

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Кафедра інженерії програмного забезпечення

КУРСОВИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Моделювання та аналіз програмного забезпечення»
на тему:
«Веб-сайт поліклініки»

студента IV курсу групи ІПЗ-19-1
спеціальності 121 «Інженерія програмного
забезпечення»
Гопчука Олександра Ярославовича
(прізвище, ім'я та по-батькові)

Керівник: доцент кафедри ІПЗ
Сугоняк І.І.

Дата захисту: " ____ " _____ 20__ р.
Національна шкала _____
Кількість балів: _____
Оцінка: ECTS _____

Члени комісії

_____	<u>І.І. Сугоняк</u>
(підпис)	(прізвище та ініціали)
_____	<u>О.В. Власенко</u>
(підпис)	(прізвище та ініціали)
_____	<u>С.М. Кравченко</u>
(підпис)	(прізвище та ініціали)

Житомир – 2022

Зміст	
ВСТУП	3
1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ	4
1.1 Технічне завдання на розробку системи.....	4
1.2 Обґрунтування вибору засобів моделювання	6
1.3 Аналіз вимог до програмного продукту	7
2 РОЗРОБКА МОДЕЛІ ПРОГРАМНОГО КОМПЛЕКСУ НА ЛОГІЧНОМУ РІВНІ	11
2.1 Алгоритм роботи та стани програмної системи	11
2.2 Об’єктно-орієнтована модель системи	13
2.3 Взаємодія об’єктів системи.....	17
3 ФІЗИЧНА МОДЕЛЬ ТА ПРОТОТИП ПРОГРАМНОГО КОМПЛЕКСУ	18
3.1 Взаємодія компонентів системи	18
3.2 Архітектура програмного комплексу та його розгортання.....	19
3.3 Генерування програмного коду для прототипу програми	21
ВИСНОВКИ	22
1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ	23
ДЖЕРЕЛА ІНФОРМАЦІЇ	23
ДОДАТКИ	25

					ДУ «Житомирська політехніка».22.121.03.000 - ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Гончук О. Я.			ФІКТ Гр. ІПЗ-19-1		
Перевір.		Сугоняк І.І.					
Керівник							
Н. контр.							
Зав. каф.							
					Лім.	Арк.	Аркушів
						1	34

ВСТУП

У сучасному світі, коли інформаційні технології набули надзвичайного значення в різних сферах діяльності, веб-сайти стали невід'ємною складовою успішного функціонування бізнесу та організацій. Сфера медицини не є винятком, адже поліклініки та медичні заклади активно використовують можливості Інтернету для забезпечення зручного та ефективного доступу до медичних послуг.

Метою курсової роботи є дослідження особливостей моделювання та аналізу програмних комплексів за визначеним темою курсової роботи напрямком.

Завданням на курсову роботу є:

- Аналіз теоретичних аспектів, пов'язаних з процесом моделювання програмного забезпечення.
- Вивчення та опис вимог користувачів до програмного продукту;
- Методи моделювання функцій та поведінки системи;
- Проектування об'єктної структури системи;
- Моделювання програмних комплексів;
- Кодогенерація з моделей.

Предметом дослідження є можливості застосування CASE-засобів проектування програмного забезпечення.

Об'єктом дослідження є методи та засоби проектування програмного забезпечення та уніфікації процесу проектування.

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ

1.1 Технічне завдання на розробку системи

1. Назва системи:

Повне найменування системи: веб-сайт поліклініки

Коротке найменування системи: Веб-додаток, Веб-сайт, сайт, система, Програмний комплекс.

2. Призначення системи:

Програмний комплекс створений для поліпшення доступу до медичних послуг та спрощення взаємодії між пацієнтами та медичним персоналом.

3. Вимоги до системи:

Програмний продукт повинен мати наступні функціональні характеристики:

- Доступ до системи за допомогою популярних в наш час браузерів;
- Реєстрація та авторизація пацієнтів та медичного персоналу.
- Онлайн запис на прийом до лікарів з відображенням доступних часових слотів.
- Перегляд розкладу роботи лікарів та їх спеціалізації
- Можливість заміни або скасування прийому
- Онлайн консультації з медичним персоналом;
- Система сповіщень та нагадувань про прийоми, зміни у розкладі або лікуванні
- Перегляд актуальної інформації про медичні послуги, процедури та програми лікарні
- Наявність шести ролей користувачів (адміністратор, модератор, лікар, консультант, пацієнт, гість).

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Вимоги до програмного забезпечення:

Програмний комплекс повинен бути побудований на клієнт-серверній архітектурі з використанням веб-технологій, що не вимагають додаткового ліцензування. Серверна частина програмного комплексу має бути реалізована з використанням Node.JS та системи управління базами даних MySQL. Для написання такої системи буде доцільно використати мову JavaScript. При розробці буде використано фреймворк Express.js для серверної частини і React.js для клієнта.

Клієнтська частина має бути побудована на стандартних веб-технологіях, що не вимагають встановлення додаткових компонентів на комп'ютер. Сторінки інтерфейсу генеруються сервером за стандартом HTML5, оформлення здійснюється за допомогою таблиць стилів CSS. Інтерфейс повинен коректно працювати у сучасних веб-браузерах Mozilla FireFox та Google Chrome. Також, є можливим використання готових бібліотек для пришвидшення розробки застосунку та покращення його швидкодії. Додатково будуть використані модулі, які спрощують процес створення системи і поліпшують користувацький досвід.

