with implementation at patients' homes.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	10
1.1 Стан розробки програмного забезпечення підтримки надання індивідуальних реабілітаційних послуг вдома.....	10
1.2 Порівняння програм підтримки надання індивідуальних реабілітаційних послуг вдома, що існують.....	15
1.3 Висновки за розділом 1	16
2 Проєктування програми.....	17
2.1 Обґрунтування вибору засобів розробки програми	17
2.2 Визначення вимог користувачів до програми підтримки надання індивідуальних реабілітаційних послуг вдома.....	18
2.3 Проєктування структури бази даних.....	25
2.4 Висновки за розділом 2	39
3 Розробка програми	40
3.1 Послідовність роботи з програмою.....	40
3.2 Реалізація програми підтримки надання індивідуальних реабілітаційних послуг вдома.....	41
3.3 Висновки за розділом 3	58
4 Експлуатація та тестування програми	59
4.1 Призначення програми	59
4.2 Опис роботи з програмою підтримки надання індивідуальних реабілітаційних послуг вдома	59
4.3 Тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома	63
4.4 Висновки за розділом 4	68
Висновки	69
Перелік джерел посилання	70

Додаток А Технічне завдання	71
Додаток Б Текст програми	77
Додаток В Слайди презентації.....	98

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ПЗ – програмне забезпечення;

ФВР – фреймворк веброзробки.

ВСТУП

Важливість реабілітаційних послуг важко переоцінити в Україні в сьогодення. Сучасний спосіб життя звичайної людини взагалі характеризується тим, що велика кількість людей зі зрілого віку починає користуватися частиною послуг з реабілітації. З іншого боку велика кількість людей в Україні зараз страждає і фізично через російську агресію. Це призводить до того, що коло таких осіб збільшується. Частині з них, зокрема зважаючи на проблему з забезпеченням мобільності людей з інвалідністю, незручним є дістатися до медичного центру. Частина з них знаходиться у складному фізичному стані в певний момент. Окрім того велика кількість інших людей може не мати часу на переміщення.

Все це актуалізує потребу в реабілітаційних послугах вдома у пацієнта. Це ж накладає і додаткові вимоги на програму, яка таку роботу має підтримувати. Важливо розуміти, що з одного боку без такої програми популяризувати цей напрямок, цю діяльність неможливо, а з іншого боку – важливо, щоб вона підтримувала особливості запису на прийом за таких послуг. Реабілітаційні послуги вдома зокрема пов'язані з тим, що лікар витрачає додатковий час на переміщення між пацієнтами. Тому необхідно будувати графік з врахуванням цього. Загалом все це вказує на актуальність, важливість і необхідність розроблення програми підтримки надання індивідуальних реабілітаційних послуг вдома. Відповідно метою роботи є розробити програмне забезпечення (ПЗ) підтримки надання індивідуальних реабілітаційних послуг для взаємодії між лікарями з реабілітації та пацієнтами на умовах реалізації реабілітаційних послуг у пацієнтів вдома.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Стан розробки програмного забезпечення підтримки надання індивідуальних реабілітаційних послуг вдома

Усе ПЗ, яке так чи інакше дотичне до надання індивідуальних реабілітаційних послуг вдома, можна розділити на 2 групи:

- таке ПЗ, що представляє клініки, які серед своїх послуг надають послуги реабілітації;
- таке ПЗ, яке дозволяє користувачам створювати оголошення щодо тих послуг, які вони надають, а серед цих послуг представлені так чи інакше і реабілітаційні послуги.

За результатами огляду було встановлено приклади і першої групи ПЗ, і другої, деякі з яких представлені нижче.

Вебсайт Bodro clinic (рис. 1.1) містить винесену сторінку з описом реабілітаційних послуг вдома [1]. Ця сторінка є частиною відповідного розділу реабілітаційних послуг, винесеного у меню вебсайту. При цьому підтримка функціонально такого розділу окрім інформування на самій сторінці реалізується через форму запису на прийом. Проте це елементарна форма зворотного зв'язку, яка не містить ніяких елементів, які б враховували особливості саме реабілітації. У ній можна задати власне ім'я та номер телефону. Тобто вся подальша робота щодо реєстрації на реабілітацію відбувається вже не через вебсайт, а по телефону. При цьому вебсайт містить розділи з інформацією про лікарів, можливість запису на прийом. У формі тоді потрібно вказати ім'я, телефон, електронну пошту та послугу, яку необхідно замовити. Також доступні відеовідгуки від пацієнтів.

Вебсайт Bodro clinic винесений як приклад засобів, доступних в Україні. Закордонний сервіс з реабілітації, який також може використовуватися українцями, зокрема через наявність україномовної версії вебсайту, представлений фізіотерапевтичним центром у Щецині, що в Польщі (рис. 1.2) [2].

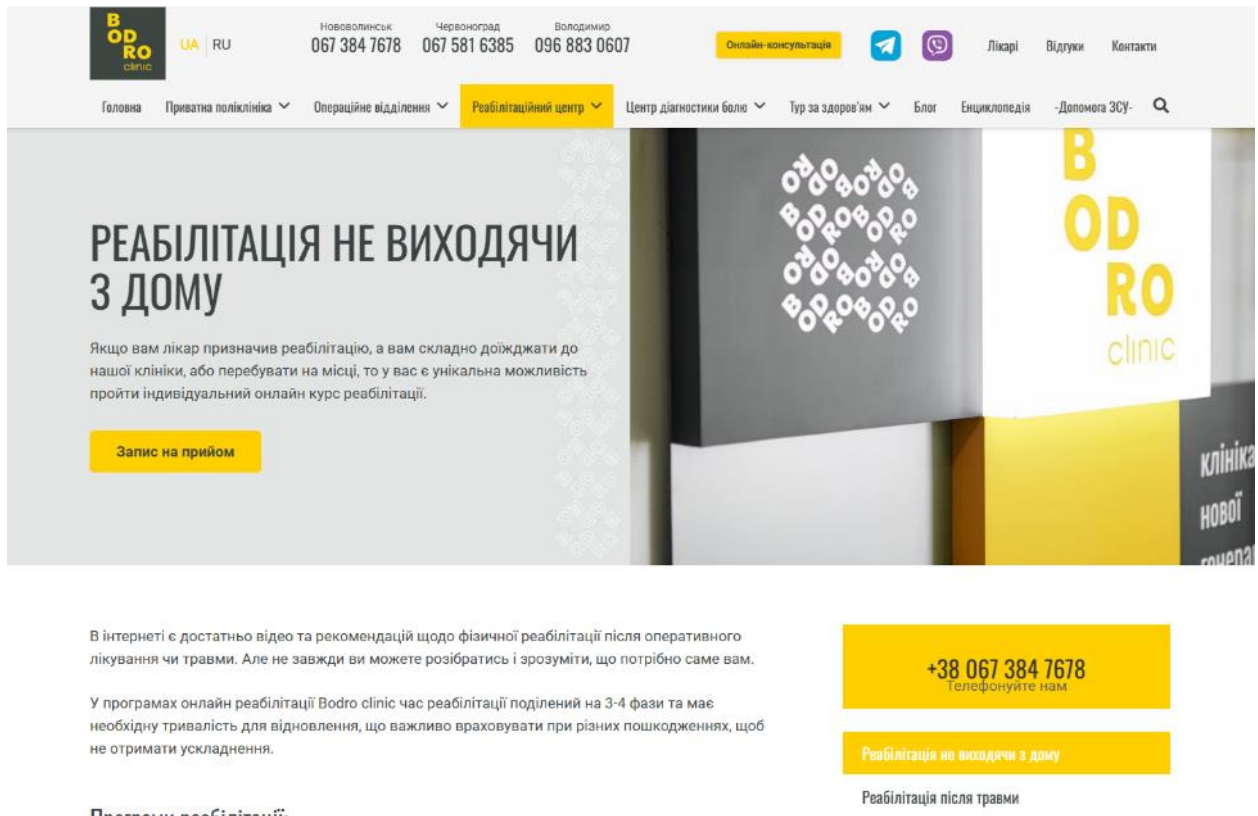


Рисунок 1.1 – Вебсайт Bodro clinic

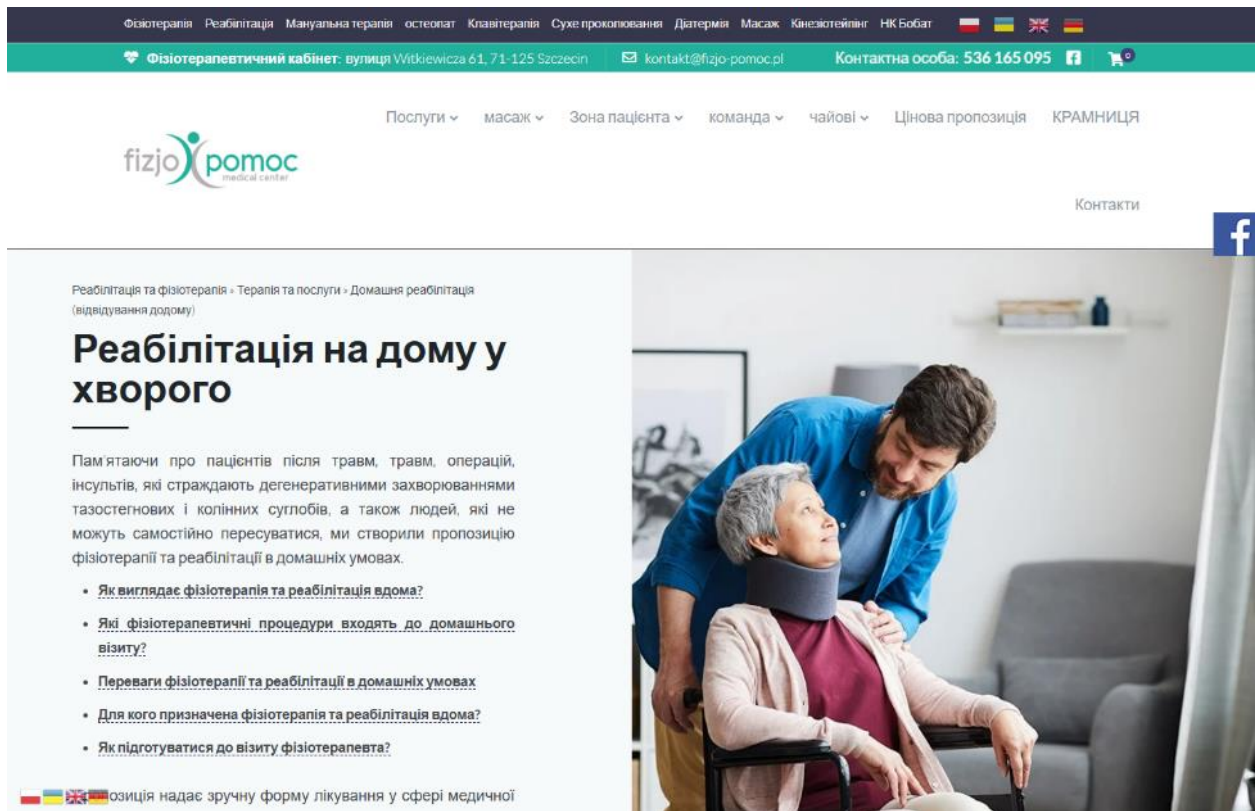


Рисунок 1.2 – Вебсайт Фізіо-Хелп (Fizjo Pomoc)

Вебсайт Фізіо-Хелп містить інформацію не тільки про лікарів, але і про вартість послуг. При цьому фактично на всіх інших сторінках особливості реабілітаційних послуг вдома у клієнта не враховано. Однак при спробі забронювати візит відкривається окремий вебсайт, де здійснюється запис з визначенням конкретного часу візиту. Бронювання відбувається за конкретними послугами. На самому вебсайті Фізіо-Хелп є також посилання на магазин, однак фактично він не працює, тобто сторінка не відкривається під час звернень.

Також були проаналізовані можливості виключно іноземних сервісів, проте вони ще більш обмежені. Так вебсайт Ramsay Connect (рис. 1.3) є представництвом служби, яка як раз займається наданням послуг вдома [3].

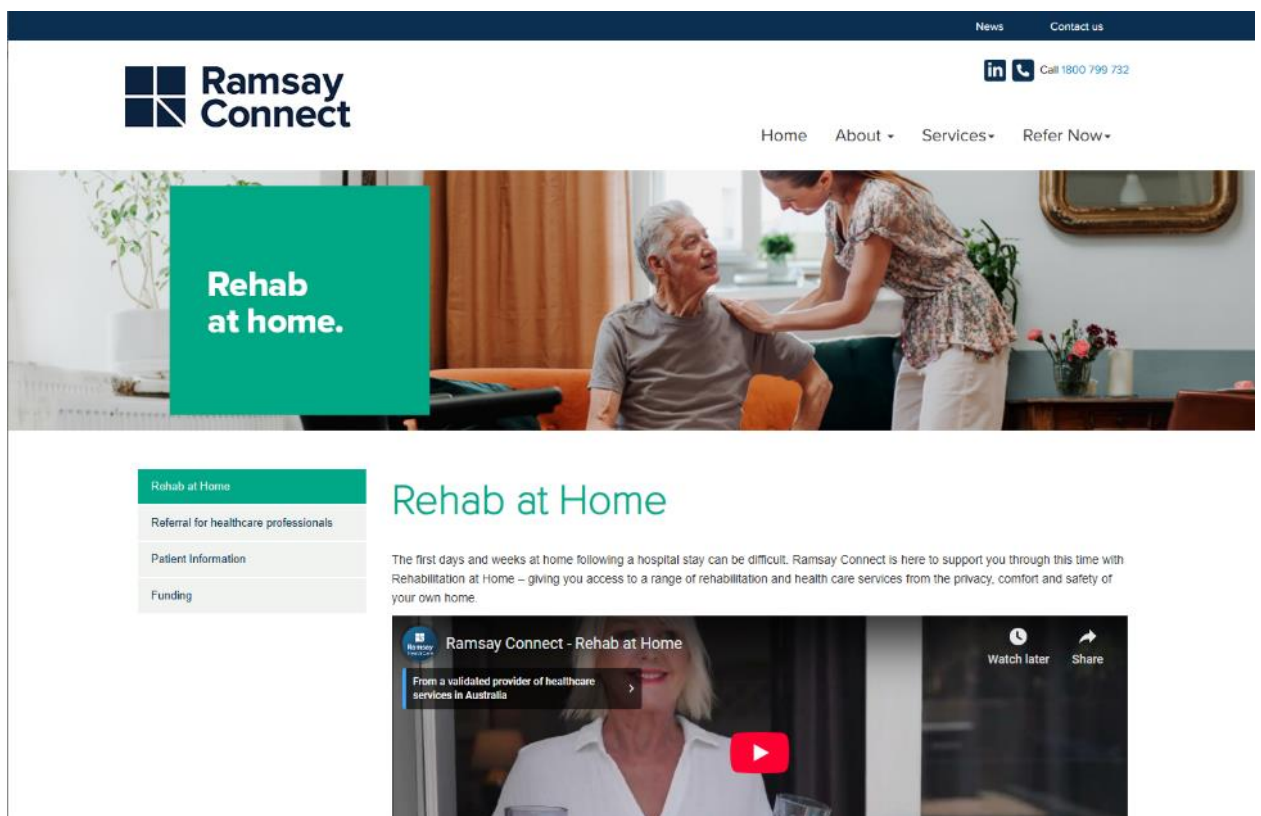


Рисунок 1.3 – Вебсайт Ramsay Connect

Вебсайт Ramsay Connect ніяких спеціалізованих функціональних можливостей окрім інформування про такі послуги не надається. Тобто записатись для реалізації таких реабілітаційних послуг, отримати інформацію

про самих лікарів або інший персонал неможливо. Це значно звужує область використання цього ПЗ, хоча воно і повністю направлено на роботу саме в сфері надання реабілітаційних послуг вдома.

Вебсайт Amplar Health (рис. 1.4) представляє медичну службу, яка зокрема надає і послуги реабілітації вдома [4]. Про надання таких послуг виділено окрему сторінку. Однак, загалом доступні засоби, як і у попередньому рішенні, обмежені значно. Записатись на прийом, отримати додаткову інформацію тут також неможливо.

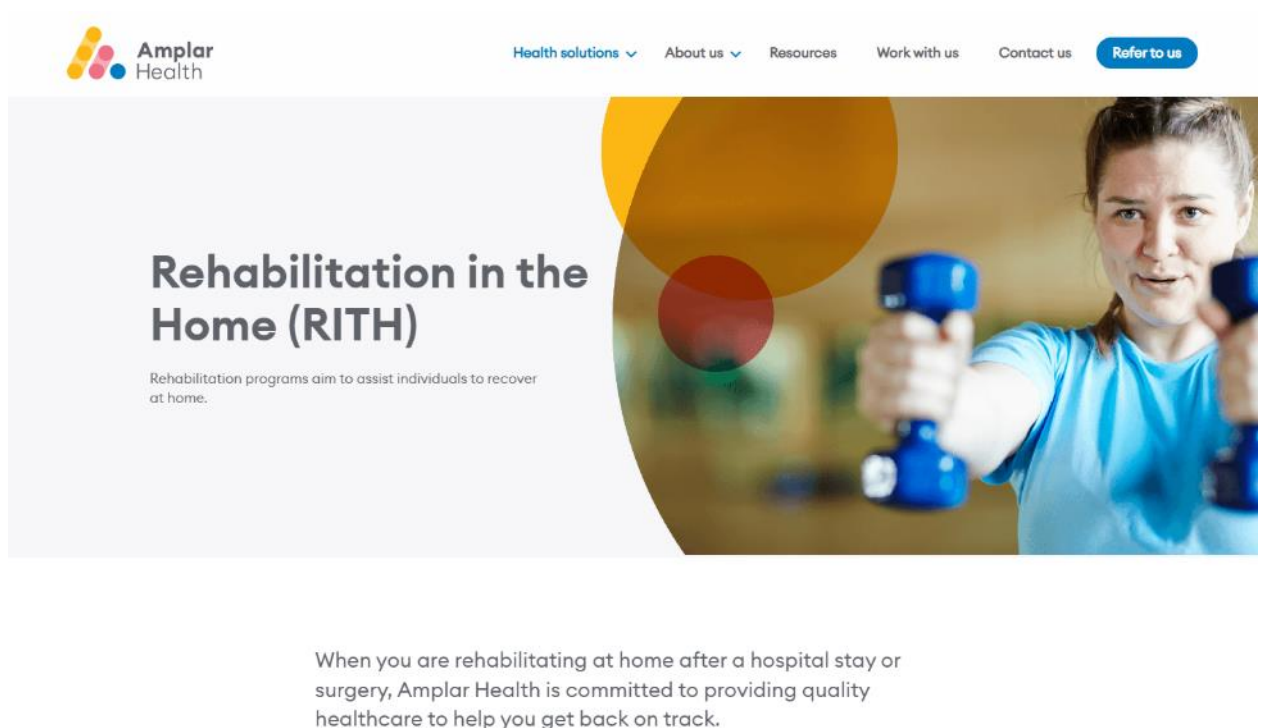


Рисунок 1.4 – Вебсайт Amplar Health

Представником другої групи ПЗ є платформа OLX (рис. 1.5), яка дозволяє виконувати пошук і за оголошеннями користувачів, які надають різні послуги, частина з яких може бути віднесена до реабілітаційних [5]. Проте такі послуги не виділені в окремий розділ. Фактично, наприклад, серед послуг стосовно краси можна знайти послуги масажу. А тоді в параметрах обрати, де саме відбувається прийом для надання послуг. Відповідно можна виконувати

пошук за назвою, а далі впливати фільтрами на вибір конкретного варіанту послуги.

