Міністерство освіти і науки України

Одеський національний політехнічний університет

Навчально-науковий інститут комп’ютерних систем

Кафедра системного програмного забезпечення

Тищенко Олександр Іванович,

студент групи АС-151

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

Сервіс управління та підтримки функціональності стоматологічної клініки, що враховує доступність інтерфейсу користувача

Спеціальність:

121 Інженерія програмного забезпечення

Спеціалізація:

Інженерія програмного забезпечення

Керівник:

Писаренко Катерина Олександрівна,

канд. техн. наук, доцент

Одеса – 2020

**ЗМІСТ**

[ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ 4](#_Toc57891276)

[ЗАВДАННЯ НА РОЗРОБКУ РОЗДІЛУ «ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 6](#_Toc57891278)

[АНОТАЦІЯ 7](#_Toc57891279)

[ВСТУП 8](#_Toc57891280)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ 10](#_Toc57891281)

[1.1 Опис предметної області 10](#_Toc57891282)

[1.2 Дослідження актуальної статистики 10](#_Toc57891283)

[1.3 Дослідження ринку 14](#_Toc57891284)

[1.4 Використання моделі NABC для опису ключової пропозиції 17](#_Toc57891285)

[1.5 Бізнес-модель Canvas 19](#_Toc57891286)

[1.6 Бізнес-модель Lean canvas 20](#_Toc57891287)

[1.7 Монетизація 21](#_Toc57891288)

[1.8 Алгоритм роботи проекту 22](#_Toc57891289)

[1.9 Масштабованість 24](#_Toc57891290)

[1.10 Swot-аналіз 25](#_Toc57891291)

[1.11 Value proposition canvas 26](#_Toc57891292)

[1.12 Аналіз конкурентів і побудова strategy canvas 28](#_Toc57891293)

[1.13 Створення стратегії блакитного океану на основі four action framework 31](#_Toc57891294)

[1.14 Формулювання стратегії із використанням методу оцінки впливу 32](#_Toc57891295)

[1.15 Аналіз останніх публікацій на тему доступності web-інтерфейсів 38](#_Toc57891296)

[2 РОЗРОБКА РОЗШИРЕНОЇ МОДЕЛІ ДОСТУПНОСТІ 39](#_Toc57891297)

[2.1 Основні стандарти і рекомендації для забезпечення доступності web-інтерфейсу 39](#_Toc57891298)

[2.2 Порушення, пов’язані зі здоров’ям користувачів 40](#_Toc57891299)

[2.3 Апаратні та програмні засоби вирішення проблеми 41](#_Toc57891300)

[2.4 Класична модель доступності 43](#_Toc57891301)

[2.5 Розширена модель доступності 45](#_Toc57891302)

[3 ФОРМУЛЮВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ 52](#_Toc57891303)

[3.1 Визначення варіантів використання системи 52](#_Toc57891304)

[4 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ 57](#_Toc57891305)

[4.1 Побудова діаграм взаємодії 57](#_Toc57891306)

[4.2 Побудова діаграми класів 71](#_Toc57891307)

[4.3 Проектування інтерфейсу користувача 73](#_Toc57891308)

[4.4 Проектування структури колекцій 77](#_Toc57891309)

[5 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ 82](#_Toc57891310)

[5.1 Набір інструментальних засобів розробки 82](#_Toc57891311)

[5.2 Тестування за допомогою Postman 86](#_Toc57891312)

[5.3 Тестування функціональності 95](#_Toc57891313)

[5.4 Приклад використання системи 100](#_Toc57891314)

[6 АПРОБАЦІЯ СИСТЕМИ 105](#_Toc57891315)

[6.1 Апробація панелі зміни типу відображення 105](#_Toc57891316)

[6.2 Апробація сервісу 106](#_Toc57891317)

[7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ 109](#_Toc57891318)

[ВИСНОВОК 111](#_Toc57891319)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 113](#_Toc57891320)

[Додаток А. ЛІСТИНГ ПРОГРАМИ 115](#_Toc57891321)

[Додаток Б. ПИТАННЯ З ОХОРОНИ ПРАЦІ ТА БЕЗПЕКИ У НАДЗВИЧАЙНИХ СИТУАЦІЯХ 128](#_Toc57891322)

[Б.1 Охорона праці. Організація та управління 128](#_Toc57891323)

[Б.2 Охорона праці. Визначення основних параметрів умов праці згідно індивідуального завдання 130](#_Toc57891324)

[Б.3 Охорона праці. Індивідуальне завдання. Розрахунок системи освітлення на робочому місці програміста, яке розташовано в офісній будівлі на 8 поверсі 131](#_Toc57891325)

[Б.4 Безпека у надзвичайних ситуаціях техногенного характеру. Індивідуальне завдання. Розрахунок пожежної безпеки технічних систем 136](#_Toc57891326)

[Б.5 Заходи з охорони праці та безпеки у надзвичайних ситуаціях 143](#_Toc57891327)

[Б.6 Безпека у суспільстві в умовах загрози COVID-19 143](#_Toc57891328)

Міністерство освіти і науки України

Одеський національний політехнічний університет

Навчально-науковий інститут комп’ютерних систем

Кафедра системного програмного забезпечення

Рівень вищої освіти: другий (магістерський)

Спеціальність: 121 – Інженерія програмного забезпечення

Спеціалізація: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Крісілов В.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р.

# **ЗАВДАННЯ**

# **НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Тищенка Олександра Івановича, група АС-151

|  |  |  |
| --- | --- | --- |
| 1. Тема роботи: | Сервіс управління та підтримки функціональності стоматологічної клініки, що враховує доступність інтерфейсу користувача | |
| Керівник роботи: | Писаренко Катерина Олександрівна, канд. техн. наук, доцент | |
| затверджені наказом ректора від «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р. № \_\_\_\_\_\_ | | |
| 2. Зміст роботи:  1. Аналіз існуючих рішень.  2. Побудова розширеної моделі доступності web-інтерфейсів.  3. Визначення вимог до програмного продукту.  4. Проектування програмного продукту.  5. Реалізація та тестування програмного продукту.  6. Апробація програмного продукту  7. Дослідження охорони праці та безпеки у надзвичайних ситуаціях.  3. Перелік ілюстративного матеріалу: презентація. | |

4. Консультанти розділів роботи

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
| завдання видав | завдання прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

5. Дата видачі завдання «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів кваліфікаційної роботи | Строк виконання | Примітка |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Здобувач вищої освіти** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | О. І. Тищенко |
| **Керівник роботи** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | К. О. Писаренко |

# **ЗАВДАННЯ**

# на розробку розділу «Охорона праці та безпека в надзвичайних ситуаціях»

Тищенка Олександра Івановича, група АС-151

Навчально-науковий інститут комп’ютерних систем

Кафедра системного програмного забезпечення

Тема роботи: Сервіс управління та підтримки функціональності стоматологічної

клініки, що враховує доступність інтерфейсу користувача

Зміст розділу:

1. Аналіз умов праці і вибір заходів і засобів захисту від небезпечних і шкідливих виробничих факторів.

2. Аналіз техногенних небезпек і вибір заходів і засобів забезпечення безпеки у надзвичайних ситуаціях.

3. Розрахунок системи освітлення на робочому місці програміста, яке розташовано в офісній будівлі на 8 поверсі. Розрахунок пожежної безпеки технічних систем.

Керівник роботи Консультант з охорони праці та БНС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. О. Писаренко \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Ю. Москалюк

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р.

# **АНОТАЦІЯ**

Метою роботи є підвищення рівня доступності інтерфейсу користувача web-застосунку за рахунок реалізації розширеної моделі доступності. Побудова моделі доступності виконується шляхом попереднього аналізу існуючих підходів до забезпечення доступності та визначення необхідності внесення нових компонент. На підставі розширеної моделі доступності була розроблена система підтримки функціональності стоматологічної клініки, яка враховує основні типи порушень здоров'я користувачів, а також дотримується рекомендацій стандарту WCAG 2.0 для web-інтерфейсу. При розробці було використано мову програмування JavaScript, СУБД MongoDB і фреймворк Express.js.

Ключові слова: стандарт, web-застосунок, доступність, web-інтерфейс, порушення здоров'я.

**ABSTRACT**

The aim of the work is to increase the level of accessibility of the user interface of a web application by implementing an extended accessibility model. Accessibility model building is performed by preliminary analysis of existing approaches to ensuring accessibility and determining the need to introduce new components. Based on the extended accessibility model, a support system for the functionality of the dental clinic was developed, which takes into account the main types of user health disorders, and also complies with the recommendations of the WCAG 2.0 standard for web interface. During the development, the programming language JavaScript, the MongoDB DBMS and the Express.js framework were used.

Keywords: standard, web application, accessibility, web interface, health problems.

# **ВСТУП**

Розробка інтерфейсів більшості сучасних web-застосунків, в тому числі CRM-систем в області стоматології, до недавнього часу була націлена лише на ту аудиторію, яка не мала порушень, пов'язаних зі здоров'ям. Однак, на сьогоднішній день, люди з обмеженими можливостями намагаються активно використовувати мережу Internet. Саме тому на перший план виходить питання про розширення користувальницької аудиторії web-застосунків. У зв'язку з цим актуальною стає задача розробки web-інтерфейсу, який буде доступний не тільки для звичайних користувачів, але і для людей з обмеженими можливостями.

Підвищення рівня доступності інтерфейсів web-сайтів і web-застосунків можливо шляхом реалізації моделі, яка враховує всі основні типи порушень, пов'язаних зі здоров'ям користувачів.

Окремо слід відзначити актуальність сервісів, які допомагають управляти та автоматизовувати процеси, які супроводжують основну діяльність стоматологічної клініки. Причини виникнення попиту на вказані програмні системи полягають в тому, що проведення ретельного аналізу діяльності клініки, який дозволяє зрозуміти тенденції розвитку, є досить трудомістким і вимагає, як залучення великої кількості співробітників, так і значних витрат часу. При цьому, на момент, коли проведення аналізу буде завершено, показники можуть бути зовсім іншими і тому прийняті на основі результатів аналізу рішення, можуть бути неправильними. Залучення сервісів управління та підтримки функціональності дозволяє вирішити дану проблему.

Метою написання даної роботи є підвищення рівня доступності інтерфейсу користувача web-застосунку за рахунок реалізації розширеної моделі доступності для людей з обмеженими можливостями.

У відповідності до вказаної мети було сформульовано наступні задачі:

– на основі аналізу існуючих web-сайтів і web-застосунків побудувати модель доступності web-інтерфейсів, котру надалі ми будемо називати класичною;

– описати компоненти класичної моделі доступності;

– побудувати розширену модель доступності web-інтерфейсів;

– описати компоненти розширеної моделі доступності;

– реалізувати розширену модель доступності;

– виконати апробацію web-застосунку, розробленого на основі розширеної моделі доступності.

Об’єкт дослідження: процес підвищення рівня доступності інтерфейсу користувача web-застосунку.

Предмет дослідження: компонети для побудови розширеної моделі доступності.

В якості методів дослідження використовувалася теорія множин для опису класичної та розширеної моделей доступності. Також використовувалися статистичні методи для збору і обробки інформації. При застосуванні даних методів дотримувалася наступна послідовність виконання дій: статистичне спостереження, зведення в єдину таблицю і угруповання даних, розрахунок узагальнюючих показників (абсолютні і відносні величини).

Наукова новизна: отримала розвиток модель доступності інтерфейсу web-застосунку.

Практична значимість: на основі запропонованої розширеної моделі доступності для інтерфейсу користувача web-застосунку, розроблена система управління і підтримки функціональності стоматологічної клініки, в рамках якої передбачено використання допоміжною панелі для людей з обмеженими можливостями.

Публікація: науковий журнал «Вісник сучасних інформаційних технологій» том 2 № 3 2019 с. 206-2014.

# **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ**

## **1.1 Опис предметної області**

В якості предметної області для дипломного проектування була обрана область стоматології. Ідея проекту полягає в розробці CRM (customer relationship management – управління відносинами з клієнтами) системи для стоматологічних клінік. Це означає, що система включає модулі підтримки функціональності стоматологічної клініки при встановленні взаємин з клієнтами. Також, система дозволяє спростити роботу стоматологів і власників стоматологічних клінік шляхом автоматизації певних процесів (складання звітів, визначення робочого навантаження для стоматолога на день). Виходячи з озвученої ідеї, було визначено назву даного продукту - DentalCare. Назва CRM-системи складається з двох слів: dental - стоматологічний, care - турбота, догляд. Аналіз доступного доменного імені обмежив вибір наступними варіантами: dentalcare.ua, dentalcare.name, dentalcare.club. Був обраний перший варіант.

Враховуючи те, що інтерфейси користувача програмного продукту визначають можливість пацієнтів знаходити необхідну інформацію і здійснювати запис на прийом за прийнятний для них час, було прийнято рішення забезпечити необхідний для цього рівень доступності інтерфейсу користувача. Це дозволить людям з порушеннями зору, слуху, проблемами з опорно-руховим апаратом, а також когнітивними порушеннями самостійно здійснювати взаємодію з програмною системою.

## **1.2 Дослідження актуальної статистики**

При виконанні аналізу враховувалася статистика за 2018-2020 роки. Зазначена статистика використовувалася для перевірки бізнес-гіпотези (доцільності розробки CRM-систем виходячи з потреб стоматологічних клінік) і аналізу розвитку CRM-маркетингу.

Опитування основних стоматологічних клінік Одеси в 2018 році на предмет потреби даних клінік в CRM-системах, для спрощення відносин з клієнтами, показав наступні результати:

* 37% клінік є користувачами тих чи інших CRM-систем;
* 29% клінік планують впроваджувати CRM-системи у найближчі 2 роки;
* 17% стоматологічних клінік розглядають можливість переходу на CRM-системи у найближчі 5 років;
* 10% клінік не планують впроваджувати CRM-системи у зв’язку із фінановим положенням;
* 7% не знають про CRM-системи.

Дана статистика дозволяє підтвердити бізнес-гіпотезу про існування потреби стоматологічних клінік у впровадженні CRM-систем. Що стосується фінансової статистики, то в даному випадку ми маємо в своєму розпорядженні показники стоматологічних клінік, які вже впровадили CRM-систему.

Динаміка доходів однієї з таких клінік після впровадження CRM-системи за період із січня по квітень 2019 року зображена на рис. 1.1.

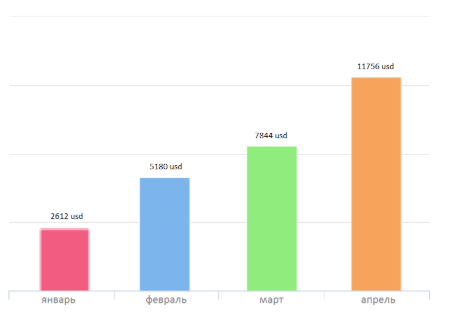


Рисунок 1.1 – Динаміка доходів в 2019 році

Як можна бачити на рис. 1.1, використання CRM-системи дозволило збільшити сумарний дохід за місяць з 2612 доларів США до 11756 доларів США. У відсотковому еквіваленті це становить 450% (збільшення в 4,5 рази).

Що стосується кількості записів, яка включає в себе як додавання нових клієнтів, так і зростання числа повторних візитів, то дану статистику можна спостерігати на рис. 1.2.

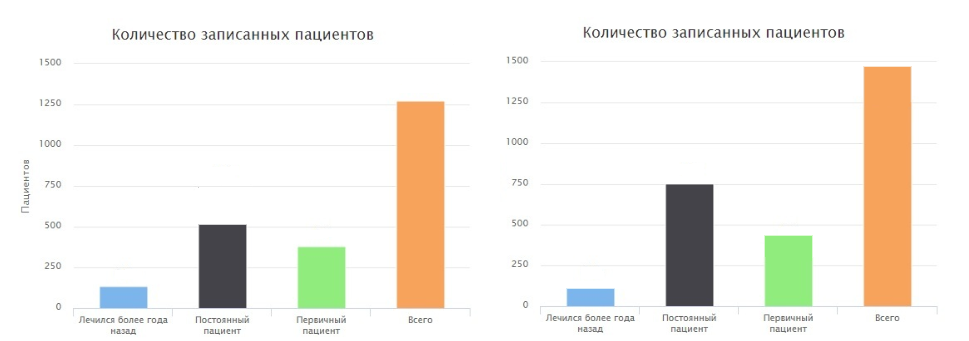


Рисунок 1.2 – Зміна кількості пацієнтів

Кількість постійних пацієнтів збільшилася з 500 до 750, кількість пацієнтів, що відвідали вперше – з 323 до 417. Це дозволило збільшити загальну кількість пацієнтів до 1476 (на 19%).

Наступним питанням є розвиток CRM-маркетингу в медицині і, зокрема, в галузі стоматології. Ситуація на момент 2018 року зображена на рис. 1.3.

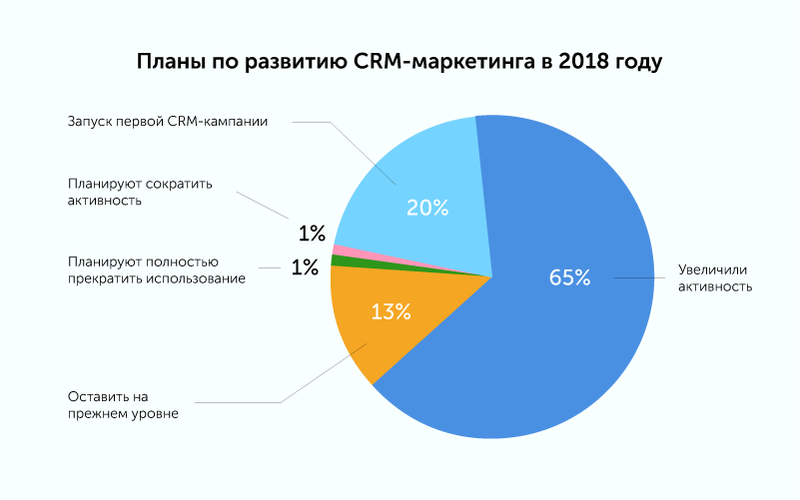


Рисунок 1.3 – Плани по розвитку CRM-маркетингу в 2018 році

Рис. 1.3 демонструє збільшення активності в області розвитку CRM-маркетингу (65%). Також, інтерес до CRM-систем призводить до того, що 20% компаній планують впроваджувати CRM-системи. І в першому і в другому випадку можна стверджувати, що користувачі CRM-систем залишаються задоволеними отриманим результатом, так як лише 2% планують скорочувати або ж повністю припиняти використання.

Аналіз витрат на CRM-маркетинг, опублікований на початку 2020 року, відображений на рис. 1.4.



Рисунок 1.4 – Затрати компаній на CRM-маркетинг

У 62% випадків CRM-маркетинг становить до 15% усього бюджету. Ще 19% витрачають на CRM-маркетинг 15-30% всього бюджету.

## **1.3 Дослідження ринку**

Були визначені галузева і географічна складові ринку. Що стосується географічної складової, то в даному випадку все достатньо зрозуміло: відправною точкою продажу є місто Одеса. Після збільшення кількості продажів можливе розширення до меж Одеської області. У разі успішного масштабування в рамках області, можна виходити на ринки півдня України і всієї України. Мовні і функціональні особливості розроблюваної CRM-системи дозволяють виходити на ринки сусідніх країн (в рамках Співдружності Незалежних Держав).

Галузева складова характеризується медичної спрямованістю, а конкретно ринком програмних продуктів для автоматизації процесів в рамках стоматологічних клінік. Отже, нашими клієнтами будуть власники приватних стоматологічних клінік.

Після визначення вартості передплати на наш продукт на місяць (30 $), була виконана оцінка обсягів ринку:

1. TAM – 140 (кількість стоматологічних клінік в Одесі) \* 30 \*12 = 50 400$ (за рік).

2. SAM – 92 (кількість стоматологічних клінік, які зацікавлені у використанні CRM-систем) \* 30 \* 12 = 33 120$ (за рік).

3. SOM – 20 (кількість CRM-систем, супроводження яких можна забезпечити на даний момент) \* 30 \* 12 = 7 200$ (за рік).

Третім етапом дослідження ринку було опитування наших потенційних клієнтів (власників стоматологічних клінік) шляхом інтерв'ю. Опитування проводилося в стоматологічних клініках різного рівня прибутковості, як в містах з великою кількістю жителів (Одеса), так і в містах з населенням 50 тисяч чоловік. Опитувальник містив такі питання:

1. Що становить робочий день власника клініки, на що витрачається найбільше часу?

2. Які проблеми існують на даний момент при управлінні стоматологічною клінікою?

3. Чи використовуєте ви CRM-системи? Якщо ні, то чому не використовуєте, а якщо так, то які?

4. Що приваблює в обраному рішенні (переваги) і яких недоліків ви би хотіли позбавитись?

5. Який функціонал ви використовуєте частіше за все?

6. Як впливає використання CRM-системи на діяльність клініки (організаційні, комунікаційні та фінансові питання)?

7. Яка вартість передплати на місяць є допустимою, а яка – бажаною (чи допустима сформована ціна у розмірі 30$ за місяць)?

Вдалось зібрати 7 різних думок. Наведемо аналіз отриманих відгуків (нумерація відгуків відповідає запитанням вище):

1. 71% витрачає основну кількість часу на комунікаційні процеси для з'ясування поточного стану речей, 29% – на аналіз документації.

2. Серед основних проблем були названі: витрати часу в сумі з використанням людських ресурсів (бухгалтери, секретарі) для отримання інформації про стан справ в клініці (вказали 86%), відсутність електронного документообігу (86%), відсутність інформації про ефективність роботи співробітників (42%).

3. 57% використовують CRM-системи, 28% не використовують їх через фінансові причини і готові почати використання після зниження ціни на ринку, інші 15% не використовують, так як вважають, що не мають потреби в CRM-системах. Серед тих систем, які були названі: denta-pro, dental4windows, dentist24, bitrix24, yclients.

4. В обраному рішенні (CRM-системі) бачать наступні переваги: можливість фіксувати прибутки і витрати, визначати ситуацію на складі. Серед недоліків були названі: відсутність електронної бази документації, недостатня деталізація звітів, відсутність можливості здійснювати оголошення і вказівки в рамках єдиного комунікаційного простору, висока ціна при під’єднанні великого штату співробітників до системи (більше 10 осіб). Серед інших рішень, якими користуються для вирішення проблем, були названі месенджери. Вони використовувалися для вирішення комунікаційних та організаційних питань, оперативного отримання інформації.

5. Всі, хто відповів ствердно на запитання «Чи використовуєте ви CRM-систему» вказали, що найчастіше виконують запит для отримання звітів стосовно бухгалтерії і складу.

6. В області фінансів спостерігалася динаміка в бік поліпшення фінансових показників, велика частина організаційних і комунікаційних питань залишилися невирішеними. При цьому, додалися деякі проблеми, пов'язані з використанням системи і навчанням роботи з нею нових співробітників.

7. Багато хто називав бажаною вартістю числа в діапазоні 20-40$, деякі доходили до суми в 50$. Допустимої ціною є 30$ -50$ за передплату на місяць.

Незважаючи на те, що стоматологи і пацієнти стоматологічних клінік є не потенційними клієнтами, а потенційними користувачами нашого ПП (оскільки не сплачують підписку на наш продукт, а лише використовують його), було проведено їх опитування для визначення проблем, з якими вони стикаються. Виконано це було з тієї причини, що від роботи стоматологів залежить діяльність всієї клініки, а від вибору клієнтів певної клініки залежить надходження або ненадходження грошей до її власника. В результаті спілкування зі стоматологами були виявлені наступні моменти:

* основні проблеми - необхідність тримати в голові інформацію про клієнта (історію лікування, особливості). необхідність відволікатися на другорядні моменти при обслуговуванні клієнтів (уточнювати новий час прийому);
* є побажання складати електронні плани лікування пацієнта, оскільки найчастіше є необхідність в лікуванні відразу декількох зубів (або ж виконанні ряду інших процедур);
* відсутність комунікацій з клієнтами призводить до того, що в ході лікування стоматолог не може контролювати стан здоров'я пацієнта і своєчасно виписувати рекомендації і призначати прийом.

Пацієнти повідомили, що основними проблемами, з якими вони стикаються при лікуванні в стоматологічних клініках є: запис, найчастіше, тільки по телефону; не попереджають своєчасно про зміну часу прийому; відсутність доступної інформації про хід лікування; відсутність можливості прямого спілкування зі своїм лікарем.

## **1.4 Використання моделі NABC для опису ключової пропозиції**

На основі аналізу отриманих відгуків (зазначених нашими потенційними клієнтами проблем, рішень, які вони використовують, переваг і недоліків, які вони відзначають в обраному рішенні), а також інформації, отриманої від потенційних користувачів – стоматологів і пацієнтів, можна описати ключову пропозицію нашого програмного продукту . Скористаємося для цього моделлю NABC.

NEEDS

Необхідно зазначити потреби користувачів у відповідності до типу, до якого вони належать.

Власники клінік мають наступні проблеми: низька інформованість про стан справ в клініці, необхідність роботи з великою кількістю паперів, відсутність інформації про зайнятість (результативності) працівників (стоматологів), відсутність шляхів донесення до співробітників оголошень і розпоряджень.

Стоматологи скаржаться на необхідність відволікатися на другорядні моменти, запам'ятовувати або постійно уточнювати особливості та історію лікування пацієнтів, виконані і заплановані процедури, враховувати витрати матеріалів.

Пацієнти мають проблеми із записом на прийом, комунікацією з лікуючим стоматологом, відсутністю інформації про хід лікування.

APPROACH

Необхідно запропонувати власний підхід до вирішення проблеми.

З огляду на те, що існують 3 групи проблем, необхідно запропонувати 3 рішення, які будуть інтегровані в один програмний продукт.

Для власників клінік пропонується розширення функціоналу «Звітність» для відстеження ефективності роботи співробітників.

Для стоматологів передбачено створення персонального кабінету, в якому буде визначатися список завдань на день / тиждень, окремо буде список пацієнтів із зазначенням все необхідної інформації про них (в тому числі, планом лікування).

Клієнтам пропонується використовувати онлайн-запис, переходити в особистий кабінет для перегляду ходу лікування, рекомендацій лікаря.

BENEFITS

Необхідно сформулювати переваги рішення, запропоноваго в рамках даного проекту, цінність для користувачів та для бізнесу.

Перевагами нашого ПП є: спрощення вирішення комунікаційних та організаційних питань в стоматологічній клініці, збільшення інформованості пацієнтів про хід лікування, часткова автоматизація задач стоматолога (при роботі з завданнями і прийомами).

Для користувачів цінність полягає в зменшенні кількості часу, яку необхідно витрачати для вирішення тих самих задач, що і раніше.

Для власників бізнесу це створює цінність у вигляді спрощення управління бізнесом через наявність необхідної інформації для прийняття рішень, а також можливостей віддаленого вирішення організаційних та комунікаційних питань. Все це, разом із поліпшенням роботи співробітників і задоволеністю клієнтів при використанні нашого ПП, призводить до поліпшення фінансових показників.

COMPETITION

Необхідно визначити альтернативні підходи, а також переваги даного рішення (це можуть бути вартість, швидкість, краща якість).

Альтернативними шляхами для часткового вирішення комунікаційних питань можуть бути месенджери. Однак, вони вирішують лише один аспект проблеми і при цьому не забезпечують відділення від загальної маси чатів робочих моментів для зручності роботи з ними.

Прямими конкурентами є CRM-системи. Серед них можна виділити denta-pro, dentist24, yclients. Вказані системи залишають невирішеними проблеми, пов'язані з відсутністю надання систематизованої інформації специфікованої під кожен тип користувача системи. При цьому, слід зазначити, що вартість передплати на місяць на програмний продукт, який розробляється, є нижчою за середню на ринку стоматологічних CRM-систем.

## **1.5 Бізнес-модель Canvas**

Сегменти користувачів: власники стоматологічних клінік, стоматологи, клієнти стоматологічних клінік.

Ціннісна пропозиція: простота запису на прийом до стоматолога (для пацієнтів), зручність роботи із запланованими візітами (для стоматологів), низька ціна і вісока ступінь деталізації звітів (для власників стоматологічних клінік).

Канали: веб-застосунок, реклама в соціальних мережах (інформування потенційних користувачів про наш ПП).

Відносини з клієнтами: персональна підтримка шляхом відповіді менеджера на запитання клієнта.

Ключові види діяльності: управління платформою, просування платформи, надання відповідей на запитання клієнтів.

Ключові ресурси: ресурси веб-сервера, людські ресурси, робоче приміщення та устаткування, яке розташовано в ньому, клієнтські бази даних.

Ключові партнери: основними партнерами є рекламодавці та хостинг-компанії. Можливим є стратегічне співробітництво із фірмами, які не є нашими конкурентами (власниками CRM-систем в суміжних областях) з метою розробки спільних проектів та виходу на нові ринки.

Потоки доходів: підписка на використання нашого програмного продукту на певний час (місяць або рік). Вартість підписки не є фіксованою, а визначається динамічно. При використанні продукту більше, ніж один рік, надається знижка у розмірі 5%, при використанні продукту більше трьох років – знижка 15%.