5. Вимоги до інтерфейсу:

Користувач має взаємодіяти з продуктом за допомогою веб-браузера. Відповідно, інтерфейс має бути функціональним, незалежно від браузера, що використовується, за умови що браузер є сучасним і популярним. Інтерфейс має бути зручним і функціональним. Час відклику на дії користувача має бути достатньо малим, щоб не викливати дискомфорту. Введення та виведення даних в систему, введення команд та результат їх виконання повинні виконуватися в інтерактивному режимі.

До інтерфейсу є наступні вимоги

- Мінімальний час відгуку системи;
- Повідомлення про критичні ситуації системи;

		Гопчук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

- Підтвердження виконання критичних дій в системі
- Відображення процесу завантаження для ресурсомістких завдань.

1.2 Обґрунтування вибору засобів моделювання

UML, або Unified Modeling Language (Об'єднана мова моделювання), є мовою позначень та побудови діаграм, яка призначена для визначення, візуалізації та документування моделей програмного забезпечення, орієнтованих на об'єкти. UML не є методом розробки, тобто вона не накладає обмежень щодо порядку дій або інструкцій щодо побудови системи. Замість цього, UML надає засоби для наочного представлення компонентів системи та полегшує співпрацю між розробниками.

Для моделювання даної курсової роботи розглянемо нижче наведені засоби UML:

- ArgoUML
- Dia
- StarUML

Зробимо аналіз кожного засобу:

ArgoUML:

- Підтримка UML - UML 1.4
- Генерація вихідних кодів можлива на основі UML діаграм для мов програмування, таких як Java, C++, C#, PHP. Також можна додати інші мови.
- Здійснюється зворотне проектування вихідного коду для мови програмування Java.
- Немає підтримки для баз даних.
- Відсутній багатокористувацький режим роботи.
- Цей програмний комплекс є відкритим програмним забезпеченням.

Dia:

- Підтримка UML - 2.0

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

- Можлива генерація вихідних кодів з використанням UML діаграм для мов програмування, таких як Ada, C, C++, Java, PHP, Python, SQL (за допомогою Dia2Code).
- Програмний комплекс забезпечує зворотне проектування вихідного коду для мов Perl, Java, PHP, C++ (за допомогою AutoDia).
- Підтримка роботи з базами даних є в програмному комплексі.
- Цей програмний комплекс є відкритим програмним забезпеченням.

StarUML:

- Підтримка UML - 2.0
- Перевірка коректності UML діаграми здійснюється.
- Генерація вихідного коду на основі UML діаграми можлива для мов програмування, таких як Ada, Java, C#, C++. Для інших мов доступні додаткові розширення на офіційному веб-сайті.
- Програмний комплекс підтримує зворотне проектування вихідного коду.
- В програмному комплексі забезпечується підтримка патернів.
- Можливий експорт документації в різних форматах, таких як DOC, PPT, TXT, XLS та інші.
- Робота з базами даних не підтримується.
- Програмний комплекс має зручний графічний редактор.
- Цей програмний комплекс є відкритим програмним забезпеченням.

Після проведення порівнянь засобів UML було обрано «StarUML», завдяки тому, що він підтримує генерацію коду на різні мови програмування, є зручним у використанні, а також має безкоштовну версію.

1.3 Аналіз вимог до програмного продукту

Розглянемо вимоги, які повинен задовільнити програмний продукт, аби мати можливість виконувати всі покладені на нього функції, тим самим задовольнивши всі визначені стандарти, специфікації та інші формальні

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

документи. Приведені можливості системи, обмеження та показники якості.

Бізнес-вимоги:

1. Основні цілі: проект створюється з метою простого та зручного доступу до медичних послуг, взаємодії лікарів та клієнтів.
2. Представлення проекту: проект буде реалізовано у вигляді сайту, що містить можливості структуровану інформацію по даній тематичній області.

Вимоги користувачів

- Зовнішні користувачі
 1. Можливість перегляду новин(не авторизований користувач)
 2. Можливість перегляду послуг (не авторизований користувач)
 3. Можливість перегляду списку лікарів (не авторизований користувач)
 4. Пошук послуг за назвами (не авторизований користувач)
 5. Можливість редагування особистого кабінету (не авторизований користувач)
- Внутрішній користувач (адміністратор, модератор, лікар, консультант, пацієнт)
 1. Публікація новин (модератор)
 2. Додавання і редагування списку лікарів, послуг (модератор)
 3. Чат між клієнтом та консультантом (клієнт, консультант)
 4. Заповнення послуг (клієнт)
 5. Зміна та перегляд графіку прийомів (лікар)
 6. Запис на прийом (консультант, клієнт)
 7. Управління ролями (адміністратор)

Функціональні вимоги

1. Авторизація користувачів в системі: в системі повинна бути представлена можливість реєстрації користувача та присвоєння йому відповідної ролі (адміністратор, модератор, лікар, консультант, пацієнт).

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Можливість збереження інформації: система повинна зберігати інформацію і надавати можливість користувачу керувати нею.
3. Створення умов онлайн спілкування: система повинна надавати можливість консультанту спілкуватися з клієнтом.