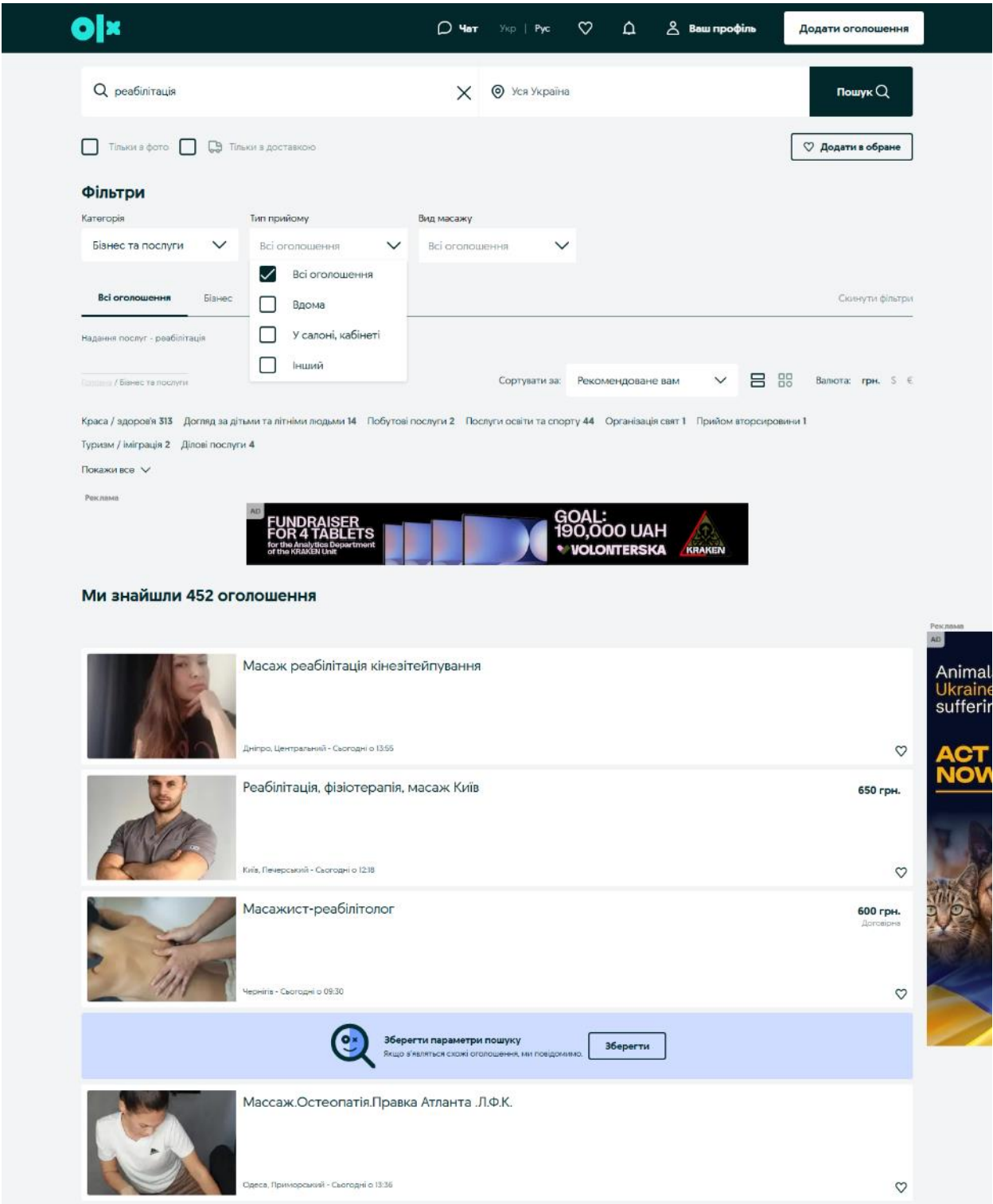


Рисунок 1.5 – Пошук оголошень про надання реабілітаційних послуг на платформі OLX

1.2 Порівняння програм підтримки надання індивідуальних реабілітаційних послуг вдома, що існують

Остаточне порівняння було зведено до запланованих засобів розробки та вебзастосунків OLX і Фізіо-Хелп (табл. 1.1), оскільки вони функціонально є найбільш пристосованими до практичного застосування порівняно з іншими проаналізованими.

Таблиця 1.1 – Порівняння програм підтримки надання індивідуальних реабілітаційних послуг вдома

Критерій порівняння	OLX	Фізіо-Хелп	Моя програма
Бронювання відвідування для реабілітації вдома	Немає	Наявне тільки для відвідування лікарні	Так
Джерело надання послуг реабілітації	Окремі лікарі та інші працівники	Медичний центр	Окремі лікарі та інші працівники, медичні центри
Збереження параметрів пошуку	Так	Немає	Так
Визначення вартості послуг з реабілітації вдома	Може вказуватись в тексті оголошення	Так	Так

Розробка цієї роботи надає безпосередньо можливість бронювання відвідування для реабілітації вдома, що в такому вигляді недоступно у аналогів. Так само як і в OLX у програмі мають бути представлені окремі лікарі та інші працівники, але також вони можуть бути об'єднані в медичні

центри. У програмі має надаватися подібно до OLX можливість збереження параметрів пошуку. Також мають визначатися для лікарів варіанти вартості послуг з реабілітації вдома.

1.3 Висновки за розділом 1

У межах роботи потрібно виконати завдання:

- аналіз стану проблеми розробки ПЗ підтримки надання індивідуальних реабілітаційних послуг вдома;
- проєктування програми підтримки надання індивідуальних реабілітаційних послуг вдома;
- реалізація, тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома.

У цьому розділі виконано аналіз стану проблеми розробки ПЗ підтримки надання індивідуальних реабілітаційних послуг вдома. Аналіз показав, що існуючі програми більшою мірою опосередковано надають підтримку реабілітаційних послуг, переважно така підтримка виражена окремими статичними сторінками з інформацією про таку можливість. Деякі дозволяють бронювати відвідування лікарів, проте ніяка специфіка реабілітаційних послуг вдома не враховується. Єдиний варіант врахування таких можливостей пов'язаний з платформою OLX, яка втім не виділяє окремо реабілітаційні послуги серед інших послуг. Таким чином, у ПЗ, яке заплановано розробити у роботі, ці недоліки мають бути враховані та подолані.

2 ПРОЄКТУВАННЯ ПРОГРАМИ

2.1 Обґрунтування вибору засобів розробки програми

Фреймворки веброзробки (ФВР) є ключовою складовою розробки вебзастосунків, оскільки без них цей процес значно ускладнюється, потребує значно більше часу та створення фактично свого власного рішення за архітектурою, використовуваними засобами, що знижує його практичну придатність у підсумку. На даний момент найбільш пристосованими для практичного використання і активно використовуваними ФВР для мови Python, яка в свою чергу є універсальною, дозволяє розширювати таким чином застосунки в подальшому так, як необхідно, є ФВР Flask [6] і Django [7] (табл. 2.1).

Таблиця 2.1 – Порівняння ФВР для розробки програми підтримки надання індивідуальних реабілітаційних послуг вдома

Критерій порівняння	Django	Flask
Тип рішення	Менш гнучке, але більш функціональне і підготовлене для застосування в обмежених умовах	Більш гнучке, але менш функціональне
Структура проєкту	Визначена фреймворком	Визначається користувачем
Час розробки	Менший у випадку застосування для великих, середніх проєктів	Більший у випадку застосування для великих, середніх проєктів
Підтримка баз даних	Готова	Можлива

Остаточний вибір ФВР був зроблений на користь Django, враховуючи, що не зважаючи на меншу гнучкість, він є більш функціональним і підготовленим для застосування в обмежених умовах для середніх і великих проєктів, до яких і можна віднести цю розробку, також час розробки відповідно є меншим, що важливо для дипломної роботи, є наявна підтримка бази даних, не має потреби встановлювати додаткові рішення, при цьому даний проєкт належить до таких, які як раз і виходять зі структури бази даних.

2.2 Визначення вимог користувачів до програми підтримки надання індивідуальних реабілітаційних послуг вдома

Неавторизований користувач стосовно підтримки надання індивідуальних реабілітаційних послуг вдома повинен бути забезпечений програмою наступними основними сценаріями:

- перегляд сторінки реабілітаційного центру;
- перегляд лікарів реабілітаційного центру;
- перегляд сторінки лікаря з реабілітації;
- авторизація;
- перегляд сторінки пацієнта;
- пошук лікаря для реабілітації;
- пошук реабілітаційних послуг;
- перегляд всіх дипломів або сертифікатів лікаря з реабілітації;
- перегляд доступних послуг з реабілітації у лікаря;
- перегляд доступних часових проміжків прийому в лікаря з реабілітації за датою;
- перегляд відгуків від лікарів з реабілітації про пацієнта;
- перегляд відгуків від пацієнтів про лікаря з реабілітації.

В якості терміну лікар з реабілітації в цих сценаріях передбачається лікар або працівник з реабілітації. Відповідно і роль такого користувача назва як лікар з реабілітації, хоча загалом передбачає, що це може бути і лікар з

реабілітації, і працівник.

Альтернативним сценарієм у неавторизованого користувача є реєстрація для авторизації, коли користувач не має облікового запису пацієнта або лікаря з реабілітації (рис. 2.1). Також він може мати необхідність використати інший обліковий запис. Тоді також задіюється цей сценарій для роботи.

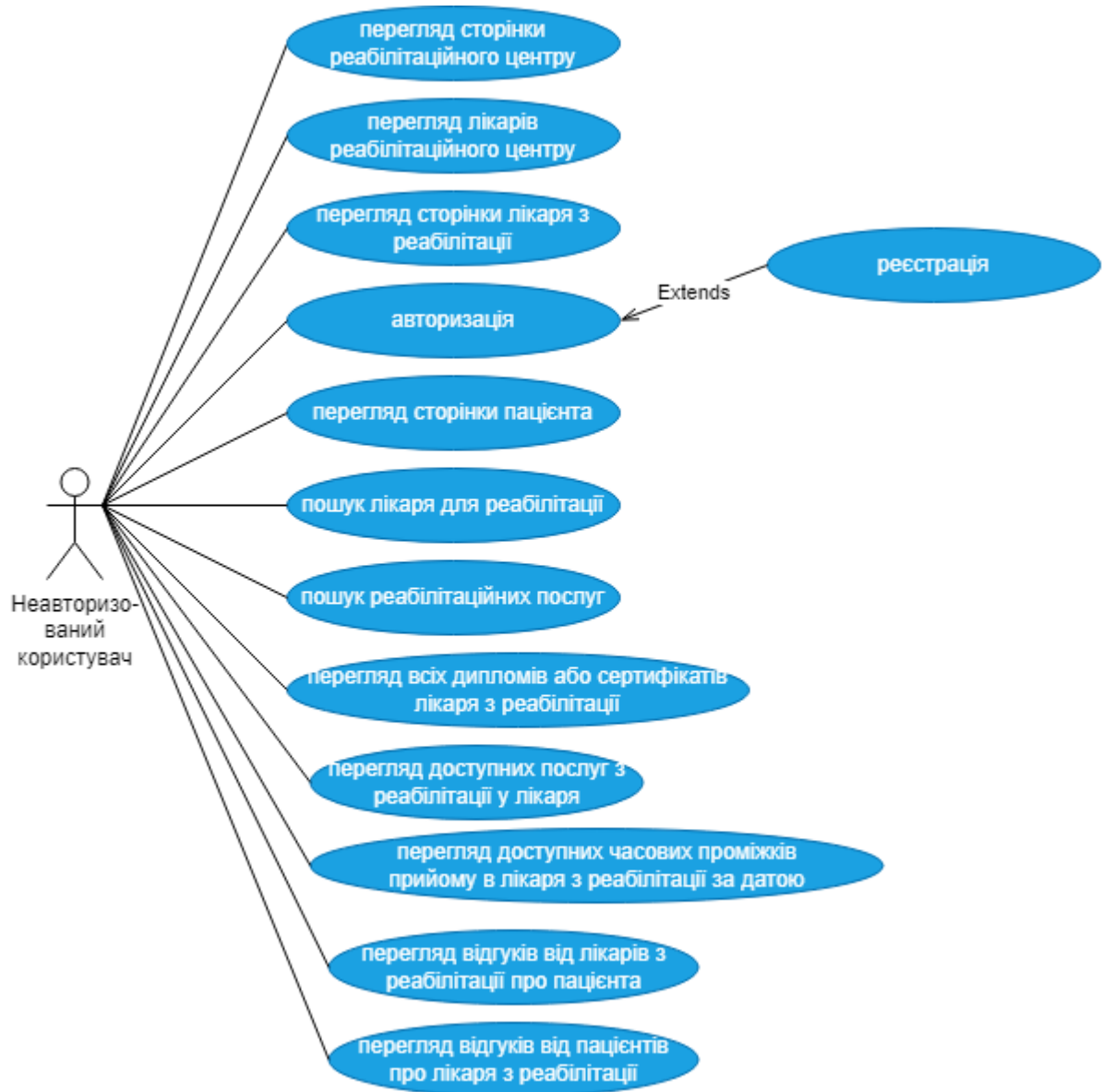


Рисунок 2.1 – Діаграма прецедентів програми підтримки надання індивідуальних реабілітаційних послуг вдома для неавторизованого користувача

Лікар з реабілітації стосовно підтримки надання індивідуальних реабілітаційних послуг вдома повинен бути забезпечений програмою наступними основним сценаріями окрім сценаріїв неавторизованого користувача:

- внесення інформації про лікаря з реабілітації;
- додавання професії з реабілітації лікарю;
- перегляд медичних записів пацієнта;
- додавання диплома або сертифіката лікаря з реабілітації;
- додавання медичного запису пацієнта;
- додавання хвороби пацієнта для реабілітації;
- додавання послуги з реабілітації для лікаря;
- визначення стандартного часу роботи лікаря з реабілітації;
- редагування доступного часового проміжку прийому в лікаря з реабілітації;
- внесення результатів відвідування пацієнтом лікаря з реабілітації;
- перегляд всіх запланованих прийомів у лікаря з реабілітації;
- перегляд всіх реалізованих прийомів у лікаря з реабілітації;
- перегляд всіх нереалізованих прийомів у лікаря з реабілітації;
- перегляд всіх скасованих прийомів у лікаря з реабілітації;
- визначення фактичного прийому пацієнта лікарем з реабілітації;
- створення відгуку про пацієнта, що потребує реабілітації, за результатами прийому.

Для лікаря з реабілітації також передбачено ряд альтернативних сценаріїв:

- вилучення професії з реабілітації лікаря до додавання професії з реабілітації лікарю, коли професія з реабілітації лікарю була присвоєна помилково або він більше не веде прийом за цією професією з реабілітації;
- вилучення диплома або сертифіката лікаря з реабілітації до додавання диплома або сертифіката лікаря з реабілітації, коли диплом або сертифікат з

реабілітації був доданий помилково або більше немає потреби його демонструвати за лікарем;

- редагування даних диплома або сертифіката лікаря з реабілітації до додавання диплома або сертифіката лікаря з реабілітації, коли деякі внесені дані за дипломом або сертифікатом лікаря з реабілітації мають бути змінені;

- редагування медичного запису пацієнта до додавання медичного запису пацієнта, коли медичний запис пацієнта містить неточності;

- вилучення медичного запису пацієнта до додавання медичного запису пацієнта, коли медичний запис пацієнта був раніше внесений помилково;

- вилучення хвороби пацієнта для реабілітації до додавання хвороби пацієнта для реабілітації, коли хвороба пацієнта для реабілітації була внесена помилково або більше не є актуальною;

- редагування послуги з реабілітації для лікаря до додавання послуги з реабілітації для лікаря, коли дані послуги з реабілітації містять неточності через помилки внесення або зміну стану;

- вилучення послуги з реабілітації для лікаря до додавання послуги з реабілітації для лікаря, коли дана послуга з реабілітації цим лікарем більше вже не надається;

- вилучення доступного часового проміжку прийому в лікаря з реабілітації до редагування доступного часового проміжку прийому в лікаря з реабілітації, коли цей часовий проміжок прийому в лікаря більше не доступний;

- визначення нереалізованого прийому пацієнта лікарем з реабілітації до визначення фактичного прийому пацієнта лікарем з реабілітації, коли прийом пацієнта лікарем з реабілітації не відбувся, хоча був запланований;

- скасування запису на прийом до лікаря з реабілітації до перегляду всіх запланованих прийомів у лікаря з реабілітації, коли лікар з реабілітації з тих чи інших причин не може прийняти цього пацієнта в цей час (рис. 2.2).

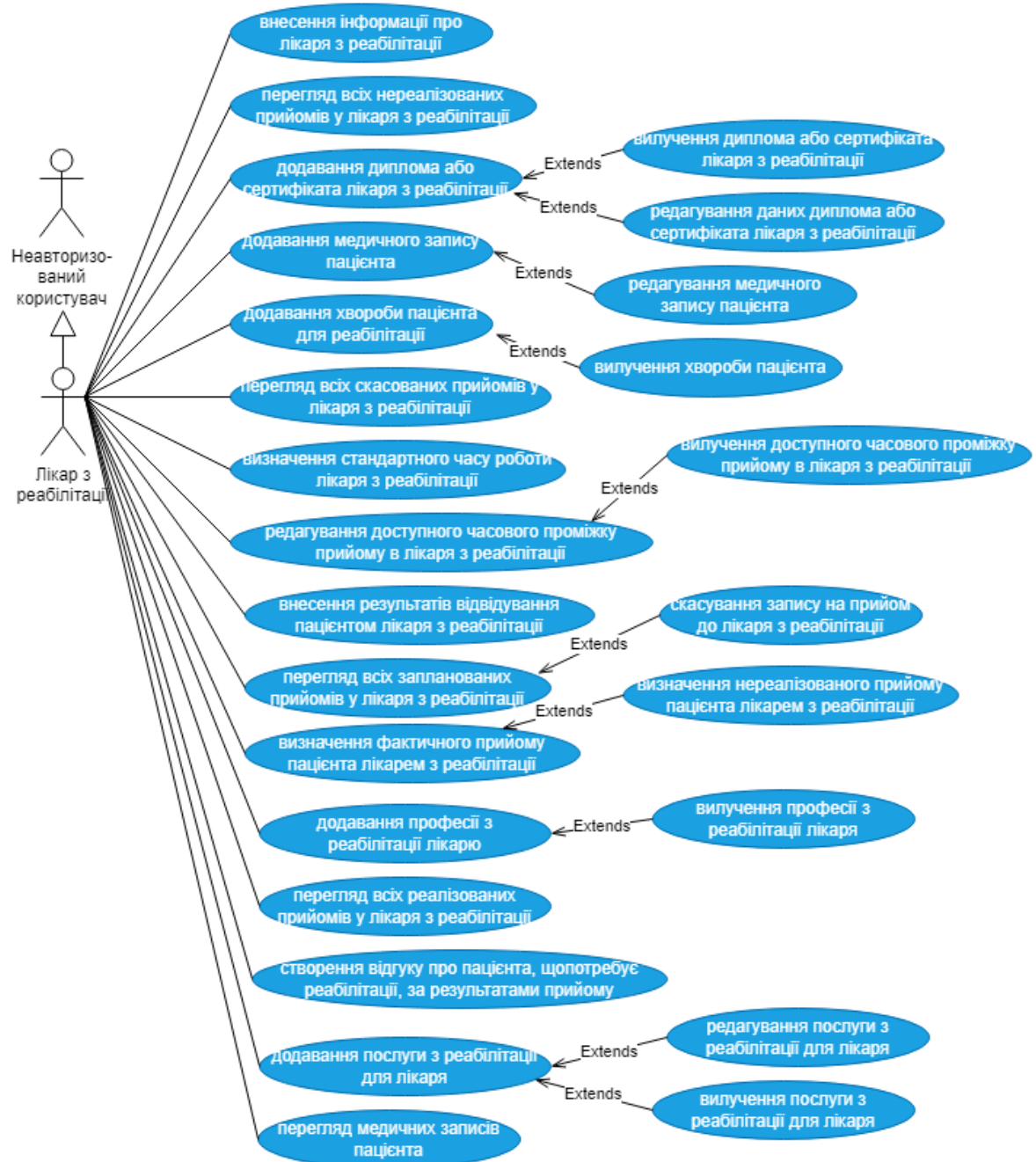


Рисунок 2.2 – Діаграма прецедентів програми підтримки надання індивідуальних реабілітаційних послуг вдома для лікаря з реабілітації

Пацієнт стосовно підтримки надання індивідуальних реабілітаційних послуг вдома повинен бути забезпечений програмою наступними основним сценаріями окрім сценаріїв неавторизованого користувача:

- внесення інформації про пацієнта, що потребує реабілітації;
- збереження параметрів пошуку лікаря для реабілітації;
- збереження параметрів пошуку реабілітаційних послуг;

- пошук лікаря для реабілітації за допомогою збережених параметрів;
- пошук реабілітаційних послуг за допомогою збережених параметрів;
- перегляд збережених параметрів пошуку пацієнтом;
- додавання адреси проживання пацієнта;
- додавання хвороби пацієнта для реабілітації;
- збереження сторінки лікаря з реабілітації пацієнтом;
- перегляд медичних записів пацієнта;
- перегляд збережених лікарів з реабілітації;
- запис на прийом до лікаря з реабілітації;
- перегляд всіх запланованих прийомів пацієнта;
- перегляд всіх реалізованих прийомів пацієнта;
- перегляд всіх нереалізованих прийомів пацієнта;
- перегляд всіх скасованих прийомів пацієнта;
- створення відгуку про лікаря з реабілітації за результатами прийому.

Для пацієнта також передбачено ряд альтернативних сценаріїв:

- вилучення збережених параметрів пошуку лікаря для реабілітації до збереження параметрів пошуку лікаря для реабілітації, коли збережені параметри пошуку лікаря для реабілітації більше не потрібні пацієнту;

- вилучення збережених параметрів пошуку реабілітаційних послуг до збереження параметрів пошуку реабілітаційних послуг, коли збережені параметри пошуку реабілітаційних послуг більше не потрібні пацієнту;

- вилучення адреси проживання пацієнта до додавання адреси проживання пацієнта, коли ця адреса проживання пацієнта більше ним не використовується для замовлення реабілітаційних послуг вдома;

- редагування адреси проживання пацієнта до додавання адреси проживання пацієнта, коли дані щодо адреси проживання пацієнта містять некоректно;

- вилучення хвороби пацієнта для реабілітації до додавання хвороби пацієнта для реабілітації, коли дані про хворобу пацієнта були додані помилково;

– скасування запису на прийом до лікаря з реабілітації до запису на прийом до лікаря з реабілітації, коли прийом за часом ще не мав відбутись, а пацієнт не зможе його відвідати;

– скасування збереження сторінки лікаря з реабілітації пацієнтом до збереження сторінки лікаря з реабілітації пацієнтом, коли сторінка лікаря з реабілітації була збережена пацієнтом, але більше не потрібна (рис. 2.3).