Структура витрат: аренда приміщення, оплата за використання хостингу та доменного імені, витрати на електроенергію, яка споживається обчислювальними ресурсами при роботі персоналу, витрати на рекламу.

## **1.6 Бізнес-модель Lean canvas**

Користувачі: власники стоматологічних клінік, стоматологи, клієнти стоматологічних клінік.

Унікальна торгівельна перевага: можливість використання асистивних інформаційних технологій при записі на прийом до стоматолога.

Прихована перевага: наявність досвіду спільних проектів зі стоматологічними клініками.

Канали: веб-застосунок, реклама в соціальних мережах (інформування потенційних користувачів про наш ПП).

Проблема: відсутність інформації про стан справ у клініці у її власника, витрати часу стоматологами при виконанні процесів, які супроводжують основні процеси, труднощі пацієнтів при записі на прийом до стоматолога.

Рішення: технічна реалізація наступних модулів: модуля для онлайн-запису, модуля особистого кабінету стоматолога, модуля надання звітів за результатами діяльності клініки.

Ключові метрики: кількість проданих підписок, місячний дохід, кількість користувачів.

Потоки виручки: підписка на використання нашого програмного продукту на певний час (місяць або рік). Вартість підписки не є фіксованою, а визначається динамічно. При використанні продукту більше, ніж один рік, надається знижка у розмірі 5%, при використанні продукту більше трьох років – знижка 15%.

Структура витрат: аренда приміщення, оплата за використання хостингу та доменного імені, витрати на електроенергію, яка споживається обчислювальними ресурсами при роботі персоналу, витрати на рекламу.

## **1.7 Монетизація**

Шляхом виконання аналізу конкурентних пропозицій та відгуків потенційних клієнтів, була сформована ціна передплати на місяць в розмірі 30 $. Споживачами даного програмного продукту є власники середніх та великих стоматологічних клінік. Ціна CRM-системи є адекватною, оскільки витрати на щомісячну підписку складають менше 3-5% від доходу стоматологічної клініки.

Враховуючи те, що можливості, які стосуються супроводу однією людиною проданих CRM-систем, обмежені 20 продуктами, а статті витрат, в сумі, складають 40$ (30$ на оплату хостингу, 10$ – витрати на електроенергію), то місячний дохід становитиме 560$ (1.1).

(1.1)

Шляхом збільшення штату співробітників до 5 осіб, можливо збільшити кількість підписок, які супроводжуються, до 100. При цьому виникає необхідність оренди приміщення (600$ на місяць), а витрати на електроенергію сягнуть 100$ на місяць. Місячний дохід становитиме 2270$ (1.2).

(1.2)

Варто зазначити, що при цьому не враховуються витрати на рекламу, оскільки вони визначаються динамічно, виходячи зі стартових можливостей та місячного доходу.

## **1.8 Алгоритм роботи проекту**

Алгоритм роботи проекту можна побачити на рис. 1.5. На даному рисунку зображено 3 типи користувачів і саму CRM-систему. Комунікація відбувається шляхом виконання запитів і отримання сформованих відповідей. Так як існує 3 типи користувачів, то і кількість взаємонезалежних процесів (потоків) дорівнює 3. У разі, якщо один з процесів виконує редагування даних, а другий намагається в той же час їх зчитувати, то зчитування буде заблоковано до того часу, поки не буде завершено редагування.

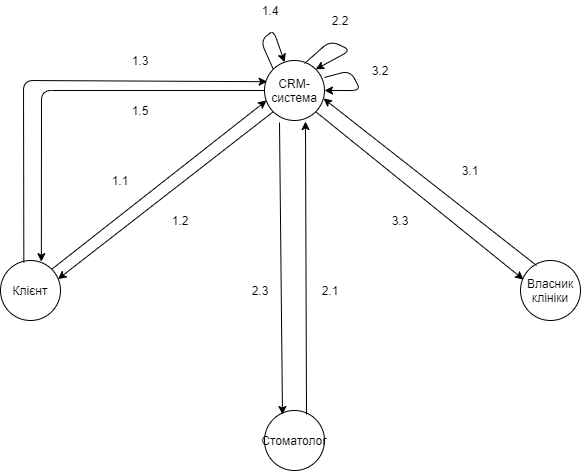


Рисунок 1.5 – Базовий алгоритм роботи програмного продукта

На рис. 1.5 зображені послідовно і паралельно виконувані процеси. Розкриємо їх суть:

* 1. – виконання клієнтом запиту на перегляд календаря записів;
  2. – надання календаря записів;
  3. – виконання запису на вільний час;
  4. – фіксація запису в БД;
  5. – сповіщення клієнта про успішне виконання запису;

2.1 – запит на отримання списку задач;

2.2 – формування списку задач;

2.3 – надання стоматологу списку задач;

3.1 – виконання власником клініки запиту на отримання звітів;

3.2 – формування звітів;

3.3 – виведення необхідної інформації.

Надання користувачам цінностей виконується на етапах 1.2, 1.3-1.5, 2.3 і 3.3.

## **1.9 Масштабованість**

У попередніх пунктах вже розглядалася тема масштабування. Було розглянуто питання географічного масштабування: у разі успішного просування системи, яка розробляється, на ринку, є можливим розширення за межі Одеси та вихід на ринки середніх і великих міст півдня України (Миколаїв, Херсон, Запоріжжя, Дніпро, Маріуполь). Після цього можна виконувати просування на північ і досягати повного покриття України.

Можливості та особливості програмного продукту, який розробляється, дозволяють пропонувати наші послуги в країнах СНД, а в разі створення англомовної версії – і в певних країнах ЄС.

Що стосується галузевого масштабування, то можливості нашої CRM-системи набагато більше обмежені. Сформованість у відповідності до специфіки роботи стоматологів не дозволяє розширювати даний проект для здійснення продажів поліклінікам або медикам інших напрямків (в силу своєї трудомісткості і низької актуальності). Те саме стосується і переходу в галузі, які не пов'язані з медициною: розваги і туризму, виробництва і сільського господарства.

Однак, цінність, яка пропонується власникам стоматологічних клінік, може бути цікава іншим представникам приватного підприємництва. Попит на оперативне отримання інформації про стан діяльності бізнес-компаній досить високий у всіх напрямках виробництва товарів і надання послуг.

Масштабування даного проекту може здійснюватися шляхом активізації рекламної діяльності. Використання різніх відів реклами, в сумі із якісним супроводом проданих підпісок, дозволяє прогнозувати поступове збільшення кількості продажів.

## **1.10 Swot-аналіз**

Swot-аналіз (табл. 1.1) використовується для формулювання сильних і слабких сторін проекту, його можливостей і загроз для нього. Аналіз даних аспектів дозволяє визначити перспективи програмного продукту.

Таблиця 1.1 – Swot-аналіз

|  |  |
| --- | --- |
| Переваги:   1. Низька ціна підписки на місяць/рік. 2. Доступність для людей із обмеженими можливостями. 3. Висока ступінь деталізації звітів. 4. Можливість додавати функціонал системи (гнучка архітектура). | Недоліки:   1. Обмежені можливості супроводу (людські ресурси). 2. Висока трудомісткість переходу в інші області медичної сфери. |
| Можливості:   1. Мала кількість прямих конкурентів. 2. Інтерес стоматологічних клінік до переходу на CRM-системи. 3. Вигідна локація (в центрі Одеси). | Загрози:   1. Обмежена кількість стоматологічних клінік в Одесі. 2. Відсутність бажання у власників клінік переходити з однієї CRM-системи на іншу. 3. Економічна нестабільність в країні. |

Виконаний аналіз свідчить про велику кількістю конкурентних переваг даного програмного продукту. При цьому, частину із сформульованих загроз можливо подолати шляхом розширення ринку збуту.

## **1.11 Value proposition canvas**

Було визначено 3 складових профілю користувача: роботи, болі та виграші. Що стосується робіт, то, з огляду на те, що ми маємо три типи користувачів, то і кількість робіт має бути не менше трьох: управління власником стоматологічної клініки її роботою, організація роботи стоматолога, запис пацієнта на прийом і подальше лікування.

Далі, було виділено основні проблеми, які існують у наших користувачів. Зазначені проблеми було згруповано у відповідності до типів користувачів: пункти 1-3 стосуються пацієнтів стоматологічних клінік, пункти 4-6 стосуються стоматологів, а пункт 7 – власника клініки. Очікувані результати та вигоди для користувачів вказано в секції «Виграші».

Для того, щоб виконувалась відповідність потреб користувачів і пропозиції, сформованої в рамках нашого програмного продукту, необхідно щоб для кожного болю був хоча б один знеболюючий засіб, а для кожного виграшу існував хоча б один творець виграшу. В результаті аналізу отриманої канви (рис. 1.7) можна відзначити, що зазначена вимога була виконана, адже спостерігається наступна відповідність: для наявності аналогів запису по телефону, передбачена можливість запису на прийом із використанням web-застосунку; календар записів та доступ до інформації про стоматологів вирішують проблему дефіциту інформації про стоматологічну клініку та її співробітників; дефіцит інформації про хід лікування вирішується шляхом перегляду пацієнтом історії лікування; особистий кабінет стоматолога дозволяє швидко отримувати систематизовану інформацію про пацієнтів; передбачена можливість створення планів лікування; власник клініки буде витрачати менше часу на отримання інформації про діяльність клініки, адже звітність автоматизована.

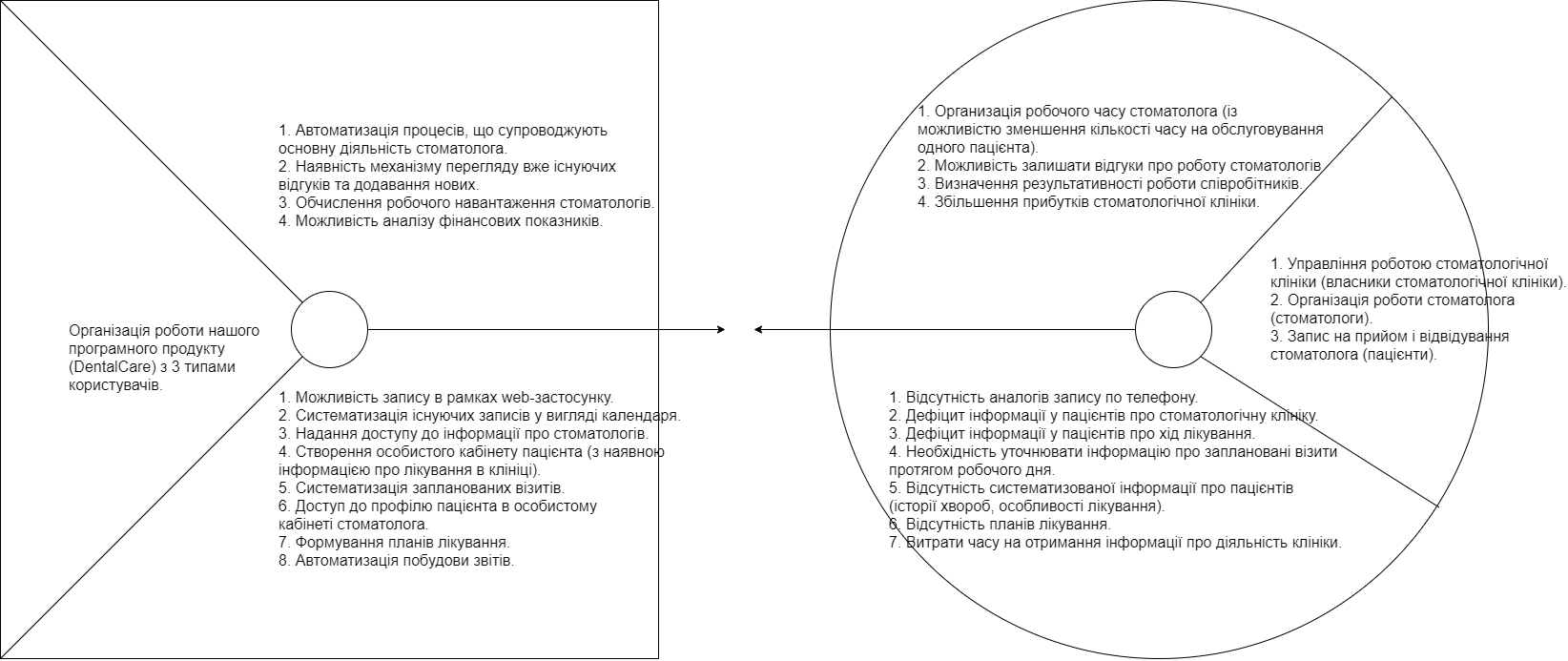


Рисунок 1.7 – Value proposition canvas

На рис. 1.7 зазначено чотири виграші. Кожному з них відповідає хоча б один творець виграшу: зменшення кількості часу, яка витрачається на обслуговування одного пацієнта досягається шляхом автоматизації процесів, які супроводжують основну діяльність стоматолога; обчислення робочого навантаження стоматолога дозволяє роботи висновки про його результативність; пацієнту надається можливість перегляду відгуків про стоматологів.

## **1.12 Аналіз конкурентів і побудова strategy canvas**

Прямими конкурентами, які надають можливості для управління відносинами стоматологічних клінік з їх клієнтами є:

* Dentist24 – 5 років на ринку CRM-систем, вартість підписки на місяць – від 25$ до 60$.
* Denta-pro – 3 роки на ринку CRM-систем, вартість підписки на місяць – від 36$ до 80$.
* Yclients – 2 роки на ринку CRM-систем, вартість підписки на місяць – 60$ (до 5 співробітників), 90$ (до 10 співробітників).

Серед CRM-систем, які не спеціалізуються на стоматології (або навіть на медицині), але при цьому надають певний набір функціоналу для управління та контролю процесів в рамках підприємства можна виділити:

* Keepin CRM – CRM-система для оптимізації роботи малого і середнього бізнесу. Система пропонує управління доходами і витратами, візуалізацію продажів, автоматизацію роботи зі складом, можливість створення картки клієнта. Ціна послуги – 45$ на місяць.
* Мегаплан – CRM-система, призначена для управління бізнесом і підвищення продажів. Серед всіх можливостей системи можна виділити: контроль робочого часу співробітників, управління задачами та дорученнями, оформлення звітів, обмін документами, облік витрат і доходів. Мінімальний тариф – 50$.
* Клієнтська база – CRM-система для автоматизації бізнес-процесів. Користувачеві системи доступні можливості обліку клієнтів, прийому заявок із сайту безпосередньо в базу, контролю за виконанням задач. Безкоштовний аккаунт – до 14 днів, базовий набір функціоналу доступний за 35$ на місяць.

Непрямими конкурентами нашої системи є традиційні шляхи вирішення проблем. Для пацієнта – це дзвінок до клініки і уточнення вільного часу. Для власника клініки – особисті комунікації зі співробітниками клініки, взаємодія з секретарями і бухгалтерією, які періодично складають паперові звіти. Стоматологи, для вирішення своїх проблем, змушені звертатися в реєстратуру для отримання списку записів на день, а також до медичних карток пацієнтів для перегляду історії хвороби.

Було виконано порівняльний аналіз нашого програмного продукту з конкурентами. В якості конкурентів були обрані 3 найбільш популярні на даний момент стоматологічні CRM-системи: Dentist24, Denta-pro і Yclients. Оцінка кожного програмного продукту виконувалась за 8 критеріями. Максимальна оцінка за кожним критерієм – 10. Результати оцінювання було систематизовано у вигляді таблиці (табл. 1.2).

Таблиця 1.2 – Порівняльний аналіз

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Наш продукт | Dentist24 | Denta-pro | Yclients |
| Календар записів | 6 | 8 | 1 | 4 |
| Прайс-лист | 8 | 6 | 8 | 8 |
| Запис на прийом | 9 | 7 | 5 | 7 |

Продовження таблиці 1.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Наш продукт | Dentist24 | Denta-pro | Yclients |
| Особистий кабінет пацієнта (перегляд історії лікування) | 8 | 6 | 2 | 3 |
| Звітність | 4 | 3 | 7 | 5 |
| Систематизація запланованих візитів | 7 | 3 | 5 | 5 |
| Створення планів лікування | 9 | 1 | 6 | 1 |
| Ціна | 6 | 8 | 4 | 2 |

На базі отриманих даних була побудована strategy canvas (рис. 1.8). На ній зображено 4 графіка, у відповідності до кількості програмних продуктів, які підпадали аналізу. По осі X відмічені фактори, на основі яких було виконано аналіз, а по осі Y – результати оцінювання.

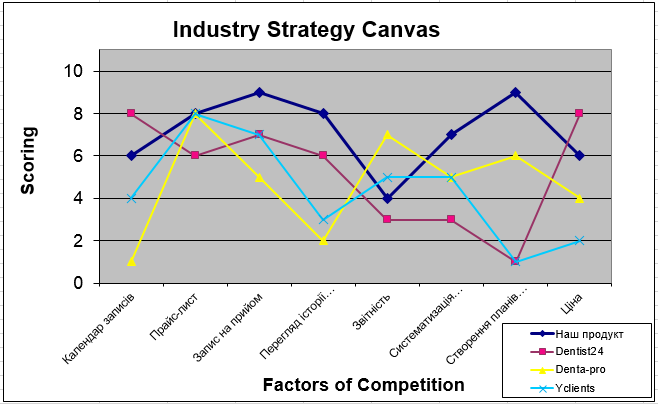


Рисунок 1.8 – Strategy canvas

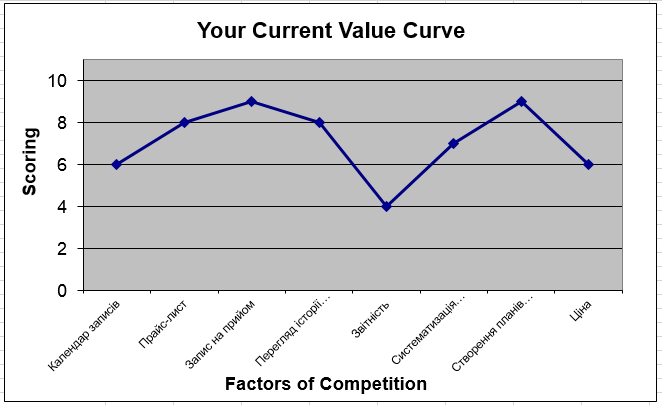


Рисунок 1.9 – Value curve

Current value curve (рис 1.9) відображає графік тільки нашого програмного продукту для розуміння того, в яких точках існують явні провали, а в яких - піки. Аналіз strategy canvas показує, що графік нашого програмного продукту, переважно, перебуває вище, ніж графіки конкурентів. Даний сигнал є позитивним.

## **1.13 Створення стратегії блакитного океану на основі four action framework**

Для створення стратегії блакитного океану (а, отже, виходу з червоного океану) необхідно проаналізувати стратегічну канву. Це дозволяє виділити окрему область, в якій немає такої великої кількості конкурентних пропозицій.

Для цього необхідно виконати наступні дії: фактори, в яких всі або майже всі графіки перетинаються, необхідно скасувати. Це дозволяє прибрати з розгляду ті характеристики, в яких наш програмний продукт нічим не відрізняється від конкурентів.

На другому етапі необхідно підвищити значення тих факторів, в яких ми явно програємо. Це дозволяє поліпшити нашу пропозицію на ринку. Вплив факторів, в яких наш графік знаходиться принципово вище всіх інших пропозицій, необхідно знизити. Це дозволяє зберегти конкурентну перевагу, але уникнути надмірної трудомісткості функціоналу, який відповідає даним факторам.

Для того, щоб розширити межі ринку, слід додати нові фактори, які відсутні у наших конкурентів. В результаті аналізу стратегічної канви, були прийняті наступні рішення:

Скасувати: фактор прайс-листа.

Знизити: фактор створення планів лікування.

Підвищити: фактор звітності і фактор ціни.

Створити: фактор відгуків про стоматолога.

## **1.14 Формулювання стратегії із використанням методу оцінки впливу**

В даному випадку мета розробки програмного продукту розглядається з точки зору performance requirements. Для їх визначення будемо використовувати наступний варіант декомпозиції для performance attributes (рис. 1.10).

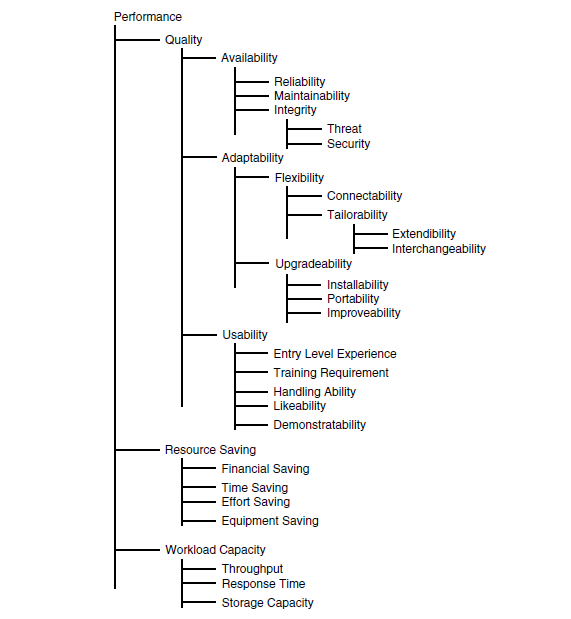


Рисунок 1.10 – Performance attributes

Стоматологи, а також власники стоматологічних клінік, використовують даний продукт при виконанні посадових обов'язків, а тому однією із цілей є підвищення надійності (reliability) CRM-системи.

Ціль: збільшення середньої кількості часу до відмови (напрацювання на відмову) зі 100 годин до 1000 годин.

Якщо вже так сталося, що виникла відмова, то її вирішення повинно займати не більше 5 хвилин, з метою зниження впливу відмови системи на роботу фахівця (стоматолога, власника клініки). Дані вимоги стосуються атрибуту Maintainability.

Ціль: зменшення кількості часу, необхідного для відновлення роботи системи, з 15 до 5 хвилин (наприклад, для введення в роботу резервної бази даних при збої в роботі із основною базою даних).

Існують три основні сегменти користувачів: клієнти стоматологічних клінік, стоматологи і власники клінік. Однак, з огляду на специфіку роботи власників стоматологічних клінік, вони можуть передоручати частину своїх обов'язків секретарям або іншим співробітникам клініки, при цьому додаючи завдання, виконання яких властиве для даних фахівців. Отже, необхідно задуматися про те, що може виникнути необхідність створювати нові типи користувачів, для яких буде характерним свій набір функціоналу. Це називається Extendibility (Adaptability.Flexibility.Tailorability.Extendibility):

Ціль: зниження витрат, необхідних для додавання нового типу користувача (до прикладу, секретаря) з 300$ до 100$.

При цьому, вже створені типи користувачів можуть потребувати періодичного оновлення функціоналу. Дана властивість системи називається Improvability (Adaptability.Upgradability.Improvability):

Ціль: зниження вартості внесення змін в функціонал (приклад таких змін – покращення функціонала звітності: збільшення варіантів вибору параметрів, ступеня деталізації та візуалізації) системи з 500$ до 200$.

Далі було розглянуто питання, яке стосується зручності використання системи користувачами. Необхідно, щоб кількість часу, яку необхідно витратити на навчання для досягнення потрібного результату, шляхом взаємодії з системою, було знижено з 10 хвилин до 3 хвилин (приклад – навчання стоматолога працювати з електронною базою пацієнтів через особистий кабінет). Це називається Training Requirement (Usability.Training Requirement):

Ще однією ціллю є підвищення рівня задоволення користувачів від використання програмного продукту з 70% (оцінка, отримана при демонстрації робочого прототипу) до 100%. Дана вимога стосується атрибуту User Opinion (Usability.User Opinion):

В якості конкретного типу користувача розглядається власник стоматологічної клініки при виконанні запиту фінансової звітності за певний період часу.

Все раніше перераховане стосувалося Performance Savings. Тепер розглянемо Resource Savings (це все ще performance attributes). Для власника стоматологічної клініки важливим є зниження витрат на клініку при використанні системи – Financial Saving.

Ціль: зниження витрат на клініку (з боку власника клініки) зі 100 000$ за рік до 80 000$ за рік, шляхом використання нашого ПП (збільшення суми зекономлених грошей з 0 до 20 000$).

Для стоматологів більш важливим моментом є заощаджений час – Time Saving.

Ціль: зменшення кількості часу, яку необхідно витратити для обслуговування 1 клієнта, на 10 хвилин, завдяки частковій автоматизації виконання процесів, які супроводжують основні процеси (збільшення виграшу у часі з 0 до 10 хвилин).

Сформулювавши цілі, можна переходити до методу оцінки впливу. Даний метод може використовуватися для вирішення цілого спектру задач в рамках проекту: порівняння альтернативних ідей для визначення кращої, оцінки стану проекту шляхом визначення відповідності вимогам, аналізу ризиків. Тому, перш ніж виконувати обчислення, потрібно визначитися для чого саме задіюється метод оцінки впливу. У нашому випадку існує лише одна ідея, а тому метод оцінки впливу було використано для визначення відповідності ідеї кожній цілі, визначення показника «Performance to Cost Ratio» і розробки стратегії, шляхом визначення необхідних для внесення в програмний продукт змін.

При виставленні оцінок можна використовувати досвід участі в інших проектах, результати роботи інших систем, припущення. У нашому випадку, в основному, використовувалися останні два варіанти. Також, при виставленні оцінок було дотримано принципу «Сумнівні числа краще, ніж їх відсутність». Результат можна спостерігати в табл. 1.3.

Таблиця 1.3 – Оцінка впливу ідеї

|  |  |  |  |
| --- | --- | --- | --- |
| Вимоги:  цілі і бюджети | Оцінка впливу нашої ідеї | Сума для вимоги (сума процентного впливу) | Сума процентних значень невизначеності |
| Reliability  100 <–> 1000 годин | 650 (550)±100 годин  61%±11% | 61% | ±11% |
| Maintainability  15 <–> 5 хвилин | 8 (7)±3 хвилин  70%±30% | 70% | ±30% |
| Extendibility  300 <–> 100 доларів США | 150 (150)±20 доларів  75%±10% | 75% | ±10% |
| Improvability  500 <–> 200 доларів США | 370 (130)±60 доларів  43%±20% | 43% | ±20% |
| Training Requirement  10 <–> 3 хвилин | 4 (6)±2 хвилини  86%±29% | 86% | ±29% |
| User Opinion  70 <–> 100 відсотків | 93 (23)±5 відсотки  77%±17% | 77% | ±17% |

Продовження таблиці 1.3

|  |  |  |  |
| --- | --- | --- | --- |
| Вимоги:  цілі і бюджети | Оцінка впливу нашої ідеї | Сума для вимоги (сума процентного впливу) | Сума процентних значень невизначеності |
| Financial Saving  100 000<–>80 000 доларів за рік | 91 000 (9000)±2000 доларів за рік  45%±10% | 45% | ±10% |
| Time Saving  0 <–> 10 хвилин на пацієнта | 3 (3)±1 хвилини  30%±10% | 30% | ±10% |
| Сумма показників виконання | 487% |  |  |
| Капітал  0 <–> 10 000 доларів США | 8 000 (8000)±1 500 доларів США  80%±15% | 80% | ±15% |
| Сума витрат | 80% |  |  |
| Співвідношення цілей до витрат | 6,09 (487/80) |  |  |

Аналізуючи отримані дані, можна зробити наступні висновки: необхідно дослідити можливість поліпшення показників для атрибутів Time Saving і Financial Saving шляхом внесення змін до програмного продукту, оскільки цілі, в яких вони фігурують, прямо пов'язані з вирішенням проблем користувачів, сформульованих раніше. Також, рекомендується переглянути архітектуру, з метою спрощення внесення змін (Improvability).

## **1.15 Аналіз останніх публікацій на тему доступності web-інтерфейсів**

Останнім часом з'явилась достатньо велика кількість публікацій на тему доступності web-інтерфейсів. У даній роботі проаналізовані 10 публікацій на зазначену тему.

Роботи [1-3] описують характеристику доступності. Вона розглядається на часовому проміжку з 2005 року до 2018 року. Відправною точкою є формування стандартів WCAG (Web Content Accessibility Guidelines – рекомендації для забезпечення доступності вмісту web-ресурсів), що дозволяє визначати єдині вимоги до доступності web-сайтів. Формулювання стандарту доступності включає в себе перерахування існуючих рівнів доступності.

Публікації [4-5] розглядають технологічні шляхи забезпечення можливості взаємодії користувачів з web-інтерфейсом шляхом використання спеціалізованих контролерів.

Наступною групою є публікації [6-9]. В якості засобів підвищення рівня доступності, пропонуються структуризація елементів html, вибір візуального представлення елементів і анімації (стилізація елементів), використання атрибутів елементів при роботі зі «Screen Reader». «Screen Reader» являє собою технологію, яка дозволяє трансформувати текстовий вміст web-інтерфейсу в аудіоформат.