Нефункціональні вимоги

1. Сприйняття
 - Час, потрібний для навчання інструментами роботи з інформаційною системою для звичайних користувачів – 20 хвилин, а для досвідчених – 10 хвилин
 - Час відповіді системи для звичайних запитів не повинен перевищувати 5 секунд, а для більш складних запитів – 10 секунд
 - Інтерфейс представлення веб-сайту має бути інтуїтивно зручним для користувача та не вимагати від нього додаткової підготовки
2. Надійність
 - Доступність – час, потрібний для обслуговування сайту не повинен перевищувати 10% від загального часу роботи
 - Середній час безперервної роботи – 20 робочих днів
 - Максимальна норма помилок та дефектів в роботі системи – 1 помилка на 1000 запитів користувача
3. Продуктивність
 - Система повинна підтримувати мінімум 50 одночасно працюючих користувачів, пов'язаних з спільною базою даних
4. Можливість експлуатації
 - Масштабування – система повинна мати можливість збільшувати потужності зі збільшенням користувачів таким чином, щоб це не відобразилось на її роботі негативно
 - Оновлення версій – Оновлення версій повинно здійснюватися в автоматичному режимі залежно від вподобань користувачів та розширення списку послуг

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Системні вимоги

1. Вимоги до середовища виконання:

Система повинна задовольняти зазначеним вимогам на комп'ютері в наступній мінімальній комплектації:

- підтримка браузера Google Chrome версії 45+ або
- підтримка браузера Mozilla Firefox версії 38+ або
- підтримка браузера Microsoft Edge версії 12+ або
- підтримка браузера Opera версії 30+ або
- підтримка браузера Safari версії 9+

2. Вимоги до СУБД та доступу до даних

- У ядрі системи повинна бути представлення СУБД реляційного доступу

Аналіз вимог користувачів дав підстави для визначення варіантів використання веб-сайту поліклініки. На рисунку 1.1 зображено побудовану діаграму варіантів використання.

На рисунку 1.4 зображено побудовану діаграму варіантів використання

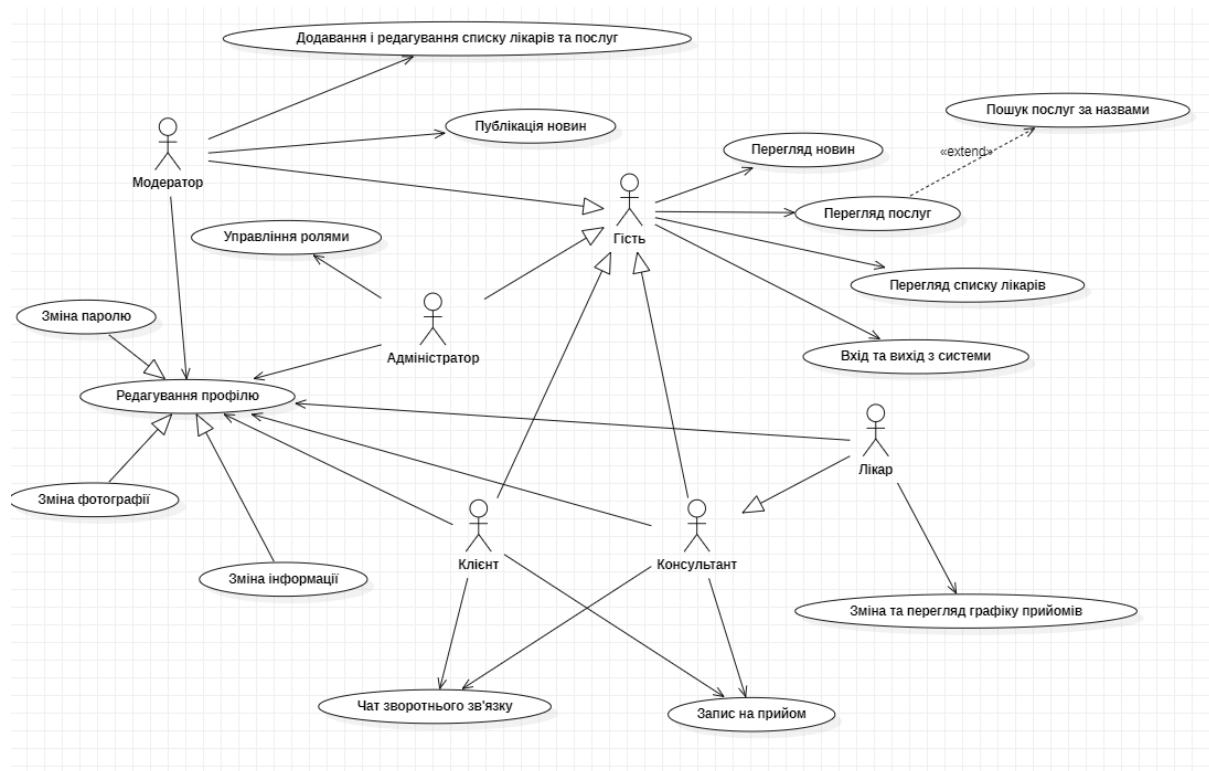


Рис. 1.3 – Діаграма варіантів використання

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

2 РОЗРОБКА МОДЕЛІ ПРОГРАМНОГО КОМПЛЕКСУ НА ЛОГІЧНОМУ РІВНІ

2.1 Алгоритм роботи та стани програмної системи

Під час планування алгоритму роботи та станів програмного додатку необхідно побудувати діаграму активності(Activity diagram).

Діаграма активності (Activity diagram) – UML-діаграма, яка дозволяє моделювати послідовності бізнес-процесів або дій, реалізованих методами класів. Зазначені послідовності можуть являти собою альтернативні галузі процесу обробки даних або галузям, які можуть виконуватися паралельно. Діаграми діяльності є аналогом блок-схеми будь-якого алгоритму.