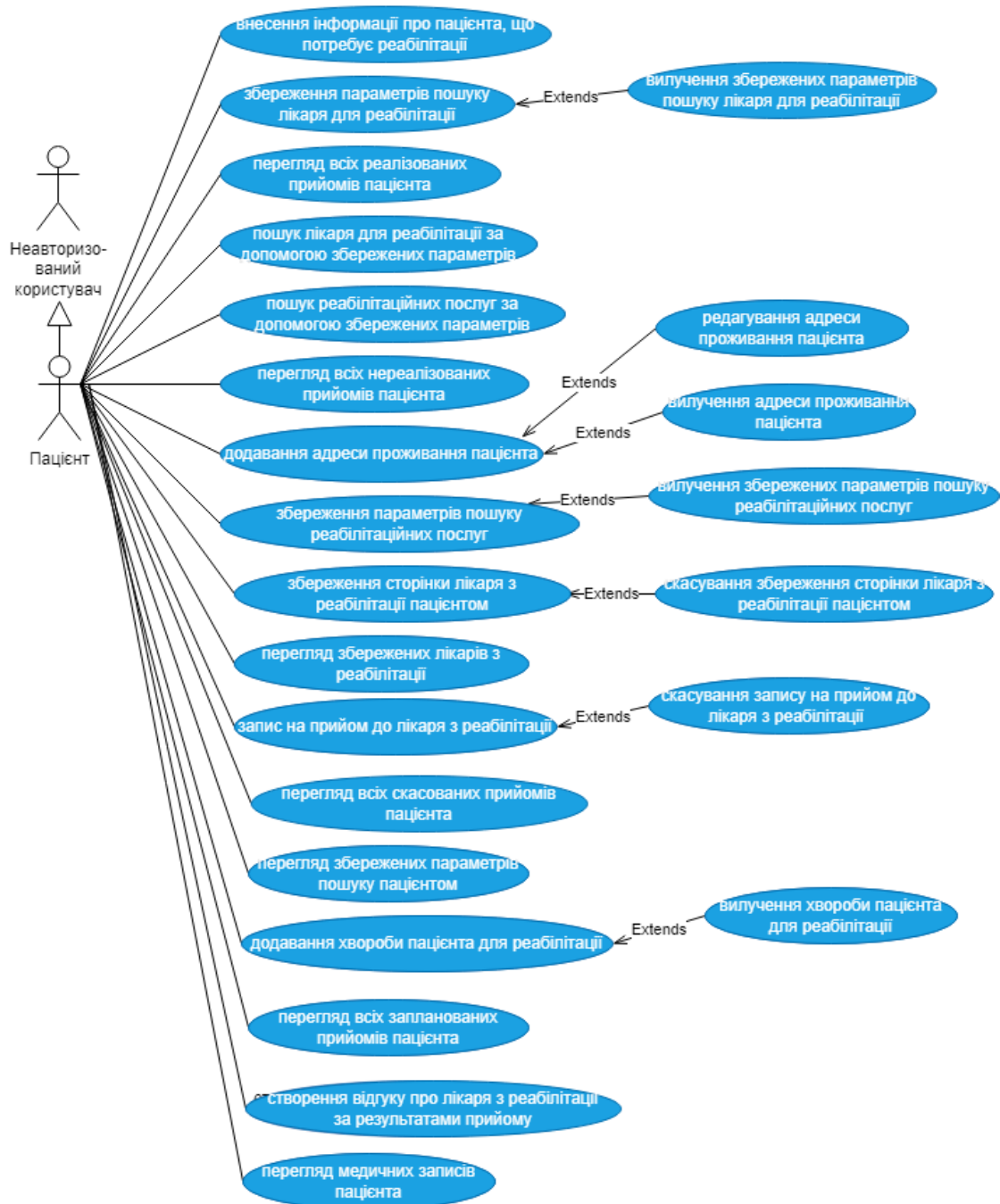


Рисунок 2.3 – Діаграма прецедентів програми підтримки надання індивідуальних реабілітаційних послуг вдома для пацієнта

У підсумку це дозволило для програми підтримки надання індивідуальних реабілітаційних послуг вдома забезпечити моделювання.

2.3 Проєктування структури бази даних

Система MySQL використовується для роботи програми підтримки надання індивідуальних реабілітаційних послуг вдома з базою даних на основі представлених нижче таблиць та зв'язків між ними.

Основним системо утворюючим показником щодо того, де саме надаються реабілітаційні послуги, є міста або населені пункти, які для систематизації були винесені в таблицю City за стовпцями:

- id – ідентифікатор міста або населеного пункту;
- cityname – назва міста або населеного пункту.

Центри реабілітації винесені в таблицю Rehabcenter за стовпцями:

- id – ідентифікатор центру реабілітації;
- rehabcity – місто або населений пункт центру реабілітації;
- rehabaddress – адреса центру реабілітації в цьому місті або населеному пункті;

- rehabcentername – назва центру реабілітації;
- rehabinfo – текстова інформація про центр реабілітації;
- rehabcenterphoto – фотографія центру реабілітації.

Стовпець rehabcity є зовнішнім ключем, з'єднуючим таблицю Rehabcenter з таблицею City.

Можливі професії лікарів або працівників з реабілітації (мануальні терапевти, остеопати) винесені в таблицю Rehabprofession за стовпцями:

- id – ідентифікатор можливої професії лікаря або працівника з реабілітації;
- rehabprofessionname – назва можливої професії лікаря або працівника з реабілітації;

– rehabprofessionphoto – фотографія, долучена для ілюстрації можливої професії лікаря або працівника з реабілітації.

Професії конкретних лікарів або працівників з реабілітації винесені в таблицю Rehabdoctorprofession за стовпцями:

– id – ідентифікатор професії конкретного лікаря або працівника з реабілітації;

– rehabdoctor – лікар або працівник, який здатен працювати за цією професією з реабілітації;

– rehabprofession – професія, за якою здатен працювати цей лікар або працівник з реабілітації;

– rehabworkstart – рік початку роботи цього лікаря або працівника з реабілітації за цією професією.

Стовпець rehabprofession є зовнішнім ключем, з'єднуючим таблицю Rehabdoctorprofession з таблицею Rehabprofession.

Стовпець rehabdoctor є зовнішнім ключем, з'єднуючим таблицю Rehabdoctorprofession з таблицею Rehabdoctor.

Дипломи, сертифікати лікаря або працівника з реабілітації винесені в таблицю Rehabdoctorprofessiondiploma за стовпцями:

– id – ідентифікатор диплома, сертифіката лікаря або працівника з реабілітації;

– rehabdoctorprofession – професія лікаря або працівника з реабілітації, за якою він отримав цей диплом або сертифікат;

– diplomaphoto – фотографія або зображення самого сертифіката або диплома лікаря або працівника з реабілітації;

– diplomadate – дата видачі цього сертифіката або диплома лікаря або працівника з реабілітації;

– diplomadetail – опис необхідних деталей, які додаються до отримання цього сертифіката або диплома лікаря або працівника з реабілітації;

– diplomaphoto – фотографія або зображення цього сертифіката або диплома лікаря або працівника з реабілітації.

Стовпець `rehabdoctorprofession` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctorprofessiondiploma` з таблицею `Rehabdoctorprofession`.

Лікарі або працівники з реабілітації винесені в таблицю `Rehabdoctor` за стовпцями:

- `id` – ідентифікатор лікаря або працівника з реабілітації;
- `doctordjangoaccount` – обліковий запис користувача, який використовує лікар або працівник з реабілітації, що є обліковим записом в Django;
- `rehabdoctorscore` – оцінка лікарю або працівнику з реабілітації, виставлена пацієнтами;
- `rehabdoctordetail` – текстова інформація про лікаря або працівника з реабілітації;
- `rehabcenter` – центр з реабілітації, де працює цей лікар або працівник з реабілітації, який проте може бути не заданий, якщо цей лікар або працівник з реабілітації діє індивідуально в межах програми;
- `rehabdoctorphoto` – фотографія лікаря або працівника з реабілітації;
- `schedulepausetime` – стандартний час на паузу між прийомами цього лікаря або працівника з реабілітації при забезпеченні послуг у пацієнта вдома;
- `rehabcity` – місто або населений пункт, де працює цей лікар з реабілітації (важливо для випадків, коли лікар працює не у складі центру з реабілітації або медичного центру).

Стовпець `rehabcenter` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctor` з таблицею `Rehabcenter`.

Стовпець `rehabcity` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctor` з таблицею `City`.

Стовпець `doctordjangoaccount` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctor` з таблицею `User`. Це стандартна таблиця Django з обліковими записами користувачів. Зв'язок реалізується як один-до-одного (на відміну від інших вказаних).

Пацієнти, які потребують реабілітації, винесені в таблицю Rehabpatient за стовпцями:

- id – ідентифікатор пацієнта, який потребує реабілітації;
- patientdjangoaccount – обліковий запис користувача, який використовує пацієнт, який потребує реабілітації, що є обліковим записом в Django;
- rehabpatientscore – оцінка, в яку пацієнта, який потребує реабілітації, оцінили лікарі або працівники з реабілітації, які працювали з цим пацієнтом;
- rehabpatientdetail – текстова інформація про пацієнта, який потребує реабілітації;
- rehabpatientphoto – фотографія пацієнта, який потребує реабілітації.

Стовпець patientdjangoaccount є зовнішнім ключем, з'єднуючим таблицю Rehabpatient з таблицею User. Зв'язок реалізується як один-до-одного.

Адреси проживання пацієнтів, які потребують реабілітації, винесені в таблицю Rehabpatienthome за стовпцями:

- id – ідентифікатор адреси проживання пацієнта, який потребує реабілітації;
- rehabpatient – пацієнт, який потребує реабілітації та має цю адресу проживання;
- rehabcity – місто або населений пункт, до проживає цей пацієнт, який потребує реабілітації;
- rehabaddress – адреса проживання цього пацієнта, який потребує реабілітації, у місті або населеному пункті;
- rehabphone – номер телефону пацієнта, який потребує реабілітації.

Стовпець rehabcity є зовнішнім ключем, з'єднуючим таблицю Rehabpatienthome з таблицею City.

Стовпець rehabpatient є зовнішнім ключем, з'єднуючим таблицю Rehabpatienthome з таблицею Rehabpatient.

Медичні записи щодо консультування або спостережень за пацієнтами, які потребують реабілітації, у лікарів винесені в таблицю Rehabpatientmedrecord за стовпцями:

- id – ідентифікатор медичного запису щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря;
- rehabpatient – пацієнт, який потребує реабілітації, щодо якого сформовано цей медичний запис щодо консультування або спостережень лікарем;
- rehabdocument – файл документу, який додається до медичного запису щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря;
- rehabdate – дата збереження медичного запису щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря;
- rehabinfo – текстова інформація щодо цього медичного запису (тобто фактично сам медичний запис) щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря.

Стовпець rehabpatient є зовнішнім ключем, з'єднуючим таблицю Rehabpatientmedrecord з таблицею Rehabpatient.

Можливі хвороби, з якими проводиться реабілітація, винесені в таблицю Rehabdisease за стовпцями:

- id – ідентифікатор можливої хвороби, з якою проводиться реабілітація;
- rehabdiseasename – назва можливої хвороби, з якою проводиться реабілітація;
- mkhcode – міжнародний код серед хвороб для даної, з якою може проводитись реабілітація.

Хвороби конкретних пацієнтів, яким потрібна реабілітація, винесені в таблицю Rehabpatientdisease за стовпцями:

- id – ідентифікатор хвороби пацієнта, якому потрібна реабілітація;

– rehabpatient – пацієнт, якому потрібна реабілітація і який має цю хворобу;

– rehabdisease – хвороба, яку має цей пацієнт, який потребує реабілітації;

– fixdate – дата, з якою зафіксована у цього пацієнта, який потребує реабілітації, ця хвороба.

Стовпець rehabpatient є зовнішнім ключем, з'єднуючим таблицю Rehabpatientdisease з таблицею Rehabpatient.

Стовпець rehabdisease є зовнішнім ключем, з'єднуючим таблицю Rehabpatientdisease з таблицею Rehabdisease.

Хвороби, яких стосуються медичні записи пацієнтів, які потребують реабілітації, винесені в таблицю Rehabpatientmedrecorddisease за стовпцями:

– id – ідентифікатор хвороби, яка стосується медичного запису пацієнта, який потребує реабілітації;

– rehabpatientmedrecord – медичний запис щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря, до якого відноситься ця хвороба;

– rehabdisease – хвороба, яка відноситься до цього медичного запису щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря.

Стовпець rehabdisease є зовнішнім ключем, з'єднуючим таблицю Rehabpatientmedrecorddisease з таблицею Rehabdisease.

Стовпець rehabpatientmedrecord є зовнішнім ключем, з'єднуючим таблицю Rehabpatientmedrecorddisease з таблицею Rehabpatientmedrecord.

Можливі послуги з реабілітації винесені в таблицю Rehabservice за стовпцями:

– id – ідентифікатор можливої послуги з реабілітації;

– rehabservicegroup – група, до якої відноситься ця послуга з реабілітації;

– rehab servicename – назва послуги з реабілітації.

Стовпець `rehabservicegroup` є зовнішнім ключем, з'єднуючим таблицю `Rehabpatientmedrecorddisease` з таблицею `Rehabservicegroup`.

Групи послуг з реабілітації винесені в таблицю `Rehabservicegroup` за стовпцями:

- `id` – ідентифікатор групи послуг з реабілітації;
- `rehabservicegroupname` – назва групи послуг з реабілітації.

Послуги з реабілітації, які надаються лікарем або працівником при роботі за однією з його професій, винесені в таблицю `Rehabdoctorprofessionservice` за стовпцями:

- `id` – ідентифікатор послуги з реабілітації, яка надається лікарем або працівником при роботі за цією професією;
- `rehabservice` – послуга з реабілітації, що надається цим лікарем або працівником з реабілітації при роботі за цією професією;
- `rehabdoctorprofession` – професія, за якою цей лікар або працівник з реабілітації надає цю послугу з реабілітації;
- `rehabserviceinfo` – текстова інформація щодо особливостей роботи цього лікаря або працівника з реабілітації за цією професією;
- `rehabprice` – стандартна вартість надання цієї послуги з реабілітації цим лікарем або працівником з реабілітації при роботі за цією професією.

Стовпець `rehabservice` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctorprofessionservice` з таблицею `Rehabservice`.

Стовпець `rehabdoctorprofession` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctorprofessionservice` з таблицею `Rehabdoctorprofession`.

Стандартний робочий час лікаря або працівника з реабілітації за певною професією з реабілітації винесений в таблицю `Rehabdoctorprofessionworktime` за стовпцями:

- `id` – ідентифікатор стандартного робочого часу лікаря або працівника з реабілітації за певною професією з реабілітації;
- `rehabdoctorprofession` – професія, за якою цей лікар або працівник з реабілітації, працює в цей час;

– rehabworkday – номер робочого дня в тижні, за яким цей лікар або працівник з реабілітації, працює за цією професією;

– rehabstartdaytime – час початку роботи цього лікаря або працівника з реабілітації за цією професією в цей робочий день;

– rehabenddaytime – час закінчення роботи цього лікаря або працівника з реабілітації за цією професією в цей робочий день.

Стовпець rehabdoctorprofession є зовнішнім ключем, з'єднуючим таблицю Rehabdoctorprofessionworktime з таблицею Rehabdoctorprofession.

Обладнання, яке може використовуватись для реабілітації, винесене в таблицю Rehabequipment за стовпцями:

– id – ідентифікатор обладнання, яке може використовуватись для реабілітації;

– rehequipmentphoto – фотографія обладнання, яке може використовуватись для реабілітації;

– rehequipmentname – назва обладнання, яке може використовуватись для реабілітації;

– rehequipmentinfo – текстовий опис цього обладнання, яке може використовуватись для реабілітації.

Реабілітаційна послуга, при виконанні якої цей лікар або працівник з реабілітації застосовує це обладнання, винесена в таблицю Rehabdoctorserviceequipment за стовпцями:

– id – ідентифікатор реабілітаційної послуги, при виконанні якої цей лікар або працівник з реабілітації застосовує це обладнання;

– rehabdoctorprofessionservice – реабілітаційна послуга, при виконанні якої за певною професією цей лікар або працівник з реабілітації використовує це обладнання;

– rehequipment – обладнання, яке використовується при виконанні цієї реабілітаційної послуги за певною професією лікаря або працівника з реабілітації;

– rehabusageinfo – тестовий опис особливостей застосування цього обладнання для виконання цієї реабілітаційної послуги цим лікарем або працівником з реабілітації.

Стовпець rehabequipment є зовнішнім ключем, з'єднуючим таблицю Rehabdoctorserviceequipment з таблицею Rehabequipment.

Стовпець rehabdoctorprofessionservice є зовнішнім ключем, з'єднуючим таблицю Rehabdoctorserviceequipment з таблицею Rehabdoctorprofessionservice.

Збережені параметри пошуку лікарів або послуг з реабілітації для пацієнта винесені в таблицю Rehabsearch за стовпцями:

– id – ідентифікатор збережених параметрів пошуку лікарів або послуг з реабілітації для пацієнта;

– rehabpatient – пацієнт, який зберіг ці параметри пошуку лікарів або послуг з реабілітації;

– rehabsearchname – назва збережених параметрів пошуку лікарів або послуг з реабілітації для пацієнта;

– rehabprofession – професія лікаря або пацієнта з реабілітації, пошук якого здійснюється пацієнтом;

– rehabdoctorscore – мінімальна оцінка, яку повинен мати лікар або пацієнт з реабілітації.

Стовпець rehabpatient є зовнішнім ключем, з'єднуючим таблицю Rehabsearch з таблицею Rehabpatient.

Стовпець rehabprofession є зовнішнім ключем, з'єднуючим таблицю Rehabsearch з таблицею Rehabprofession.

Хвороби, за якими пацієнт потребує реабілітації та за якими виконує пошук лікарів або послуг з реабілітації, винесені в таблицю Rehabsearchdisease за стовпцями:

– id – ідентифікатор запису про хворобу, за якою пацієнт потребує реабілітації та за якою виконує пошук лікарів або послуг з реабілітації;

– rehabsearch – збережені параметри пошуку лікарів або послуг з реабілітації для пацієнта, відносно яких визначена ця хвороба;

– rehabdisease – хвороба, за якою пацієнт потребує реабілітації та за якою виконує пошук лікарів або послуг з реабілітації.

Стовпець rehabsearch є зовнішнім ключем, з'єднуючим таблицю Rehabsearchdisease з таблицею Rehabsearch.

Стовпець rehabdisease є зовнішнім ключем, з'єднуючим таблицю Rehabsearchdisease з таблицею Rehabdisease.

Послуги з реабілітації, за якими здійснюється пошук пацієнтом у збережених параметрах пошуку, винесені в таблицю Rehabsearchservice за стовпцями:

– id – ідентифікатор запису про послуги з реабілітації, за якими здійснюється пошук пацієнтом у збережених параметрах пошуку;

– rehabsearch – збережені параметри пошуку лікарів або послуг з реабілітації для пацієнта, відносно яких визначена ця послуга з реабілітації;

– rehabservice – послуга з реабілітації, за якою здійснюється пошук пацієнтом у збережених параметрах пошуку.

Стовпець rehabsearch є зовнішнім ключем, з'єднуючим таблицю Rehabsearchservice з таблицею Rehabsearch.

Стовпець rehabservice є зовнішнім ключем, з'єднуючим таблицю Rehabsearchservice з таблицею Rehabservice.

Обладнання, яке має застосовуватись при виконанні послуг з реабілітації, за збереженими пацієнтом параметрами пошуку винесено в таблицю Rehabsearchserviceequipment за стовпцями:

– id – ідентифікатор запису про обладнання, яке має застосовуватись при виконанні послуг з реабілітації, за збереженими пацієнтом параметрами пошуку;

– rehabsearchservice – запис про послуги з реабілітації, за якими здійснюється пошук пацієнтом у збережених параметрах пошуку та за якими має використовуватися це обладнання;

– `reabequipment` – обладнання, яке має застосовуватись при виконанні послуг з реабілітації, за збереженими пацієнтом параметрами пошуку.

Стовпець `rehabsearchservice` є зовнішнім ключем, з'єднуючим таблицю `Rehabsearchserviceequipment` з таблицею `Rehabsearchservice`.

Стовпець `reabequipment` є зовнішнім ключем, з'єднуючим таблицю `Rehabsearchserviceequipment` з таблицею `Reabequipment`.