В роботі [10] розглядається залежність кількості продажів від рівня доступності web-сайту. При забезпеченні певного рівня доступності web-інтерфейсу, розширюється аудиторія людей, які можуть взаємодіяти з даним сайтом. У випадку інтернет-магазину, мова йде по розширення переліку користувачів. Однак, хоча дана публікація аргументує необхідність забезпечення доступності, вона не вказує спосіб для досягнення цільового рівня доступності.

# **2 РОЗРОБКА РОЗШИРЕНОЇ МОДЕЛІ ДОСТУПНОСТІ**

## **2.1 Основні стандарти і рекомендації для забезпечення доступності web-інтерфейсу**

Функціонування програмного забезпечення в рамках мережі Internet передбачає дотримання стандартів. За розробку і впровадження даних стандартів відповідає організація W3C (World Wide Web Consortium). Одним з таких стандартів, є WCAG. Він являє собою набір рекомендацій, реалізація яких дозволяє забезпечити доступність web-контенту. Доступність – це характеристика web-інтерфейсу, яка означає можливість отримання інформації при взаємодії з інтерфейсом усіма групами користувачів, в тому числі людьми, які мають порушення, пов’язані зі здоров’ям. В першу чергу, доступність орієнтується саме на людей з обмеженими можливостями. WCAG 2.0 був опублікований в грудні 2008 року і став стандартом ISO, ISO / IEC 40500: 2012 у жовтні 2012 року. WCAG 2.1 став рекомендацією W3C в червні 2018 року [11].

Стандарт WCAG 2.0 виділяє 4 основні вимоги до web-сайтів: зрозумілість, надійність, сприймаємість, керованість. На підставі відповідності об'єкта, аналіз якого виконується, даним критеріям, виділяють 3 рівня доступності інтерфейсів: мінімальний (A), середній (АА), високий (ААА) [12]. Можливий перехід від одного рівня доступності до іншого.

Зазначені вище положення стандартів і, як наслідок, рівні доступності, формувалися на підставі аналізу варіантів обмеження можливостей людини при взаємодії з контентом web-сторінки. Комплексне вивчення даних моментів дозволяє побудувати модель доступності, яка, у випадку реалізації, призведе до задіяння, в якості цільової аудиторії, всіх основних груп людей з обмеженими можливостями. Це буде виконано шляхом підвищення рівня доступності web-інтерфейсу.

## **2.2 Порушення, пов’язані зі здоров’ям користувачів**

Було виділено чотири основних типи порушень здоров'я користувачів web-сайтів:

*проблеми із зором*

Беручи до уваги поточний стан ресурсів мережи Internet, можна визначити, що найбільше проблем з доступністю виникає у людей, які мають порушення зору. Причиною цього є той факт, що 80% інформації про навколишній світ людина отримує через зоровий аналізатор [13]. Інформація, яка міститься на web-сторінці, не є винятком. Розглядаючи питання дальтонізму, можна зазначити, що люди, які страждають від цього захворювання, відчувають проблеми у тому випадку, якщо комбінації кольорів фону і тих елементів, які на ньому розташовані, підібрані неправильно [14].

*проблеми зі слухом*

Люди, які страждають на глухоту або ж на інші слухові порушення, у порівнянні з першою категорією людей, набагато рідше стикаються з проблемою доступності. Однак, останнім часом можна спостерігати тенденцію до збільшення кількості мультимедіа елементів на web-сайтах. При цьому, в більшості випадків, відео-контент не супроводжується синхронізованими субтитрами. Це робить неможливим повноцінне сприйняття інформації даною категорією людей.

*порушення у роботі опорно-рухового апарату*

Велика кількість користувачів відчувають труднощі, певного характеру, у тих випадках, коли виникає необхідність здійснювати точні і швидкі переміщення курсора миші [15]. Також, люди, які мають порушення в роботі опорно-рухового апарату, дуже часто не можуть утримувати одночасно кілька клавіш клавіатури.

*когнітивні проблеми*

На превеликий жаль, когнітивні порушення у людей, які взаємодіють з інтерфейсами web-сайтів і web-застосунків, враховуються набагато рідше, ніж інші фізичні вади. В результаті маємо ситуацію, при якій люди, які відчувають проблеми з короткочасною пам'яттю, а також люди, які мають порушення просторового мислення, відчувають проблеми в ті моменти, коли навігація по вмісту web-сторінки достатньо складна. Користувачі з дислексією можуть відчувати проблеми при спробах прочитати тексти великого обсягу, особливо, якщо вони не структуровані за абзацами [16].

## **2.3 Апаратні та програмні засоби вирішення проблеми**

Слід відзначити, що для поліпшення якості взаємодії користувача з web-інтерфейсом можна застососувати три абсолютно різних підходи, кожен з яких має свої переваги та недоліки. Першим таким підходом є використання спеціалізованої техніки, яка під’єднується до персонального комп'ютера. Серед такої техніки можна виділити джойстики, програмовані клавіатури, виносні комп'ютерні кнопки, оптичні миші, що одягаються на палець.

Користувачі підбирають технічні рішення у відповідності до індивідуальних потреб. Для тих, хто може використовувати тільки одну руку, розроблено спеціальні клавіатури під ліву чи праву руку. Розташування клавіш на такій клавіатурі зручніше, ніж на тій, до якої ми всі звикли.

При порушеннях дрібної моторики може допомогти клавіатура, яка за розмірами більша за звичайну. Клавіші на ній більші і розташовані на більших видстанях одна від одної. Міні-клавіатури підходять тим групам людей, для яких проблемою є не дрібна моторика, а обмежений обсяг рухів, що не дозволяє використовувати клавіатуру звичайних розмірів.

Для людей з важкими руховими порушеннями існують мімічні рішення. Замість звичної роботи з клавіатурою і комп'ютерної мишею пропонуються сканування, азбука Морзе і стеження очима. Крім того, до комп'ютера можна під’єднати спеціальні перемикачі, які дозволять користуватися комп'ютером за допомогою єдиної м'язи, якою людина в стані свідомо керувати (наприклад, палець, м'язи ніг, рот).

Однак, як вже було зазначено раніше, даний підхід, окрім позитивних сторін, має і негативні сторони. Основним мінусом, в даному випадку, є вартість покупки спеціалізованого пристрою. Ціна може коливатися від двох тисяч гривень, при покупці виносної комп'ютерної миші, і до сімдесяти тисяч гривень для обладнання автоматизованого робочого місця для учнів з порушеннями зору або порушеннями слуху [17].

Розглянувши перший варіант вирішення проблеми, можна переходити до другого варіанту. Він базується на використанні програмного забезпечення, яке спрямоване на індивідуальний підхід до кожного користувача. Індивідуальний підхід реалізується шляхом попереднього налаштування даного ПЗ і подальшого його використання. Серед таких ПЗ можна виділити «Screen Reader». Він являє собою форму ассистивної технології, яка дозволяє людям з порушеннями зору організувати роботу з текстом. Програмне забезпечення виконує читання зазначених уривків і перехід між ними при отриманні вказівок від користувача [18]. Слід зазначити, що даний підхід також має певні недоліки. Спеціалізоване програмне забезпечення не завжди доступне у безкоштовному варіанті. Не передбачається зміна відображення web-інтерфейсу для зручності роботи користувача. При використанні програм, користувач змушений виконувати додаткові дії і, які у більшості, вимагають витрачання додаткового часу.

Найкращим варіантом є використання вбудованих програмних рішень, спеціалізованих під певні мови програмування, які дозволяють, враховуючи потреби кінцевого користувача, організувати взаємодію з web-інтерфейсом, шляхом налаштування необхідного формату відображення. Деякі підходи до створення web-інтерфейсів вже активно використовуються, інші ж, на даний момент, зустрічаються досить рідко. Наявність даних компонент і визначає доступність web-інтерфейсів. Для кращого розуміння забезпечення доступності шляхом використання готових програмних рішень, у відповідності до стандарту WCAG 2.0, доцільно виконати побудову моделі доступності.

## **2.4 Класична модель доступності**

На основі проведеного аналізу 100 web-сайтів і web-застосунків, різних за тематикою, вдалося визначити і побудувати класичну модель доступності (classic accessibility model – CAM). Ця модель складається з компонент, реалізації яких присутні на більшості досліджених web-ресурсів. Вона дозволяє, в деякій мірі, спростити взаємодію користувача з web-інтерфейсом, а також поліпшити сприйняття користувачем інформації, вказаної на ньому.

(2.1)

де c – правильний підбір кольорів,

n – правильне іменування посилань,

t – розділення тексту на тематичні блоки,

h – допомога користувачу при виправлені помилок,

d – відсутність небезпечних елементів та анімацій.

Надамо короткий опис даних компонент:

* правильний підбір кольорів

Необхідно уникати поєднання яскравих кольорів, а також комбінованого використання жовтого, синього і зеленого кольорів, так як дане рішення буде створювати серйозні труднощі для користувачів, які страждають на дальтонізм. Слід продумати яким буде контраст тексту та фону. Чорний текст на білому фоні – це найкраще рішення, оскільки воно гарно сприймається більшістю аудиторії.

* правильне іменування посилань

Варіант, при якому в реченні «Щоб дізнатися більше про ОНПУ натисніть тут» посиланням є лише останнє слово є неправильним. Більш коректним буде наступне формулювання: «Відвідайте web-сайт ОНПУ для отримання більш детальної інформації» (посиланням є весь текст, розташований у лапках). Також, слід виділяти посилання на тлі решти тексту. Це можна виконати за допомогою підкреслення і вибору кольору, відмінного від того, який використовується для решти тексту.

* розділення тексту на тематичні блоки

Рекомендовано розділяти великі текстові блоки інформації на більш дрібні (абзаци). Це дозволить поліпшити сприйняття і запам'ятовування прочитаної інформації всіма категоріями користувачів. Особливо це буде доцільно для web-сайтів, які відвідують, переважно, користувачі похилого віку та люди з обмеженими здібностями до навчання.

* допомога користувачу при виправленні помилок

Даний підхід активно використовується при виконанні користувачем заповнення форми. У тому випадку, якщо користувач неправильно заповнив поле, біля цього поля повинна з'явитись текстова підказка з вимогами до заповнення. У тому випадку, якщо не будуть виводитись текстові підказки, а сповіщення про неправильне заповнення поля буде обмежуватися зміною кольору його кордонів, ми будемо мати ситуацію, при якій людина з вадами зору зможе не помітити цього факту. В описаній ситуації час заповнення форми значно збільшується.

* відсутність небезпечних елементів та анімацій

У цьому пункті йдеться про зміну слайдів з великою частотою, несподівані спалахи, різкі звуки, появу та зникнення елементів. Все перераховане вище є вкрай шкідливим для людей з психічними розладами.

## **2.5 Розширена модель доступності**

З орієнтуванням на виділені раніше порушення здоров'я користувачів, була виконана побудова розширеної моделі доступності web-сайтів (extended accessibility model – EAM). Дана модель розширює класичну. Це означає, що всі компоненти класичної моделі були додані до розширеної моделі. Крім цього, було досліджено і конкатеновано до моделі доступності ще сім компонент.

(2.2)

де | – операція конкатенації двух кортежів,

a – використання атрибутів alt і title,

m – забезпечення можливості керування за допомогою клавіатури,

r – створення часових резервів,

s – визначення допустимих розмірів елементів інтерфейсу,

mc – забезпечення субтитрами медіа-контенту,

d – розташування роздільників в абревіатурах,

i – надання інформації про способи забезпечення доступності.

Наведемо опис кожної із компонент розширеної моделі доступності web-сайту:

* використання атрибутів alt і title

Для людини з порушеннями зору, яка використовує програму прочитання тексту, відображеного на екрані (програмне забезпечення, яке читає вголос текст, розташований на web-сторінці), значення атрибутів alt і title озвучуються, що дозволяє дізнатися про вміст та направленість виконання подальших дій.

* забезпечення можливості керування за допомогою клавіатури

Взаємодія з елементами web-інтерфейсу за допомогою клавіатури активно використовується користувачами, які мають порушення опорно-рухового апарату. Завдяки заздалегідь продуманій розробниками роботі з фокусом, досягається можливість зручного переміщення між елементами форми, випадаючих вікон та панелей навігації.

* створення часових резервів

Резервування певної кількості часу є збільшенням кількості часу, який виділяється користувачу для виконання запланованих дій. Воно може бути застосовано при проведенні онлайн-аукціонів, або ж онлайн-тестів. Це доповнення дозволяє поліпшити якість взаємодії літніх людей, а також людей, які мають проблеми із зором, з інтерфейсом web-сайту.

* визначення допустимих розмірів елементів інтерфейсу

Користувачам, які мають проблеми з опорно-руховим апаратом, може бути достатньо важко натискати, за допомогою комп’ютерної миші, на елементи інтерфейсу невеликого розміру (наприклад, кнопки). Для вирішення вказаної проблеми, необхідно збільшити розміри області, з якої буде взаємодіяти користувач. Це, також, дозволить спростити роботу з web-інтерфейсом людям, які мають порушення зору.

* забезпечення субтитрами медіа-контенту

У тому випадку, якщо web-сайт містить велику кількість різноманітного відео-контенту, буде доцільним надати субтитри до цього матеріалу. У разі, якщо відеоматеріал власного виробництва, то можна скористатися інструментами таких відео-хостингів, як YouTube для того, щоб додати субтитри.

* розташування роздільників в абревіатурах

Якщо в тексті web-сторінки наявні абревіатури, то рекомендовано ставити крапки між літерами абревіатур. Наприклад, якщо є необхідність посилатися на Інститут комп'ютерних систем, то буде більш правильним записати «І.К.С.», а не «ІКС». Програма прочитання тексту з екрану не розпізнає абревіатуру без точок і прочитає її як слово (І-К-С буде прочитано як «ікс»).

* надання інформації про способи забезпечення доступності

У тому випадку, якщо розробникам web-сайту вдалося забезпечити певний рівень доступності, слід створити окремий розділ, в якому будуть детально викладені основні можливості для людей, які мають порушення здоров’я, при взаємодії з web-інтерфейсом. Це дозволить найбільш повно використовувати закладений функціонал. Також, можна включити в даний розділ інформацію про те, як налаштувати свій персональний комп'ютер і браузер, для коректної роботи з вбудованими атрибутами програмного коду.

На підставі аналізу зв'язків між порушеннями здоров'я користувачів і компонентами розширеної моделі, які роблять можливим вирішення проблеми обмеженої взаємодії з web-інтерфейсом, була виділена наступна статистична інформація:

– за наявності проблем із зором у користувача, йому допомагають 10 із 12 компонент (83% від загальної кількості), а саме: правильний підбір кольорів, правильне іменування посилань, розділення тексту на тематичні блоки, допомога користувачу при виправленні помилок, відсутність небезпечних елементів та анімацій, використання атрибутів alt і title, створення часових резервів, визначення допустимих розмірів елементів інтерфейсу, розташування роздільників в абревіатурах, надання інформації про способи забезпечення доступності;

– за наявності у користувача проблем зі слухом, йому допомагають 4 з 12 компонент (33%), а саме: забезпечення субтитрами медіа-контенту, створення часових резервів, надання інформації про способи забезпечення доступності, допомога користувачу при виправленні помилок;

– за наявності у користувача проблем з опорно-руховим апаратом, йому допомагають 5 з 12 компонент (42%), а саме: забезпечення можливості керування за допомогою клавіатури, створення часових резервів, визначення допустимих розмірів елементів інтерфейсу, допомога користувачу при виправленні помилок, надання інформації про способи забезпечення доступності;

– за наявності у користувача когнітивних проблем, йому допомагають 6 з 12 компонент (50%), а саме: розділення тексту на тематичні блоки, допомога користувачу при виправленні помилок, відсутність небезпечних елементів та анімацій, правильний підбір кольорів, створення часових резервів, надання інформації про способи забезпечення доступності.

Для порівняння, класична модель доступності web-сайтів дозволяє досягнути наступних показників:

– 5 компонент допомагають користувачам із порушеннями зору взаємодіяти з web-інтерфейсом, а саме: правильний підбір кольорів, правильне іменування посилань, розділення тексту на тематичні блоки, допомога користувачу при виправленні помилок, відсутність небезпечних елементів та анімацій;

– 1 компонента допомагає користувачам із порушеннями слуху взаємодіяти з web-інтерфейсом, а саме: допомога користувачу при виправленні помилок (візуальні сигнали);

– 1 компонента допомагає користувачам із порушеннями опорно-рухового апарату взаємодіяти з web-інтерфейсом, а саме: допомога користувачу при виправленні помилок;

– 4 компоненти допомагають користувачам з когнітивними проблемами взаємодіяти з web-інтерфейсом, а саме: правильний підбір кольорів, розділення тексту на тематичні блоки, відсутність небезпечних елементів та анімацій, допомога користувачу при виправленні помилок.

Результати виконання аналізу наведено в табл. 2.1. Комірки, виділені сірим кольором, демонструють переваги розширеної модели доступності.

Таблиця 2.1 – Використання елементів моделі доступності при різних типах порушення здоров’я

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Порушення зору | Порушення слуху | Порушення опорно-рухового апарату | Когнітивні порушення |
| Правильний підбір кольорів | CAM / EAM | - | - | CAM / EAM |
| Використання атрибутів alt і title | EAM | - | - | - |
| Правильне іменування посилань | CAM / EAM | - | - | - |
| Забезпечення можливості керування за допомогою клавіатури | - | - | EAM | - |
| Створення часових резервів | EAM | EAM | EAM | EAM |
| Розділення тексту на тематичні блоки | CAM / EAM | - | - | CAM / EAM |

Продовження таблиці 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Порушення зору | Порушення слуху | Порушення опорно-рухового апарату | Когнітивні порушення |
| Допомога користувачу при виправленні помилок | CAM / EAM | CAM / EAM | CAM / EAM | CAM / EAM |
| Визначення допустимих розмірів елементів інтерфейсу | EAM | - | EAM | - |
| Забезпечення субтитрами медіа-контенту | - | EAM | - | - |
| Розташування роздільників в абревіатурах | EAM | - | - | - |
| Відсутність небезпечних елементів та анімацій | CAM / EAM | - | - | CAM / EAM |

Продовження таблиці 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Порушення зору | Порушення слуху | Порушення опорно-рухового апарату | Когнітивні порушення |
| Надання інформації про способи забезпечення доступності | EAM | EAM | EAM | EAM |

де EAM – елемент тільки розширеної моделі доступності,

CAM / EAM – елемент і класичної і розширеної моделей доступності.

За результатами статистики, можна зробити наступний висновок: розширення моделі доступності дозволило збільшити кількість задіяних компонент, що в свою чергу розширило функціональні можливості користувачів, у яких спостерігаються порушення, пов'язані зі здоров'ям. Окремо слід зазначити, що розширена модель доступності передбачає перехід до високого рівня доступності (ААА) саме шляхом покращення сприймаємості, зрозумілості та керованості. Сприймаємість та зрозумілість забезпечуються шляхом додавання компонент: правильний підбір кольорів, використання атрибутів alt і title, створення часових резервів, правильне іменування посилань, розділення тексту на тематичні блоки, визначення допустимих розмірів елементів інтерфейсу, забезпечення субтитрами медіа-контенту, розташування роздільників в абревіатурах, відсутність небезпечних елементів та анімацій, надання інформації про способи забезпечення доступності, допомога користувачу при виправленні помилок. Покращення керованості web-інтерфейсу забезпечується за допомогою компоненти «Забезпечення можливості керування за допомогою клавіатури».

# **3 ФОРМУЛЮВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ**

## **3.1 Визначення варіантів використання системи**

В результаті виконання аналізу предметної області та потреб, які існують у пацієнтів стоматологічних клінік, власників даних клінік, а також стоматологів, вдалось визначити 13 варіантів використання системи (рис. 3.1).

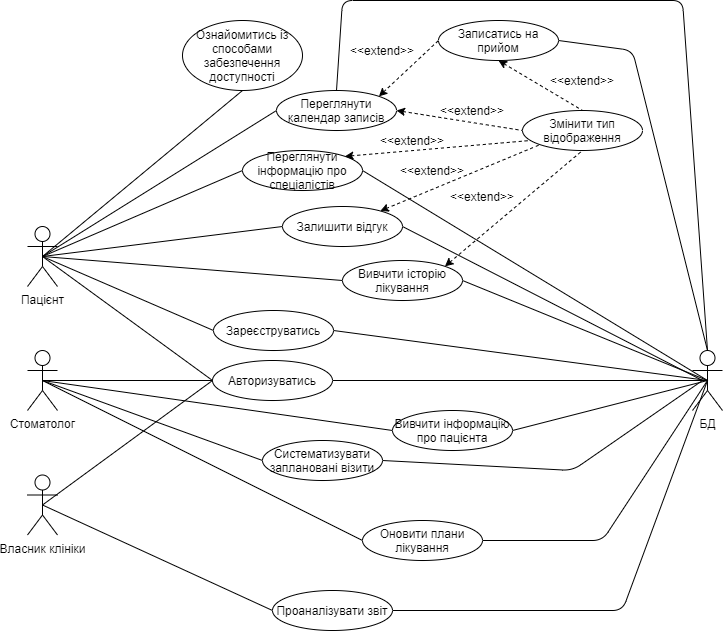


Рисунок 3.1 – Діаграма варіантів використання

Деталізуємо сценарії всіх варіантів використання, зазначених на діаграмі прецедентів (К – користувач, С – система).

ВВ «Зареєструватись»

1. К. запитує реєстрацію нового пацієнта.

2. С. надає форму для реєстрації.

3. К. заповнює форму необхідними даними.

4. Якщо К. не зміг заповнити форму, то С. виводить повідомлення з інструкціями.

5. Якщо К. зміг заповнити форму, то С. перевіряє відсутність раніше зареєстрованого користувача з вказаними даними. Якщо користувач з вказаними даними вже був зареєстрований, то С. повідомляє про це і пропонує ввести інші дані. Інакше, С. виконує реєстрацію К.

ВВ «Авторизуватись»

1. К. запитує авторизацію в системі.

2. С. надає форму для авторизації.

3. К. заповнює форму даними, вказаними при реєстрації.

4. Якщо К. не зміг правильно заповнити форму, то С. виводить повідомлення з інструкціями.

5. Якщо К. зміг правильно заповнити форму, то С. зчитує дані з БД і перевіряє коректність. Якщо дані були вказані коректно, то С. виконує авторизацію К. Інакше, С. повідомляє, що К. із вказаними даними не зареєстрован і пропонує повторно заповнити форму.

ВВ «Переглянути календар записів»

1. К. виконує запит на перегляд календаря записів.

2. C. завантажує із БД і визначає доступні прізвища стоматологів.

3. С. запитує прізвище стоматолога та дату.

4. К. обирає прізвище стоматолога і вказує необхідну дату.

5. С. завантажує із БД і структурує необхідні дані.

6. С. надає доступ К. до записів на обрану дату.

7. Якщо К. бажає змінити формат відображення інформації, то виклик ВВ «Змінити тип відображення». Якщо К. бажає виконати запис на вказану дату, то виклик ВВ «Записатись на прийом».

ВВ «Записатись на прийом»

1. С. визначає доступний для запису час.

2. Якщо К. бажає змінити формат відображення інформації, то виклик ВВ «Змінити тип відображення». Інакше, К. обирає необхідний час.

3. С. зберігає дані в БД і повідомляє К. про успішне створення запису.

ВВ «Переглянути інформацію про спеціалістів»

1. К. запитує інформацію про стоматологів.

2. С. завантажує необхідні дані з БД.

3. С. надає доступ К. до інформації про стоматологів. Якщо К. бажає змінити формат відображення інформації, то виклик ВВ «Змінити тип відображення».

ВВ «Залишити відгук»

1. К. виконує запит на створення відгуку про роботу стоматолога.

2. С. надає форму для відгуку.

3. Якщо К. бажає змінити формат відображення інформації, то виклик ВВ «Змінити тип відображення». Інакше, К. заповнює форму необхідними даними.

4. С. зберігає відгук в БД і повідомляє К. про успішне створення відгуку.

ВВ «Вивчити історію лікування»

1. К. виконує запит на перегляд історії лікування.

2. C. завантажує необхідні дані із БД.

3. С. надає доступ К. до історії лікування. Якщо К. бажає змінити формат відображення інформації, то виклик ВВ «Змінити тип відображення».

ВВ «Змінити тип відображення»

1. С. виводить доступні типи зміни відображення.

2. К. обирає необхідний тип змін.

3. C. визначає поточне значення параметру і всі можливі варіанти.

4. К. обирає нове значення параметру.

5. С. зберігає нове значення параметру і змінює тип відображення у відповідності до параметру.

ВВ «Систематизувати заплановані візити»

1. К. виконує запит на перегляд запланованих візитів.

2. С. запитує дату.

3. К. обирає необхідну дату.

4. С. завантажує з БД і структурує необхідні дані.

5. С. надає доступ К. до запланованих візитів на обрану дату.

ВВ «Вивчити інформацію про пацієнта»

1. К. виконує запит на перегляд інформації про пацієнта.

2. C. завантажує з БД і визначає доступні прізвища пацієнтів.

3. C. запитує прізвище пацієнта.

4. К. обирає необхідне прізвище.

5. С. завантажує дані про пацієнта з БД.

6. С. надає доступ К. до інформації про пацієнта.

7. К. запитує план лікування пацієнта.

8. С. відображає діючий план лікування. Якщо доступні більш ранні плани лікування, то С. активує можливість переходу до попередніх планів.

ВВ «Оновити плани лікування»

1. К. виконує запит на оновлення планів лікування.

2. C. завантажує з БД і визначає доступні прізвища пацієнтів.

3. С. запитує прізвище пацієнта.

4. К. обирає прізвище зі списку.

5. С. визначає визначає наявність вже існуючих планів лікування для обраного пацієнта. Якщо таких планів немає, то С. повідомляє про це. Інакше, С. виводить останній доступний план і надає доступ до його редагування.

6. Якщо К. запитує створення нового плану, то С. надає форму для плану лікування. К. заповнює форму необхідними даними.

7. С. зберігає оновлені плани лікування в БД і повідомляє К. про успішне виконання дії.

ВВ «Проаналізувати звіт»

1. К. виконує запит на отримання звіту.

2. С. завантажує необхідні дані з БД.

3. С. виконує обчислення агрегованих показників.

4. С. виконує побудову діаграм для агрегованих показників.

5. С. відображає загальну звітність і можливість вибору деталізованого типу звітності (бухгалтерія / склад / робота стоматологів).

6. К. обирає необхідний тип.

7. С. завантажує необхідні дані з БД.

8. С. виконує обчислення деталізованих показників.

9. С. виконує побудову діаграм для деталізованих показників.

10. С. відображає деталізовану звітність.

ВВ «Ознайомитись із способами забезпечення доступності»

1. К. виконує запит на ознайомлення із способами забезпечення доступності.

2. С. надає доступ К. до списку способів.

3. К. обирає необхідний спосіб.

4. С. відображає детальну інформацію про обраний спосіб.

# **4 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ**

## **4.1 Побудова діаграм взаємодії**

**4.1.1 Застосування архітектурних стилів.** При побудові діаграм взаємодії необхідно використати діаграму варіантів використання, а також сценарії всіх варіантів використання. Враховуючи те, що система не передбачає наявності розподілених обчислень, то архітектурні стилі, такі як «Broker» не застосовувались. Те саме стосується і шаблонів адаптивних систем («Microkernel»), адже задача розробки декількох програмних продуктів, побудованих на одній функціональності ядра, кожен з яких використовує схожі програмні інтерфейси не була поставлена. У відповідності до специфіки роботи ПЗ, будемо застосовувати шаблон «MVC» для розробки гнучкого людино-машинного інтерфейсу [5], а також архітектурний стиль «Layers», котрий буде представлений в неявному вигляді, але при цьому буде виконувати задачу структуризації системи. В цій структурі будуть наявні наступні рівні: інтерфейс користувача, бізнес-логіка, сховище даних. Інтерфейс буде приймати дані від користувача і відображати результати виконання обробки дій користувача, блок логіки буде реалізовувати виконання операцій над даними, а сховище буде зберігати дані у вигляді колекцій.

**4.1.2 Застосування тактик.** Виконавши аналіз нефункціональних вимог до системи, я вирішив використати наступні тактики для реалізації вимог: «Контрольна точка і відкат» – дозволить забезпечити відновлення працездатності серверу і гарантувати надання системою відповіді на усі запити користувача [5]; «Активна надмірність» – для відновлення даних за прийнятну кількість часу у випадку збою бази даних; «Авторизація» – для забезпечення захищеності даних. Результат застосування архітектурних стилів і тактик відображено на рис. 4.1-4.13.