Було створено 2 діаграми активності для можливих сценаріїв в роботі системи.

На рисунку 2.1 зображено діаграму активності та відповідності стану дій для процесу замовлення послуги клієнтом, та подальше коригування дати виконання послуги.

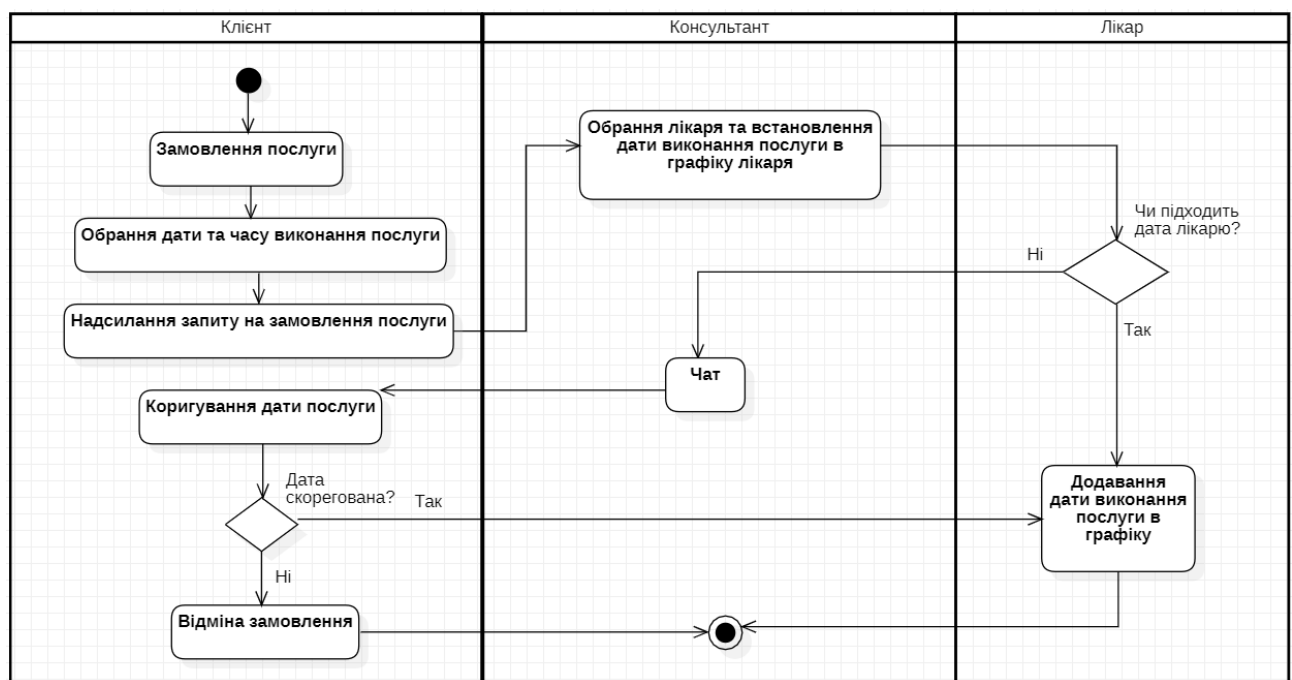


Рис. 2.1. Діаграма активності для процесу замовлення послуги клієнтом

Опис діаграми активностей:

- Початковий стан - чорний круг
- Кінцевий стан - чорний круг в колі

- Рішення - ромби, а саме:
 - Чи підходить дата лікарю?
 - Чи дата скорегована?
- Округлені прямокутники позначають дії. На діаграмі наявні наступні
 - Замовлення послуги
 - Обрання дати та часу виконання послуги
 - Надсилання запиту на замовлення послуги
 - Обрання лікаря та встановлення дати виконання послуги в графіку лікаря
 - Додавання дати виконання послуги в графіку
 - Коригування дати послуги
 - Відміна замовлення

На рис. 2.3 наведено діаграму активності та відповідності стану дій для зміщення дати виконання послуги у графіку лікаря, та подальша взаємодія з іншими ролями.