Доступні часові проміжки у роботі лікаря або працівника з реабілітації винесені в таблицю `Rehabdoctorprofessiontimeslot` за стовпцями:

– `id` – ідентифікатор доступного часового проміжку в роботі лікаря або працівника з реабілітації;

– `rehabdoctorprofession` – професія лікаря або працівника з реабілітації, за якою доступний цей часовий проміжок;

– `timeslotstart` – час початку часового проміжку в роботі лікаря або працівника з реабілітації;

– `timeslotend` – час закінчення часового проміжку в роботі лікаря або працівника з реабілітації.

Стовпець `rehabdoctorprofession` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctorprofessiontimeslot` з таблицею `Rehabdoctorprofession`.

Записи пацієнта на прийом до лікаря або працівника з реабілітації винесені в таблицю `Rehabdoctortimeslotapplication` за стовпцями:

– `id` – ідентифікатор запису пацієнта на прийом до лікаря або працівника з реабілітації;

– `rehabdoctorprofessiontimeslot` – доступний часовий проміжок у роботі лікаря або працівника з реабілітації, за яким відбувся запис пацієнта на прийом;

– `rehabpatient` – пацієнт, який записався на прийом до цього лікаря або працівника з реабілітації;

– `rehabstatus` – стан запису пацієнта на прийом до лікаря або працівника з реабілітації;

– rehabprescription – рекомендація щодо лікування пацієнта за результатами прийому, надана лікарем або працівником з реабілітації;

– rehabdoctorprofessionservice – послуга з реабілітації, яка має бути надана лікарем або працівником з реабілітації пацієнту;

– rehabpatienthome – адреса проживання пацієнта, за якою має бути надано цю послугу з реабілітації лікарем або працівником з реабілітації.

Стовпець rehabpatienthome є зовнішнім ключем, з'єднуючим таблицю Rehabdoctortimeslotapplication з таблицею Rehabpatienthome.

Стовпець rehabpatient є зовнішнім ключем, з'єднуючим таблицю Rehabdoctortimeslotapplication з таблицею Rehabpatient.

Обладнання, яке запланував для використання під час цього прийому лікар або працівник з реабілітації, винесене в таблицю Rehabdoctortimeslotapplicationequipment за стовпцями:

– id – ідентифікатор обладнання, яке запланував для використання під час цього прийому лікар або працівник з реабілітації;

– rehabdoctortimeslotapplication – запис пацієнта на прийом до лікаря або працівника з реабілітації, за яким заплановано застосувати це обладнання;

– rehabdoctorserviceequipment – обладнання, яке застосовується лікарем або працівником з реабілітації для цієї послуги з реабілітації та яке заплановано для застосування під час цього прийому;

– rehabdoctorown – визначення, чи це власне обладнання лікаря, чи це обладнання пацієнта.

Стовпець rehabdoctortimeslotapplication є зовнішнім ключем, з'єднуючим таблицю Rehabdoctortimeslotapplicationequipment з таблицею Rehabdoctortimeslotapplication.

Стовпець rehabdoctorserviceequipment є зовнішнім ключем, з'єднуючим таблицю Rehabdoctortimeslotapplicationequipment з таблицею Rehabdoctorserviceequipment (рис. 2.4).

Стовпець `rehabdoctorprofessiontimeslot` є зовнішнім ключем, з'єднуючим таблицю `Rehabdoctortimeslotapplication` з таблицею `Rehabdoctorprofessiontimeslot`.

Відгуки за результатами прийомів пацієнтів у лікарів або працівників з реабілітації для надання реабілітаційних послуг винесені в таблицю Rehabview за стовпцями:

- id – ідентифікатор відгуку за результатом прийому пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг;
- rehabdoctortimeslotapplication – прийом пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг, за яким створено відгук;
- rehabdoctororpatient – визначення, чи це лікар створив відгук про пацієнта, чи навпаки пацієнт про лікаря або працівника з реабілітації за результатами прийому;
- rehabviewscore – оцінка відгуку за результатом прийому пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг;
- rehabviewtext – текст відгуку за результатом прийому пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг;
- rehabviewtime – час, коли збережено відгук за результатом прийому пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг.

Збереження лікарів або працівників з реабілітації пацієнтами винесені в таблицю Rehabdoctorsave за стовпцями:

- id – ідентифікатор збереження лікаря або працівника з реабілітації пацієнтом;
- rehabpatient – пацієнт, який зберіг цього лікаря або працівника з реабілітації;
- rehabdoctor – лікар або працівник з реабілітації, якого зберіг цей пацієнт.

Стовпець rehabpatient є зовнішнім ключем, з'єднуючим таблицю Rehabdoctorsave з таблицею Rehabpatient.

Стовпець rehabdoctor є зовнішнім ключем, з'єднуючим таблицю Rehabdoctorsave з таблицею Rehabdoctor.

2.4 Висновки за розділом 2

У розділі детально представлені результати проєктування бази даних для програми підтримки надання реабілітаційних послуг вдома, визначено необхідні для реалізації сценарії для базових ролей користувачів, включаючи пацієнтів та лікарів (або працівників з реабілітації), неавторизованих користувачів. Визначено Python-ФВР Django для реалізації.

3 РОЗРОБКА ПРОГРАМИ

3.1 Послідовність роботи з програмою

Основою взаємодії для пацієнта та лікаря з реабілітації є прийом лікарем пацієнта (рис. 3.1). Прийом здійснюється за попереднім записом, а для цього пацієнту доступні часові проміжки, на які він може записатися. Коли пацієнт обирає один з них і зберігає для себе, то запис на прийом для цього створюється і переходить до стану запланованого. Після цього лікар з реабілітації або сам пацієнт можуть скасувати це відвідування, тоді запис на прийом переходить до стану скасованого. Якщо запис на прийом до самого прийому не був скасований, то якщо пацієнт прибув на прийом до лікаря з реабілітації, то лікар визначає його як реалізований. Якщо ні, то як нереалізований.

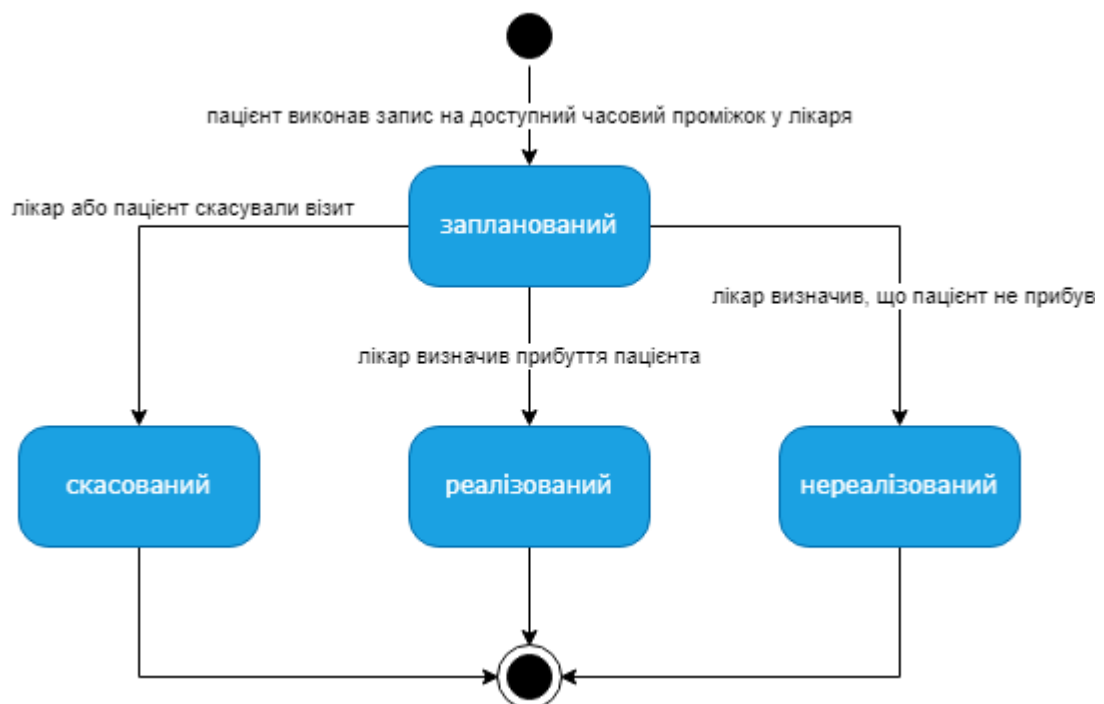


Рисунок 3.1 – Діаграма станів запису на прийом до лікаря з реабілітації

3.2 Реалізація програми підтримки надання індивідуальних реабілітаційних послуг вдома

Реалізація програми підтримки надання індивідуальних реабілітаційних послуг вдома забезпечена через використання ФВР Django. У цьому ФВР головним чином функціональність забезпечується через підготовку на основі відповідних функцій представлень даних для роботи з відповідними сторінками. Тому були розроблені самі функції представлень, самі сторінки вебсторінок у вигляді шаблонів, а для визначення дизайну шаблонів цих вебсторінок використано вільний шаблон [8].

Перегляд сторінки реабілітаційного центру реалізований функцією представлення `rehab_center_view` з параметром `rehabcenter`, який задає ідентифікатор медичного центру.

Перегляд лікарів реабілітаційного центру реалізований функцією представлення `rehab_center_doctors_view` з параметром `rehabcenter`, який задає ідентифікатор медичного центру.

Перегляд сторінки лікаря з реабілітації реалізований функцією представлення `rehab_doctor_page_view` з параметром `rehabdoctor`, який задає логін лікаря з реабілітації.

Перегляд сторінки пацієнта реалізований функцією представлення `rehab_patient_page_view`, а логін пацієнта задається параметром `rehabpatient`.

Внесення інформації про пацієнта, що потребує реабілітації, реалізоване функцією представлення `rehab_patient_edit_view`. Використовується створений для цього клас форми `RehabpatientForm` за даними авторизації.

Внесення інформації про лікаря з реабілітації реалізоване функцією представлення `rehab_doctor_edit_view`. Використовується створений для цього клас форми `RehabdoctorForm` за даними авторизації.

Пошук лікаря для реабілітації реалізований функцією представлення `rehab_doctor_search_view` з передачею параметрів пошуку через ключі запиту.

Зокрема до ключів пошуку належать `rehabprof` з ідентифікатором професії лікаря з реабілітації, `rehabexp` з кількістю років досвіду роботи лікаря, `rehabcity` з ідентифікатором міста, де має проводитись реабілітація, `rehabscore` з мінімальною оцінкою, яку повинен мати лікар з реабілітації, тощо.

Пошук реабілітаційних послуг реалізований функцією представлення `rehab_service_search_view` з передачею параметрів пошуку через ключі запиту.

Збереження параметрів пошуку лікаря для реабілітації реалізоване функцією представлення `rehab_doctor_search_save_view` на основі класу форми `RehabsearchForm`.

Пошук лікаря для реабілітації за допомогою збережених параметрів реалізований функцією представлення `rehab_doctor_search_saved_view` з параметром `rehabsearch`, який задає ідентифікатор збережених параметрів пошуку лікаря.

Пошук реабілітаційних послуг за допомогою збережених параметрів реалізований функцією представлення `rehab_service_search_saved_view` з параметром `rehabservice`, який задає ідентифікатор збережених параметрів пошуку реабілітаційних послуг у складі пошуку лікаря.

Перегляд збережених параметрів пошуку пацієнтом реалізований функцією представлення `rehab_searches_view`.

Додавання хвороби пацієнта для реабілітації реалізоване функцією представлення `rehabpatientdisease_add_view` з параметром `rehabpatient`, що задає логін пацієнта, на основі класу форми `RehabpatientdiseaseForm`.

Вилучення хвороби пацієнта для реабілітації реалізоване функцією представлення `rehabpatientdisease_delete_view` з параметром `rehabpatientdisease`, що задає ідентифікатор хвороби пацієнта для реабілітації.

Редагування доступного часового проміжку прийому в лікаря з реабілітації реалізоване функцією представлення `rehabdoctorprofessiontimeslot_edit_view` з параметром `rehabdoctorprofessiontimeslot`, що задає часовий проміжок прийому в лікаря, на основі класу форми `RehabdoctorprofessiontimeslotForm`.

Вилучення доступного часового проміжку прийому в лікаря з реабілітації реалізоване функцією представлення `rehabdoctorprofessiontimeslot_delete_view` з параметром `rehabdoctorprofessiontimeslot`, що задає часовий проміжок прийому в лікаря.

Перегляд доступних часових проміжків прийому в лікаря з реабілітації за датою реалізований функцією представлення `rehabdoctorprofessiontimeslots_view` з параметрами `timeslotstart` і `timeslotend`, що задають дати початку і закінчення цього проміжку часу, та `rehabdoctorprofession`, що задає ідентифікатор професії лікаря.

Вилучення збережених параметрів пошуку лікаря для реабілітації реалізоване функцією представлення `rehabsearch_delete_view` з параметром `rehabsearch`, який задає ідентифікатор збережених параметрів пошуку лікаря.

Вилучення збережених параметрів пошуку реабілітаційних послуг реалізоване функцією представлення `rehabsearch_service_delete_view` з параметром `rehabsearchservice`, який задає ідентифікатор збережених параметрів пошуку реабілітаційних послуг.

Додавання професії з реабілітації лікарю реалізоване функцією представлення `rehabdoctorprofession_add_view` на основі класу форми `RehabdoctorprofessionForm`.

Вилучення професії з реабілітації лікаря реалізоване функцією представлення `rehabdoctorprofession_delete_view` з параметром `rehabdoctorprofession`, що задає ідентифікатор професії з реабілітації лікаря.

Додавання диплома або сертифіката лікаря з реабілітації реалізоване функцією представлення `rehabdoctorprofession_diploma_add_view` з параметром `rehabdoctorprofession`, що задає ідентифікатор професії лікаря з реабілітації, та на основі класу форми `RehabdoctorprofessiondiplomaForm`.

Перегляд всіх дипломів або сертифікатів лікаря з реабілітації реалізований функцією представлення `rehab_doctor_diplomas_view` з параметром `rehabdoctor`, який задає логін лікаря з реабілітації, та `rehabdoctprof`,

який задає ідентифікатор професії лікаря з реабілітації, за якою потрібно відобразити ці дипломи або сертифікати.

Вилучення диплома або сертифіката лікаря з реабілітації реалізоване функцією представлення `rehabdoctorprofession_diploma_delete_view` з параметром `rehabdiploma`, що задає ідентифікатор диплома або сертифіката лікаря з реабілітації.

Редагування даних диплома або сертифіката лікаря з реабілітації реалізоване функцією представлення `rehabdoctorprofession_diploma_edit_view` з параметром `rehabdiploma`, що задає ідентифікатор диплома або сертифіката лікаря з реабілітації, та на основі класу форми `RehabdoctorprofessiondiplomaForm`.

Додавання адреси проживання пацієнта реалізоване функцією представлення `rehabpatienthome_add_view` на основі класу форми `RehabpatienthomeForm`.

Вилучення адреси проживання пацієнта реалізоване функцією представлення `rehabpatienthome_delete_view` з параметром `rehabhome`, що задає ідентифікатор адреси проживання пацієнта.

Редагування адреси проживання пацієнта реалізоване функцією представлення `rehabpatienthome_edit_view` з параметром `rehabhome`, що задає ідентифікатор адреси проживання пацієнта.

Додавання лікарем з реабілітації медичного запису пацієнта реалізоване функцією представлення `rehabpatientmedrecord_add_view` з параметром `rehabpatient`, що задає логін пацієнта, на основі класу форми `RehabpatientmedrecordForm`.

Редагування лікарем з реабілітації медичного запису пацієнта реалізоване функцією представлення `rehabpatientmedrecord_edit_view` з параметром `rehabpatientmedrecord`, що задає ідентифікатор медичного запису пацієнта, на основі класу форми `RehabpatientmedrecordForm`.

Вилучення лікарем з реабілітації медичного запису пацієнта реалізоване функцією представлення `rehabpatientmedrecord_delete_view` з

параметром `rehabpatientmedrecord`, що задає ідентифікатор медичного запису пацієнта.

Додавання послуги з реабілітації для лікаря реалізоване функцією представлення `rehabdoctorprofession_service_add_view` з параметром `rehabdoctprof`, який задає ідентифікатор професії лікаря з реабілітації, за якою потрібно додати нову послугу з реабілітації, на основі класів форм `RehabdoctorprofessionserviceForm` та `RehabdoctorserviceequipmentForm`.

Редагування послуги з реабілітації для лікаря реалізоване функцією представлення `rehabdoctorprofession_service_edit_view` з параметром `rehabdoctorprofession`, який задає ідентифікатор послуги з реабілітації для лікаря, на основі класів форм `RehabdoctorprofessionserviceForm` та `RehabdoctorserviceequipmentForm`.

Вилучення послуги з реабілітації для лікаря реалізоване функцією представлення `rehabdoctorprofession_service_delete_view` з параметром `rehabdoctorprofession`, який задає ідентифікатор послуги з реабілітації для лікаря

Перегляд доступних послуг з реабілітації у лікаря реалізований функцією представлення `rehabdoctorprofession_services_view` з параметром `rehabdoctor`, який задає логін лікаря з реабілітації, та `rehabdoctprof`, який задає ідентифікатор професії лікаря з реабілітації, за якою потрібно відобразити ці послуги з реабілітації.

Збереження сторінки лікаря з реабілітації пацієнтом реалізоване функцією представлення `rehab_doctor_save_view` з параметром `rehabdoctor`, який задає логін лікаря з реабілітації, а якщо такий лікар цим пацієнтом був збережений, то реалізується скасування збереження сторінки лікаря з реабілітації пацієнтом.

Перегляд збережених лікарів з реабілітації реалізований функцією представлення `rehab_saved_doctors_view`.

Запис на прийом до лікаря з реабілітації реалізований функцією представлення `rehabdoctortimeslotapplication_set_visit_view` з параметрами

rehabdoctorprofessiontimeslot, який задає ідентифікатор часового проміжку в лікаря, rehabdoctorprofessionservice, що задає ідентифікатор послуги з реабілітації, rehabpatienthome, що задає ідентифікатор адреси проживання пацієнта.

Внесення результатів відвідування пацієнтом лікаря з реабілітації реалізоване функцією представлення rehabapplication_to_add_view з параметром rehabdoctorapplication, який задає ідентифікатор запису на прийом, та на основі класів форм RehabdoctortimeslotapplicationForm та RehabdoctortimeslotapplicationequipmentForm.

Скасування запису на прийом до лікаря з реабілітації реалізоване функцією представлення rehabapplication_to_cancel_view з параметром rehabdoctorapplication, який задає ідентифікатор запису на прийом.

Перегляд всіх запланованих прийомів у лікаря з реабілітації реалізований функцією представлення rehab_planned_doctor_applications_view з параметрами rehabstart, що визначає дату початку часового проміжку, що відображається, rehabend, що визначає дату закінчення часового проміжку, що відображається, sortordering, що задає варіант сортування. Кожне представлення має окремий шаблон вебсторінки. У даному випадку це plandoctorapps.html.

Перегляд всіх реалізованих прийомів у лікаря з реабілітації реалізований функцією представлення rehab_finalized_doctor_applications_view з параметрами rehabstart, що визначає дату початку часового проміжку, що відображається, rehabend, що визначає дату закінчення часового проміжку, що відображається, sortordering, що задає варіант сортування.

Перегляд всіх нереалізованих прийомів у лікаря з реабілітації реалізований функцією представлення rehab_unfinalized_doctor_applications_view з параметрами rehabstart, що визначає дату початку часового проміжку, що відображається, rehabend, що

визначає дату закінчення часового проміжку, що відображається, `sortordering`, що задає варіант сортування.

Перегляд всіх скасованих прийомів у лікаря з реабілітації реалізований функцією представлення `rehab_cancelled_doctor_applications_view` з параметрами `rehabstart`, що визначає дату початку часового проміжку, що відображається, `rehabend`, що визначає дату закінчення часового проміжку, що відображається, `sortordering`, що задає варіант сортування.

Перегляд всіх запланованих прийомів пацієнта реалізований функцією представлення `rehab_planned_patient_applications_view` з параметрами `rehabstart`, що визначає дату початку часового проміжку, що відображається, `rehabend`, що визначає дату закінчення часового проміжку, що відображається, `sortordering`, що задає варіант сортування.

Перегляд всіх реалізованих прийомів пацієнта реалізований функцією представлення `rehab_finalized_patient_applications_view` з параметрами `rehabstart`, що визначає дату початку часового проміжку, що відображається, `rehabend`, що визначає дату закінчення часового проміжку, що відображається, `sortordering`, що задає варіант сортування.

Перегляд всіх нереалізованих прийомів пацієнта реалізований функцією представлення `rehab_unfinalized_patient_applications_view` з параметрами `rehabstart`, що визначає дату початку часового проміжку, що відображається, `rehabend`, що визначає дату закінчення часового проміжку, що відображається, `sortordering`, що задає варіант сортування.

Перегляд всіх скасованих прийомів пацієнта реалізований функцією представлення `rehab_cancelled_patient_applications_view` з параметрами `rehabstart`, що визначає дату початку часового проміжку, що відображається, `rehabend`, що визначає дату закінчення часового проміжку, що відображається, `sortordering`, що задає варіант сортування.