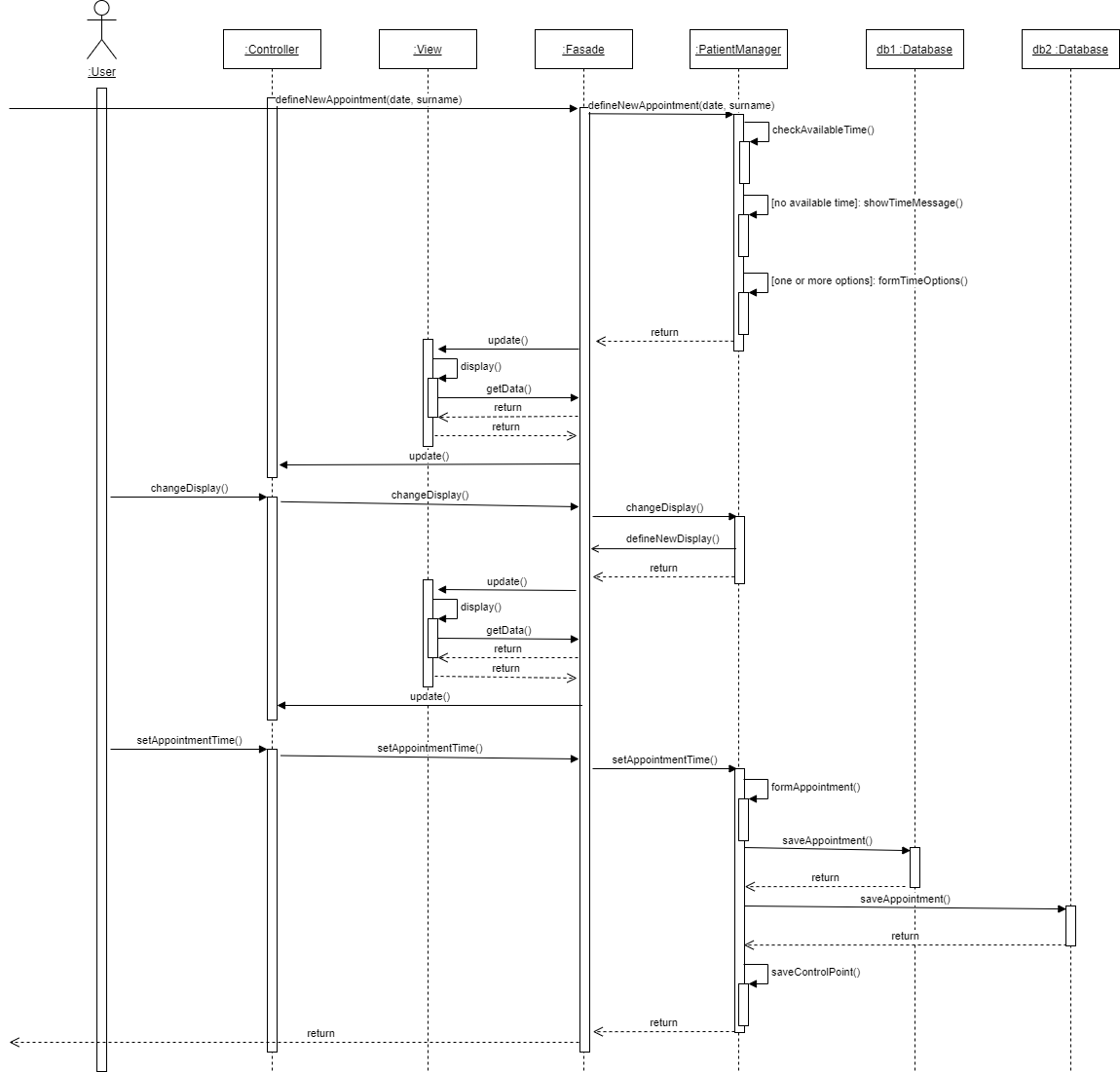


Рисунок 4.1 – Діаграма взаємодії для ВВ «Записатись на прийом»

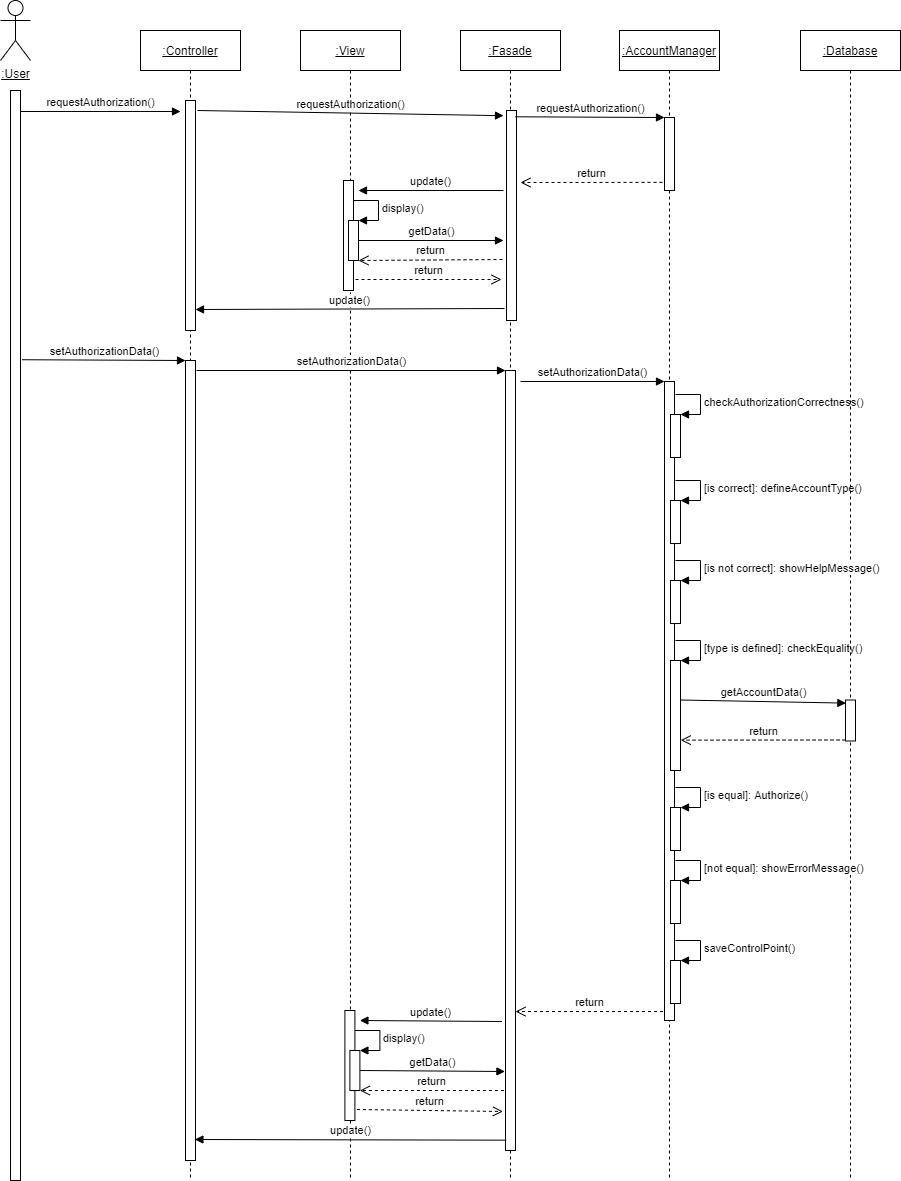


Рисунок 4.2 – Діаграма взаємодії для ВВ «Авторизуватись»

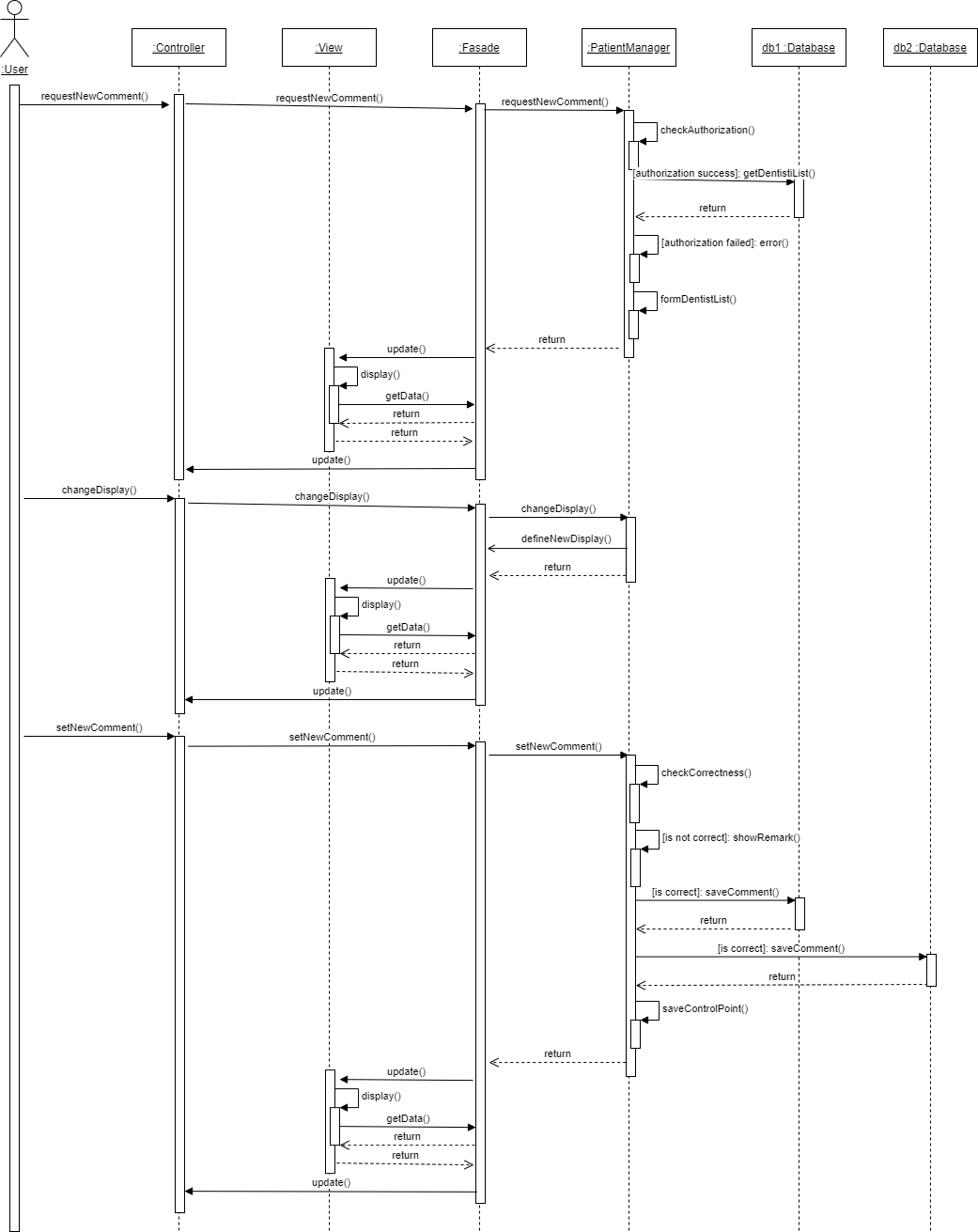


Рисунок 4.3 – Діаграма взаємодії для ВВ «Залишити відгук»

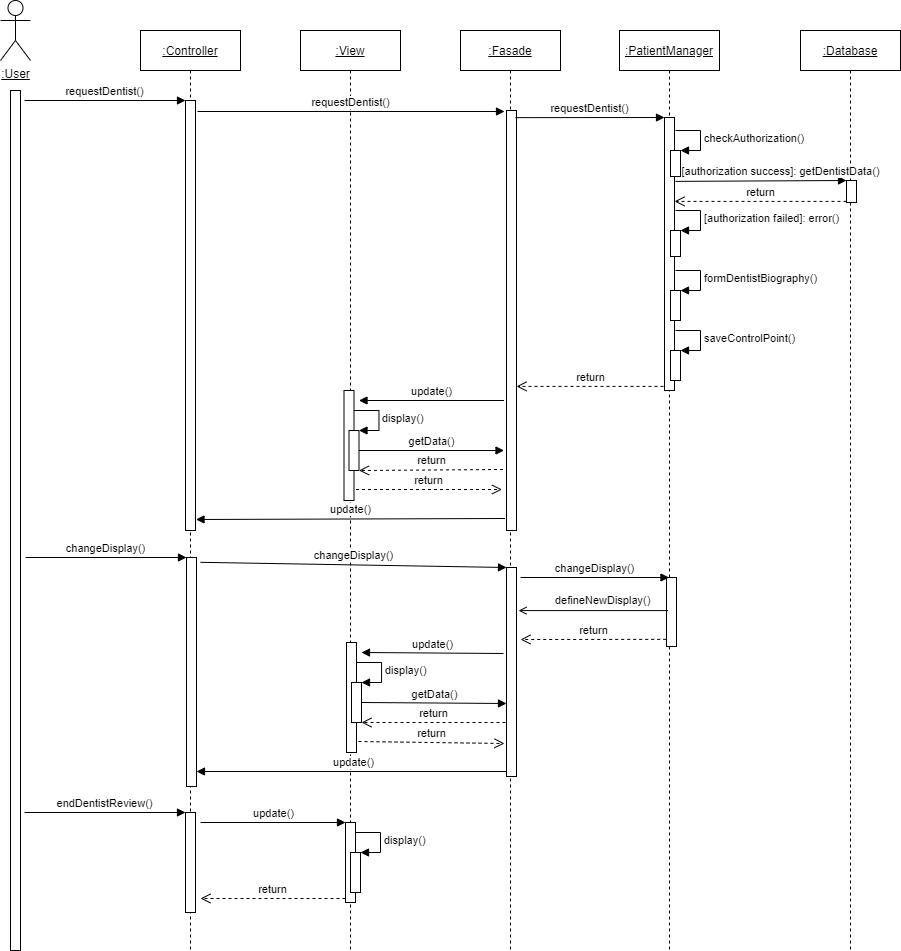


Рисунок 4.4 – Діаграма взаємодії для ВВ «Переглянути інформацію про спеціалістів»

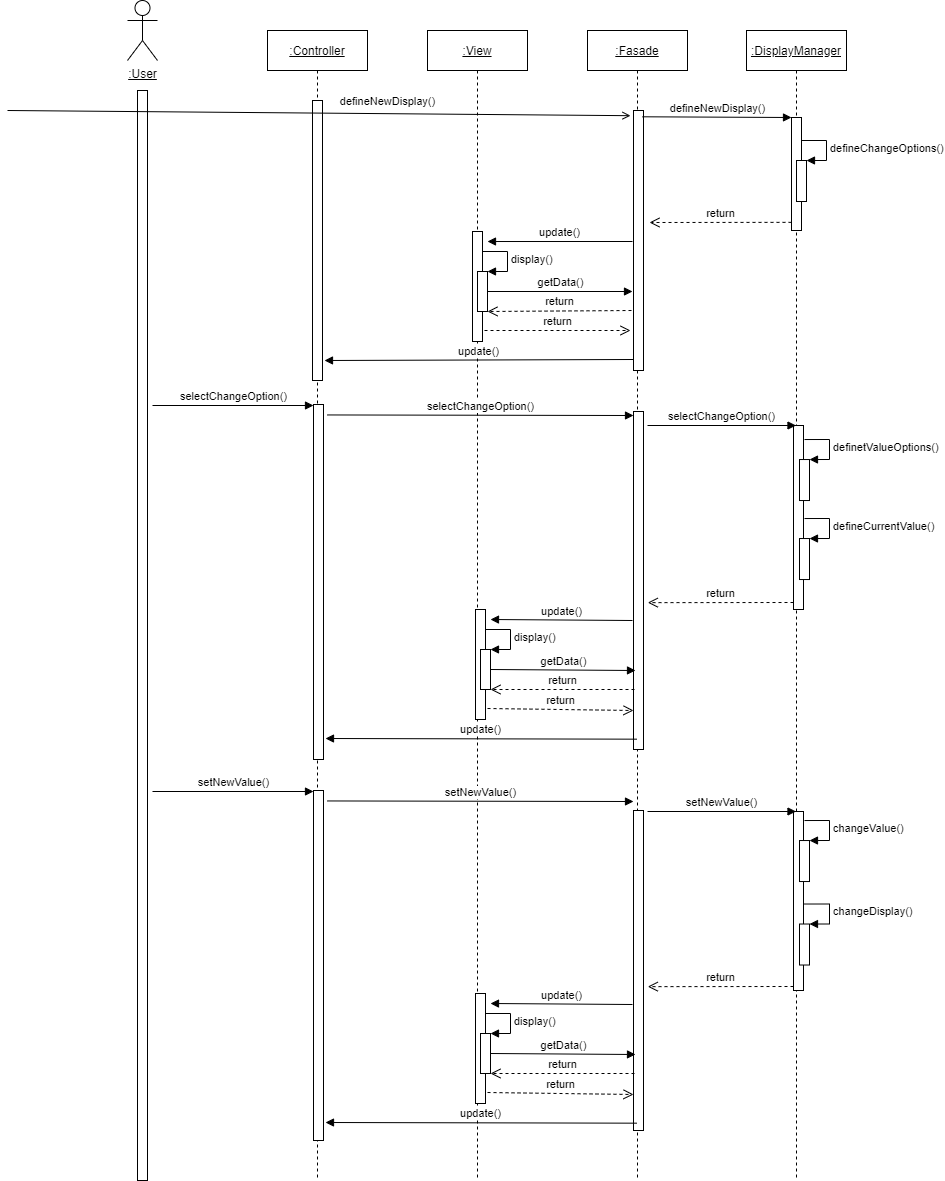


Рисунок 4.5 – Діаграма взаємодії для ВВ «Змінити тип відображення»

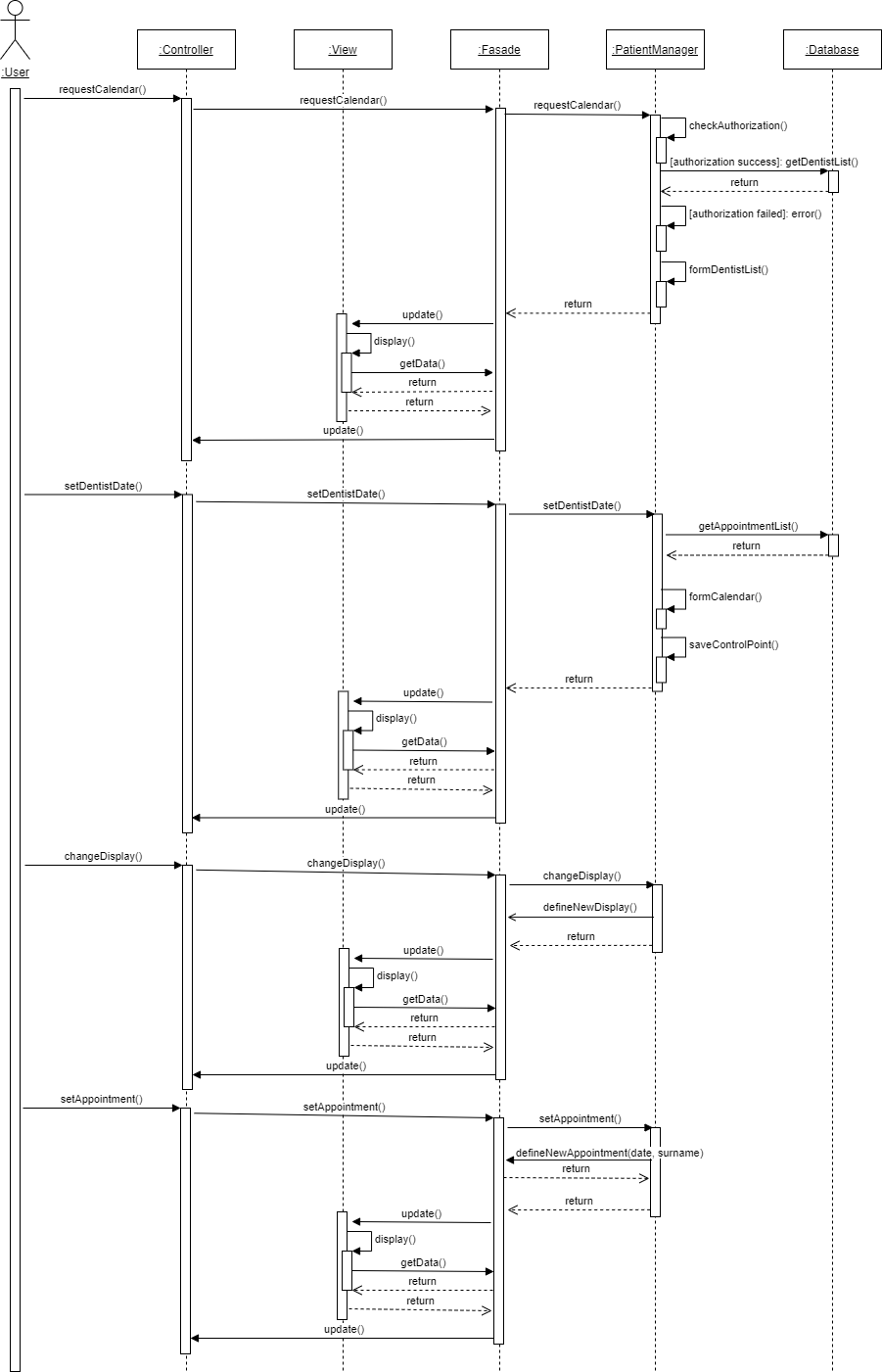


Рисунок 4.6 – Діаграма взаємодії для ВВ «Переглянути календар записів»

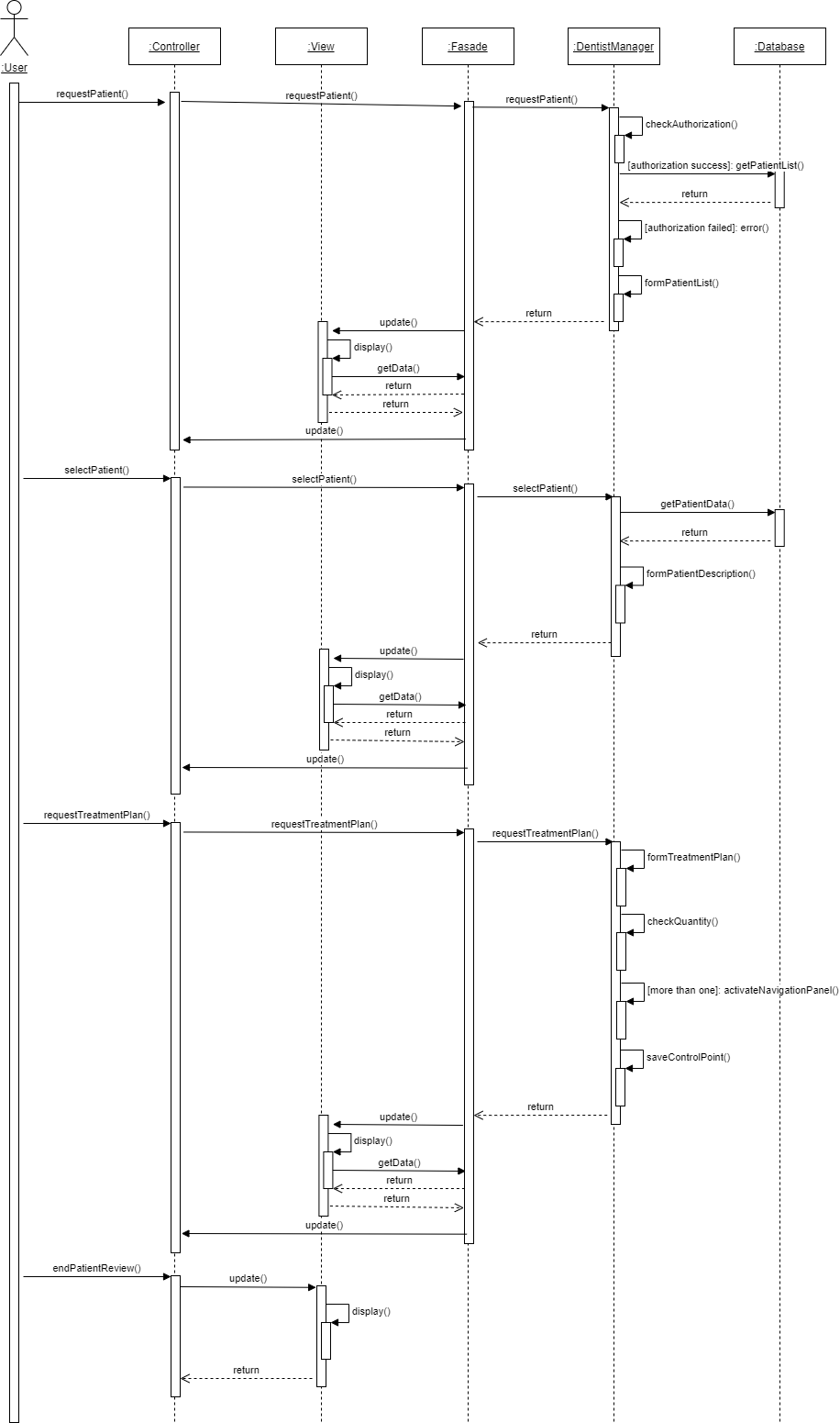


Рисунок 4.7 – Діаграма взаємодії для ВВ «Вивчити інформацію про пацієнта»

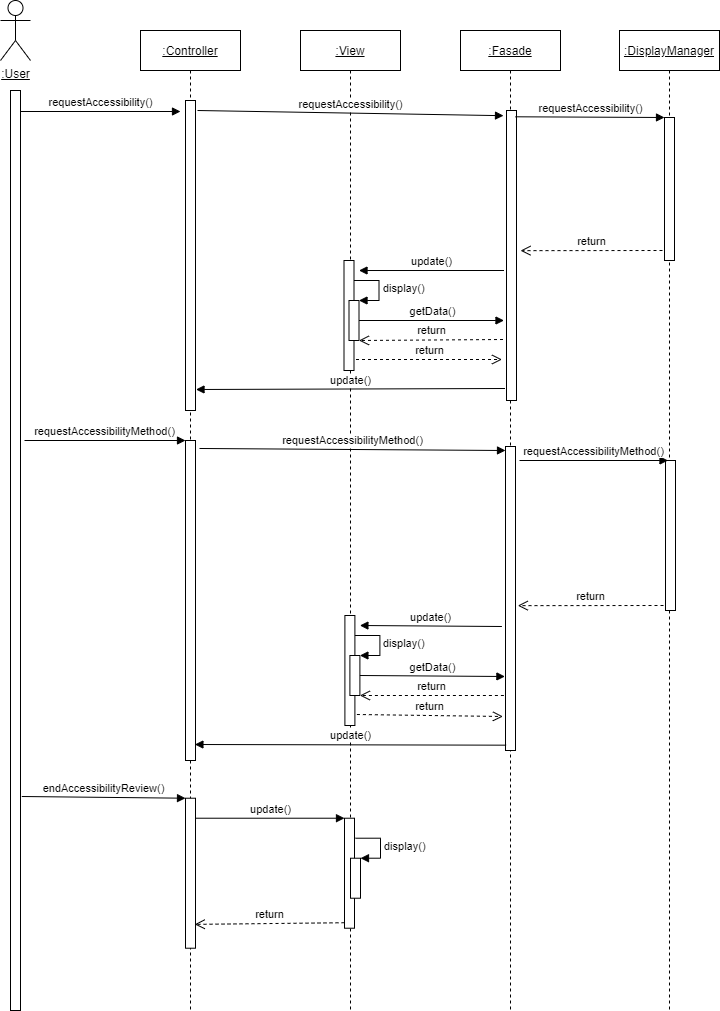


Рисунок 4.8 – Діаграма взаємодії для ВВ «Ознайомитись із способами забезпечення доступності»