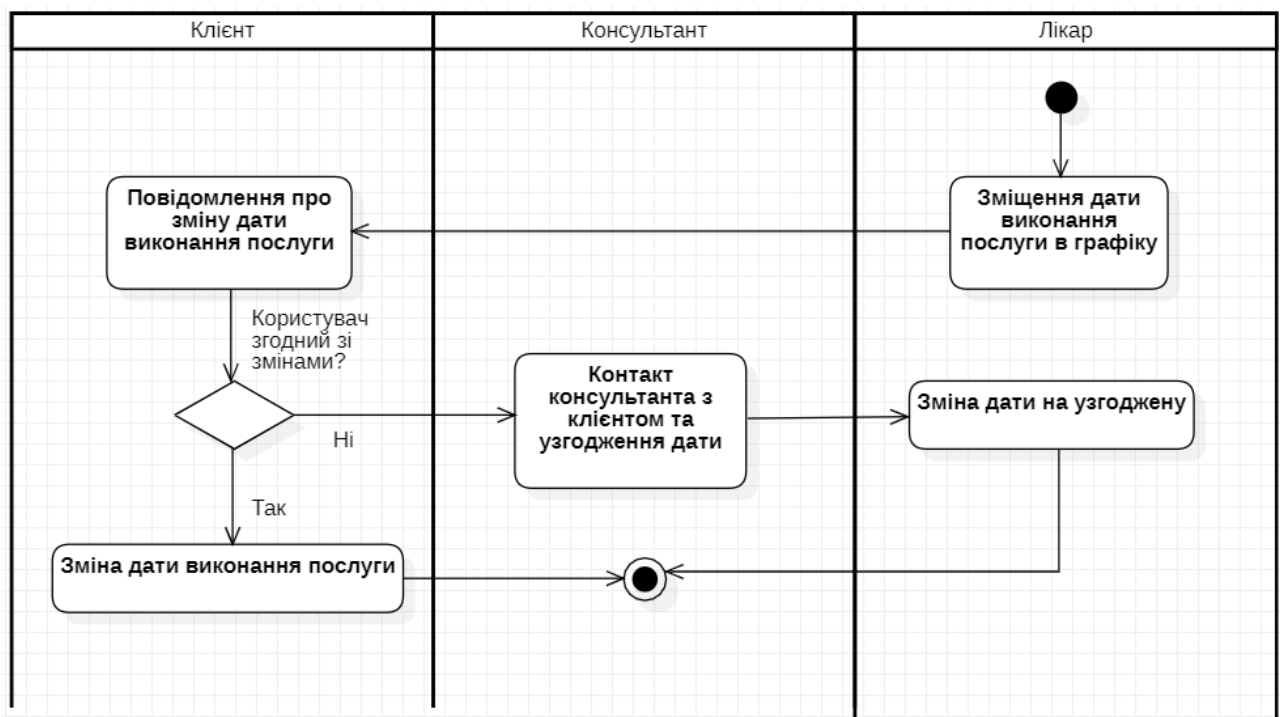


Рис. 2.3. Діаграма дій для процесу зміни дати виконання послуги

Опис діаграми активностей:

- Початковий стан - чорний круг

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

- Кінцевий стан - чорний круг в колі
- Рішення - ромб, а саме:
 - Чи користувач згодний зі змінами?
- Округлені прямокутники позначають дії. На діаграмі наявні наступні
 - Зміщення дати виконання послуги в графіку
 - Обрання дати та часу виконання послуги
 - Повідомлення про зміну дати виконання послуги
 - Обрання лікаря та встановлення дати виконання послуги в графіку лікаря
 - Контакт консультанта з клієнтом та узгодження дати
 - Коригування дати послуги
 - Зміна дати на узгоджену
 - Зміна дати виконання послуги

2.2 Об'єктно-орієнтована модель системи

Об'єктно-орієнтовану модель системи прийнято розглядати у вигляді діаграми класів (class diagram). Діаграма класів - основна діаграма для створення коду додатка. За допомогою діаграми класів створюється внутрішня структура системи, описується спадкування й взаємне положення класів один відносно другого.