Визначення фактичного прийому пацієнта лікарем з реабілітації реалізоване функцією представлення `rehabapplication_to_finish_view` з

параметром `rehabdoctorapplication`, який задає ідентифікатор запису на прийом.

Визначення нереалізованого прийому пацієнта лікарем з реабілітації реалізоване функцією представлення `rehabapplication_to_unfinish_view` з параметром `rehabdoctorapplication`, який задає ідентифікатор запису на прийом.

Створення відгуку про лікаря з реабілітації за результатами прийому реалізоване функцією представлення `new_rehab_review_view` з параметром `rehabdoctorapplication`, який задає ідентифікатор запису на прийом, за яким створюється відгук.

Створення відгуку про пацієнта, що потребує реабілітації, за результатами прийому лікарем реалізоване функцією представлення `new_patient_rehab_review_view` з параметром `rehabdoctortimeslotapplication`, який задає ідентифікатор запису на прийом, за яким створюється відгук.

Перегляд відгуків від лікарів з реабілітації про пацієнта реалізований функцією представлення `rehab_patient_views_view`, а логін пацієнта задається параметром `rehabpatient`.

Перегляд відгуків від пацієнтів про лікаря з реабілітації реалізований функцією представлення `rehab_doctor_views_view`, а логін лікаря з реабілітації задається параметром `rehabdoctor`.

Перегляд медичних записів пацієнта реалізований функцією представлення `rehab_patient_medrecords_view`, а логін пацієнта задається параметром `rehabpatient`.

Модель `City` є класом з назвою міста або населеного пункту в полі `cityname` класу.

Модель центру реабілітації реалізована класом `Rehabcenter`.

Поле `rehabcity` в класі визначає місто або населений пункт центру реабілітації, реалізоване типом `ForeignKey` на основі моделі `City`.

Поле `rehabaddress` в класі визначає адресу центру реабілітації в цьому місті або населеному пункті, реалізовану типом `CharField`.

Поле `rehabcentername` в класі визначає назву центру реабілітації, реалізовану типом `CharField`.

Поле `rehabinfo` в класі визначає текстову інформацію про центр реабілітації, реалізовану типом `TextField`.

Поле `rehabcenterphoto` в класі визначає фотографію центру реабілітації, реалізовану типом `ImageField`.

Модель можливих професій лікарів або працівників з реабілітації реалізована класом `Rehabprofession`.

Поле `rehabprofessionname` в класі визначає назву можливої професії лікаря або працівника з реабілітації, реалізовану типом `CharField`.

Поле `rehabprofessionphoto` в класі визначає фотографію, долучену для ілюстрації можливої професії лікаря або працівника з реабілітації, реалізовану типом `ImageField`.

Модель професії конкретних лікарів або працівників з реабілітації реалізована класом `Rehabdoctorprofession`.

Поле `rehabdoctor` в класі визначає лікаря або працівника, який здатен працювати за цією професією з реабілітації, реалізованого типом `ForeignKey` на основі моделі `Rehabdoctor`.

Поле `rehabprofession` в класі визначає професію, за якою здатен працювати цей лікар або працівник з реабілітації, реалізовану типом `ForeignKey` на основі моделі `Rehabprofession`.

Поле `rehabworkstart` в класі визначає рік початку роботи цього лікаря або працівника з реабілітації за цією професією, реалізований типом `DateField`.

Модель дипломів, сертифікатів лікаря або працівника з реабілітації реалізована класом `Rehabdoctorprofessiondiploma`.

Поле `rehabdoctorprofession` в класі визначає професію лікаря або працівника з реабілітації, за якою він отримав цей диплом або сертифікат, реалізовану типом `ForeignKey` на основі моделі `Rehabdoctorprofession`.

Поле `diplomaphoto` в класі визначає фотографію або зображення самого сертифіката або диплома лікаря або працівника з реабілітації, реалізовану типом `ImageField`.

Поле `diplomadate` в класі визначає дату видачі цього сертифіката або диплома лікаря або працівника з реабілітації, реалізовану типом `DateField`.

Поле `diplomadetail` в класі визначає опис необхідних деталей, які додаються до отримання цього сертифіката або диплома лікаря або працівника з реабілітації, реалізований типом `TextField`.

Поле `diplomaphoto` в класі визначає фотографію або зображення цього сертифіката або диплома лікаря або працівника з реабілітації, реалізовану типом `ImageField`.

Модель лікарів або працівників з реабілітації реалізована класом `Rehabdoctor`.

Поле `doctordjangoaccount` в класі визначає обліковий запис користувача, який використовує лікар або працівник з реабілітації, що є обліковим записом в Django, реалізований типом `OneToOneField` на основі моделі `User`.

Поле `rehabdoctorscore` в класі визначає оцінку лікарю або працівнику з реабілітації, виставлену пацієнтами, реалізовану типом `PositiveSmallIntegerField`.

Поле `rehabdoctordetail` в класі визначає текстову інформацію про лікаря або працівника з реабілітації, реалізовану типом `TextField`.

Поле `rehabcenter` в класі визначає центр з реабілітації, де працює цей лікар або працівник з реабілітації, реалізований типом `ForeignKey` на основі моделі `Rehabcenter`.

Поле `rehabdoctorphoto` в класі визначає фотографію лікаря або працівника з реабілітації, реалізовану типом `ImageField`.

Поле `schedulepausetime` в класі визначає стандартний час на паузу між прийомом цього лікаря або працівника з реабілітації при забезпеченні послуг у пацієнта вдома, реалізований типом `TimeField`.

Поле `rehabcity` в класі визначає місто або населений пункт, де працює цей лікар з реабілітації, реалізований типом `ForeignKey` на основі моделі `City`.

Модель пацієнтів, які потребують реабілітації, реалізована класом `Rehabpatient`.

Поле `patientdjangoaccount` в класі визначає обліковий запис користувача, який використовує пацієнт, який потребує реабілітації, що є обліковим записом в Django, реалізований типом `OneToOneField` на основі моделі `User`.

Поле `rehabpatientscore` в класі визначає оцінку, в яку пацієнта, який потребує реабілітації, оцінили лікарі або працівники з реабілітації, які працювали з цим пацієнтом, реалізовану типом `PositiveSmallIntegerField`.

Поле `rehabpatientdetail` в класі визначає текстову інформацію про пацієнта, який потребує реабілітації, реалізовану типом `CharField`.

Поле `rehabpatientphoto` в класі визначає фотографію пацієнта, який потребує реабілітації, реалізовану типом `ImageField`.

Поле `rehabdoctors` в класі визначає збережених лікарів або працівників з реабілітації пацієнтом, реалізованих типом `ManyToManyField` на основі моделі `Rehabdoctor`, таким чином забезпечуючи зв'язок багато-до-багатьох.

Модель адрес проживання пацієнтів, які потребують реабілітації, реалізована класом `Rehabpatienthome`.

Поле `rehabpatient` в класі визначає пацієнта, який потребує реабілітації та має цю адресу проживання, реалізовану типом `ForeignKey` на основі моделі `Rehabpatient`.

Поле `rehabcity` в класі визначає місто або населений пункт, до проживає цей пацієнт, який потребує реабілітації, реалізовану типом `ForeignKey` на основі моделі `City`.

Поле `rehabaddress` в класі визначає адресу проживання цього пацієнта, який потребує реабілітації, у місті або населеному пункті, реалізовану типом `CharField`.

Поле `rehabphone` в класі визначає номер телефону пацієнта, який потребує реабілітації, реалізований типом `CharField`.

Модель медичних записів щодо консультування або спостережень за пацієнтами, які потребують реабілітації, у лікарів реалізована класом `Rehabpatientmedrecord`.

Поле `rehabpatient` в класі визначає пацієнта, який потребує реабілітації, щодо якого сформовано цей медичний запис щодо консультування або спостережень лікарем, реалізованого типом `ForeignKey` на основі моделі `Rehabpatient`.

Поле `rehabdocument` в класі визначає файл документу, який додається до медичного запису щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря, реалізований типом `FileField`.

Поле `rehabdate` в класі визначає дату збереження медичного запису щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря, реалізовану типом `DateField`.

Поле `rehabinfo` в класі визначає текстову інформацію щодо цього медичного запису (тобто фактично сам медичний запис) щодо консультування або спостереження за пацієнтом, який потребує реабілітації, у лікаря, реалізовану типом `TextField`.

Поле `rehabdiseases` в класі визначає хвороби, яких стосується цей медичний запис пацієнта, який потребує реабілітації, реалізовані типом `ManyToManyField` на основі моделі `Rehabdisease`, таким чином забезпечуючи зв'язок багато-до-багатьох.

Модель можливих хвороб, з якими проводиться реабілітація, реалізована класом `Rehabdisease`.

Поле `rehabdiseasename` в класі визначає назву можливої хвороби, з якою проводиться реабілітація, реалізовану типом `CharField`.

Поле `mkhcode` в класі визначає міжнародний код серед хвороб для даної, з якою може проводитись реабілітація, реалізований типом `CharField`.

Модель хвороб конкретних пацієнтів, яким потрібна реабілітація, реалізована класом Rehabpatientdisease.

Поле rehabpatient в класі визначає пацієнта, якому потрібна реабілітація і який має цю хворобу, реалізованого типом ForeignKey на основі моделі Rehabpatient.

Поле rehabdisease в класі визначає хворобу, яку має цей пацієнт, який потребує реабілітації, реалізовану типом ForeignKey на основі моделі Rehabdisease.

Поле fixdate в класі визначає дату, з якої зафіксована у цього пацієнта, який потребує реабілітації, ця хвороба, реалізовану типом DateField.

Модель можливих послуг з реабілітації реалізована класом Rehabservice.

Поле rehabservicegroup в класі визначає групу, до якої відноситься ця послуга з реабілітації, реалізовану типом ForeignKey на основі моделі Rehabservicegroup.

Поле rehabservicename в класі визначає назву послуги з реабілітації, реалізовану типом CharField.

Модель групи послуг з реабілітації реалізована класом Rehabservicegroup.

Поле rehabservicegroupname в класі визначає назву групи послуг з реабілітації, реалізовану типом CharField.

Модель послуг з реабілітації, які надаються лікарем або працівником при роботі за однією з його професій, реалізована класом Rehabdoctorprofessionservice.

Поле rehabservice в класі визначає послугу з реабілітації, що надається цим лікарем або працівником з реабілітації при роботі за цією професією, реалізовану типом ForeignKey на основі моделі Rehabservice.

Поле rehabdoctorprofession в класі визначає професію, за якою цей лікар або працівник з реабілітації надає цю послугу з реабілітації, реалізовану типом ForeignKey на основі моделі Rehabdoctorprofession.

Поле `rehabserviceinfo` в класі визначає текстову інформацію щодо особливостей роботи цього лікаря або працівника з реабілітації за цією професією, реалізовану типом `TextField`.

Поле `rehabprice` в класі визначає стандартну вартість надання цієї послуги з реабілітації цим лікарем або працівником з реабілітації при роботі за цією професією, реалізовану типом `DecimalField`.

Модель стандартного робочого часу лікаря або працівника з реабілітації за певною професією з реабілітації реалізована класом `Rehabdoctorprofessionworktime`.

Поле `rehabdoctorprofession` в класі визначає професію, за якою цей лікар або працівник з реабілітації, працює в цей час, реалізовану типом `ForeignKey` на основі моделі `Rehabdoctorprofession`.

Поле `rehabworkday` в класі визначає номер робочого дня в тижні, за яким цей лікар або працівник з реабілітації працює за цією професією, реалізований типом `PositiveSmallIntegerField`.

Поле `rehabstartdaytime` в класі визначає час початку роботи цього лікаря або працівника з реабілітації за цією професією в цей робочий день, реалізований типом `TimeField`.

Поле `rehabenddaytime` в класі визначає час закінчення роботи цього лікаря або працівника з реабілітації за цією професією в цей робочий день, реалізований типом `TimeField`.

Модель обладнання, яке може використовуватись для реабілітації, реалізована класом `Rehabequipment`.

Поле `rehabequipmentphoto` в класі визначає фотографію обладнання, яке може використовуватись для реабілітації, реалізовану типом `ImageField`.

Поле `rehabequipmentname` в класі визначає назву обладнання, яке може використовуватись для реабілітації, реалізовану типом `CharField`.

Поле `rehabequipmentinfo` в класі визначає текстовий опис цього обладнання, яке може використовуватись для реабілітації, реалізований типом `TextField`.

Модель реабілітаційної послуги, при виконанні якої цей лікар або працівник з реабілітації застосовує це обладнання, реалізована класом `Rehabdoctorserviceequipment`.

Поле `rehabdoctorprofessionservice` в класі визначає реабілітаційну послугу, при виконанні якої за певною професією цей лікар або працівник з реабілітації використовує це обладнання, реалізовану типом `ForeignKey` на основі моделі `Rehabdoctorprofessionservice`.

Поле `rehabequipment` в класі визначає обладнання, яке використовується при виконанні цієї реабілітаційної послуги за певною професією лікаря або працівника з реабілітації, реалізоване типом `ForeignKey` на основі моделі `Rehabequipment`.

Поле `rehabusageinfo` в класі визначає тестовий опис особливостей застосування цього обладнання для виконання цієї реабілітаційної послуги цим лікарем або працівником з реабілітації, реалізований типом `TextField`.

Модель збережених параметрів пошуку лікарів або послуг з реабілітації для пацієнта реалізована класом `Rehabsearch`.

Поле `rehabpatient` в класі визначає пацієнта, який зберіг ці параметри пошуку лікарів або послуг з реабілітації, реалізованого типом `ForeignKey` на основі моделі `Rehabpatient`.

Поле `rehabsearchname` в класі визначає назву збережених параметрів пошуку лікарів або послуг з реабілітації для пацієнта реалізовану типом `CharField`.

Поле `rehabprofession` в класі визначає професію лікаря або пацієнта з реабілітації, пошук якого здійснюється пацієнтом, реалізовану типом `ForeignKey` на основі моделі `Rehabprofession`.

Поле `rehabdoctorscore` в класі визначає мінімальну оцінку, яку повинен мати лікар або пацієнт з реабілітації, реалізовану типом `DecimalField`.

Поле `rehabdiseases` в класі визначає хвороби, за якими пацієнт потребує реабілітації та за якими виконує пошук лікарів або послуг з реабілітації,

реалізовані типом `ManyToManyField` на основі моделі `Rehabdisease`, таким чином забезпечуючи зв'язок багато-до-багатьох.

Модель послуг з реабілітації, за якими здійснюється пошук пацієнтом у збережених параметрах пошуку, реалізована класом `Rehabsearchservice`.

Поле `rehabsearch` в класі визначає збережені параметри пошуку лікарів або послуг з реабілітації для пацієнта, відносно яких визначена ця послуга з реабілітації, реалізовані типом `ForeignKey` на основі моделі `Rehabsearch`.

Поле `rehabservice` в класі визначає послугу з реабілітації, за якою здійснюється пошук пацієнтом у збережених параметрах пошуку, реалізовану типом `ForeignKey` на основі моделі `Rehabservice`.

Поле `rehabequipments` в класі визначає обладнання, яке має застосовуватись при виконанні послуг з реабілітації, за збереженими пацієнтом параметрами пошуку, реалізоване типом `ManyToManyField` на основі моделі `Rehabequipment`, таким чином забезпечуючи зв'язок багато-до-багатьох.

Модель доступних часових проміжків у роботі лікаря або працівника з реабілітації реалізована класом `Rehabdoctorprofessiontimeslot`.

Поле `rehabdoctorprofession` в класі визначає професію лікаря або працівника з реабілітації, за якою доступний цей часовий проміжок, реалізовану типом `ForeignKey` на основі моделі `Rehabdoctorprofession`.

Поле `timeslotstart` в класі визначає час початку часового проміжку в роботі лікаря або працівника з реабілітації реалізований типом `DateTimeField`.

Поле `timeslotend` в класі визначає час закінчення часового проміжку в роботі лікаря або працівника з реабілітації, реалізований типом `DateTimeField`.

Модель записів пацієнта на прийом до лікаря або працівника з реабілітації реалізована класом `Rehabdoctortimeslotapplication`.

Поле `rehabdoctorprofessiontimeslot` в класі визначає доступний часовий проміжок у роботі лікаря або працівника з реабілітації, за яким відбувся запис пацієнта на прийом, реалізований типом `ForeignKey` на основі моделі `Rehabdoctorprofessiontimeslot`.

Поле `rehabpatient` в класі визначає пацієнта, який записався на прийом до цього лікаря або працівник з реабілітації, реалізованого типом `ForeignKey` на основі моделі `Rehabpatient`.

Поле `rehabstatus` в класі визначає стан запису пацієнта на прийом до лікаря або працівника з реабілітації, реалізований типом `CharField`.

Поле `rehabprescription` в класі визначає рекомендацію щодо лікування пацієнта за результатами прийому, надану лікарем або працівником з реабілітації, реалізовану типом `CharField`.

Поле `rehabdoctorprofessionservice` в класі визначає послуга з реабілітації, яка має бути надана лікарем або працівником з реабілітації пацієнту, реалізовану типом `ForeignKey` на основі моделі `Rehabdoctorprofessionservice`.

Поле `rehabpatienthome` в класі визначає адресу проживання пацієнта, за якою має бути надано цю послугу з реабілітації лікарем або працівником з реабілітації, реалізовану типом `ForeignKey` на основі моделі `Rehabpatienthome`.

Модель обладнання, яке запланував для використання під час цього прийому лікар або працівник з реабілітації, реалізована класом `Rehabdoctortimeslotapplicationequipment`.

Поле `rehabdoctortimeslotapplication` в класі визначає запис пацієнта на прийом до лікаря або працівника з реабілітації, за яким заплановано застосувати це обладнання, реалізований типом `ForeignKey` на основі моделі `Rehabdoctortimeslotapplication`.

Поле `rehabdoctorserviceequipment` в класі визначає обладнання, яке застосовується лікарем або працівником з реабілітації для цієї послуги з реабілітації та яке заплановано для застосування під час цього прийому, реалізоване типом `ForeignKey` на основі моделі `Rehabdoctorserviceequipment`.

Поле `rehabdoctorown` в класі визначає визначення, чи це власне обладнання лікаря, чи це обладнання пацієнта, реалізоване типом `BooleanField`.

Модель відгуків за результатами прийомів пацієнтів у лікарів або працівників з реабілітації для надання реабілітаційних послуг реалізована класом Rehabview.

Поле rehabdoctortimeslotapplication в класі визначає прийом пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг, за яким створено відгук, реалізований типом ForeignKey на основі моделі Rehabdoctortimeslotapplication.

Поле rehabdoctororpatient в класі встановлює визначення, чи це лікар створив відгук про пацієнта, чи навпаки пацієнт про лікаря або працівника з реабілітації, реалізоване типом BooleanField.

Поле rehabviewscore в класі визначає оцінку відгуку за результатом прийому пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг, реалізовану типом PositiveSmallIntegerField.

Поле rehabviewtext в класі визначає текст відгуку за результатом прийому пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг, реалізований типом CharField.

Поле rehabviewtime в класі визначає час відгуку за результатом прийому пацієнта у лікаря або працівника з реабілітації для надання реабілітаційних послуг, реалізований типом DateTimeField.

3.3 Висновки за розділом 3

Повна реалізація програми надання індивідуальних реабілітаційних послуг вдома виконана, порядок роботи з сутністю запису на прийом до лікаря з реабілітації визначено, що враховано під час реалізації функцій програми, які зокрема описані в цьому розділі.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми

Розроблена програма призначена для підтримки надання індивідуальних реабілітаційних послуг з врахуванням особливостей реалізації послуг вдома у пацієнта. Для цього програма забезпечує пацієнтів засобами пошуку лікарів з реабілітації, послуг з реабілітації, запису на прийом до лікарів з врахуванням графіку роботи з виїздами додому до пацієнтів, лікарів – засобами контролю свого графіку роботи, керування записами до них на прийом та відслідковування плану роботи, визначення рекомендованих процедур для подальшого лікування.

4.2 Опис роботи з програмою підтримки надання індивідуальних реабілітаційних послуг вдома

Організація роботи користувача з програмою підтримки надання індивідуальних реабілітаційних послуг вдома виконана наступним чином. Після початкового звернення користувач до виконання авторизації має одразу можливість працювати над пошуком лікарів з реабілітації, а також перемкнутися на пошук послуг з реабілітації, виконуючи це за допомогою меню, яке розташоване зліва. Відповідно з результатів пошуку можна переходити на окремі сторінки. Зокрема це сторінки лікаря з реабілітації (рис. 4.1), медичного центру.

Сторінка лікаря з реабілітації містить можливості збереження цього лікаря до свого списку, переходу до пошуку часу прийому, переходу до переліку всіх послуг реабілітації, що надаються лікарем, при цьому обираючи відповідну професію, переходу до відгуків про лікаря, до дипломів або сертифікатів. При цьому частина відгуків, частина послуг з реабілітації та частина дипломів або сертифікатів знаходиться безпосередньо на сторінці лікаря з реабілітації.