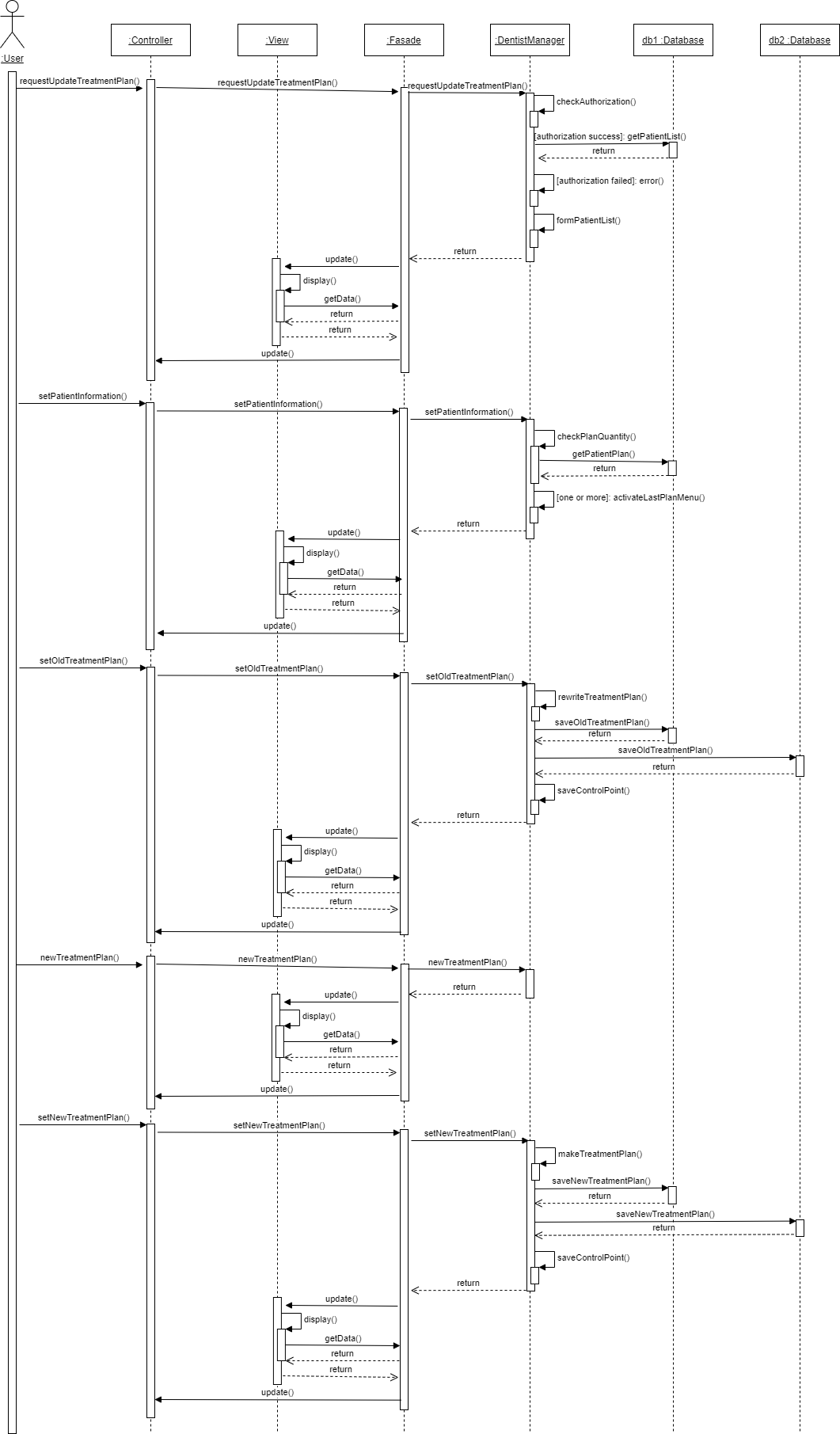


Рисунок 4.9 – Діаграма взаємодії для ВВ «Оновити плани лікування»

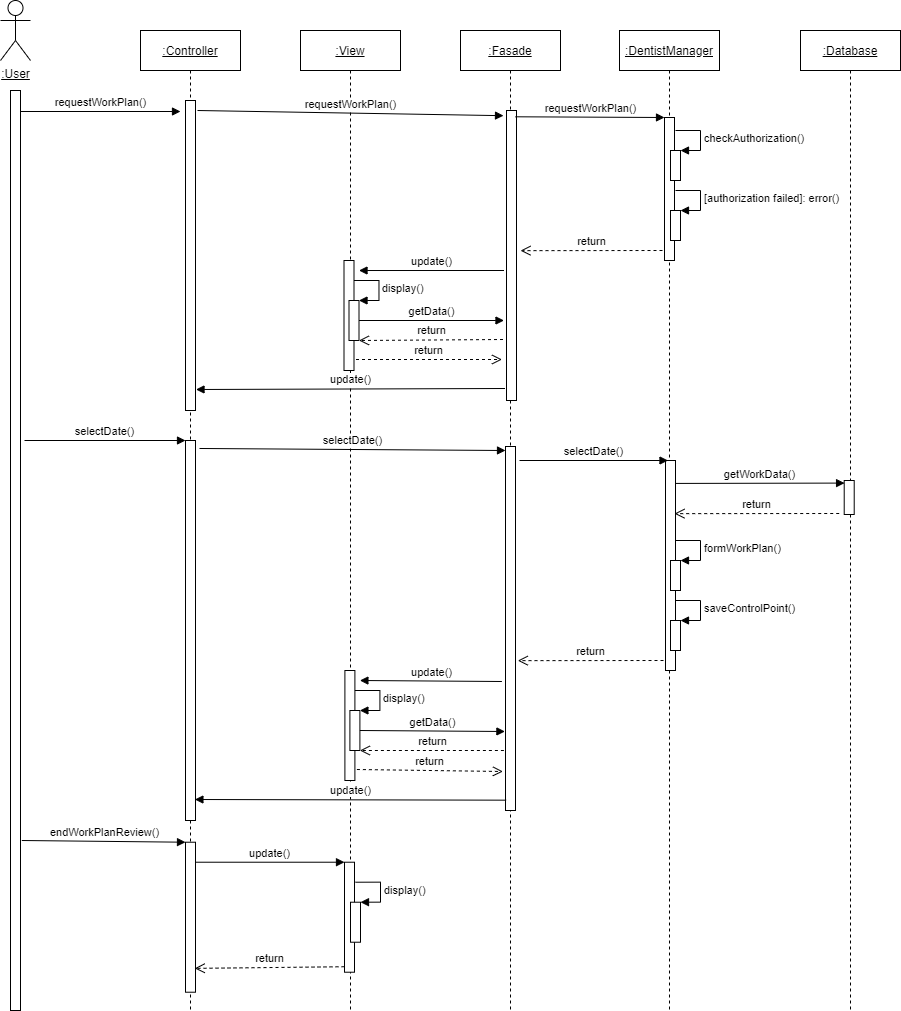


Рисунок 4.10 – Діаграма взаємодії для ВВ «Систематизувати заплановані візити»

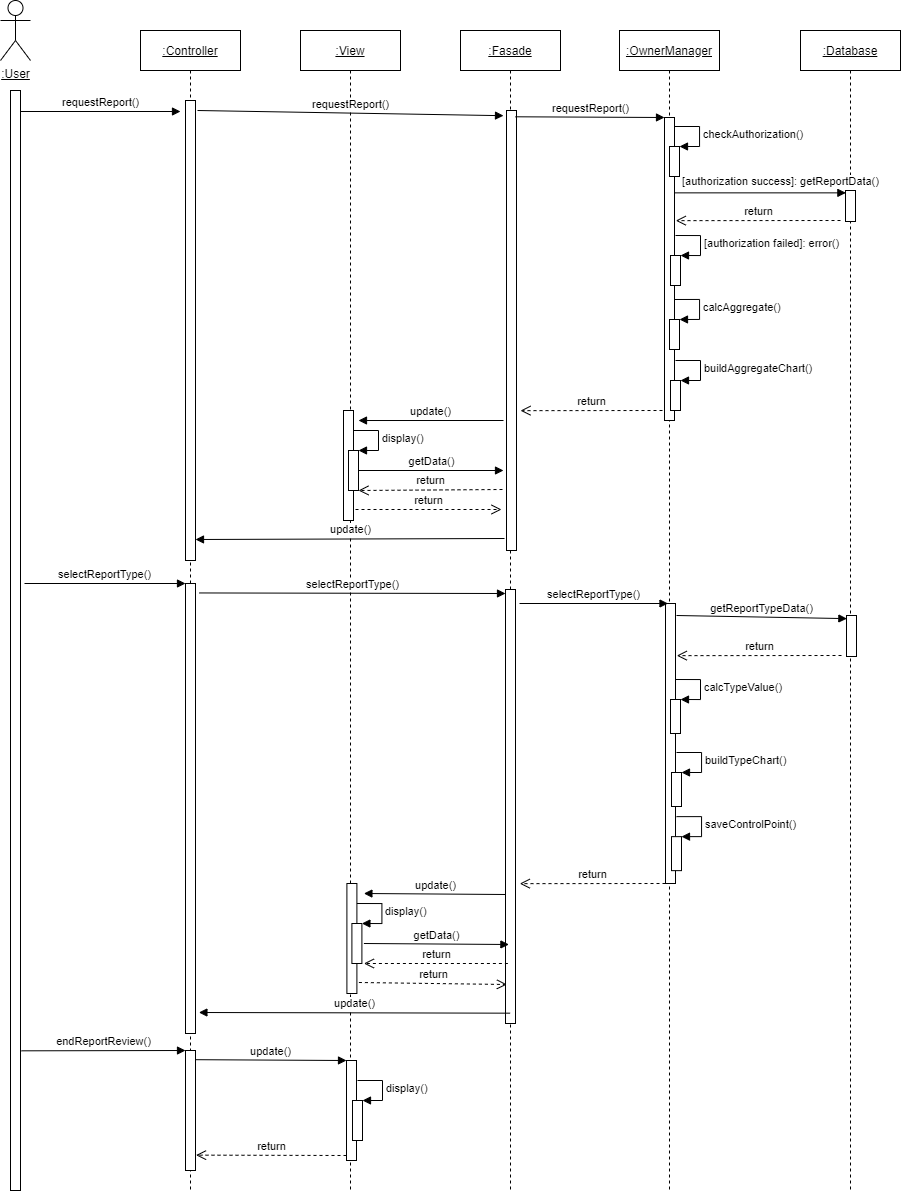


Рисунок 4.11 – Діаграма взаємодії для ВВ «Проаналізувати звіт»

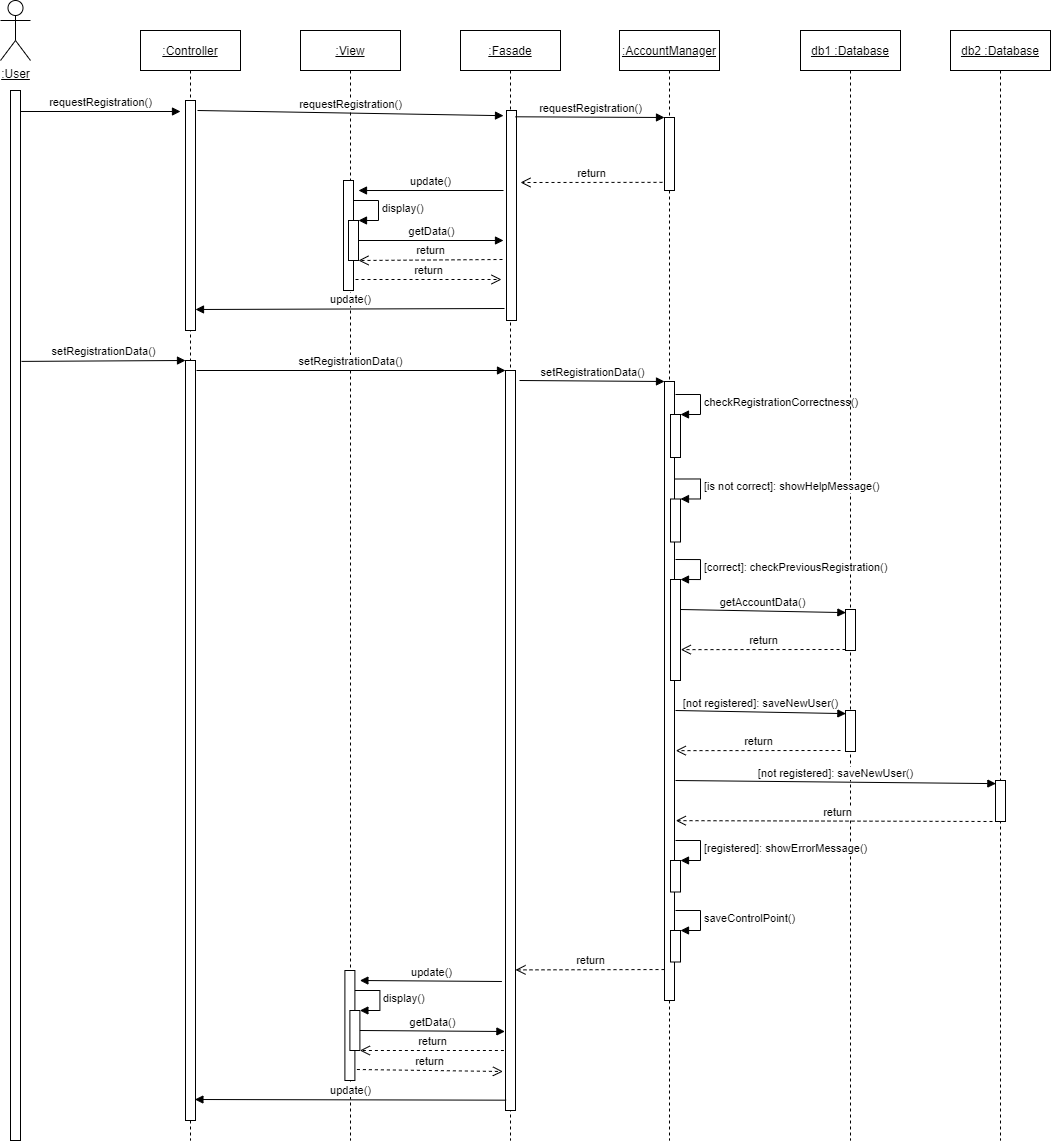


Рисунок 4.12 – Діаграма взаємодії для ВВ «Зареєструватись»

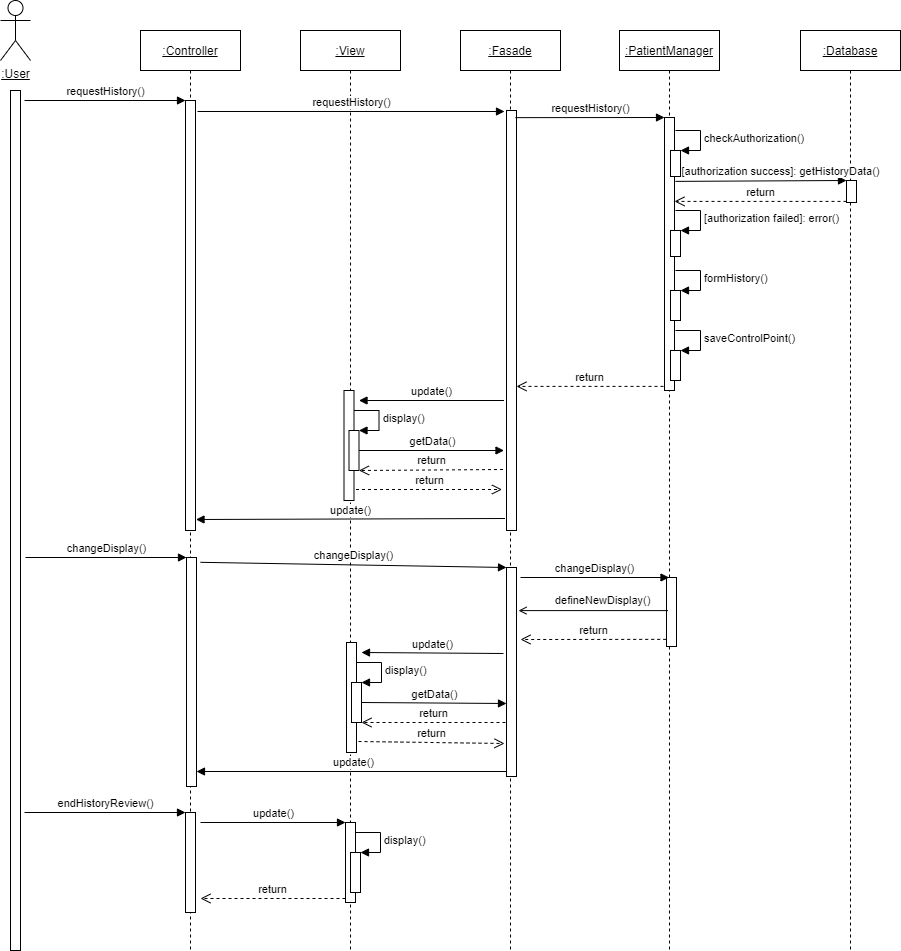


Рисунок 4.13 – Діаграма взаємодії для ВВ «Вивчити історію лікування»

Як можна побачити на рис. 4.1-4.13, реалізація моделі, наявність якої передбачено архітектурним стилем «MVC», досягається шляхом використання «менеджерів»: «PatientManager», «DentistManager», «OwnerManager», «AccountManager», «DisplayManager».

## **4.2 Побудова діаграми класів**

Діаграма класів використовується для візуалізації структури компонентів системи. На ній відображають класи, їх методи, а також зв’язки між класами. У випадку, коли всі діаграми взаємодії побудовано, можна переходити до етапу побудови діаграми класів. Для цього необхідно виконати певну послідовність кроків. Перш за все, всі сутності, які були виокремлені як компоненти системи на діаграмах взаємодії, ми виокремлюємо в якості класів. Далі, визначається приналежність методів до певних класів. Для цього необхідно відслідкувати виклик методів компоненту (визначити стрілки, які направлені на лінію життя компонента і не є внутрішніми методами, або ж стрілками типу «return»). Що стосується внутрішніх методів, то їх можна як відображати на діаграмі класів, так і ні. Для того, щоб не роботи діаграму класів надто громіздкою, внутрішні методи записувати не будемо. Третім кроком при побудові діаграми класів є визначення зв’язків між ними. Якщо компонент одного класу, на діаграмі взаємодії, викликає метод компоненту іншого класу, то на діаграмі класів необхідно вказати асоціативний зв’язок між даними класами. В результаті виконання даної послідовності кроків було отримано результат, відображений на рис. 4.14.

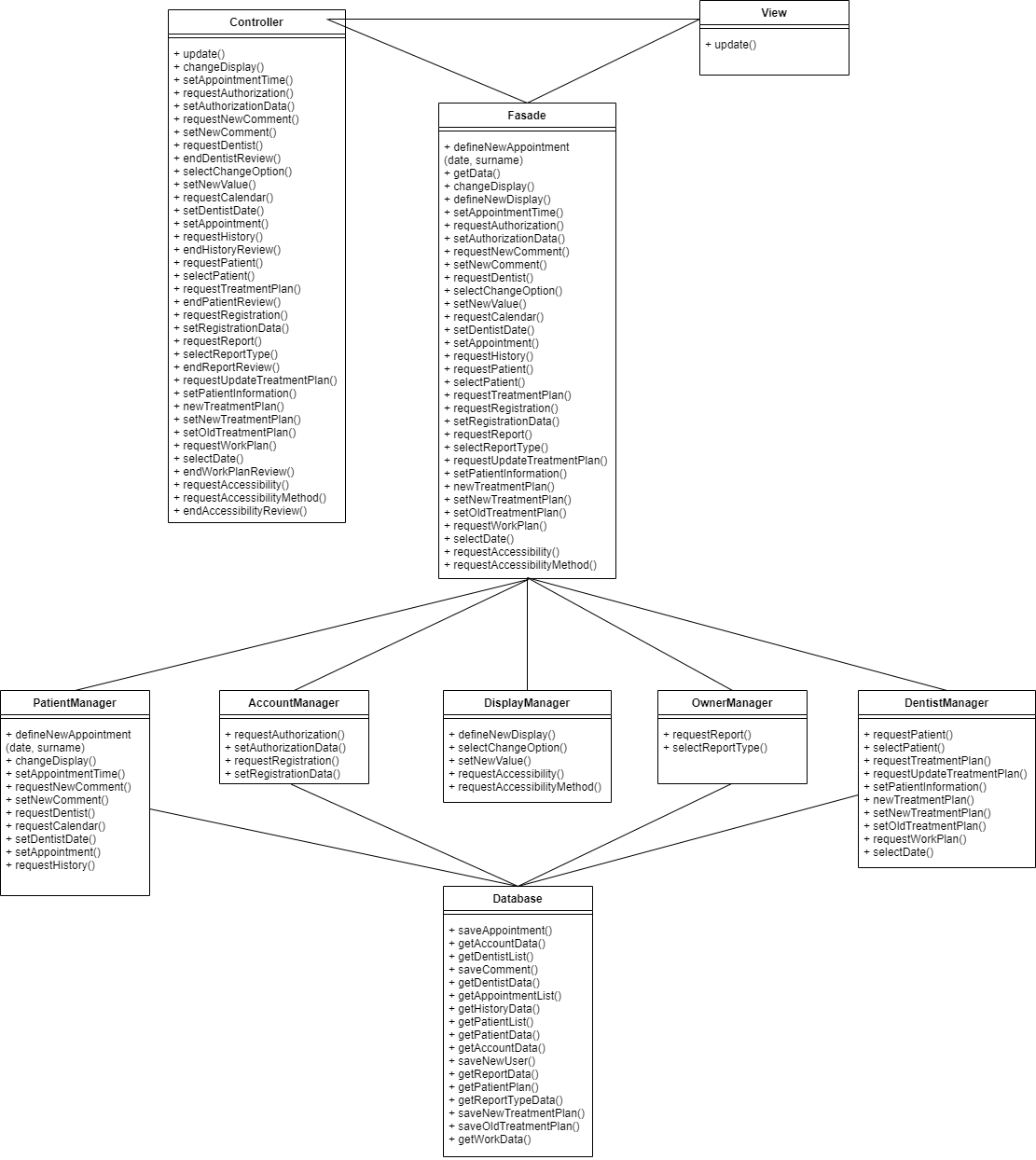


Рисунок 4.14 – Діаграма класів

Дана діаграма класів дозволяє перевірити відповідність структури системи архітектурному стилю «Layers». Класи «Controller» і «View» реалізують інтерфейс користувача. Клас «Database» реалізує сховище даних. Всі інші класи виконують ролі бізнес-логіки.

## **4.3 Проектування інтерфейсу користувача**

Проектування інтерфейсу користувача було виконано за допомогою програмного забезпечення «iPlotz». Даний інструментарій дозволяє побудувати дизайн інтерфейсу різного рівня складності: від скетчу до мокапу. Для проектування інтерфейсу користувача було виконано побудову вайфреймів. Цей рівень деталізації передбачає наявність чорно-білого плану інтерфейсу, на якому відображено основні елементи, їх розташування та основний функціонал, який виконується при взаємодії із елементами інтерфейсу. Побудову вайфрейму було виконано для 7 основних варіантів інтерфейсу користувача: переглянути інформацію про спеціалістів (рис. 4.15), переглянути календар записів (рис. 4.16), вивчити інформацію про пацієнта (рис. 4.17), зареєструватись (рис. 4.18), оновити плани лікування (рис. 4.19), авторизуватись (рис. 4.20), проаналізувати звіт (рис. 4.21).

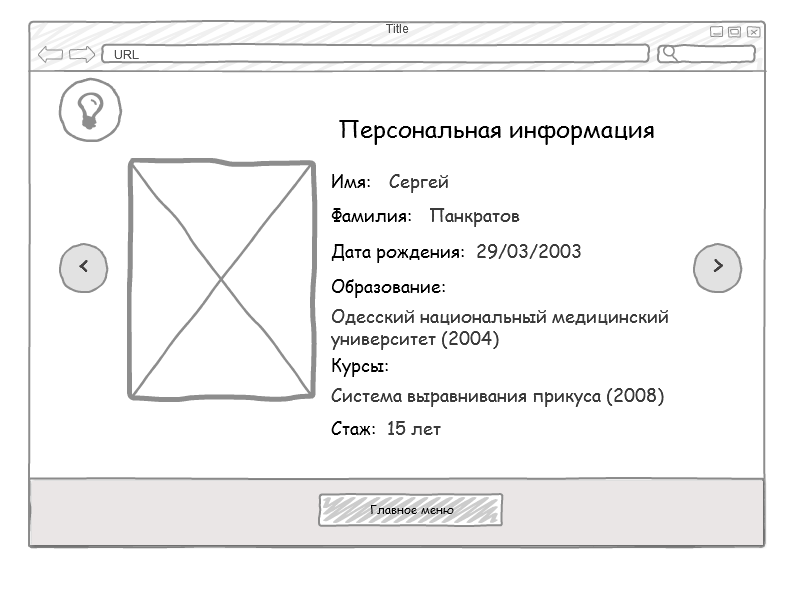


Рисунок 4.15 – Вайфрейм перегляду інформації про спеціалістів

Календар записів повинен у зручній формі надавати інформацію про наявність чи відсутність годин для запису (рис. 4.16).

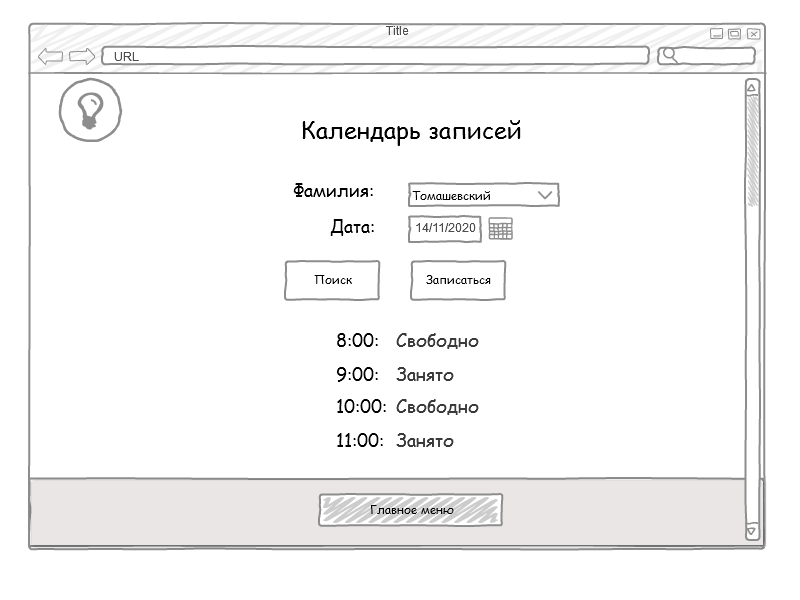


Рисунок 4.16 – Вайфрейм перегляду календаря записів

Інформація про пацієнтів має містити як дані про особливості пацієнта, так і плани лікування (рис. 4.17).

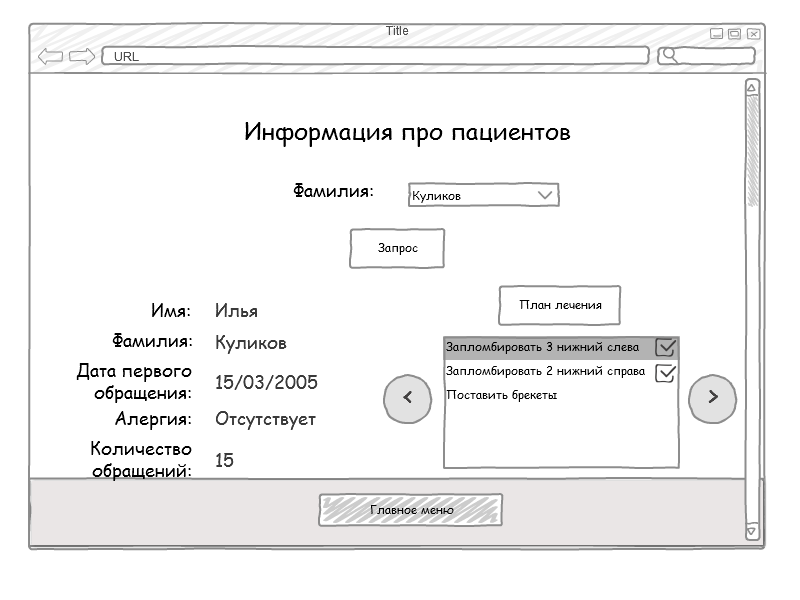


Рисунок 4.17 – Вайфрейм вивчення інформації про пацієнтів

Реєстрація передбачає заповнення форми, яка містить такі поля, як «Ім’я», «Прізвище», «Дата народження» та інші (рис. 4.18).

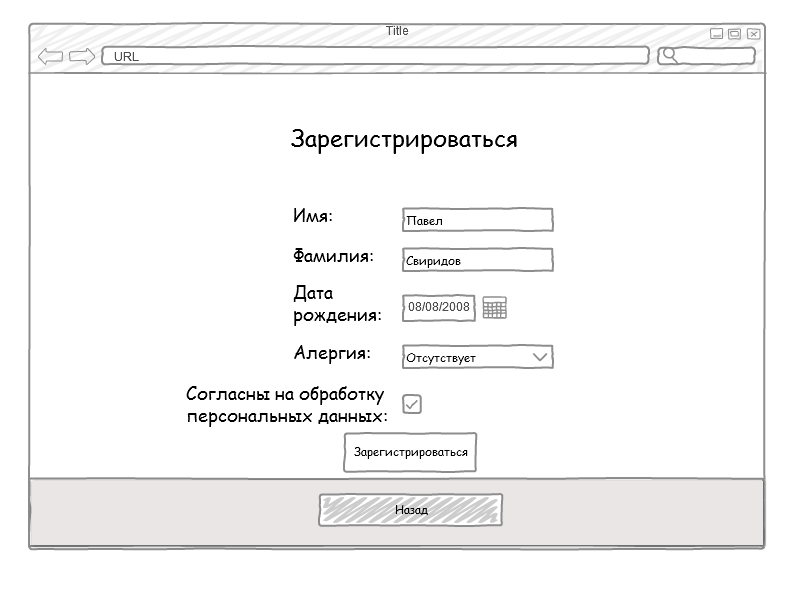


Рисунок 4.18 – Вайфрейм реєстрації

Оновлення планів лікування може стосуватись як редагування вже створених планів, так і додавання нових (рис. 4.19).