На рисунку 2.3 зображено побудовану діаграму класів

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

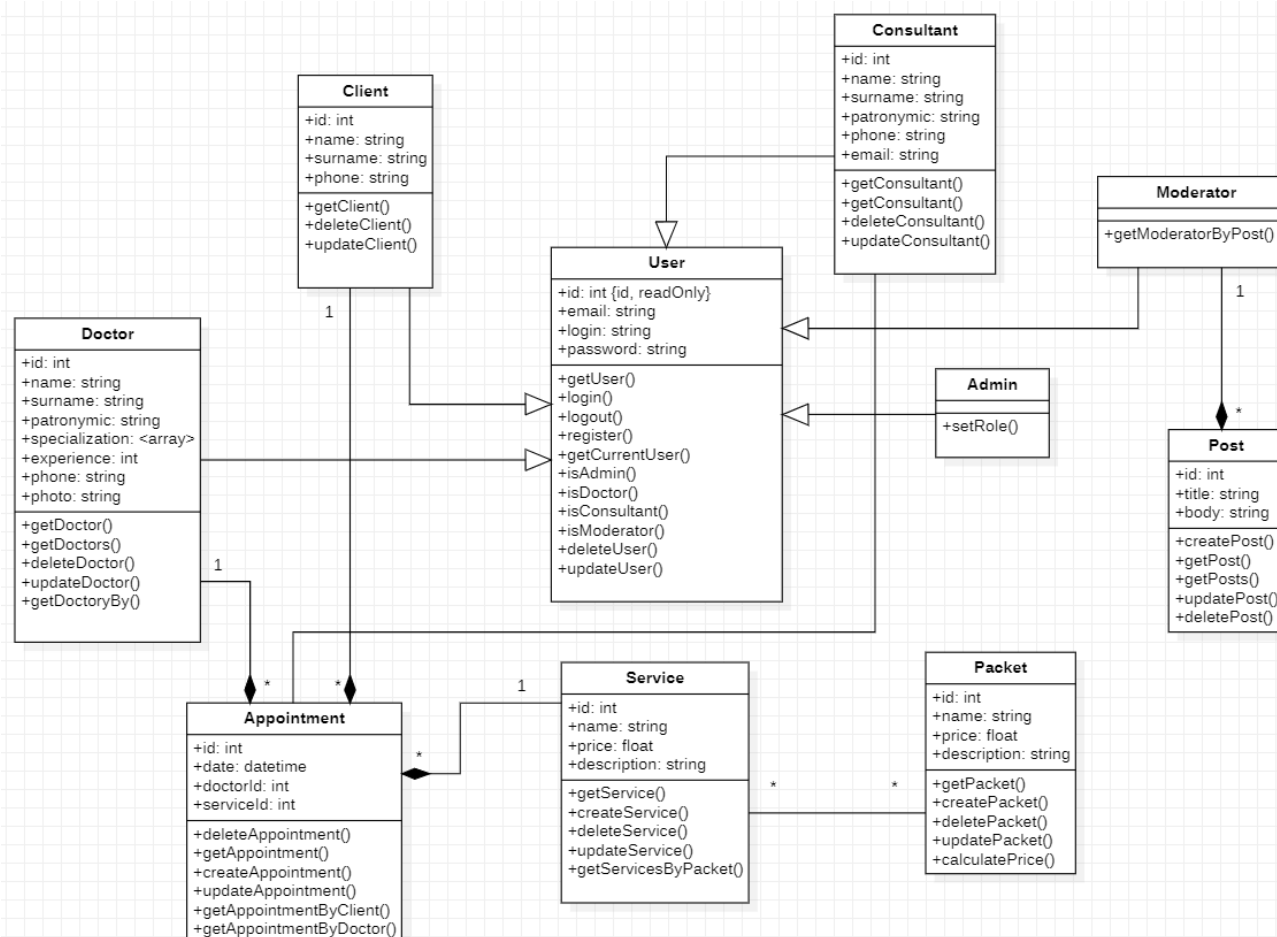


Рис. 2.3. Діаграма класів

Нижче наведено опис класів, а також їх атрибутів та методів:

- Doctor є моделлю лікаря. Містить властивості про лікаря, а саме: ім'я, прізвище, спеціалізація, досвід, телефон, фото. Наявні методи для управління моделлю (getDoctor, updateDoctor, deleteDoctor)
- User є моделлю користувача. Містить деякі дані, такі як: електронна пошта, логін, пароль. Є методи для керування користувачем (login, logout, register, deleteUser, updateUser)
- Client є моделлю клієнта. Містить дані про клієнта: ім'я, прізвище, номер телефону. Також методи для роботи з клієнтом (отримання, видалення, редагування клієнта)
- Consultant є моделлю консультанта. Містить властивості про консультанта, такі як: ім'я, прізвище, телефон, електронна пошта.

Наявні методи для роботи з консультантом(отримання, зміна, видалення)

- Appointment є моделлю запису на прийом до лікаря. Створюється консультантом, містить зв'язки між послугою та лікарем. Також наявні методи для керуванням запису.
- Service є моделлю послуги. Має наступні атрибути: назва, опис, ціна. Також є методи для керуванням послугою(отримання послуги за пакетом послуг).
- Packet є моделлю пакету послуг. Містить дані про пакет послуг: назва, ціна, опис. Також методи для роботи з пакетом(отримання, видалення, редагування, обрахування ціни).
- Post є моделлю новини на сайті. Має наступні атрибути: заголовок, внутрішній текст. Є методи для роботи з новиною, а саме, створення, видалення, редагування, отримання.
- Admin є моделлю адміністратора, створена для керуванням ролями, тобто видача ролі, зміна.
- Moderator – є моделлю модератора, який має можливість вносити нових лікарів на сайт, та писати новини.

Патерни проектування

Як згадувалося раніше, у проектуванні використовуються патерни. Патерни проектування є рішеннями для поширених проблем, які можуть виникнути. Вони сприяють прискоренню процесу розробки, адже створюються для усунення відомих проблем ще до їх появи в процесі реалізації.

Патерн - це загальний принцип вирішення проблеми, і його застосування потребує налаштування під конкретну програму, оскільки не можна просто копіювати код. Хоча знання патернів не є обов'язковими, можна успішно працювати і без них. Ви, можливо, вже використовували певні патерни у своїй роботі, навіть не особливо про це думаючи.

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Проте, знання патернів дозволяє більш ефективно вирішувати завдання та розробляти програмне забезпечення. Патерни надають можливість для:

- Перевірені рішення;
- Стандартизація коду;
- Загальний словник програмістів.

Патерни відрізняються за рівнем складності, деталізації та охоплення проєктованої системи. Розглянемо основні групи патернів:

- Породжуючі патерни піклуються про гнучке створення об'єктів без внесення в програму зайвих залежностей;
- Структурні патерни показують різні способи побудови зв'язків між об'єктами;
- Поведінкові патерни піклуються про ефективну комунікацію між об'єктами.

Прототип – це породжуючий патерн проєктування, що дає змогу копіювати об'єкти, не вдаючись у подробиці їхньої реалізації. Клас User є прототипом до класів Admin, Client, Doctor, Moderator та Consultant. Реалізацію патерну представлено на рис.2.4