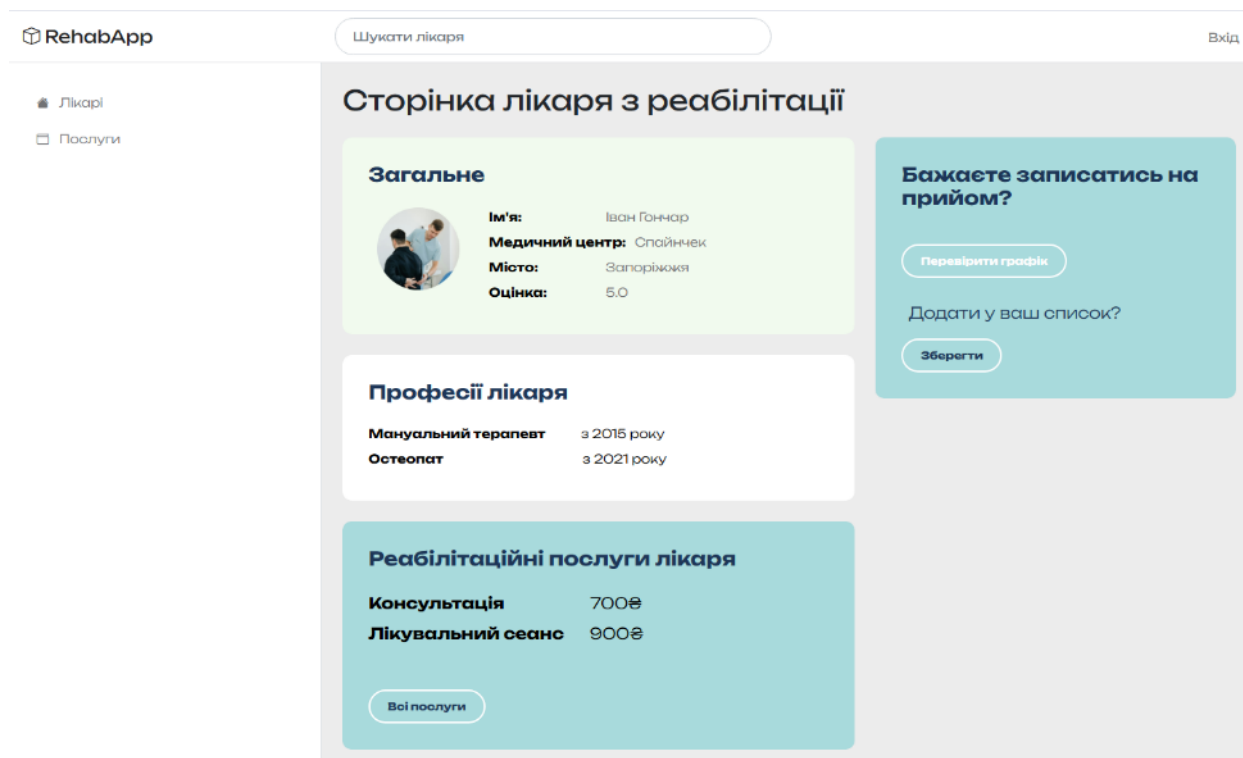


Рисунок 4.1 – Сторінка лікаря з реабілітації для неавторизованого користувача

Для авторизації використовується посилання «Вхід» у верхньому рядку застосунку у правому крайньому кутку. Після авторизації, якщо користувач є лікарем з реабілітації, то він працює з відповідним меню і засобами (рис. 4.2). Наприклад, у такому випадку його особиста сторінка лікаря з реабілітації доступна йому з меню за розділом «Про себе», а безпосередньо на ній доступні засоби редагування, додавання даних. Так можна відредагувати дані про себе як про лікаря загалом, додати дипломи або сертифікати, послуги, професії, вилучити існуючі.

З розділу прийомів лікарю доступний нижче вибір конкретного стану, включаючи заплановані, реалізовані, нереалізовані та скасовані (рис. 4.3). При цьому він має обрати за необхідності проміжок дат, прийоми у нього за якими потрібно відобразити. Якщо не обрано, то відображаються всі. Також обирається напрямок сортування. Якщо записи на прийоми заплановані, то можна визначити прибуття пацієнта (реалізований), неприбуття (нереалізований), скасувати візит, визначити дані візиту (рекомендації тощо).

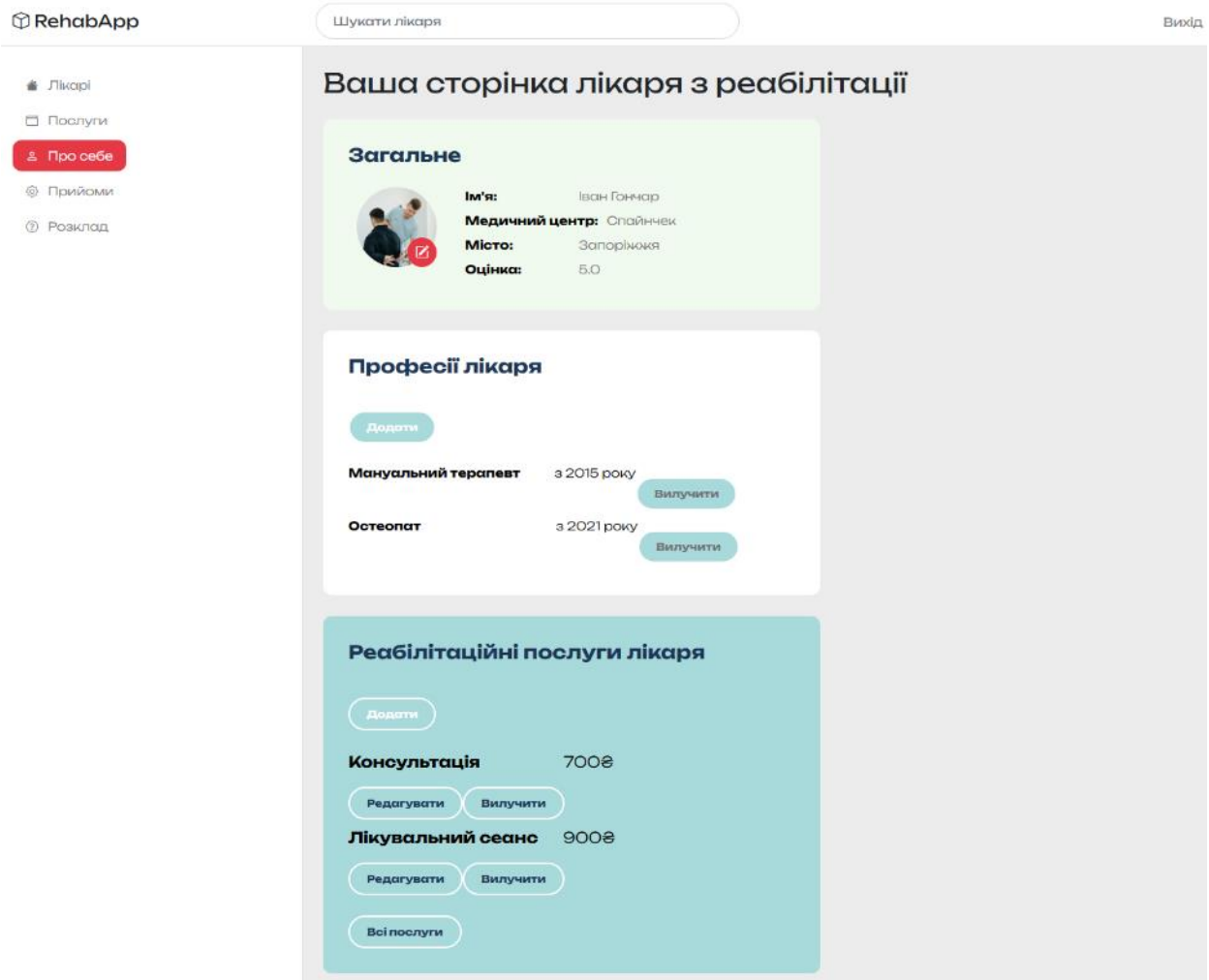


Рисунок 4.2 – Сторінка лікаря з реабілітації для самого лікаря

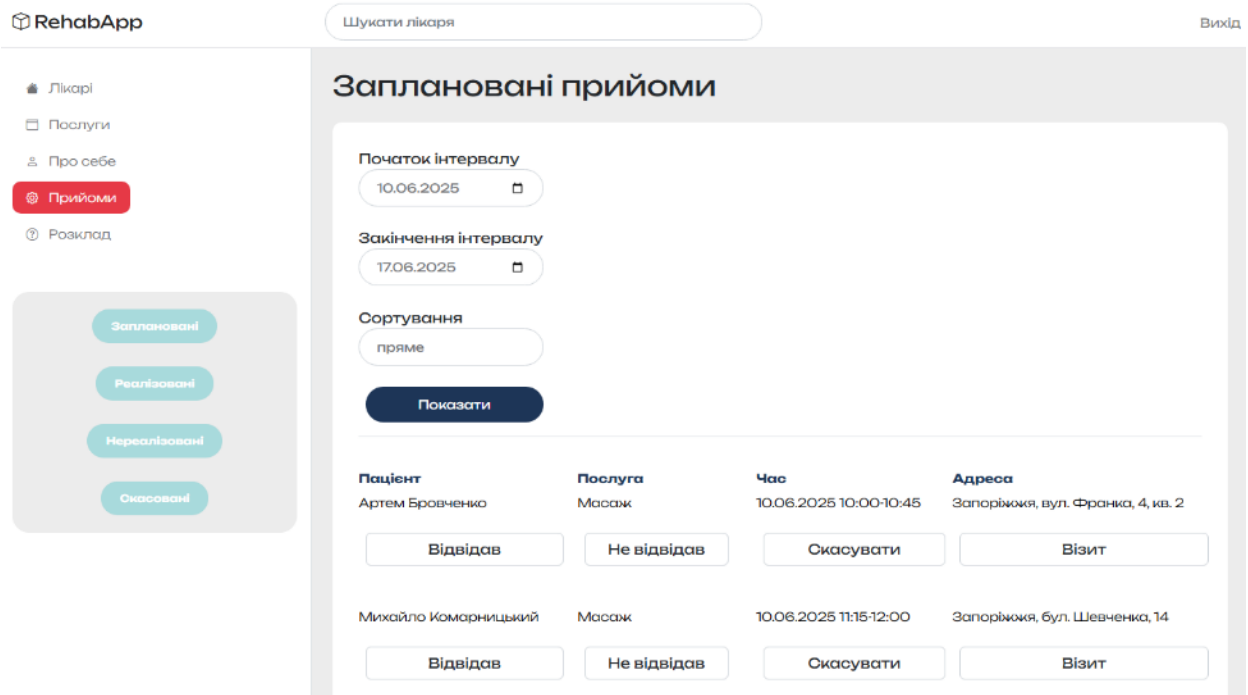


Рисунок 4.3 – Перегляд всіх запланованих прийомів у лікаря з реабілітації

У випадку, якщо запис на прийом є в одному з кінцевих станів (скасований, реалізований, нереалізований), лікар з реабілітації та пацієнт можуть залишити відгук про іншу сторону.

Пацієнт також може переглядати прийоми у лікарів з реабілітації за такими ж станами, але вже тільки власні. Якщо для лікаря з реабілітації доступний графік його роботи, звідки він може задати для кожного дня тижня стандартні години прийому, передивитись доступні на поточний момент часові проміжки для прийому, вилучити або змінити такі проміжки, то для пацієнта такі засоби звісно не представлені, але доступний розділ меню «Мої лікарі», звідки він може переглянути раніше збережених лікарів з реабілітації, вилучити такого лікаря, якщо до нього більше не потрібно повертатись (рис. 4.4). З розділу «Про себе» пацієнт може перейти зокрема до власних медичних засобів, що в такому випадку через посилання на сторінці доступне йому для роботи.

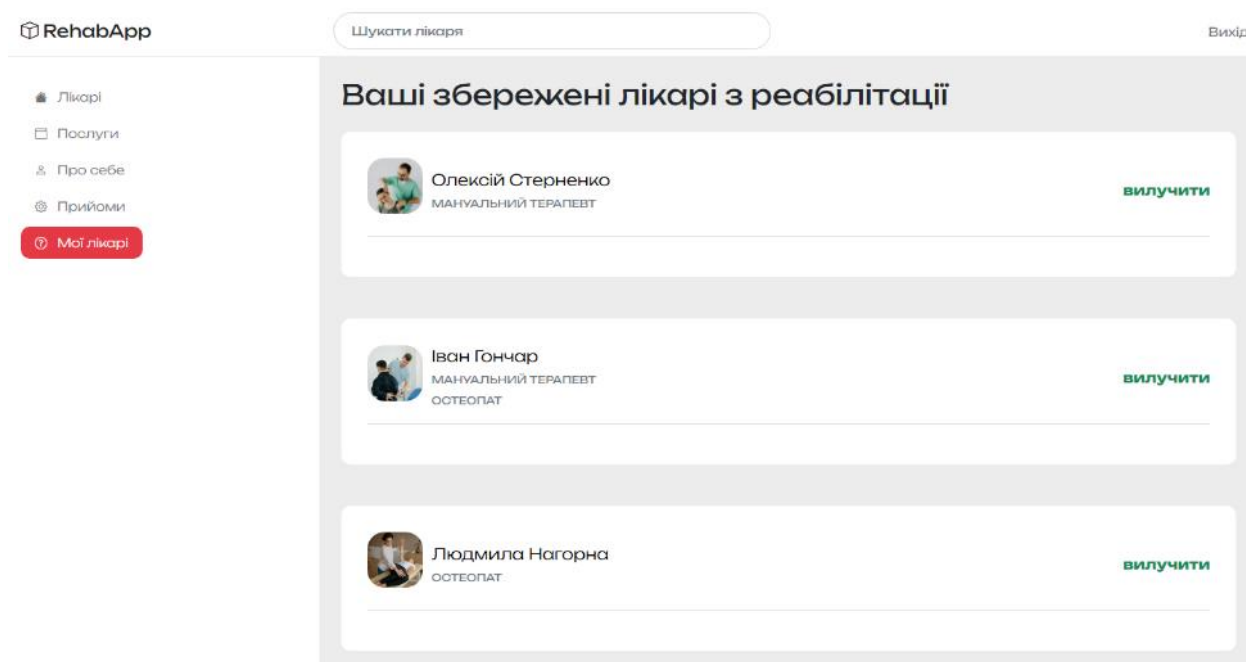


Рисунок 4.4 – Перегляд збережених лікарів з реабілітації пацієнтом

4.3 Тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома

Важливим для перевірки виконання роботи було забезпечення тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома. Оскільки набір функцій є обширним, то за кожною окремо проводилось тестування. Якщо функція доступна для різних ролей користувачів, то відповідно окремо вона виконувалась для кожної такої ролі. Результати були такими, що кожна функція в підсумку під час тестування дала успішний результат (табл. 4.1).

Таблиця 4.1 – Результати тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома

Функція	Неавто- ризований користувач	Лікар з реабі- літації	Пацієнт
1	2	3	4
Перегляд сторінки реабілітаційного центру	+	+	+
Перегляд лікарів реабілітаційного центру	+	+	+
Перегляд сторінки лікаря з реабілітації	+	+	+
Авторизація	+	+	+
Перегляд сторінки пацієнта	+	+	+
Пошук лікаря для реабілітації	+	+	+
Пошук реабілітаційних послуг	+	+	+
Реєстрація	+	+	+

Продовження таблиці 4.1

1	2	3	4
Перегляд всіх дипломів або сертифікатів лікаря з реабілітації	+	+	+
Перегляд медичних записів пацієнта	+	+	+
Перегляд доступних послуг з реабілітації у лікаря	+	+	+
Перегляд доступних часових проміжків прийому в лікаря з реабілітації за датою	+	+	+
Перегляд відгуків від лікарів з реабілітації про пацієнта	+	+	+
Додавання медичного запису пацієнта		+	
Додавання хвороби пацієнта для реабілітації		+	
Додавання послуги з реабілітації для лікаря		+	
Визначення стандартного часу роботи лікаря з реабілітації		+	
Редагування доступного часового проміжку прийому в лікаря з реабілітації		+	
Внесення результатів відвідування пацієнтом лікаря з реабілітації		+	
Перегляд всіх запланованих прийомів у лікаря з реабілітації		+	

Продовження таблиці 4.1

1	2	3	4
Перегляд всіх реалізованих прийомів у лікаря з реабілітації		+	
Перегляд всіх нереалізованих прийомів у лікаря з реабілітації		+	
Перегляд всіх скасованих прийомів у лікаря з реабілітації		+	
Визначення фактичного прийому пацієнта лікарем з реабілітації		+	
Створення відгуку про пацієнта, що потребує реабілітації, за результатами прийому		+	
Вилучення професії з реабілітації лікаря		+	
Вилучення диплома або сертифіката лікаря з реабілітації		+	
Редагування даних диплома або сертифіката лікаря з реабілітації		+	
Редагування медичного запису пацієнта		+	
Вилучення медичного запису пацієнта		+	
Вилучення хвороби пацієнта для реабілітації		+	
Редагування послуги з реабілітації для лікаря		+	
Вилучення послуги з реабілітації для лікаря		+	

Продовження таблиці 4.1

1	2	3	4
Вилучення доступного часового проміжку прийому в лікаря з реабілітації		+	
Визначення нереалізованого прийому пацієнта лікарем з реабілітації		+	
Скасування запису на прийом до лікаря з реабілітації		+	+
Внесення інформації про пацієнта, що потребує реабілітації			+
Збереження параметрів пошуку лікаря для реабілітації			+
Збереження параметрів пошуку реабілітаційних послуг			+
Пошук лікаря для реабілітації за допомогою збережених параметрів			+
Пошук реабілітаційних послуг за допомогою збережених параметрів			+
Перегляд збережених параметрів пошуку пацієнтом			+
Додавання адреси проживання пацієнта			+
Додавання хвороби пацієнта для реабілітації			+
Збереження сторінки лікаря з реабілітації пацієнтом			+

Кінець таблиці 4.1

1	2	3	4
Перегляд збережених лікарів з реабілітації			+
Запис на прийом до лікаря з реабілітації			+
Перегляд всіх запланованих прийомів пацієнта			+
Перегляд всіх реалізованих прийомів пацієнта			+
Перегляд всіх нереалізованих прийомів пацієнта			+
Перегляд всіх скасованих прийомів пацієнта			+
Створення відгуку про лікаря з реабілітації за результатами прийому			+
Вилучення збережених параметрів пошуку лікаря для реабілітації			+
Вилучення збережених параметрів пошуку реабілітаційних послуг			+
Вилучення адреси проживання пацієнта			+
Редагування адреси проживання пацієнта			+
Вилучення хвороби пацієнта для реабілітації			+
Скасування збереження сторінки лікаря з реабілітації пацієнтом			+

4.4 Висновки за розділом 4

Роботу всіх функцій програми підтримки надання індивідуальних реабілітаційних послуг вдома, які було необхідно реалізувати за результатами моделювання в другому розділі, реалізовано повністю. Цей факт перевірено на основі виконаного тестування. Його забезпечено шляхом перевірки роботи кожної функції для відповідної ролі користувача.

ВИСНОВКИ

Розроблено ПЗ підтримки надання індивідуальних реабілітаційних послуг для взаємодії між лікарями з реабілітації та пацієнтами на умовах реалізації реабілітаційних послуг у пацієнтів вдома. Для цього спочатку було виконано аналіз стану проблеми розробки ПЗ підтримки надання індивідуальних реабілітаційних послуг вдома. Аналіз показав, що існуючі програми більшою мірою опосередковано надають підтримку реабілітаційних послуг, переважно така підтримка виражена окремими статичними сторінками з інформацією про таку можливість. Деякі дозволяють бронювати відвідування лікарів, проте ніяка специфіка реабілітаційних послуг вдома не враховується. Єдиний варіант врахування таких можливостей пов'язаний з платформою OLX, яка втім не виділяє окремо реабілітаційні послуги серед інших послуг.

У роботі детально представлені результати проєктування бази даних для програми підтримки надання реабілітаційних послуг вдома, визначено необхідні для реалізації сценарії для базових ролей користувачів, включаючи пацієнтів та лікарів (або працівників з реабілітації), неавторизованих користувачів. Визначено Python-ФВР Django для реалізації.

Розроблена програма призначена для підтримки надання індивідуальних реабілітаційних послуг з врахуванням особливостей реалізації послуг вдома у пацієнта. Для цього програма забезпечує пацієнтів засобами пошуку лікарів з реабілітації, послуг з реабілітації, запису на прийом до лікарів з врахуванням графіку роботи з виїздами додому до пацієнтів, лікарів – засобами контролю свого графіку роботи, керування записами до них на прийом та відслідковування плану роботи, визначення рекомендованих процедур для подальшого лікування.