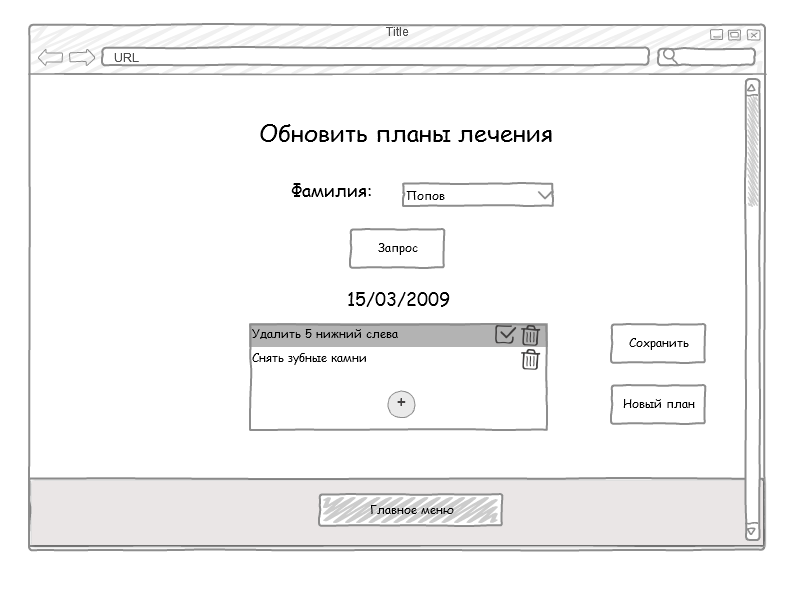


Рисунок 4.19 – Вайфрейм оновлення планів лікування

Для авторизації (рис. 4.20) необхідно використати дані, зазначені при проходжені реєстрації.

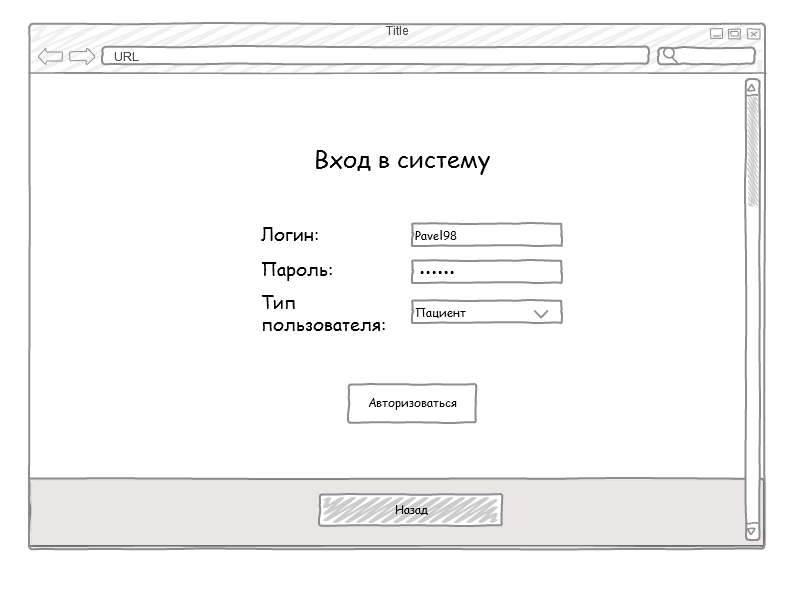


Рисунок 4.20 – Вайфрейм авторизації

Звіт про діяльність стоматологів демонструє хто зі співробітників працює більше за інших (рис. 4.21).

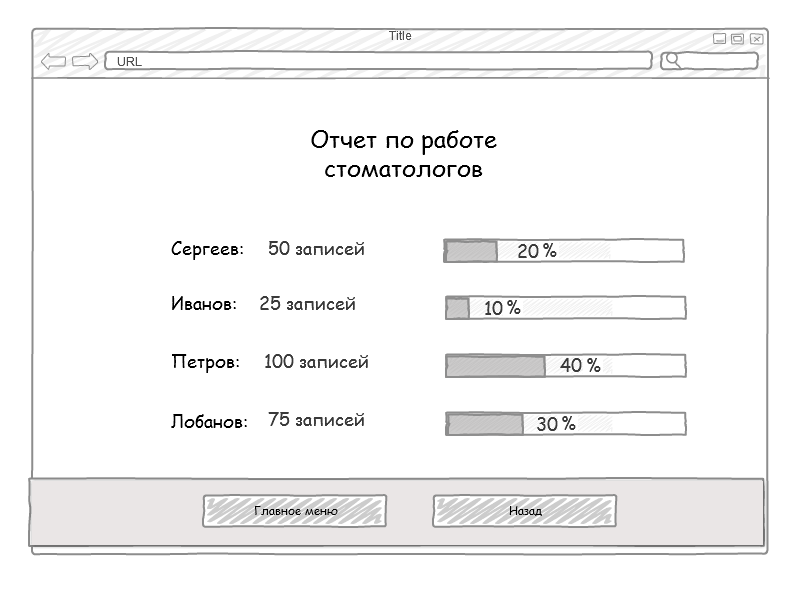


Рисунок 4.21 – Вайфрейм звітності

Необхідно відзначити, що вайфрейм реєстрації (рис. 4.18), з причин обмеженої площі, не включає в себе всі необхідні для заповнення поля. Це стосується логіну, електронної пошти та паролю. Однак їх присутність є необхідною, адже ці дані потім використовуються при авторизації.

## **4.4 Проектування структури колекцій**

Основними структурними компонентами бази даних, з якими відбувається взаємодія шляхом виконання запитів певного типу, є колекції. Проаналізувавши діаграми взаємодії для кожного з варіантів використання, вдалось виокремити 9 колекцій: відгук (рис. 4.22), стоматолог (рис. 4.23), пункт плану лікування (рис. 4.24), тип робіт (рис. 4.25), власник клініки (рис. 4.26), пацієнт (рис. 4.27), план лікування (рис. 4.28), запланований візит (рис. 4.29), склад (рис. 4.30).

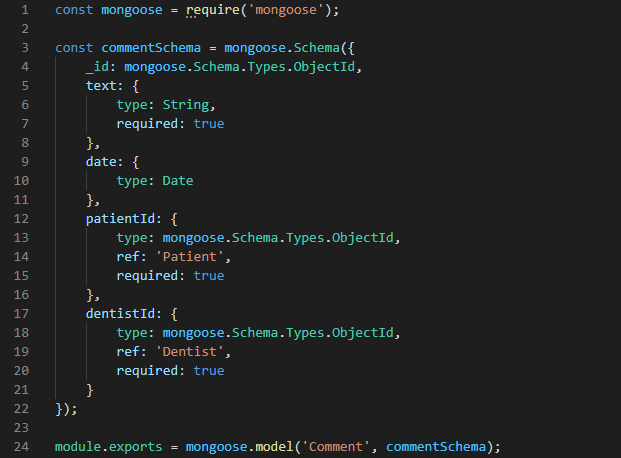
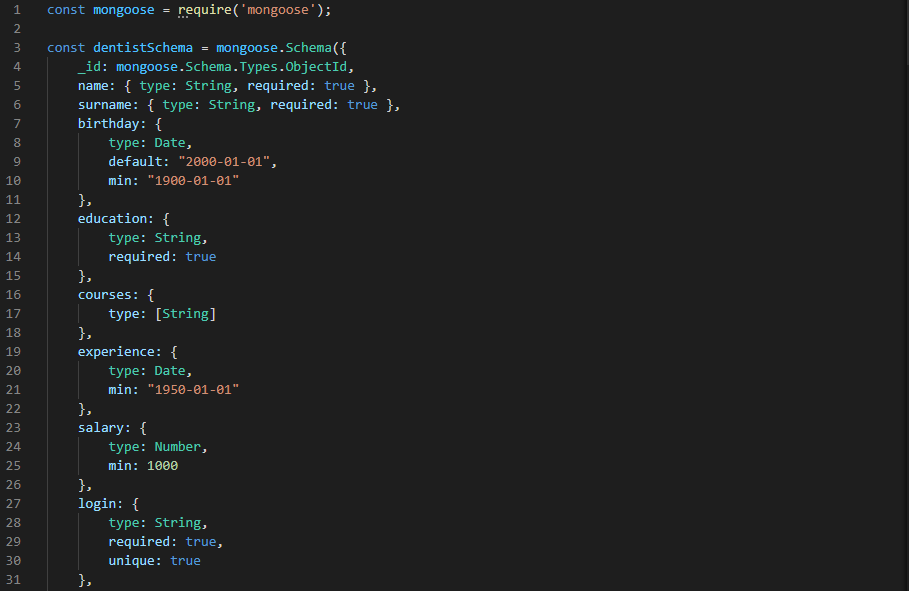


Рисунок 4.22 – Колекція для відгуків

Колекція для стоматологів відрізняється тим, що необхідно передбачати роботу із зображеннями (рис. 4.23).



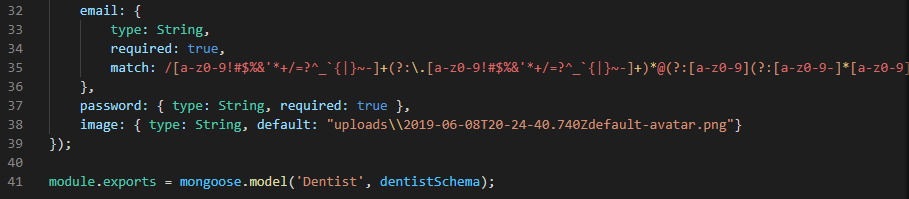


Рисунок 4.23 – Колекція для стоматологів

Для пункту плану лікування (рис. 4.24) необхідно зазначати ідентифікатор плану та ідентифікатор типу робіт.

****

Рисунок 4.24 – Колекція для пункту плану лікування

Колекція для типу робіт (рис. 4.25) містить наступну інформацію: тип робіт та вартість її виконання.

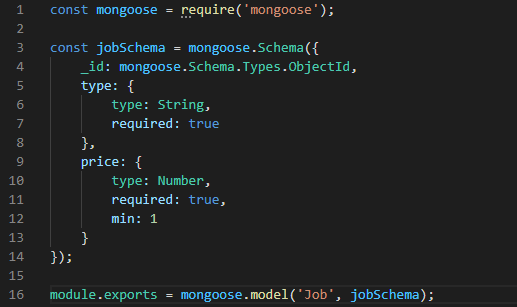


Рисунок 4.25 – Колекція для пункту типу робіт

Дані про власника клініки не вказуються у жодному з варіантів використання, а тому відповідна колекція має більщ просту структуру (рис. 4.26).



Рисунок 4.26 – Колекція для власника клініки

Для запобігання винекнення ускладнень, пацієнти мають відповідально заповнювати поле «Алергія» (рис. 4.27).

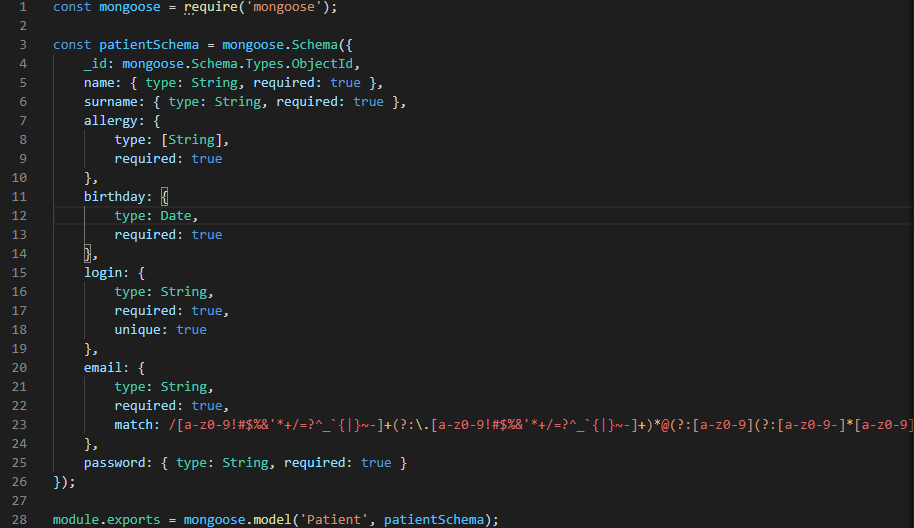


Рисунок 4.27 – Колекція для пацієнта

План лікування (рис. 4.28) складається із певної кількості пунктів, для кожного з яких визначено тип робіт.

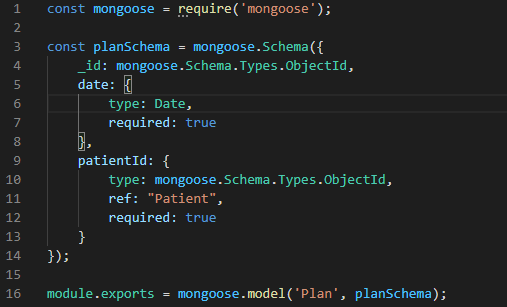


Рисунок 4.28 – Колекція для плану лікування



Рисунок 4.29 – Колекція для запланованого візиту

Колекція для запланованих візитів (рис. 4.29) накладає обмеження на час запису, адже робочий день стоматологів обмежений.



Рисунок 4.30 – Колекція для складу

Для кожного з типів користувачів (стоматолог, власник клініки, пацієнт) передбачена власна колекція. Це є наслідком відсутності єдиної структури, котра могла б забезпечувати працездатність всіх запитів для користувачів. Дана ситуація дозволяє пришвидшити процес авторизації, адже сегменти користувачів розділені по різним колекціям і для перевірки наявності зареєстрованого користувача певного типу буде витрачено менше часу.

# **5 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ**

## **5.1 Набір інструментальних засобів розробки**

При розробці програмного забезпечення використовувались наступні інструментальні засоби:

* Редактор коду «Visual Studio Code» («VS Code»).
* Бібліотеки «React» та «Redux».
* Платформа «Node.js».
* Фреймворк «Express JS».
* СУБД «MongoDB».

Опис зазначених інструментів:

Visual Studio Code позиціонує себе як «легкий» редактор коду. Застосовується переважно для розробки веб-застосунків та застосунків у сфері хмарних технологій. Серед особливостей даного редактору можна виділити наступні: вбудований налагоджувач, інструменти для роботи з Git, підсвічування синтаксису, навігація по коду, засоби для рефакторингу коду. Для полегшення роботи веб-розробників із файлами з розширенням «.css», «.html», «.js», а також sass-препроцесорами наявна технологія InteliSense, котра дозволяє автоматично дописувати назви функцій при наборі розробником її перших літер [19].

Visual Studio Code передбачає можливість інсталяції плагінів, доступних в центральному репозиторії. Серед таких плагінів можна виділити наступні дві групи: розширення можливостей редактору коду, підтримка мов програмування. Особливістю VS Code є можливість створення власних розширень, котрі дозволяють додавати підтримку нових мов програмування, тем та налагоджувачів, передбачають можливість виконання статистичного аналізу коду, підключати додаткові сервіси.

React використовується в якості заміни для звичайних сайтів, взаємодія з якими виконується шляхом переходу від однієї web-сторінки до іншої. В даному випадку пропонується рішення, яке дозволяє виділяти блоки HTML-коду в якості компонент для їх повторного використання, а зміна інтерфейсу користувача буде виконуватись в рамках однієї сторінки. Це дозволяє налагодити коректну роботу із сервером, який не перебачає роботу із сесіями [20].

Для управління станами системи передбачається використання бібліотеки Redux. Зазвичай, дану бібліотеку використовують разом із React або Angular. Саме Redux дозволяє організувати взаємодію клієнтської частини програми із серверною. При цьому на сервері розташовано RestFul API. Взаємодія реалізується наступним чином: знаючи документацію API розробники клієнтської частини виконують запит (вказуючи адресу, тип запиту та додаткові опції). Сервер повертає JSON-об’єкт, властивості котрого зберігаються у певній змінній. Після цього виконується оновлення інтерфейсу компоненту.

Для створення серверу використовуються можливості платформи Node.js та фреймворку Express.js. Node.js дозволяє використовувати мову програмування JavaScript поза браузером. Таким чином, для створення серверу, який буде прослуховувати певний порт, необхідно використати лише 3 команди. Перш за все, необхідно підключити модуль http. Для цього використовується команда require. Після цього, розробнику доступні усі методи модулю http, серед яких createServer(). Після того, як сервер створено, необхідно встановити прослуховування порту. Для цього слід використати метод listen() і вказати номер порту. Повний код для створення серверу наведено на рис. 5.1



Рисунок 5.1 – Створення серверу

Після того, як встановлено прослуховування порту, слід налаштувати сервер за допомогою фреймворку Express.js. Він дозволяє встановлювати routes (методи обробки запитів в залежності від вказаної адреси), вирішувати проблеми крос доменних запитів, зміни доступу до певних каталогів (використовуються при збереженні зображень користувача і запитах на отримання даної інформації за необхідності відображення інформації про користувача), а також використання інших фреймворків для журналювання параметрів запиту (morgan) і роботи із контентом запиту у вигляді json-об’єкту (body-parser) [21].

В якості системи управління базами даних використовується MongoDB. Це документоорієнтована СУБД, котра класифікується як NoSQL. Колекції, які зберігаються у базі даних мають JSON-формат, роботу із яким легко організувати за допомогою мови програмування JavaScript, використовуючи при цьому модуль Mongoose. Mongoose дозволяє під’єднатись до серверу MongoDB, виконати запит і отримати результат виконання запиту. Окремою частиною роботи із MongoDB є створення схеми. Для того, щоб додати схему, в рамках серверної частини необхідно створити модель даної колекції. При цьому вказуються тип даних, обмеження на значення, якщо вони потрібні, необхідність передачі певного параметру і його значення за замовчуванням.

У відповідності до розширеної моделі доступності web-інтерфейсів, слід розглянути додавання атрибутів «alt» та «title» до зображень. На рис. 5.2 відображено елемент області інтерфейсу користувача, яка дозволяє переходити від інформації про одного спеціаліста до інформації про іншого спеціаліста, шляхом натискання кнопок із зображенням напрямку переміщення елементів. Складовою частиною даного блоку є фотографія стоматолога. Для більш зручної роботи із фотографією, до зображення було додано атрибут «alt» зі значенням «Зображення першого стоматолога» та атрибут «title» зі значенням «Перший стоматолог». Вміст першого атрибуту буде відображено у тому випадку, якщо, з певних причин, не буде виведено зображення. Значення другого атрибуту буде відображатись при наведенні курсору на даний елемент.

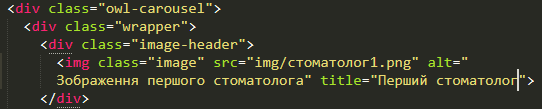


Рисунок 5.2 – Додавання атрибутів «alt» та «title»

Для коректної роботи із сторінкою людей, які мають проблеми із зором або порушення моторики, слід використовувати селектор «focus». Він дозволяє відзначати де саме розташовується користувач при переміщенні по сторінці. Прикладом цього є робота з формами. Якщо користувач перейшов до елементу «input», це можна відзначити зміною кольору даного елементу (рис. 5.3). Таким чином, шляхом натисканням клавіші «Tab» на клавіатурі, можна виконувати переміщення по елементам форми. Даний принцип діє не тільки при реєстрації, авторизації та зміні типу відображення, а й при виконанні інших дій, оскільки селектор «focus» можна встановити не тільки для «input», а й для посилань, кнопок та пунктів панелі навігації.



Рисунок 5.3 – Робота із фокусом

Окремо слід виділити роботу із ARIA-атрибутами, як за допомогою HTML, так і за допомогою JavaScript. Цю групу атрибутів слід використовувати у тих випадках, якщо елемент використовується як стилізована заміна елементу іншого типу, але інформація про це не вказується. Тоді слід зазначати роль атрибуту за допомогою «role». За допомогою «aria-label» можна вказувати інформацію, яку буде прочитано за допомогою таких технологій, як «Screen Reader».

## **5.2 Тестування за допомогою Postman**

Необхідно виконати тестування програмних модулів, розроблених на етапі конструювання програмного забезпечення. Першим етапом тестування буде тестування методом «білого ящика». Даний метод передбачає знання коду програмних модулів, які підпадають під тестування. Для виконання тестування використовувався інструмент «Postman», який передбачає можливість формування запиту до програмного модуля і отримання відповіді. Тестування було виконано не для всіх модулів, а лише для декількох основних функцій.

Першою колекцією, яка підпадає тестуванню буде «Owner». Вона створена для реєстрації та авторизації власників стоматологічних клінік, отримання даних про них, а також видалення їхніх даних, якщо співпраця з даною клінікою завершується. Виконаємо реєстрацію нового власника. Необхідно вказати ім’я, прізвище, адресу електронної пошти, логін, пароль. Логін має бути унікальним. Виконаємо запит «Post» вказавши «localhost:3000/owners/signup» (рис. 5.4).

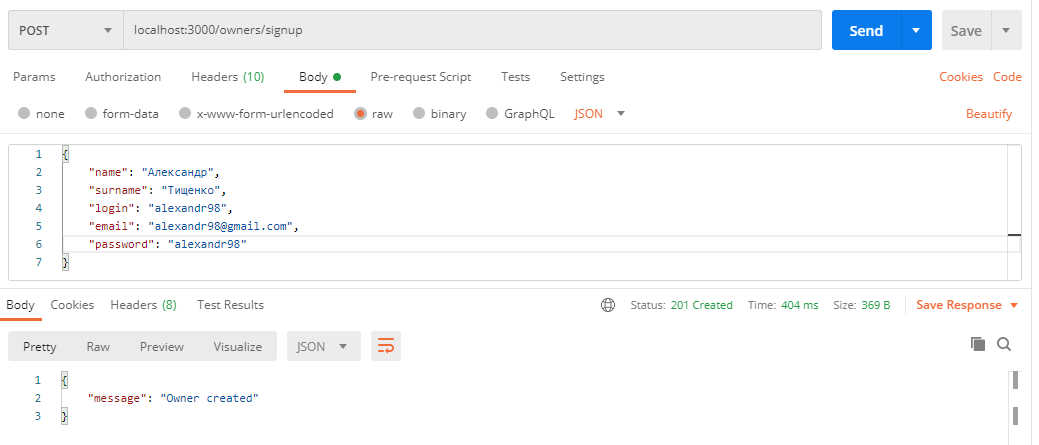


Рисунок 5.4 – Реєстрація власника

Далі, необхідно виконати авторизацію. Авторизація передбачає зазначення логіну та паролю (рис. 5.5).

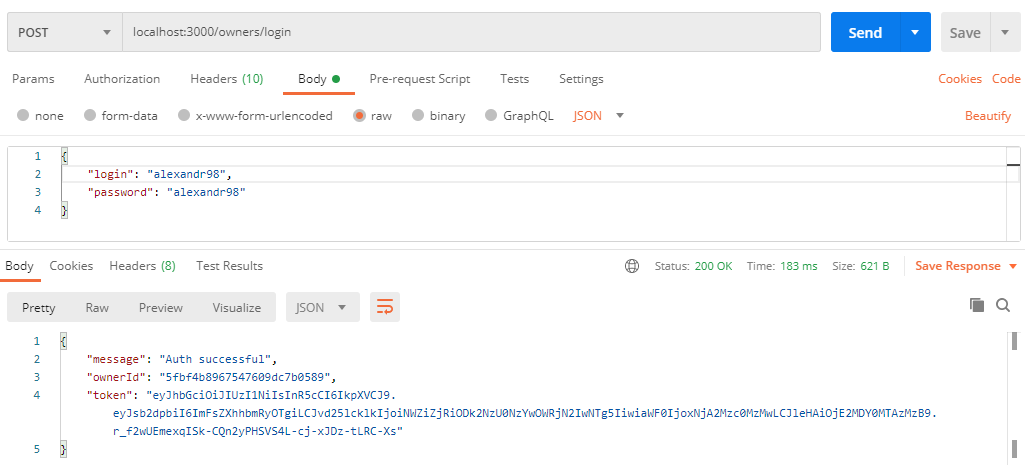


Рисунок 5.5 – Авторизація власника

Як можна побачити на рис. 5.5, у разі успішної авторизації, повертається повідомлення, яке вказує, що авторизація успішно виконана, ідентифікатор власника, токен. Токен знадобиться для виконання усіх наступних запитів. Тривалість дії токену – 10 годин. Передавати його на сторону серверу необхідно буде через headers.

Враховуючи те, що авторизація виконана успішно, можна перейти до колекції «Job» і спробувати усі доступні типи запитів: створення, оновлення, видалення, отримання роботи за її ідентифікатором, отримання усіх робіт.

Спочатку необхідно створити декілька робіт (рис. 5.6-5.8).

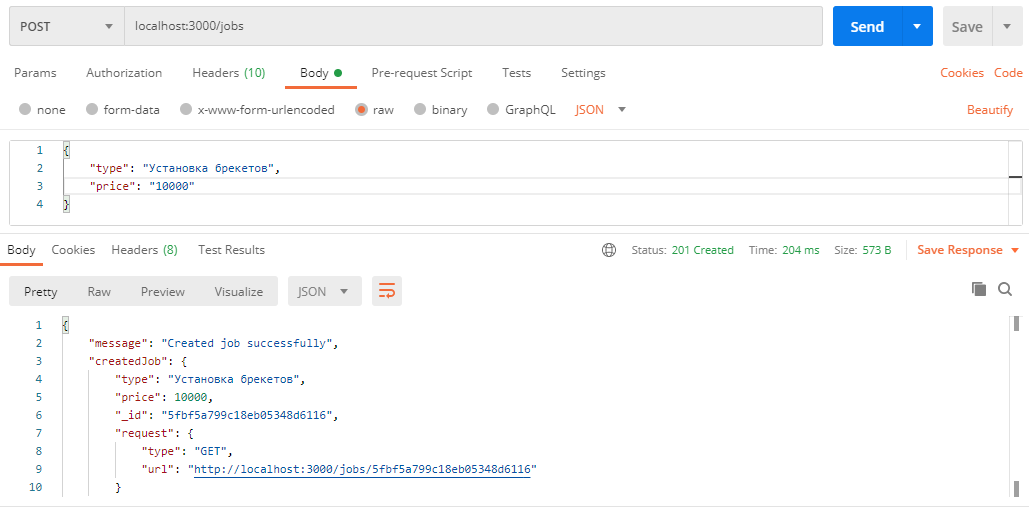


Рисунок 5.6 – Створення першої роботи

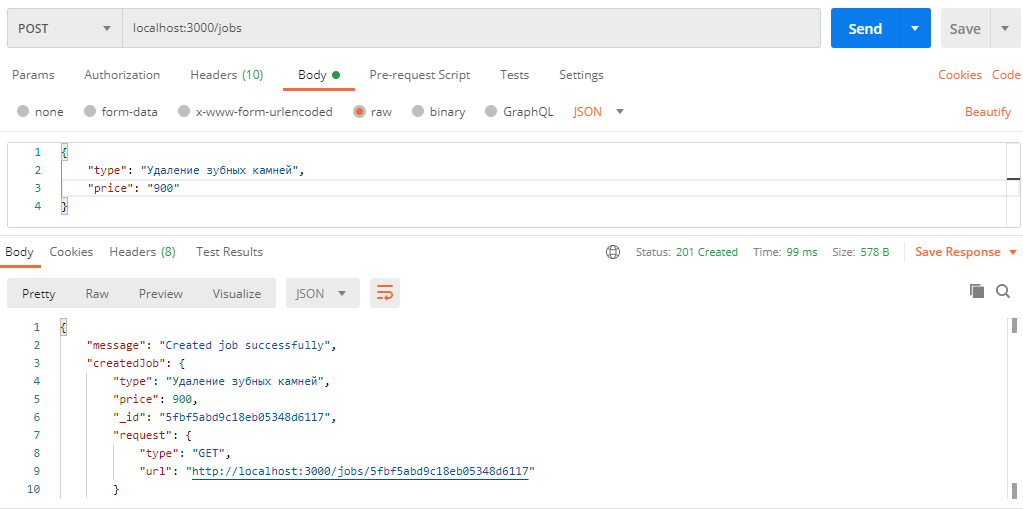


Рисунок 5.7 – Створення другої роботи

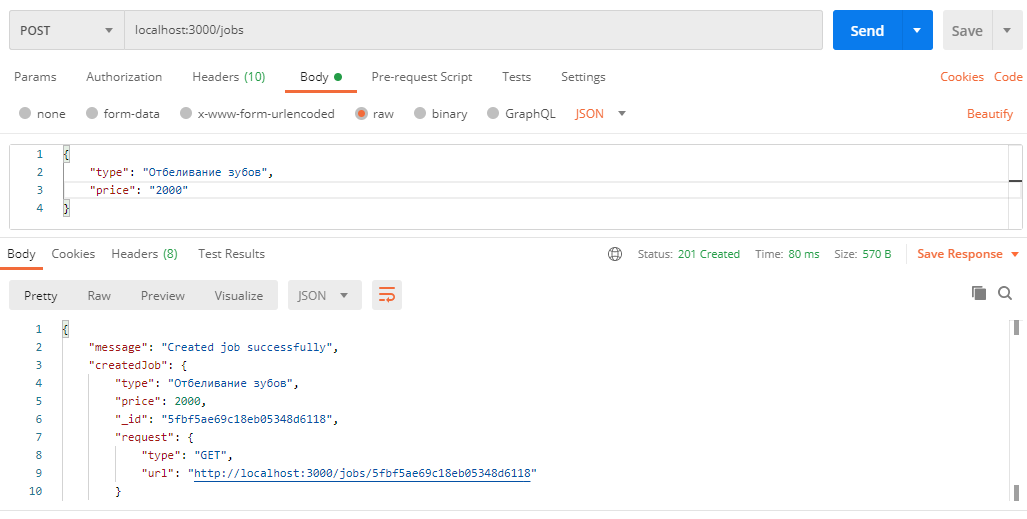


Рисунок 5.8 – Створення третьої роботи

Як бачимо, після виконання запиту, сервер повертає не лише відповідь зі статусом 201, в тілі якої міститься повідомлення про успішне створення нової роботи, а й допоміжну інформацію: поля, які були вказані клієнтською стороною, ідентифікатор створеної роботи, тип запиту та адреса, які необхідно вказати для отримання інформації про дану роботу. Результат виконання даної дії відображено на рис. 5.9.