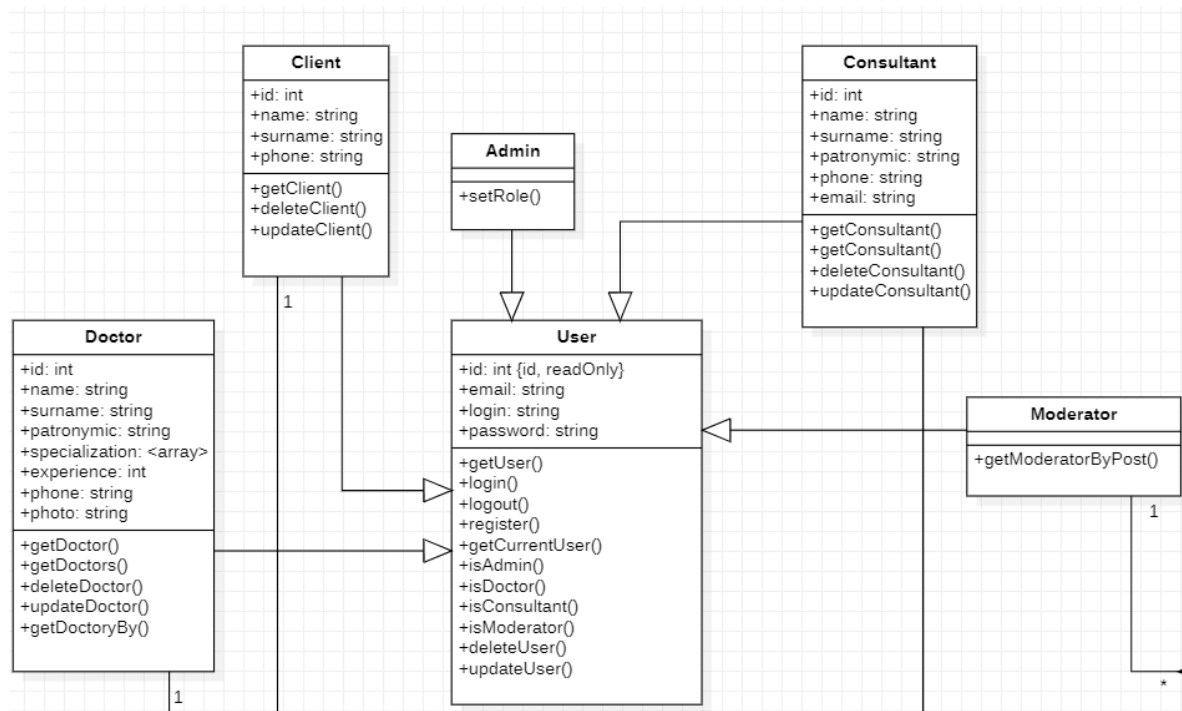


Рис. 2. 4. Загальний вигляд патерну «Прототип»

2.3 Взаємодія об'єктів системи

Взаємодію об'єктів системи в UML прийнято зображати у вигляді діаграми послідовності.

Діаграми послідовності є діаграмами взаємодії, які детально описують виконання операцій. Вони уявно відтворюють взаємодію між об'єктами під час співпраці. Ці діаграми графічно відображають порядок взаємодії та дії, які будуть виконані.

Було вирішено побудувати діаграму послідовності для замовлення послуги

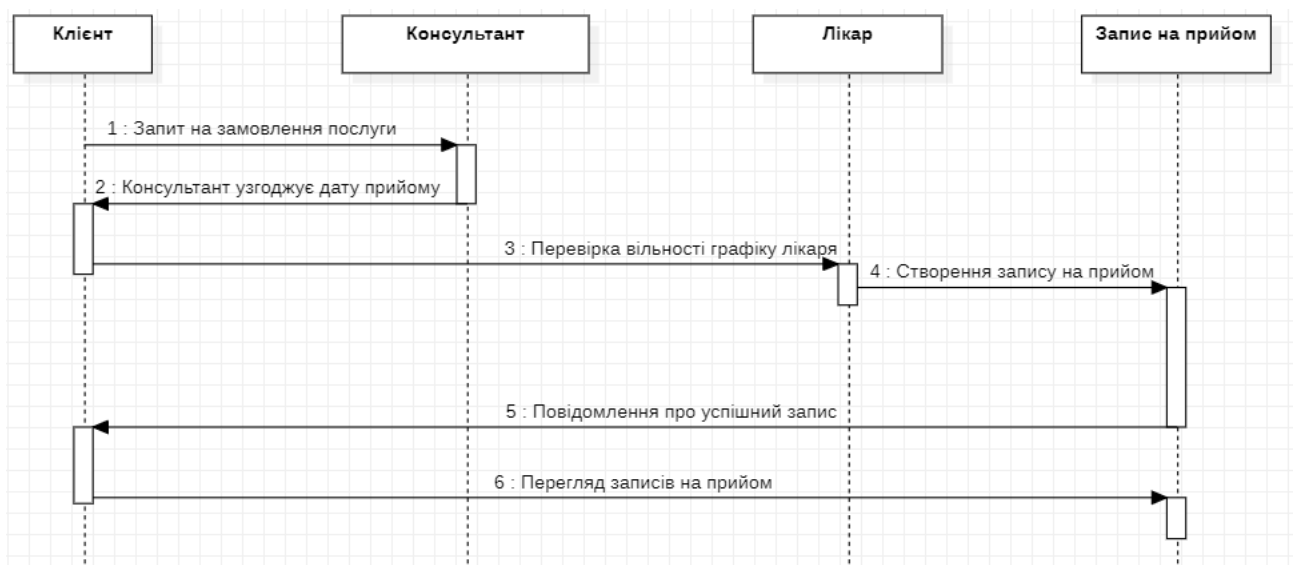


Рис. 2.5. Діаграма послідовності для замовлення послуги клієнтом

Опис діаграми послідовності:

1. Клієнт обирає послугу, та замовлює її
2. Клієнтська частина шле запит на сервер
3. Консультант отримує запит на замовлення послуги
4. Консультант узгоджує дату прийому та перевіряє вільність графіку лікаря та створює новий запис на прийом до лікаря
5. На сервері створюється новий запис, надсилається повідомлення про створення запису
6. Клієнт отримує повідомлення про успішний запис на прийом

7. Клієнт переглядає його записи на прийом

3 ФІЗИЧНА МОДЕЛЬ ТА ПРОТОТИП ПРОГРАМНОГО КОМПЛЕКСУ

3.1 Взаємодія компонентів системи

Діаграма компонентів – статична структурна діаграма, що показує розбиття програмної системи на структурні компоненти та зв'язки (залежності) між компонентами. Як фізичні компоненти можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети.