Роботу всіх функцій програми реалізовано повністю. Цей факт перевірено на основі виконаного тестування. Його забезпечено шляхом перевірки роботи кожної функції для відповідної ролі користувача.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Реабілітація не виходячи з дому. Bodro Clinic [Електрон. ресурс]. – Режим доступу : https://bodroclinic.com.ua/reabilitaciya-ne-vyhodyachy-z-domu/?srsltid=AfmBOoo95o5MI_3KgXyI599Aw_WadjxslN6iZx7ugiklK0ymmuz3xtlY.
2. Домашня реабілітація. Вїзди реабілітолога на дому у хворого. Фізіо-Хелп [Електрон. ресурс]. – Режим доступу : <https://fizjo-pomoc.pl/uk/terapije-i-uslugi/rehabilitacja-domowa/>.
3. Rehab at Home [Electronic resource]. – Access mode : <https://www.ramsayconnect.com.au/Services/Rehab-at-Home>.
4. Rehabilitation in the Home – Amplar Health [Electronic resource]. – Access mode : <https://amplarhealth.com.au/health-solutions/at-home-care/rehabilitation-in-the-home/>.
5. Реабілітація – Бізнес та послуги – OLX.ua [Електрон. ресурс]. – Режим доступу : <https://www.olx.ua/uk/uslugi/q-%D1%80%D0%B5%D0%B0%D0%B1%D1%96%D0%BB%D1%96%D1%82%D0%B0%D1%86%D1%96%D1%8F/?currency=UAH>.
6. Flask Tutorial. GeeksforGeeks [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/flask-tutorial/>.
7. Django Tutorial. Learn Django Framework. GeeksforGeeks [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/django-tutorial/>.
8. Mini Finance HTML Template by Tooplate [Electronic resource]. – Access mode : <https://www.tooplate.com/view/2135-mini-finance>.

ДОДАТОК А
Технічне завдання

Вступ

Важливість реабілітаційних послуг важко переоцінити в Україні в сьогодення. Сучасний спосіб життя звичайної людини взагалі характеризується тим, що велика кількість людей зі зрілого віку починає користуватися частиною послуг з реабілітації. З іншого боку велика кількість людей в Україні зараз страждає і фізично через російську агресію. Це призводить до того, що коло таких осіб збільшується. Частині з них, зокрема зважаючи на проблему з забезпеченням мобільності людей з інвалідністю, незручним є дістатися до медичного центру. Частина з них знаходиться у складному фізичному стані в певний момент. Окрім того велика кількість інших людей може не мати часу на переміщення. Все це актуалізує потребу в реабілітаційних послугах вдома у пацієнта. Це ж накладає і додаткові вимоги на програму, яка таку роботу має підтримувати. Важливо розуміти, що з одного боку без такої програми популяризувати цей напрямок, цю діяльність неможливо, а з іншого боку – важливо, щоб вона підтримувала особливості запису на прийом за таких послуг. Реабілітаційні послуги вдома зокрема пов'язані з тим, що лікар витрачає додатковий час на переміщення між пацієнтами. Тому необхідно будувати графік з врахуванням цього. Загалом все це вказує на актуальність, важливість і необхідність розроблення програми підтримки надання індивідуальних реабілітаційних послуг вдома.

A.1 Підстави для розробки

Наказ № 209 від 28 квітня 2025 року був виданий у Національному університеті «Запорізька політехніка», визначивши для виконання цієї дипломної кваліфікаційної роботи бакалавра тему «Програмне забезпечення підтримки надання індивідуальних реабілітаційних послуг вдома», яку було далі уточнено у вигляді завдання на дипломну кваліфікаційну роботу бакалавра.

A.2 Призначення розробки

Програма, що розробляється, призначена для підтримки надання індивідуальних реабілітаційних послуг з врахуванням особливостей реалізації послуг вдома у пацієнта. Для цього програма повинна забезпечувати пацієнтів засобами пошуку лікарів з реабілітації, послуг з реабілітації, запису на прийом до лікарів з врахуванням графіку роботи з виїздами додому до пацієнтів, лікарям – контролювати свій графік роботи, керувати записами до них на прийом та відслідковувати план роботи, визначити рекомендовані процедури для подальшого лікування.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма підтримки надання індивідуальних реабілітаційних послуг вдома повинна надавати функції:

- перегляд сторінки реабілітаційного центру;
- перегляд лікарів реабілітаційного центру;
- перегляд сторінки лікаря з реабілітації;
- авторизація лікарів з реабілітації та пацієнтів;
- реєстрація лікарів з реабілітації та пацієнтів;
- перегляд сторінки пацієнта;
- внесення інформації про пацієнта, що потребує реабілітації;
- внесення інформації про лікаря з реабілітації;
- пошук лікаря для реабілітації;
- пошук реабілітаційних послуг;
- збереження параметрів пошуку лікаря для реабілітації;
- збереження параметрів пошуку реабілітаційних послуг;
- пошук лікаря для реабілітації за допомогою збережених параметрів;
- пошук реабілітаційних послуг за допомогою збережених параметрів;

- перегляд збережених параметрів пошуку пацієнтом;
- вилучення збережених параметрів пошуку лікаря для реабілітації;
- вилучення збережених параметрів пошуку реабілітаційних послуг;
- додавання професії з реабілітації лікарю;
- вилучення професії з реабілітації лікаря;
- додавання диплома або сертифіката лікаря з реабілітації;
- перегляд всіх дипломів або сертифікатів лікаря з реабілітації;
- вилучення диплома або сертифіката лікаря з реабілітації;
- редагування даних диплома або сертифіката лікаря з реабілітації;
- додавання адреси проживання пацієнта;
- вилучення адреси проживання пацієнта;
- редагування адреси проживання пацієнта;
- додавання лікарем з реабілітації медичного запису пацієнта;
- перегляд медичних записів пацієнта;
- редагування лікарем з реабілітації медичного запису пацієнта;
- вилучення лікарем з реабілітації медичного запису пацієнта;
- додавання хвороби пацієнта для реабілітації;
- вилучення хвороби пацієнта для реабілітації;
- додавання послуги з реабілітації для лікаря;
- редагування послуги з реабілітації для лікаря;
- вилучення послуги з реабілітації для лікаря;
- перегляд доступних послуг з реабілітації у лікаря;
- визначення стандартного часу роботи лікаря з реабілітації;
- збереження сторінки лікаря з реабілітації пацієнтом;
- скасування збереження сторінки лікаря з реабілітації пацієнтом;
- перегляд збережених лікарів з реабілітації;
- редагування доступного часового проміжку прийому в лікаря з реабілітації;
- вилучення доступного часового проміжку прийому в лікаря з реабілітації;

- перегляд доступних часових проміжків прийому в лікаря з реабілітації за датою;
- запис на прийом до лікаря з реабілітації;
- внесення результатів відвідування пацієнтом лікаря з реабілітації;
- скасування запису на прийом до лікаря з реабілітації;
- перегляд всіх запланованих прийомів у лікаря з реабілітації;
- перегляд всіх реалізованих прийомів у лікаря з реабілітації;
- перегляд всіх нереалізованих прийомів у лікаря з реабілітації;
- перегляд всіх скасованих прийомів у лікаря з реабілітації;
- перегляд всіх запланованих прийомів пацієнта;
- перегляд всіх реалізованих прийомів пацієнта;
- перегляд всіх нереалізованих прийомів пацієнта;
- перегляд всіх скасованих прийомів пацієнта;
- визначення фактичного прийому пацієнта лікарем з реабілітації;
- визначення нереалізованого прийому пацієнта лікарем з реабілітації;
- створення відгуку про лікаря з реабілітації за результатами прийому;
- створення відгуку про пацієнта, що потребує реабілітації, за результатами прийому лікарем;
- перегляд відгуків від лікарів з реабілітації про пацієнта;
- перегляд відгуків від пацієнтів про лікаря з реабілітації.

А.3.2 Умови експлуатації

Програма не має особливих вимог до користувачів, до клієнтської сторони. При цьому на сервері, враховуючи наявність сервера зокрема, має бути забезпечена роботою адміністратора. Така вимога обумовлена необхідністю початкового розгортання застосунку на сервері, а для цього потрібні спеціалізовані знання, а також необхідність керування застосунком через адміністративну панель.

А.3.3 Вимоги до складу та параметрів технічних засобів

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

ДОДАТОК Б
Текст програми

```

from .models import City
from .models import Rehabcenter
from .models import Rehabprofession
from .models import Rehabdoctorprofession
from .models import Rehabdoctorprofessiondiploma
from .models import Rehabdoctor
from .models import Rehabpatient
from .models import Rehabpatienthome
from .models import Rehabpatientmedrecord
from .models import Rehabdisease
from .models import Rehabpatientdisease
from .models import Rehabservice
from .models import Rehabservicegroup
from .models import Rehabdoctorprofessionservice
from .models import Rehabdoctorprofessionworktime
from .models import Rehabequipment
from .models import Rehabdoctorserviceequipment
from .models import Rehabsearch
from .models import Rehabsearchservice
from .models import Rehabdoctorprofessiontimeslot
from .models import Rehabdoctortimeslotapplication
from .models import Rehabdoctortimeslotapplicationequipment
from .models import Rehabview
from django.core.paginator import Paginator
from django.shortcuts import render
from datetime import datetime

```

```

def rehab_planned_doctor_applications_view(request, rehabstart,
rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = request.user)
        except Rehabdoctor.DoesNotExist:

```

```

        rehab_doctor = None
    else:
        rehab_doctor = None
    if not rehab_doctor:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати заплановані прийоми лікаря тільки
тоді, коли ви увійшли для роботи як лікар. Інакше це неможливо."})
        rehab_doctor_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession
",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabdoctorprofessiontimeslot__rehabdoctorp
rofession__rehabdoctor = rehab_doctor)
        rehab_doctor_applications =
rehab_doctor_applications.filter(rehabstatus = "0")
        if rehabstart:
            rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otstart__gte = rehabstart)
        if rehabend:
            rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otend__lte = rehabend)
        if sortordering == 0:
            rehab_doctor_applications =
rehab_doctor_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 1:
            rehab_doctor_applications =
rehab_doctor_applications.order_by("rehabdoctorprofessiontimeslot__tim
eslotstart")
        elif sortordering == 2:
            current_now_time = datetime.now().date()
            rehab_doctor_applications =
rehab_doctor_applications.filter(("rehabdoctorprofessiontimeslot__time
slotstart__gte = current_now_time)

```

```

        rehab_doctor_applications =
rehab_doctor_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 3:
            current_now_time = datetime.now().date()
            rehab_doctor_applications =
rehab_doctor_applications.filter(("rehabdoctorprofessiontimeslot__time
slotstart__gte = current_now_time)
            rehab_doctor_applications =
rehab_doctor_applications.order_by("rehabdoctorprofessiontimeslot__tim
eslotstart")
        return render(request, "rehabapp/plandoctorapps.html",
{"applications": rehab_doctor_applications })

```

```

def rehab_finalized_doctor_applications_view(request,
rehabstart, rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = request.user)
        except Rehabdoctor.DoesNotExist:
            rehab_doctor = None
    else:
        rehab_doctor = None
    if not rehab_doctor:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати реалізовані прийоми лікаря тільки
тоді, коли ви увійшли для роботи як лікар. Інакше це неможливо."})
        rehab_doctor_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession

```

```

",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabdoctorprofessiontimeslot__rehabdoctorp
rofession__rehabdoctor = rehab_doctor)
    rehab_doctor_applications =
rehab_doctor_applications.filter(rehabstatus = "2")
    if rehabstart:
        rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otstart__gte = rehabstart)
    if rehabend:
        rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otend__lte = rehabend)
    if sortordering == 0:
        rehab_doctor_applications =
rehab_doctor_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
    elif sortordering == 1:
        rehab_doctor_applications =
rehab_doctor_applications.order_by("rehabdoctorprofessiontimeslot__tim
eslotstart")
    return render(request, "rehabapp/finddoctorapps.html",
{"applications": rehab_doctor_applications })

```

```

def rehab_unfinalized_doctor_applications_view(request,
rehabstart, rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = request.user)

```

```

        except Rehabdoctor.DoesNotExist:
            rehab_doctor = None
    else:
        rehab_doctor = None
    if not rehab_doctor:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати нереалізовані прийоми лікаря тільки
тоді, коли ви увійшли для роботи як лікар. Інакше це неможливо."})
        rehab_doctor_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession
",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabdoctorprofessiontimeslot__rehabdoctorp
rofession__rehabdoctor = rehab_doctor)
        rehab_doctor_applications =
rehab_doctor_applications.filter(rehabstatus = "3")
        if rehabstart:
            rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otstart__gte = rehabstart)
        if rehabend:
            rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otend__lte = rehabend)
        if sortordering == 0:
            rehab_doctor_applications =
rehab_doctor_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 1:
            rehab_doctor_applications =
rehab_doctor_applications.order_by("rehabdoctorprofessiontimeslot__tim
eslotstart")
        return render(request, "rehabapp/unfindoctorapps.html",
{"applications": rehab_doctor_applications })

```

```

def rehab_cancelled_doctor_applications_view(request,
rehabstart, rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = request.user)
        except Rehabdoctor.DoesNotExist:
            rehab_doctor = None
    else:
        rehab_doctor = None
    if not rehab_doctor:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати скасовані прийоми лікаря тільки
тоді, коли ви увійшли для роботи як лікар. Інакше це неможливо."})
        rehab_doctor_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession
",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabdoctorprofessiontimeslot__rehabdoctorp
rofession__rehabdoctor = rehab_doctor)
        rehab_doctor_applications =
rehab_doctor_applications.filter(rehabstatus = "1")
        if rehabstart:
            rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otstart__gte = rehabstart)
        if rehabend:

```

```

        rehab_doctor_applications =
rehab_doctor_applications.filter(rehabdoctorprofessiontimeslot__timesl
otend__lte = rehabend)
        if sortordering == 0:
            rehab_doctor_applications =
rehab_doctor_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 1:
            rehab_doctor_applications =
rehab_doctor_applications.order_by("rehabdoctorprofessiontimeslot__tim
eslotstart")
        elif sortordering == 2:
            current_now_time = datetime.now().date()
            rehab_doctor_applications =
rehab_doctor_applications.filter(("rehabdoctorprofessiontimeslot__time
slotstart__gte = current_now_time)
            rehab_doctor_applications =
rehab_doctor_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 3:
            current_now_time = datetime.now().date()
            rehab_doctor_applications =
rehab_doctor_applications.filter(("rehabdoctorprofessiontimeslot__time
slotstart__gte = current_now_time)
            rehab_doctor_applications =
rehab_doctor_applications.order_by("rehabdoctorprofessiontimeslot__tim
eslotstart")
        return render(request, "rehabapp/canceldoctorapps.html",
{"applications": rehab_doctor_applications })

```

```

def rehab_planned_patient_applications_view(request, rehabstart,
rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:

```

```

        rehab_patient =
Rehabpatient.objects.get(patientdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    else:
        rehab_patient = None
    if not rehab_patient:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати заплановані прийоми пацієнта тільки
тоді, коли ви увійшли для роботи як пацієнт. Інакше це неможливо."})
        rehab_patient_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession
",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabpatient = rehab_patient)
        rehab_patient_applications =
rehab_patient_applications.filter(rehabstatus = "0")
        if rehabstart:
            rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotstart__gte = rehabstart)
        if rehabend:
            rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotend__lte = rehabend)
        if sortordering == 0:
            rehab_patient_applications =
rehab_patient_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 1:
            rehab_patient_applications =
rehab_patient_applications.order_by("rehabdoctorprofessiontimeslot__ti
meslotstart")
        elif sortordering == 2:
            current_now_time = datetime.now().date()

```

```

        rehab_patient_applications =
rehab_patient_applications.filter(("rehabdoctorprofessiontimeslot__tim
eslotstart__gte = current_now_time)
        rehab_patient_applications =
rehab_patient_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 3:
            current_now_time = datetime.now().date()
            rehab_patient_applications =
rehab_patient_applications.filter(("rehabdoctorprofessiontimeslot__tim
eslotstart__gte = current_now_time)
            rehab_patient_applications =
rehab_patient_applications.order_by("rehabdoctorprofessiontimeslot__ti
meslotstart")
        return render(request, "rehabapp/planpatapps.html",
{"applications": rehab_patient_applications })

```

```

def rehab_finalized_patient_applications_view(request,
rehabstart, rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:
            rehab_patient =
Rehabpatient.objects.get(patientdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    else:
        rehab_patient = None
    if not rehab_patient:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати реалізовані прийоми пацієнта тільки
тоді, коли ви увійшли для роботи як пацієнт. Інакше це неможливо."})
    rehab_patient_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",

```

```

"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession
",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabpatient = rehab_patient)
        rehab_patient_applications =
rehab_patient_applications.filter(rehabstatus = "2")
        if rehabstart:
            rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotstart__gte = rehabstart)
            if rehabend:
                rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotend__lte = rehabend)
            if sortordering == 0:
                rehab_patient_applications =
rehab_patient_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
            elif sortordering == 1:
                rehab_patient_applications =
rehab_patient_applications.order_by("rehabdoctorprofessiontimeslot__ti
meslotstart")
        return render(request, "rehabapp/finpatapps.html",
{"applications": rehab_patient_applications })

```

```

def rehab_unfinalized_patient_applications_view(request,
rehabstart, rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:

```

```

        rehab_patient =
Rehabpatient.objects.get(patientdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    else:
        rehab_patient = None
    if not rehab_patient:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати нереалізовані прийоми пацієнта
тільки тоді, коли ви увійшли для роботи як пацієнт. Інакше це
неможливо."})
        rehab_patient_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession
",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabpatient = rehab_patient)
        rehab_patient_applications =
rehab_patient_applications.filter(rehabstatus = "3")
        if rehabstart:
            rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotstart__gte = rehabstart)
            if rehabend:
                rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotend__lte = rehabend)
            if sortordering == 0:
                rehab_patient_applications =
rehab_patient_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
            elif sortordering == 1:
                rehab_patient_applications =
rehab_patient_applications.order_by("rehabdoctorprofessiontimeslot__ti
meslotstart")
        return render(request, "rehabapp/unfinpatapps.html",
{"applications": rehab_patient_applications })

```

```

def rehab_cancelled_patient_applications_view(request,
rehabstart, rehabend, sortordering = 0):
    if rehabstart:
        rehabstart = datetime.strptime(rehabstart, '%d-%m-%Y')
    else:
        rehabstart = None
    if rehabend:
        rehabend = datetime.strptime(rehabend, '%d-%m-%Y')
    else:
        rehabend = None
    if request.user.is_authenticated:
        try:
            rehab_patient =
Rehabpatient.objects.get(patientdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    else:
        rehab_patient = None
    if not rehab_patient:
        return render(request, "rehabapp/error.html",
{"caption": "Ви можете переглядати нереалізовані прийоми пацієнта
тільки тоді, коли ви увійшли для роботи як пацієнт. Інакше це
неможливо."})
    rehab_patient_applications =
Rehabdoctortimeslotapplication.objects.select_related("rehabdoctorprof
essiontimeslot",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabprofession
",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__do
ctordjangoaccount",
"rehabdoctorprofessiontimeslot__rehabdoctorprofession__rehabdoctor__re
habcity", "rehabpatient", "rehabpatient__patientdjangoaccount",
"rehabpatienthome").filter(rehabpatient = rehab_patient)
    rehab_patient_applications =
rehab_patient_applications.filter(rehabstatus = "1")
    if rehabstart:
        rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotstart__gte = rehabstart)
    if rehabend:

```

```

        rehab_patient_applications =
rehab_patient_applications.filter(rehabdoctorprofessiontimeslot__times
lotend__lte = rehabend)
        if sortordering == 0:
            rehab_patient_applications =
rehab_patient_applications.order_by("-
rehabdoctorprofessiontimeslot__timeslotstart")
        elif sortordering == 1:
            rehab_patient_applications =
rehab_patient_applications.order_by("rehabdoctorprofessiontimeslot__ti
meslotstart")
        return render(request, "rehabapp/cancelpatapps.html",
{"applications": rehab_patient_applications })

```

```

def rehab_patient_page_view(request, rehabpatient = None):
    if not rehabpatient:
        rehabpatient_start = False
    else:
        rehabpatient_start = True
    if rehabpatient:
        try:
            rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").get(patien
tdjangoaccount__username = rehabpatient)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
        elif request.user.is_authenticated:
            try:
                rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").get(patien
tdjangoaccount = request.user)
            except Rehabpatient.DoesNotExist:
                rehab_patient = None
        else:
            rehab_patient = None
    if not rehab_patient:
        return render(request, "rehabapp/error.html",
{"caption": "Такий пацієнт невідомий або потрібно переглядати власну
сторінку пацієнта. Інакше це неможливо."})
    if not rehabpatient_start:
        rehab_patient_homes =
Rehabpatienthome.objects.filter(rehabpatient = rehab_patient)