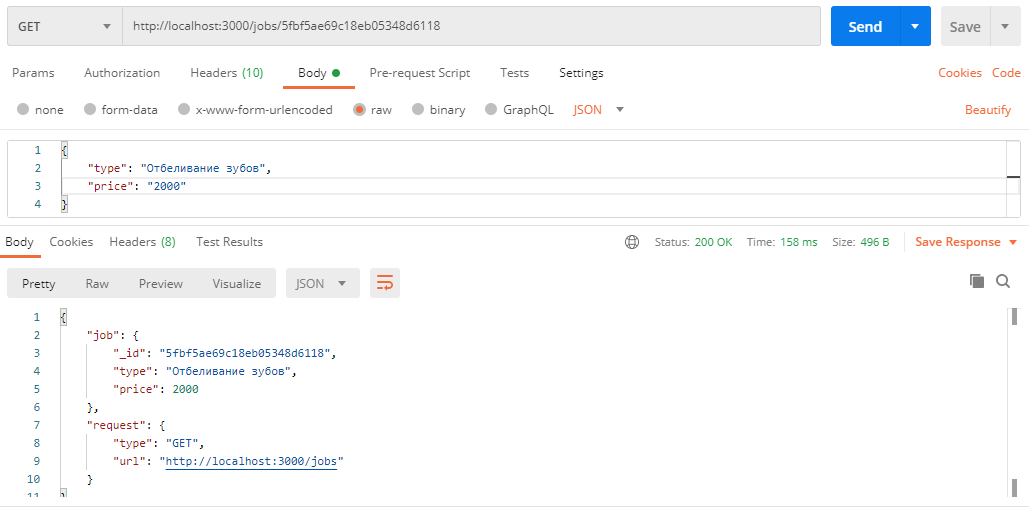


Рисунок 5.9 – Отримання інформації про роботу

Результат містить інформацію про запит, який необхідно виконати для того, щоб переглянути не лише дану роботу, а й всі інші (рис. 5.10).

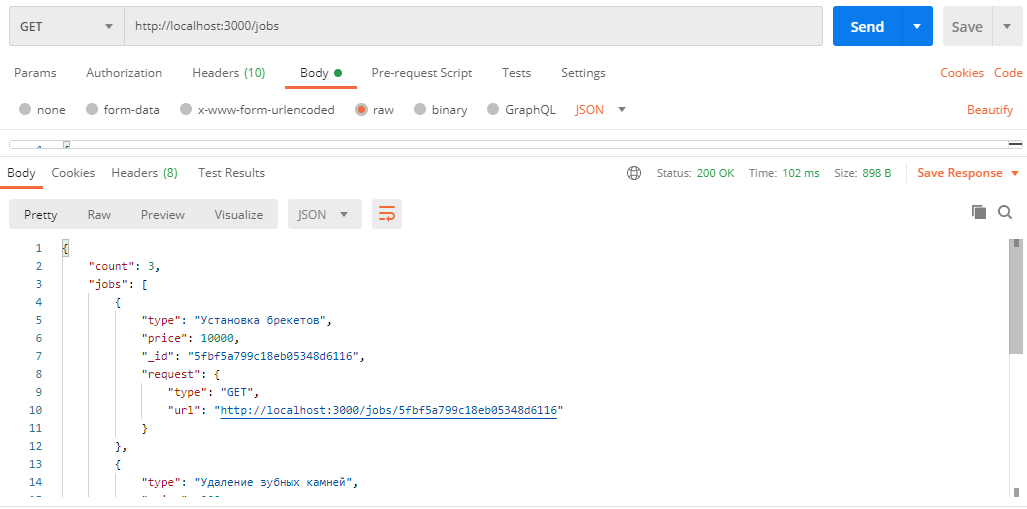


Рисунок 5.10 – Отримання інформації про всі типи робіт

Окрім детальної інформації про кожну роботу: її тип, вартість та ідентифікатор, вказано кількість елементів в масиві.

Тепер, необхідно перевірити запити на оновлення та видалення робіт. Виберемо поле «Ціна» для роботи з типом «Встановлення брекетів». Вартість послуги збільшилась і становить 20000. Виконаємо оновлення (рис. 5.11) і перевіром чи зберіглась нова інформація (рис. 5.12).

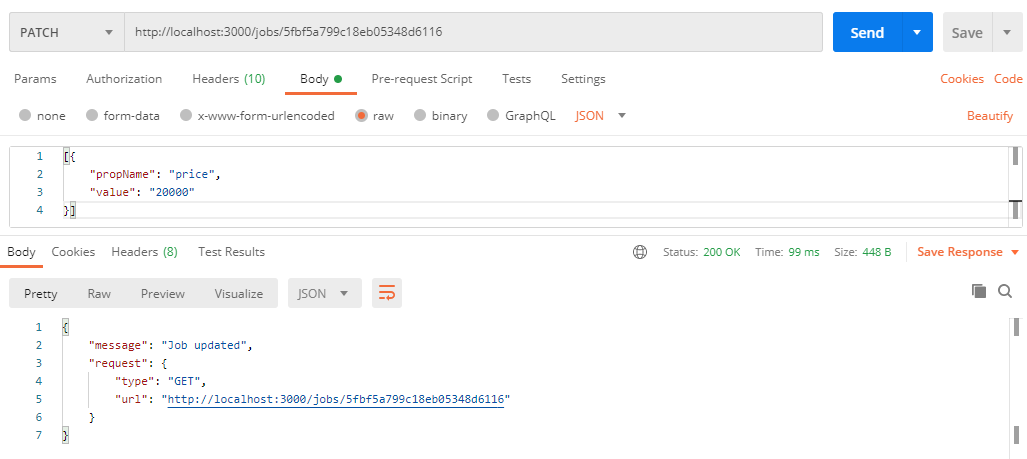


Рисунок 5.11 – Оновлення інформації

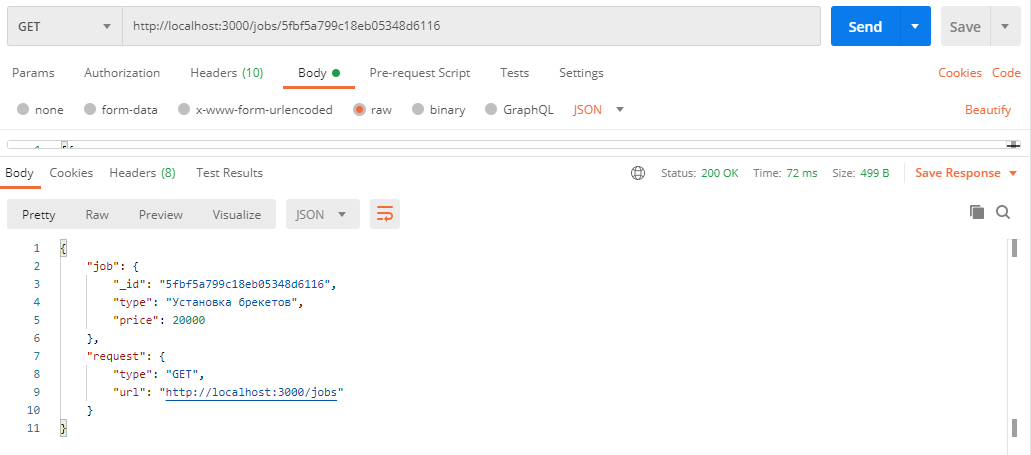


Рисунок 5.12 – Запит оновленої роботи

Тепер, видалимо дану роботу (рис. 5.13).

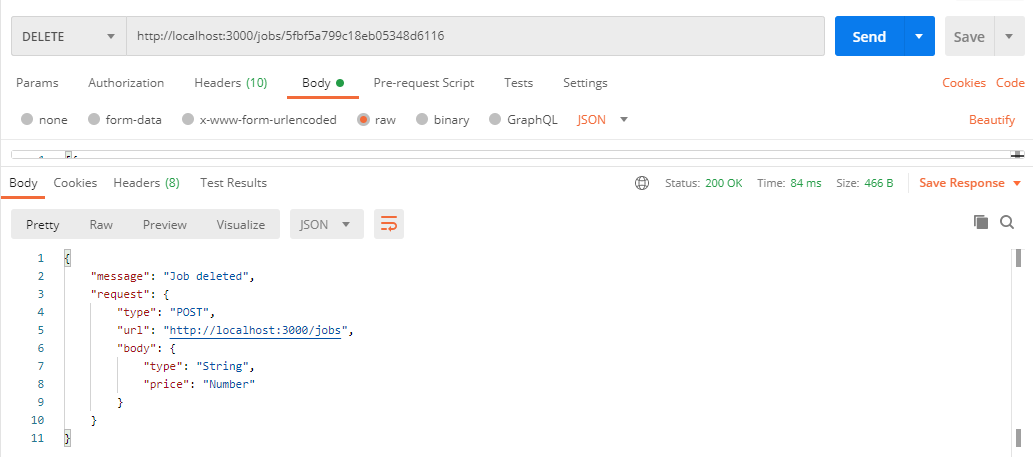


Рисунок 5.13 – Видалення роботи

Перевіримо її відсутність у загальному списку (рис. 5.14).

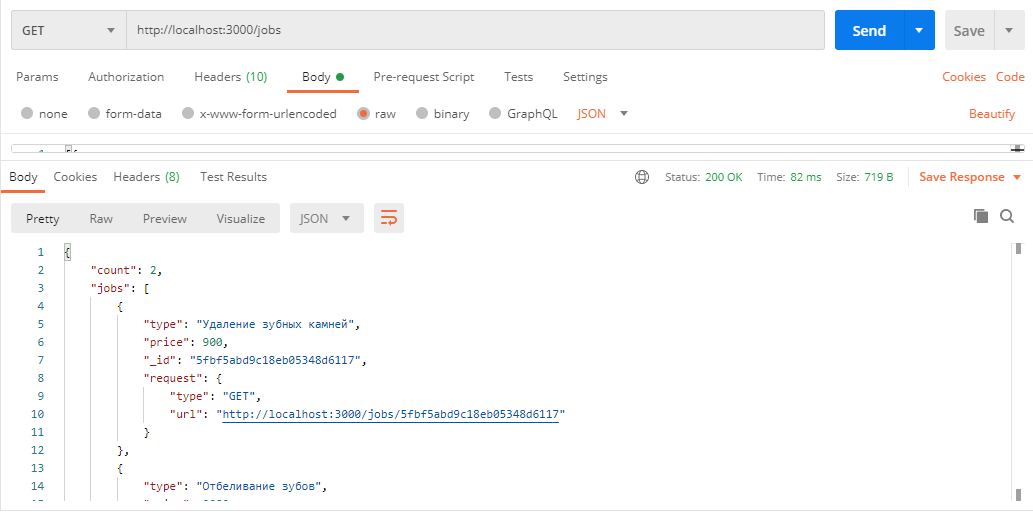


Рисунок 5.14 – Перевірка відсутності видаленої роботи

Як бачимо, кількість зменшилась і стала дорівнювати 2. Перевіримо коректність виконання запитів для ще 2 колекцій: «Patient» та «Visit». Для спрощення пояснення, було заздалегідь додано нового стоматолога за аналогією до створення нового власника. Залишилось зареєструвати пацієнта (рис. 5.15) та виконати авторизацію (рис. 5.16).

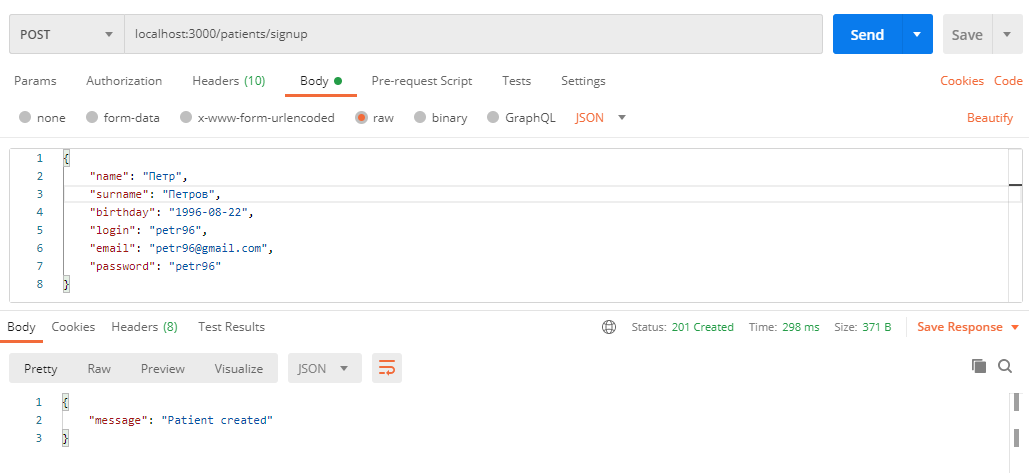


Рисунок 5.15 – Реєстрація пацієнта

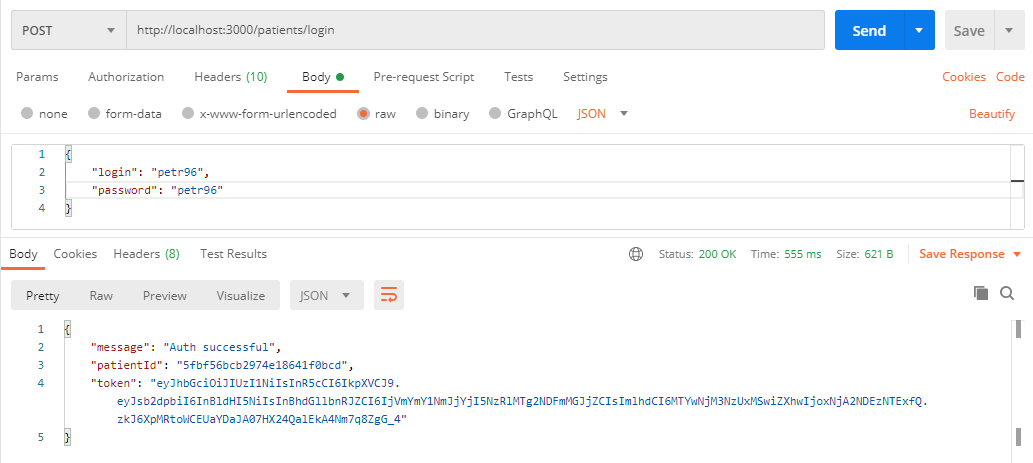


Рисунок 5.16 – Авторизація пацієнта

Створемо новий візит (рис. 5.17). При цьому вкажемо ідентифікатор новоствореного пацієнта та раніше створеного стоматолога.

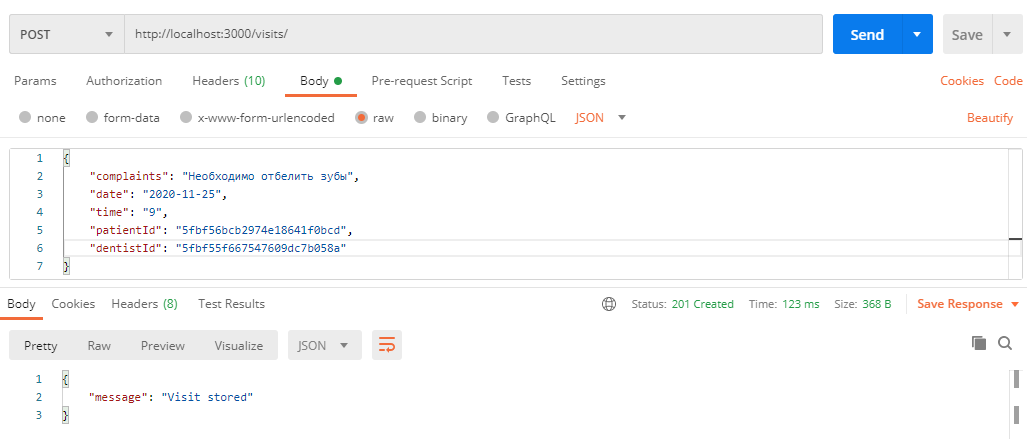


Рисунок 5.17 – Створення візиту

Спробуємо записатьсь на час, який не перебачає прийом стоматолога (19:00) (рис. 5.18).

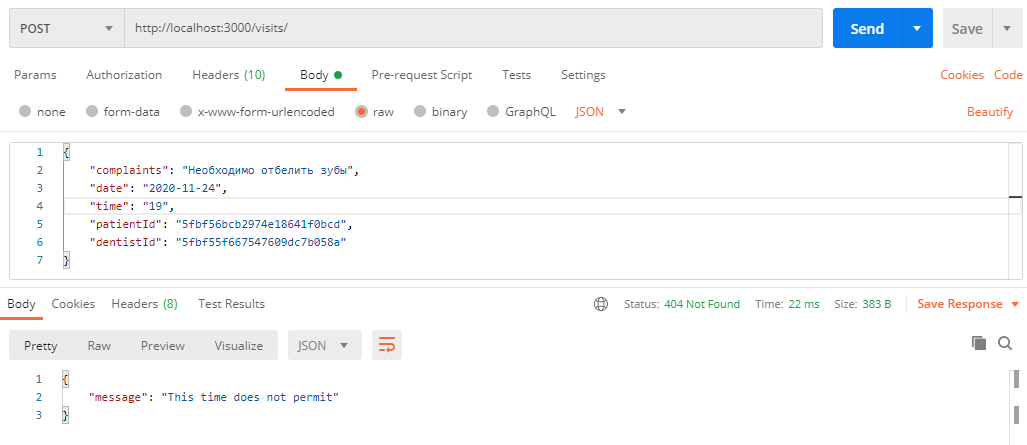


Рисунок 5.18 – Запис на заборонений час

Результатом є відповідь зі статусом 404, яка повідомляє, що даний час не є доступним для запису. Запит усіх візитів відтверджує той факт, що запис не було збережено (рис. 5.19).

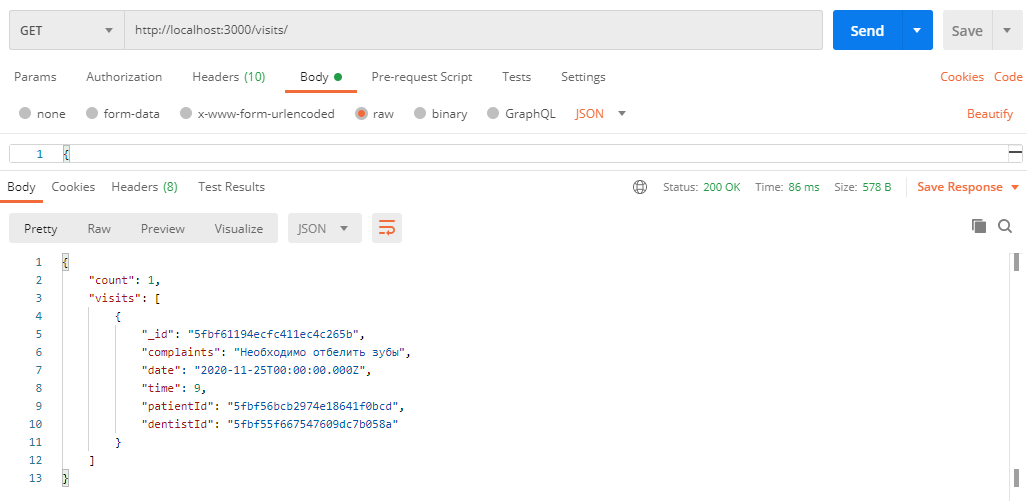


Рисунок 5.19 – Запит усіх візитів

Залишилось перевірити коректність роботи з токеном. Якщо виконати цей самий запит, але не вказати при цьому токен в заголовку із назвою «Authorization», то в запиті має бути відмовлено (рис. 5.20).

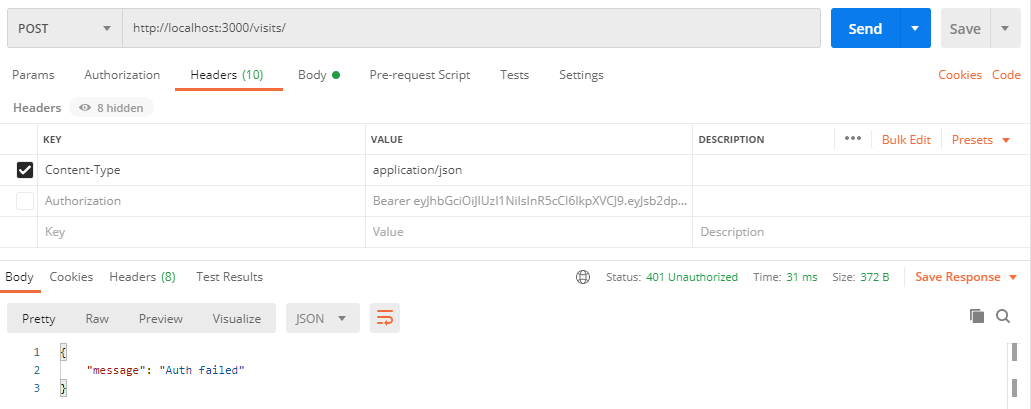


Рисунок 5.20 – Відмова в виконанні запиту

Запит спрацював правильно. Рис. 5.20 відображає, що для передачі зі сторони клієнта на сторону сервера даних, попередньо було зазначено заголовок «Content-Type» зі значенням «application/json» для визначення формату даних, що передаються.

## **5.3 Тестування функціональності**

Другим етапом тестування є тестування, для якого важливим є не програмний код модулів, а результат виконання. Очікуваний результат має співпадати із фактичним результатом. Зазначена трійка «Дія – Очікуваний результат – Фактичний результат» називається тест-кейсом. Кожен тест-кейс може містити певні передумови та постумови. Враховуючи те, що restful api не передбачає наявності станів системи, дані пункти вказувались лише для роботи з токеном (користувач авторизований або ні). Тестування функціональності було проведено для 4 варіантів використання системи: зареєструватись (табл. 5.1), авторизуватись (табл. 5.2), переглянути календар записів (табл. 5.3), залишити відгук (табл. 5.4) (із задіянням п’ятого варіанту використання – «Змінити ти відображення»).

Таблиця 5.1 – Тестування функціональності «Зареєструватись»

|  |  |  |
| --- | --- | --- |
| Дія | Очікуваний результат | Фактичний результат |
| Передумова: відсутня |  |  |
| Пацієнт виконує реєстрацію, неправильно заповнивши поля | Зазначення для всіх полів, які заповнені неправильно, вимог до їх заповнення | Зазначення для всіх полів, які заповнені неправильно, вимог до їх заповнення (успішно) |
| Пацієнт виконує реєстрацію, вказавши логін, який вже зареєстровано в системі | Повідомлення про те, що необхідно обрати інший логін, оскільки вказаний логін вже зареєстровано | Повідомлення про те, що необхідно обрати інший логін, оскільки вказаний логін вже зареєстровано (успішно) |

Продовження таблиці 5.1

|  |  |  |
| --- | --- | --- |
| Дія | Очікуваний результат | Фактичний результат |
| Пацієнт виконує реєстрацію, правильно заповнивши всі поля | Авторизація пацієнта, перехід до головного меню | Авторизація пацієнта, перехід до головного меню (успішно) |
| Постумова: відсутня |  |  |

Таблиця 5.2 – Тестування функціональності «Авторизуватись»

|  |  |  |
| --- | --- | --- |
| Дія | Очікуваний результат | Фактичний результат |
| Передумова: користувач зареєстрований в системі |  |  |
| Користувач виконує авторизацію, неправильно заповнивши поля | Зазначення для всіх полів, які заповнені неправильно, вимог до їх заповнення | Зазначення для всіх полів, які заповнені неправильно, вимог до їх заповнення (успішно) |
| Користувач вказує логін, який не зареєстровано в системі | Повідомлення, що вказаний логін не зареєстровано | Повідомлення, що вказаний логін не зареєстровано  (успішно) |
| Користувач виконує авторизацію, правильно заповнивши всі поля | Авторизація користувача | Авторизація користувача  (успішно) |
| Постумова: відсутня |  |  |

Таблиця 5.3 – Тестування функціональності «Переглянути календар записів»

|  |  |  |
| --- | --- | --- |
| Дія | Очікуваний результат | Фактичний результат |
| Передумова: пацієнт авторизований в системі |  |  |
| Пацієнт виконує запит на перегляд календаря записів | Перехід до вибору прізвища стоматолога та дати запису | Перехід до вибору прізвища стоматолога та дати запису  (успішно) |
| Пацієнт обирає пункт змінити тип виведення інформації | Відображення панелі зміни типу виведення інформації | Відображення панелі зміни типу виведення інформації  (успішно) |
| Пацієнт обирає необхідний тип змін | Відображення поточного значення параметру і всіх можливих варіантів | Відображення поточного значення параметру і всіх можливих варіантів  (успішно) |
| Пацієнт обирає нове значення параметру | Зміна типу відображення у відповідності до параметру | Зміна типу відображення у відповідності до параметру  (успішно) |
| Пацієнт вибрав прізвище стоматолога та дату | Перехід до записів на обрану пацієнтом дату | Перехід до записів на обрану пацієнтом дату  (успішно) |

Продовження таблиці 5.3

|  |  |  |
| --- | --- | --- |
| Дія | Очікуваний результат | Фактичний результат |
| Постумова: відсутня |  |  |
| Передумова: пацієнт неавторизований в системі (термін дії токену завершився) |  |  |
| Пацієнт виконує запит на перегляд календаря записів | Повідомлення про необхідність авторизації для перегляду календаря записів | Повідомлення про необхідність авторизації для перегляду календаря записів  (успішно) |
| Постумова: пацієнт авторизувався в системі |  |  |

Таблиця 5.4 – Тестування функціональності «Залишити відгук»

|  |  |  |
| --- | --- | --- |
| Дія | Очікуваний результат | Фактичний результат |
| Передумова: пацієнт авторизований в системі |  |  |
| Пацієнт виконує запит на створення відгуку | Відображення форми для відгуку | Відображення форми для відгуку  (успішно) |
| Пацієнт обирає пункт змінити тип виведення інформації | Відображення панелі зміни типу виведення інформації | Відображення панелі зміни типу виведення інформації  (успішно) |

Продовження таблиці 5.4

|  |  |  |
| --- | --- | --- |
| Дія | Очікуваний результат | Фактичний результат |
| Пацієнт обирає необхідний тип змін | Відображення поточного значення параметру і всіх можливих варіантів | Відображення поточного значення параметру і всіх можливих варіантів  (успішно) |
| Пацієнт обирає нове значення параметру | Зміна типу відображення у відповідності до параметру | Зміна типу відображення у відповідності до параметру  (успішно) |
| Пацієнт заповнює форму і натискає «Зберегти» | Повідомлення про успішне створення відгуку | Повідомлення про успішне створення відгуку  (успішно) |
| Постумова: відсутня |  |  |
| Передумова: пацієнт неавторизований в системі (термін дії токену завершився) |  |  |
| Пацієнт виконує запит на створення відгуку | Повідомлення про необхідність авторизації для створення відгуку | Повідомлення про необхідність авторизації для створення відгуку (успішно) |
| Постумова: пацієнт авторизувався в системі |  |  |

Табл. 5.1-5.4 демонструють коректну роботу для всіх 4 варіантів використання системи, адже для кожної дії очікуваний результат відповідає фактичному результату.