```

```

        rehab_patient_diseases =
Rehabpatientdisease.objects.filter(rehabpatient = rehab_patient)
        rehab_patient_diseases =
rehab_patient_diseases.order_by("-fixdate")
    else:
        rehab_patient_homes = []
        rehab_patient_diseases = []
        rehab_patient_views =
Rehabview.objects.select_related("rehabdoctortimeslotapplication",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabdoctor",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabprofession",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabdoctor__doctordjangoaccount").filter(rehabdoctor
timeslotapplication__rehabdoctortimeslotapplication__rehabpatient =
rehab_patient, rehabdoctororpatient = False)
        rehab_patient_views = rehab_patient_views.order_by("-
rehabviewtime")
        rehab_patient_views = rehab_patient_views[:5]
        return render(request, "rehabapp/patient.html", {"patient":
rehab_patient, "views": rehab_patient_views, "homes":
rehab_patient_homes, "diseases": rehab_patient_diseases})

```

```

def rehab_patient_views_view(request, rehabpatient = None):
    if request.user.is_authenticated:
        try:
            own_rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").prefetch_r
elated("rehabdoctors").get(patientdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            own_rehab_patient = None
    else:
        own_rehab_patient = None
    if not rehabpatient:
        rehabpatient_start = False
    else:
        rehabpatient_start = True
    if rehabpatient:

```

```

        try:
            rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").get(patien
tdjangoaccount__username = rehabpatient)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    elif request.user.is_authenticated:
        try:
            rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").get(patien
tdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    else:
        rehab_patient = None
    if not rehab_patient:
        return render(request, "rehabapp/error.html",
{"caption": "Такий пацієнт невідомий або потрібно переглядати власну
сторінку пацієнта. Інакше це неможливо."})
    rehab_patient_views =
Rehabview.objects.select_related("rehabdoctortimeslotapplication",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabdoctor",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabprofession",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabdoctor__doctordjangoaccount").filter(rehabdoctor
timeslotapplication__rehabdoctortimeslotapplication__rehabpatient =
rehab_patient, rehabdoctororpatient = True)
    rehab_patient_views = rehab_patient_views.order_by("-
rehabviewtime")
    rehab_patient_views = Paginator(rehab_patient_views, 10)
    rehab_patient_views_page = request.GET.get('page')
    rehab_patient_views =
rehab_patient_views.get_page(rehab_patient_views_page)
    for rehab_patient_view in rehab_patient_views:
        if
rehab_patient_view.rehabdoctortimeslotapplication.rehabdoctorprofessio
ntimeslot.rehabdoctorprofession.rehabdoctor in
own_rehab_patient.rehabdoctors:

```

```

        rehab_patient_view.saved = True
    else:
        rehab_patient_view.saved = False
    return render(request, "rehabapp/patientviews.html",
{"patient": rehab_patient, "views": rehab_patient_views })

def rehab_doctor_views_view(request, rehabdoctor = None):
    if request.user.is_authenticated:
        try:
            rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").prefetch_r
elated("rehabdoctors").get(patientdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    if not rehabdoctor:
        rehabdoctor_start = False
    else:
        rehabdoctor_start = True
    if rehabdoctor:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = rehabdoctor)
        except Rehabdoctor.DoesNotExist:
            rehab_doctor = None
        elif request.user.is_authenticated:
            try:
                rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = request.user)
            except Rehabdoctor.DoesNotExist:
                rehab_doctor = None
        else:
            rehab_doctor = None
    if not rehab_doctor:
        return render(request, "rehabapp/error.html",
{"caption": "Такий лікар з реабілітації невідомий або потрібно
переглядати власну сторінку лікаря з реабілітації. Інакше це
неможливо."})
    rehab_doctor_views =
Rehabview.objects.select_related("rehabdoctortimeslotapplication",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession",

```

```

"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabdoctor",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabprofession",
"rehabdoctortimeslotapplication__rehabdoctorprofessiontimeslot__rehabd
octorprofession__rehabdoctor__doctordjangoaccount").filter(rehabdoctor
timeslotapplication__rehabdoctorprofessiontimeslot__rehabdoctorprofess
ion__rehabdoctor = rehab_doctor, rehabdoctororpatient = False)
    rehab_doctor_views = rehab_doctor_views.order_by("-
rehabviewtime")
    rehab_doctor_views = Paginator(rehab_patient_views, 10)
    rehab_doctor_views_page = request.GET.get('page')
    rehab_doctor_views =
rehab_doctor_views.get_page(rehab_doctor_views_page)
    rehab_doctor_save = False
    if rehab_doctor in rehab_patient.rehabdoctors:
        rehab_doctor_save = True
    return render(request, "rehabapp/doctorviews.html",
{"doctor": rehab_doctor, "views": rehab_doctor_views, "save":
rehab_doctor_save })

```

```

def rehab_patient_medrecords_view(request, rehabpatient = None):
    if request.user.is_authenticated:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = rehabdoctor)
            except Rehabdoctor.DoesNotExist:
                rehab_doctor = None
        else:
            rehab_doctor = None
        if not rehabpatient:
            rehabpatient_start = False
        else:
            rehabpatient_start = True
        if rehabpatient:
            try:
                rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").get(patien
tdjangoaccount__username = rehabpatient)
            except Rehabpatient.DoesNotExist:
                rehab_patient = None
            elif request.user.is_authenticated:

```

```

        try:
            rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").get(patien
tdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    else:
        rehab_patient = None
    if not rehab_patient:
        return render(request, "rehabapp/error.html",
{"caption": "Такий пацієнт невідомий або потрібно переглядати власну
сторінку пацієнта. Інакше це неможливо."})
        rehab_patient_medrecords = []
        if not rehabpatient_start or rehab_doctor:
            rehab_patient_medrecords =
Rehabpatientmedrecord.objects.filter(rehabpatient = rehab_patient)
            rehab_patient_medrecords =
rehab_patient_medrecords.order_by("-rehabdate")
            rehab_patient_medrecords =
Paginator(rehab_patient_medrecords, 10)
            rehab_patient_medrecords_page = request.GET.get('page')
            rehab_patient_medrecords =
rehab_patient_medrecords.get_page(rehab_patient_medrecords_page)
            return render(request, "rehabapp/patientmedrecords.html",
{"patient": rehab_patient, "medrecords": rehab_patient_medrecords})

```

```

    def rehab_doctor_diplomas_view(request, rehabdoctor = None,
rehabdoctprof = None):
        if rehabdoctprof:
            try:
                rehab_doctor_prof =
Rehabdoctorprofession.objects.select_related("rehabdoctor",
"rehabprofession", "rehabdoctor__doctordjangoaccount").get(id =
rehabdoctprof)
            except Rehabdoctor.DoesNotExist:
                rehab_doctor_prof = None
        else:
            rehab_doctor_prof = None
        if request.user.is_authenticated:
            try:

```

```

        rehab_patient =
Rehabpatient.objects.select_related("patientdjangoaccount").prefetch_r
elated("rehabdoctors").get(patientdjangoaccount = request.user)
        except Rehabpatient.DoesNotExist:
            rehab_patient = None
    if not rehabdoctor:
        rehabdoctor_start = False
    else:
        rehabdoctor_start = True
    if rehabdoctor:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = rehabdoctor)
        except Rehabdoctor.DoesNotExist:
            rehab_doctor = None
    elif request.user.is_authenticated:
        try:
            rehab_doctor =
Rehabdoctor.objects.get(doctordjangoaccount = request.user)
        except Rehabdoctor.DoesNotExist:
            rehab_doctor = None
    else:
        rehab_doctor = None
    if not rehab_doctor:
        return render(request, "rehabapp/error.html",
{"caption": "Такий лікар з реабілітації невідомий або потрібно
переглядати власну сторінку лікаря з реабілітації. Інакше це
неможливо."})
        rehab_doctor_diplomas = []
    if rehabdoctprof:
        rehab_doctor_diplomas =
Rehabdoctorprofessiondiploma.objects.select_related("rehabdoctorprofes
sion", "rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofession__rehabprofession",
"rehabdoctorprofession__rehabdoctor__doctordjangoaccount").filter(reha
bdoctorprofession = rehabdoctprof)
    elif rehab_doctor:
        rehab_doctor_diplomas =
Rehabdoctorprofessiondiploma.objects.select_related("rehabdoctorprofes
sion", "rehabdoctorprofession__rehabdoctor",
"rehabdoctorprofession__rehabprofession",
"rehabdoctorprofession__rehabdoctor__doctordjangoaccount").filter(reha
bdoctorprofession__rehabdoctor = rehab_doctor)

```

```

        rehab_doctor_diplomas = rehab_doctor_diplomas.order_by("-
diplomadate")
        rehab_doctor_diplomas = Paginator(rehab_doctor_diplomas, 10)
        rehab_doctor_diplomas_page = request.GET.get('page')
        rehab_doctor_diplomas =
rehab_doctor_diplomas.get_page(rehab_doctor_diplomas_page)
        rehab_doctor_save = False
        if rehabdoctprof and rehabdoctprof.rehabdoctor in
rehab_patient.rehabdoctors:
            rehab_doctor_save = True
        elif rehab_doctor and rehab_doctor in
rehab_patient.rehabdoctors:
            rehab_doctor_save = True
        if rehabdoctprof:
            return render(request, "rehabapp/diplomas.html",
{"doctorprof": rehabdoctprof, "diplomas": rehab_doctor_diplomas,
"save": rehab_doctor_save })
        elif rehab_doctor:
            return render(request, "rehabapp/diplomas.html",
{"doctor": rehab_doctor, "diplomas": rehab_doctor_diplomas, "save":
rehab_doctor_save })

```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Титульний слайд

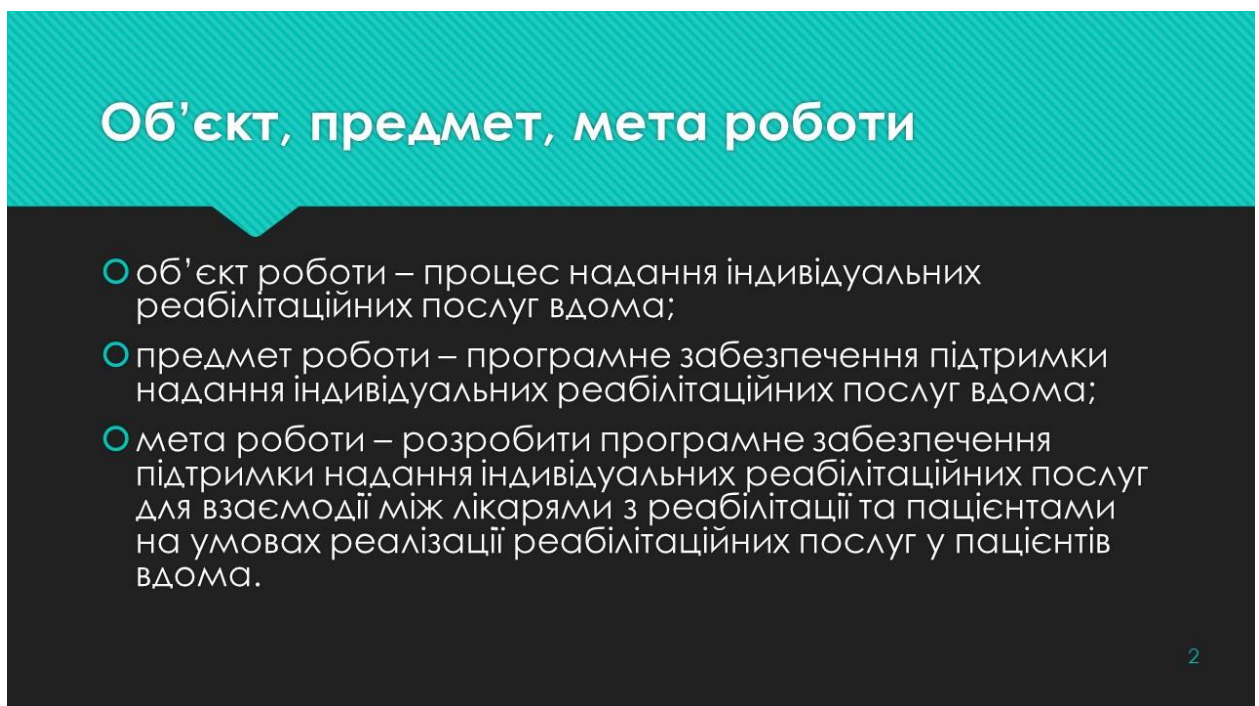


Рисунок В.2 – Об'єкт, предмет, мета роботи

Завдання роботи

- аналіз стану проблеми розробки програмного забезпечення підтримки надання індивідуальних реабілітаційних послуг вдома;
- проектування програми підтримки надання індивідуальних реабілітаційних послуг вдома;
- реалізація, тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома.

3

Рисунок В.3 – Завдання роботи

Порівняння програм підтримки надання індивідуальних реабілітаційних послуг вдома

Критерій порівняння	OLX	Фізіо-Хелп	Моя програма
Бронювання відвідування для реабілітації вдома	Немає	Наявне тільки для відвідування лікарні	Так
Джерело надання послуг реабілітації	Окремі лікарі та інші працівники	Медичний центр	Окремі лікарі та інші працівники, медичні центри
Збереження параметрів пошуку	Так	Немає	Так
Визначення вартості послуг з реабілітації вдома	Може вказуватись в тексті оголошення	Так	Так

4

Рисунок В.4 – Порівняння програм підтримки надання індивідуальних реабілітаційних послуг вдома

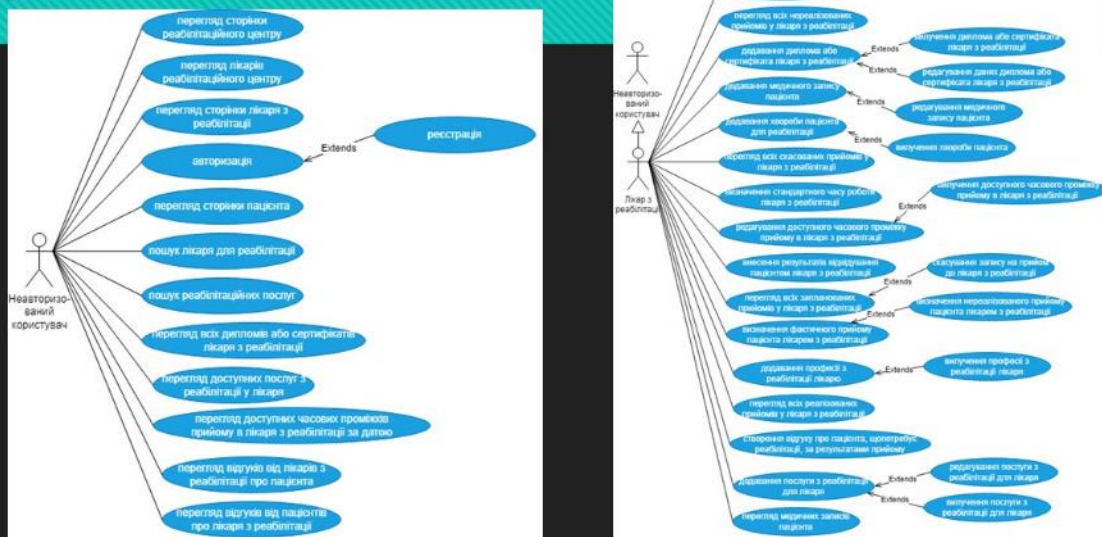
Порівняння фреймворків для розробки програми підтримки надання індивідуальних реабілітаційних послуг вдома

Критерій порівняння	Django	Flask
Тип рішення	Менш гнучке, але більш функціональне і підготовлене для застосування в обмежених умовах	Більш гнучке, але менш функціональне
Структура проєкту	Визначена фреймворком	Визначається користувачем
Час розробки	Менший у випадку застосування для великих, середніх проєктів	Більший у випадку застосування для великих, середніх проєктів
Підтримка баз даних	Готова	Можлива

5

Рисунок В.5 – Порівняння фреймворків для розробки програми підтримки надання індивідуальних реабілітаційних послуг вдома

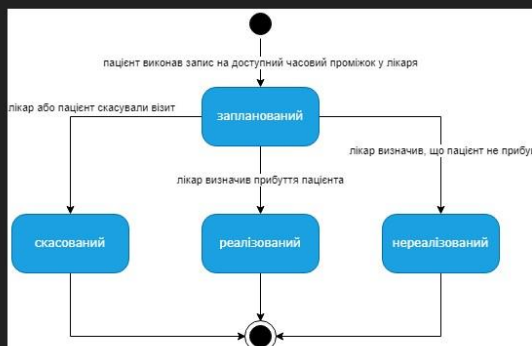
Діаграми прецедентів програми підтримки надання індивідуальних реабілітаційних послуг вдома



6

Рисунок В.6 – Діаграми прецедентів програми підтримки надання індивідуальних реабілітаційних послуг вдома

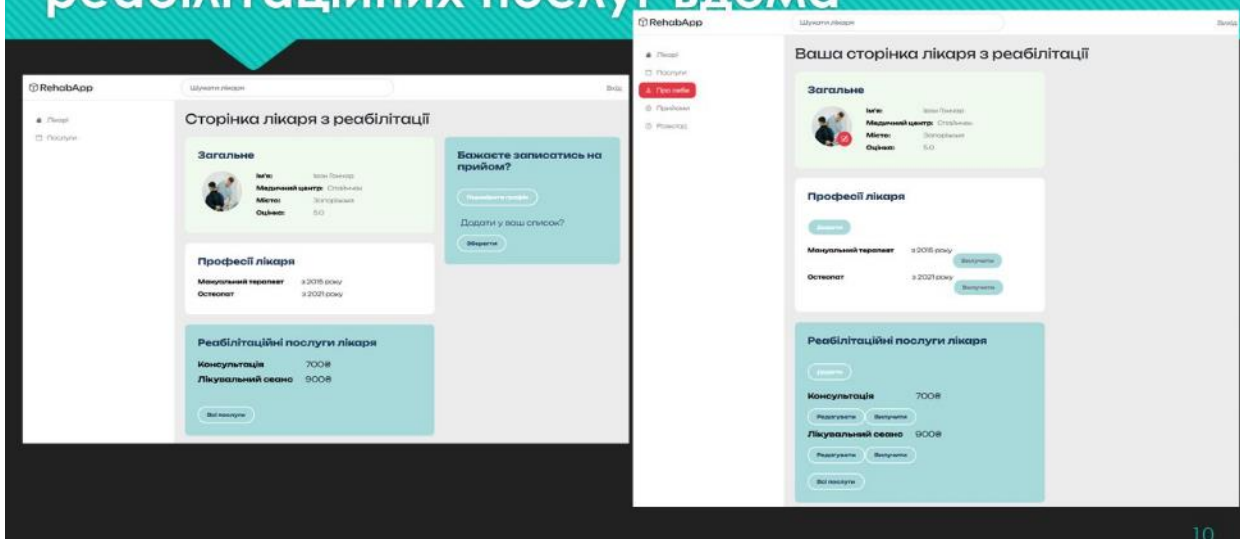
Діаграма станів запису на прийом до лікаря з реабілітації



9

Рисунок В.9 – Діаграма станів запису на прийом до лікаря з реабілітації

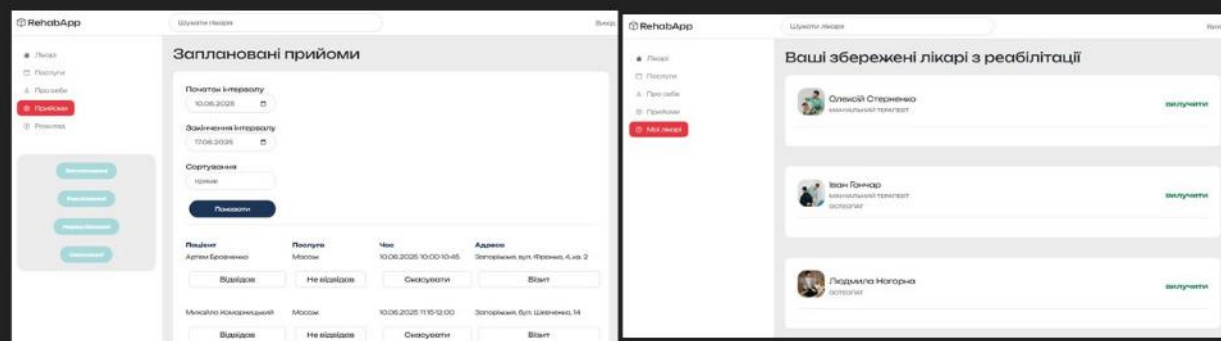
Результати тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома



10

Рисунок В.10 – Результати тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома

Результати тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома



11

Рисунок В.11 – Результати тестування програми підтримки надання індивідуальних реабілітаційних послуг вдома

Висновки

- Розроблено програмне забезпечення підтримки надання індивідуальних реабілітаційних послуг для взаємодії між лікарями з реабілітації та пацієнтами на умовах реалізації реабілітаційних послуг у пацієнтів вдома.
- Розроблена програма призначена для підтримки надання індивідуальних реабілітаційних послуг з врахуванням особливостей реалізації послуг вдома у пацієнта. Для цього програма забезпечує пацієнтів засобами пошуку лікарів з реабілітації, послуг з реабілітації, запису на прийом до лікарів з врахуванням графіку роботи з виїздами додому до пацієнтів, лікарів – засобами контролю свого графіку роботи, керування записами до них на прийом та відслідковування плану роботи, визначення рекомендованих процедур для подальшого лікування.

12

Рисунок В.12 – Висновки