## **5.4 Приклад використання системи**

Для кращого розуміння принципів функціонування даного сервісу, користувачу слід попередньо ознайомитись із прикладом використання системи. Розглянемо наступні функції: отримання звітності разного типу для власника клініки, перегляд пацієнтом інформації про спеціалістів, із використання панелі зміни типу відображення. На рис. 5.21 відображено отримання загальної звітності. Перехід на дану сторінку відбувається при натисканні відповідної кнопки головного меню. Після цього власник клініки може спостерігати стовпчаті діаграми, які відповідають кількості візитів до стоматологічної клініки. Дана кількість стосується останніх 4 місяців. Для спрощення роботи із діаграмами, кожному місяцю було поставлено у відповідність свій колір. На панелі навігації, яка розташована в нижній частині екрану, можна побачити наступні кнопки: «Бахгалтерія», «Склад», «Стоматологи», «Головне меню». При переході за посиланням «Стоматологи», відбувається перехід до звітності, яка стосується діяльності стоматологів (рис. 5.22). Відображаються всі стоматологи клініки. Біля ім’я та прізвища кожного із стоматологів відображається кількість записів до нього. В правій частині наведено панелі прогресу, які розраховують процентні частки візитів до певного стоматолога до загальної кількісті візитів.

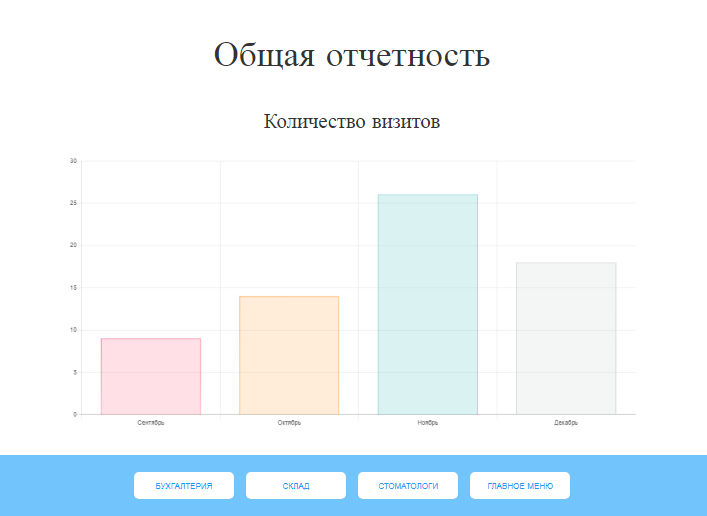


Рисунок 5.21 – Приклад перегляду загальної звітності

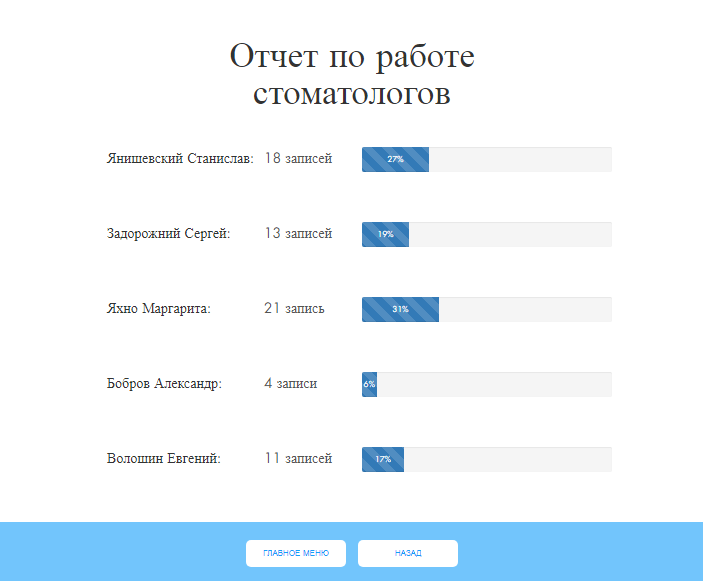


Рисунок 5.22 – Приклад перегляду звітності про діяльність стоматологів

Перегляд звітності інших типів (бухгалтерія та склад) відбувається за таким самим принципом. Для того, щоб повернутись до загальної звітності, слід натиснути на кнопку «Назад».

Далі, необхідно пояснити як взаємодіяти пацієнтам із панеллю зміни типу відображення. Для початку роботи із панеллю, слід обрати один із пунктів головного меню. Нехай було обрано перегляд інформації про спеціалістів. Окрім інформації про стоматолога, яка містить його персональні дані та фотографію, панелі навігації та кнопок переходу до інших спеціалістів, на рис. 5.23 можна спостерігати панель зміни типу відображення. Їй відповідає кнопка, яка виділена помаранчовим кольором та містить зображення лампочки.

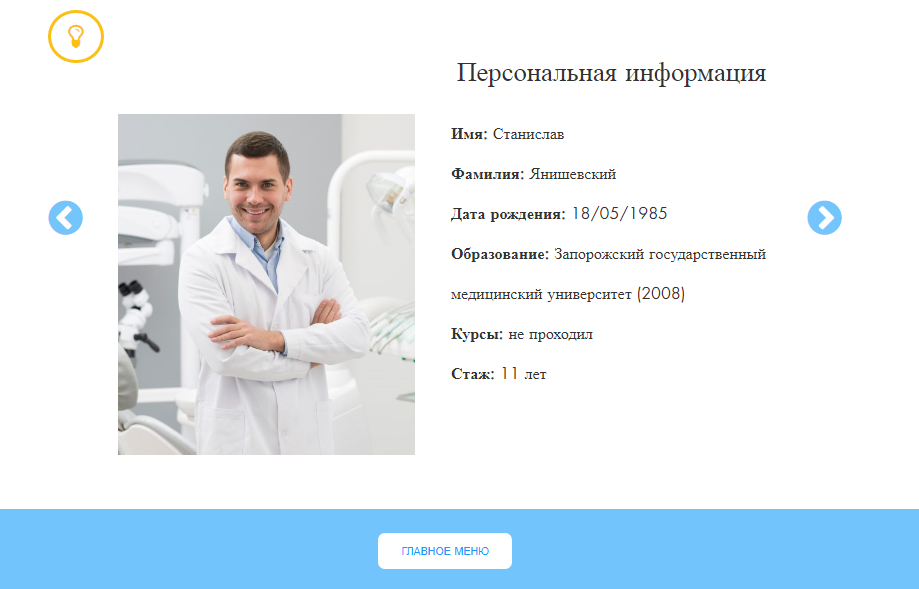


Рисунок 5.23 – Приклад перегляду інформації про спеціалістів

При натисканні на панель зміни типу відображення, з’являється меню вибору типу змін (рис. 5.24). Меню містить наступні пункти: «A» – налаштування анімації, «D» – зміна відстані між абзацами та відстані між рядками одного абзацу, «S» – зміна розміру тексту, «C» – вибір кольорів для фону та тексту, «F» – зміна шрифту. Для того, щоб прибрати меню, слід повторно натиснути на панель.

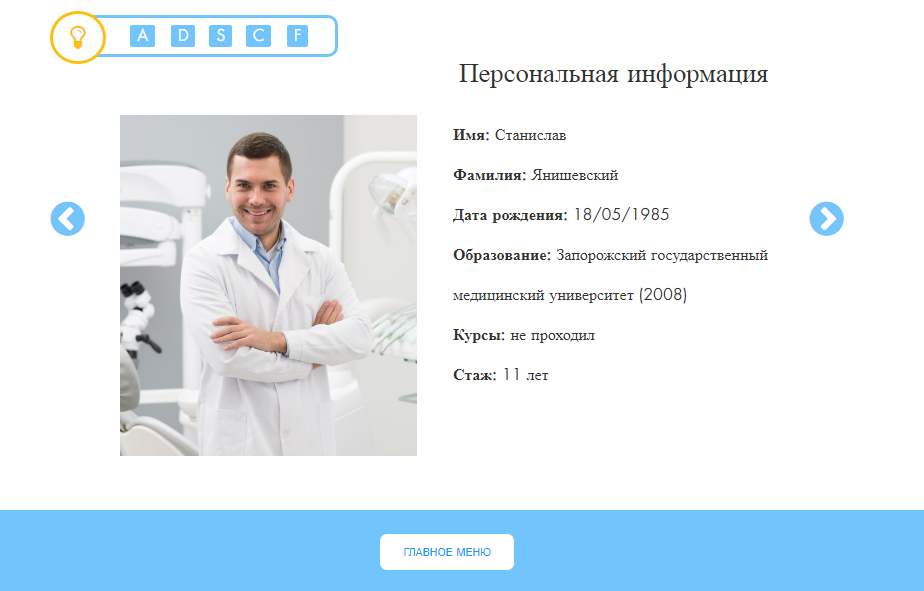


Рисунок 5.24 – Приклад взаємодії з панеллю зміни типу відображення

Для роз’яснення подальших дій, було обрано зміну шрифту. Після цього було відображено назву шрифту, який використовується для оформлення текстових елементів web-інтерфейсу, а також кнопки, за натисканням на які можна змінити вимальовування тексту на напівжирне, курсив або підкреслене (рис. 5.25).

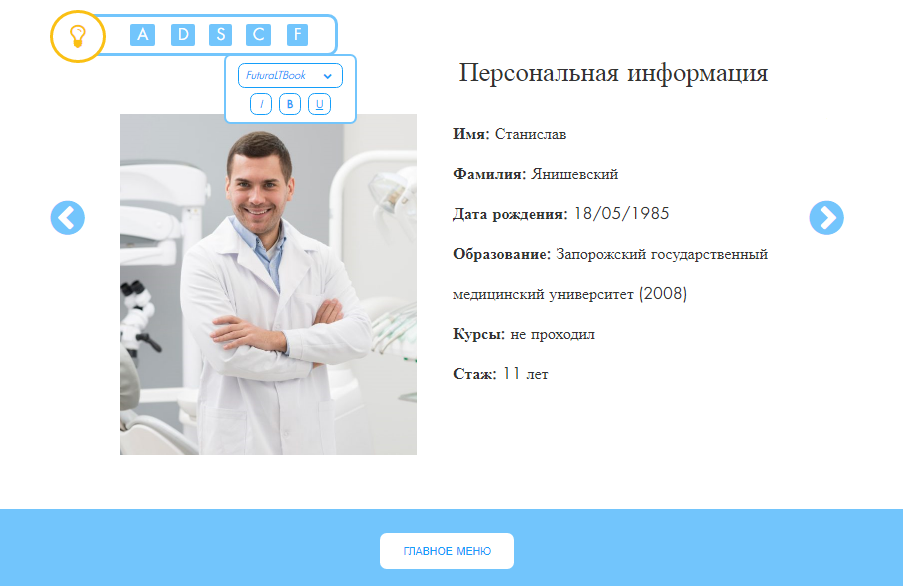


Рисунок 5.25 – Приклад перегляду змін, які стосуються шрифту

Після обрання іншого шрифту (MonotypeCorsiva), вимальовування тексту було змінено у відповідності до обраного значення параметру (рис. 5.26). Слід зазначити, що зміни не зачіпають панель навігації та панель зміни типу відображення, оскільки деякі елементи інтерфейсу мають залишатись незмінними для спрощення взаємодії з ними.

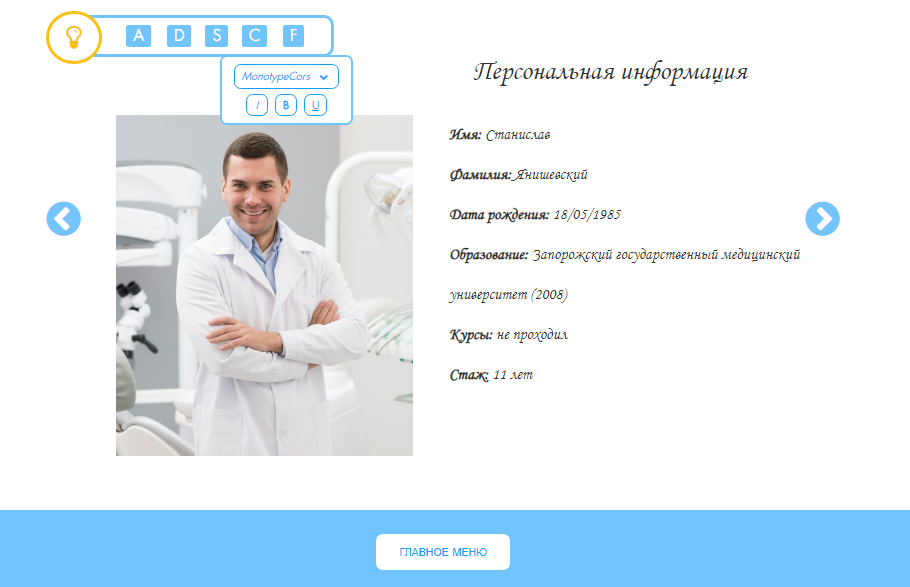


Рисунок 5.26 – Приклад зміни шрифту

# **6 АПРОБАЦІЯ СИСТЕМИ**

## **6.1 Апробація панелі зміни типу відображення**

Перший варіант апробації було виконано в 2019 році і стосувався виключно панелі зміни типу відображення. В якості реалізації розширеної моделі, було створено web-сайт, який інформує користувачів про способи захисту прав людини, а також про специфіку соціального захисту. Особливістю даного web-сайту була наявність розробленої та описаної панелі. Для визначення рівня доступності web-інтерфейсу, була сформована аудиторія, яка складалася з 30 людей. Із зазначених 30 осіб, 8 мали порушення здоров’я (4 людини мали порушення зору, 2 людини – порушення слуху, 1 – порушення опорно-рухового апарату, 1 – когнітивні порушення). На першому етапі апробації, аудиторії для оцінювання було надано версію сайту, яка реалізовувала лише класичну модель доступності. Після того, як час для ознайомлення з web-сайтом (30 хвилин), сплив, аудиторії було запропоновано відповісти на 2 запитання: «Чи зручно вам взаємодіяти з інтерфейсом web-сайту?», «Інформація на web-сайті подається в зручному для сприйняття форматі?». На кожне питання необхідно було відповісти «Так» або «Ні». В результаті, на перше запитання відповіли позитивно 20 людей (67%), а на друге – 24 людини (80%). На другому етапі апробації, аудиторія була ознайомлена з оновленою версією web-сайту, що реалізовувала розширену модель доступності. Учасникам експерименту було задано ті самі питання. У підсумку, на перше запитання відповіли позитивно 29 людей (97%), а на друге запитання – 27 людей (90%). Отримана інформація систематизована у вигляді таблиці (табл. 6.1).

Таблиця 6.1 – Результати першого експерименту

|  |  |  |
| --- | --- | --- |
|  | Перший етап | Другий етап |
| Взаємодія з web-інтерфейсом | 67% | 97% |
| Сприйняття інформації | 80% | 90% |

Графічне відображення зазначеної інформації надано на рис. 6.1.

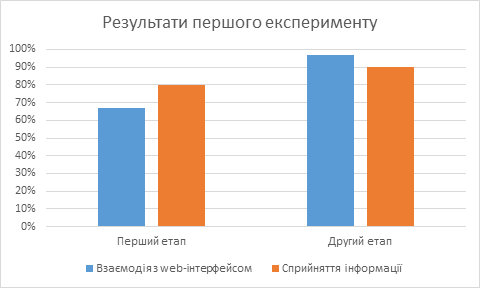


Рисунок 6.1 – Результати першого експерименту

## **6.2 Апробація сервісу**

Другий варіант апробації стосувався всієї системи. Базою для проведення експеримента слугувала 1 стоматологічна клініка. Експеримент був аналогічний тому, який виконувався в 2019 році. Участь брали: 1 власник клініки, 5 співробітників клініки та 50 пацієнтів. У цьому випадку тривалість експерименту була більшою, адже було враховано час, який необхідно учасникам експерименту для ознайомлення із всім функціоналом системи, доступним в рамках визначеного типу користувача. Перший варіант реалізовував класичну модель і не надавав доступу до панелі зміни типу відображення. Всі учасники експерименту повинні були відповісти на наступні запитання: «Чи зручно вам взаємодіяти з інтерфейсом сервісу?», «Інформація подається в зручному для сприйняття форматі?». На перше запитання дали позитивну відповідь 24 пацієнти та 2 співробітники, а на друге – 36 пацієнтів та 3 співробітників. Після цього, було реалізовано розширену модель доступності web-інтерфейсу, в рамках якої було надано доступ до панелі зміни типу відображення. Після цього всім учасникам експерименту було надано час на ознайомлення із сервісом, після чого було задано ті самі запитання. На цьому етапі відповіді були наступні: на перше запитання відповіли «Так» 44 пацієнти, 4 співробітники та власник клініки; на друге запитання відповіли «Так» 49 пацієнтів, 4 співробітники та власник клініки. Результати другого варіанту апробації систематизовано у вигляді табл. 6.2 та табл. 6.3. Перша таблиця враховує лише кількість відповідей, а друга – кількість відповідей та тип користувача.

Таблиця 6.2 – Результати другого експерименту (без урахування типу)

|  |  |  |
| --- | --- | --- |
|  | Перший етап | Другий етап |
| Взаємодія з web-інтерфейсом | 46% | 88% |
| Сприйняття інформації | 70% | 96% |

Графічне відображення зазначеної інформації надано на рис. 6.2.

Таблиця 6.3 – Результати другого експерименту (з урахуванням типу)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Перший етап | | | Другий етап | | |
|  | П | С | В | П | С | В |
| Взаємодія з web-інтерфейсом | 48% | 40% | 0% | 88% | 80% | 100% |
| Сприйняття інформації | 72% | 60% | 0% | 98% | 80% | 100% |

де П – пацієнт,

С – стоматолог,

В – власник стоматологічної клініки.

Графічне відображення зазначеної інформації надано на рис. 6.3.

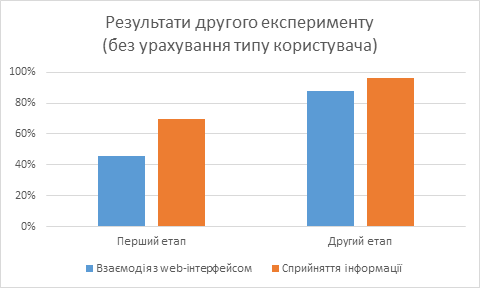


Рисунок 6.2 – Результати другого експерименту (без урахування типу користувача)

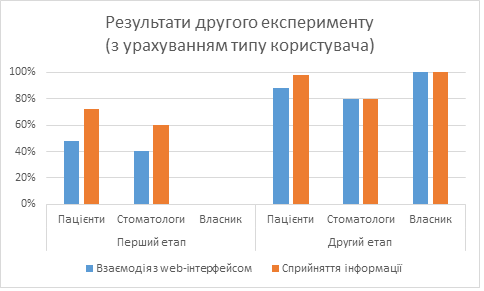


Рисунок 6.3 – Результати другого експерименту (з урахуванням типу користувача)

Результати проведеного експерименту дозволяють зробити висновок, що реалізація розширеної моделі доступності дозволила покращити взаємодію користувачів з web-інтерфейсом, а також їхнє сприйняття інформації, поданої в рамках розробленого сервісу.

# **7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

В даному розділі розглядається робота програміста, який займається web-розробкою, на робочому місці, яке розташовано на 8 поверсі офісного приміщення фірми «Norse digital». Охорона праці – невід’ємна складова діяльності будь-якого підприємства. В рамках охорони праці, необхідно визначити відповідність умов праці (освітлення) нормам. Наступною частиною є розрахунок параметрів пожежної безпеки технічних систем. Результатом виконаних обчислень мають бути сформульовані рекомендації щодо запобігання виникнення надзвичайних ситуацій в області пожежної безпеки, а також визначення можливості евакуації людей та усунення наслідків пожежі.

Більш детально питання охорони праці та безпеки у надзвичайних ситуаціях розкрито у Додатку Б. Даний розділ містить наступну інформацію:

– управління та організація охорони праці на фірмі «Norse digital»;

– вивчення основних умов праці на робочому місці помічника директора виробничого підприємства;

– розрахунок системи освітлення на робочому місці програміста, яке розташовано на фірмі «Norse digital»;

– розрахунок системи освітлення в приміщенні;

– розрахунок пожежної безпеки та особливостей локалізації наслідків вибуху, вивчення можливості використання конструкцій будівлі, яка залишилась;

– запропоновані заходи, які мають на меті покращення стану охорони праці на підприємстві;

– інструкції, які допомагають уникнути захворювання на вірусну інфекцію.

В якості висновків, можна зазначити наступне: було розглянуто організацію та управління охороною праці у фірмі «Norse digital». Структура СУОТ відповідає вимогам, встановленим чинним законодавством. Співробітники фірми «Norse digital» виконують усі положення, які стосуються охорони праці. Далі було визначено нормуємі параметри мікроклімату на робочому місці (температура, вологість, швидкість руху повітря).

Були визначені вимоги до шуму для створення сприятливих умов на робочому місці. У відповідності до індивідуального завдання було виконано розрахунок освітленості на робочому місці програміста. Можна відзначити, що фактична освітленість є більшою за нормовану. Було запропоновано комплекс заходів при проектуванні щодо розміщення світильників у робочому приміщенні у відповідності до побудованої схеми.

Далі, у відповідності до індивідуального завдання, було виконано розрахунок пожежної безпеки технічних систем, що передбачало визначення категорій пожежонебезпечних офісних приміщень, визначення категорій вибухопожежонебезпечності приміщень промислових будівель, визначення відстаней, які передбачають проведення евакуації людей, визначення витрат води на тушіння пожежі, визначення дії надлишкового тиску на об’єкт господарської діяльності при вибуху газоповітряної суміші.

Результати виконаних обчислень демонструють наступне: за умов вибуху, будівля зазнає настільки сильного деструктивного ефекту, що повторне використання залишків буде майже неможливим. Це стосується як елементів конструкції, так і устаткування Евакуація персоналу при виникненні пожежі має проходити нормально, адже усі робочі місця знаходяться на такій відстані від найближчого евакуаційного виходу, що люди, які їх займають, мають покинути приміщення вчасно.

На основі отриманих даних, було сформульовано заходи з охорони праці та безпеки у надзвичайних ситуаціях. У контексті безпеки у суспільстві в умовах загрози COVID-19, були формалізовані норми та вимоги до громадян та закладів навчання.

# **ВИСНОВОК**

У відповідності до поставлених задач було виконано наступні пункти:

– побудовано класичну модель доступності на основі аналізу існуючих web-сайтів і web-застосунків;

– описано компоненти класичної моделі доступності;

– побудувано розширену модель доступності web-інтерфейсів;

– описано компоненти розширеної моделі доступності;

– реалізовано розширену модель доступності в рамках сервісу управління та підтримки функціональності стоматологічної клініки;

– виконано апробацію сервісу, розробленого на основі розширеної моделі доступності.

Проаналізувавши особливості найбільш поширених типів порушення здоров'я у користувачів мережі Internet, вдалось розробити розширену модель доступності web-інтерфейсу. В якості основи для побудови розширеної моделі була використана класична модель доступності.

В класичну модель було включено компоненти, реалізація яких використовується більшістю сучасних web-сайтів та web-застосунків. Шляхом аналізу структури зазначених web-ресурсів, обраних ними рішень для кольорового оформлення, а також розмірів текстових та медіа-елементів, вдалось виділити 5 основних компонент моделі. Серед них: розділення тексту на тематичні блоки, допомога користувачу при виправленні помилок (найбільш часто застосовується при заповненні користувачем форм реєстрації та авторизації), відсутність небезпечних елементів та анімацій.

Шляхом додавання ще 7 компонент було отримано розширену модель доступності web-інтерфейсу. Дана модель передбачає можливість взаємодії з web-інтерфейсом користувачів з наступними типами порушень здоров’я: порушеннями зору, порушеннями слуху, порушеннями в роботі опорно-рухового апарату, когнітивними порушеннями. Для зазначених типів вад вдалось досягти наступних результатів: кількість компонент, які можуть бути задіяні при роботі з web-інтерфейсом людьми з порушеннями зору, вдалося збільшити з 5 до 10; кількість компонент, які можуть бути задіяні при роботі з web-інтерфейсом, людьми з порушеннями слуху – з 1 до 4; кількість компонент, які можуть бути задіяні людьми з проблемами у роботі опорно-рухового апарату – з 1 до 5; а для людей з когнітивними порушеннями, кількість компонент вдалось збільшити з 4 до 6.

Попередня апробація панелі зміни типу відображення продемонструвала наступні результати: покращення якості взаємодії користувача з web-інтерфейсом на 30%, покращення сприйняття користувачем інформації на 10%.

Результати апробації розробленого сервісу демонструють підвищення рівня доступності web-інтерфейсу на 34%, що підтвердило сформульовану на початку виконання робіт гіпотезу про можливість підвищення рівня доступності web-інтерфейсу при переході від класичної до розширеної моделі доступності. Зазначена величина була обчислена як середнє значення між наступними показниками:

1. Покращення якості взаємодії користувача з web-інтерфейсом – 42%.

2. Покращення сприйняття користувачем інформації – 26%.

Враховуючи, що найбільшою групою, яка брала участь в експерименті, була та, яку сформували пацієнти стоматологічної клініки, то можна зробити висновок, який стосується саме цієї групи: підвищення рівня доступності відповідає поставленим цілям і сягає 33% (покращення якості взаємодії пацієнтів з web-інтерфейсом – 40%, покращення сприйняття пацієнтами інформації – 26%).

# **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Carter, J., Markel, M. Web accessibility for people with disabilities: an introduction for Web developers. *IEEE Transactions on Professional Communication*. 2001. Vol. 44, No 4. P. 225-233.

2. King, M., Thatcher, J. W., Bronstad, P. M., Easton R. Managing usability for people with disabilities in a large Web presence. *IBM Systems Journal*. 2005. Vol. 44, No 3. P. 519-535.

3. Huang, C. J. Usability of e-government Web-sites for people with disabilities. *36th Annual Hawaii International Conference on System Sciences*, Big Island, January 06-09, 2003. Big Island, HI, USA, 2003. P. 50-53.

4. Тищенко, О. І., Лозінський, В. О., Оніщенко, Т. В., Писаренко, К. О. Система автоматичного визначення вимог до web-інтерфейсу для людей з порушеннями зору. *Сучасні інформаційні технології – 2019*: матеріали IX міжнар. конф. студентів і молодих вчених, м. Одеса, 23-24 травн. 2019 р. Одеса, 2019. C. 25-26.

5. Ursutiu, D., Iordache, D., Coftas, P., Coftas, D., Samolia, C. Web development techniques and remote laboratories. *International Journal of Online Engineering*. 2009. Vol. 5. No 1. P. 81-83.

6. Kulakov, K., Zavyalova, Y. Navigation infrastructure for people with disabilities. *20th Conference of open Innovations Association (FRUCT)*, St. Petersburg, April 03-07, 2017. St. Petersburg, Russia, 2017. P. 78-84.

7. Moreno, L., Martinez, P., Ruiz-Mezcua, B. Disability Standards for Multimedia on the Web. *IEEE MultiMedia*. 2008. Vol. 15. No 4. P. 52-54.

8. Navarrete, R., Lujan-Mora, S. OER-based learning and people with disabilities. *2015 International Conference on Interactive Collaborative and Blended Learning (ICBL)*, Mexico City, December 09-11, 2015. Mexico City, Mexico, 2015. P. 104-112.

9. Kowtko, M. Using assistive technologies to improve lives of older adults and people with disabilities. *2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, May 04, 2012. Farmingdale, NY, USA, 2012. P. 11-35.

10. Abul, K. Internet accessible remote experimentation: setting the right course of action. *International Journal of Online Engineering*. 2010. Vol. 6. No 3. P. 4-12.

11. Understanding the Web Content Accessibility Guidelines. URL: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG> (дата звернення: 03.05.2019).

12. Standard: Web Content Accessibility Guidelines (WCAG) 2.0. URL: <https://www.w3.org/TR/WCAG20/> (дата звернення: 02.05.2019).

13. The future of vision and eye care. URL: <https://medical-futurist.com/future-of-vision-and-eye-care> (дата звернення: 25.04.2019).

14. Design for colorblind persons. URL: <https://infogra.ru/ui/dizajn-dlya-daltonikov-5-faktorov-uspeha> (дата звернення: 05.05.2019).

15. Lubahn John D., Williams D. Patrick. The Hand and Wrist. Greene Netter's Orthopaedics. Saunders, 2005. 219 p.

16. Njokiktyen Charles. Pediatric behavioral neurology: in two volumes. Moscow: Terevinf, 2010. Vol. 2. P. 311-312.

17. Correctional computer equipment. URL: <https://invacenter.ru/сorrectional-computer-equipment/> (дата звернення: 02.05.2019).

18. Screen readers. URL: <https://www.afb.org/blindness-and-low-vision/using-technology/assistive-technology-products/screen-readers> (дата звернення: 07.05.2019).

19. An overview of Visual Studio Code for front-end developers. URL: <https://www.freecodecamp.org/news/an-overview-of-visual-studio-code-for-front-end-developers-49a4aa0771fb/> (дата звернення: 01.09.2020).

20. React V16.0. URL: https://reactjs.org/blog/2017/09/26/react-v16.0.html (дата звернення: 04.09.2020).

21. Node.js – Express Framework. URL: [https://www.tutorialspoint.com/nodejs/ nodejs\_express](https://www.tutorialspoint.com/nodejs/%20nodejs_express)\_framework.html (дата звернення: 05.09.2020).