На рисунку 3.1 зображено діаграму компонентів.

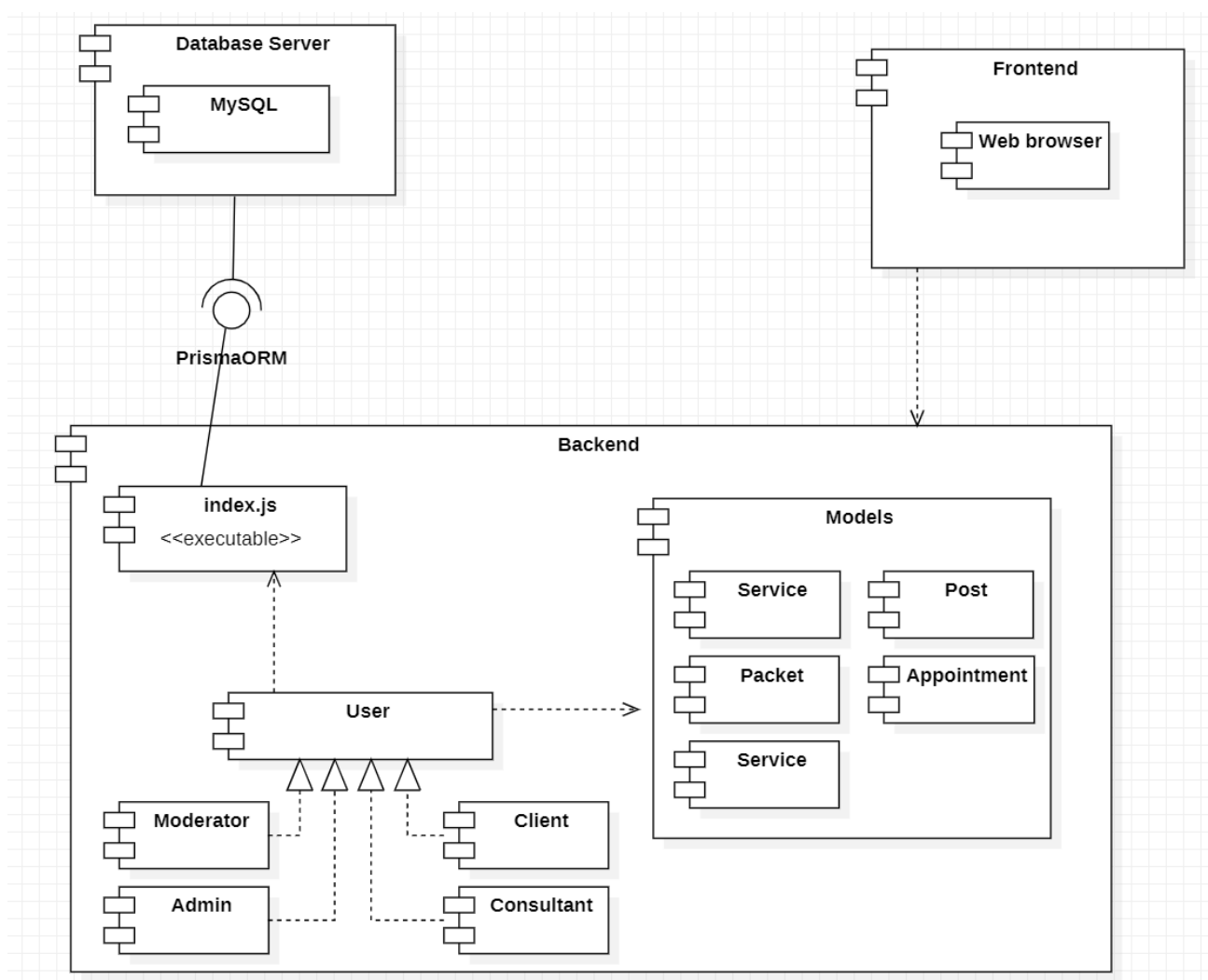


Рис. 3.1. Діаграма компонентів

Опис діаграми компонентів:

- Компоненти – фізично існуюча частина системи, яка забезпечує реалізацію класів і відносин, а також функціональної поведінки модельованої програмної системи.
- На цій діаграмі це:
 - Models – компоненти, які є модулями в програмному коді:
 - Залежності
 - Відношення залежності
 - Database Server – локальний сервер бази даних
 - База даних MySQL
 - PrismaORM – ORM для зв'язку між базою даних та веб-застосунком
 - Frontend – пристрій користувач з якого він може користуватись додатком:
 - Веб-браузер

3.2 Архітектура програмного комплексу та його розгортання

Діаграма розгортання – це діаграма на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонентів. Діаграма розгортання відображає робочі екземпляри компонентів, а діаграма компонентів, натомість, відображає зв'язки між типами компонентів.

Діаграма розгортання в UML моделює фізичне розгортання артефактів на вузлах. Вузли представляються, як прямокутні паралелепіпеди з артефактами, розташованими в них, зображеними у вигляді прямокутників. Вузли можуть мати підвузли, які представляються, як вкладені прямокутні паралелепіпеди. Один вузол діаграми розгортання може концептуально представляти безліч фізичних вузлів, таких, як кластер серверів баз даних.

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

Діаграма розгортання (Рисунок 3.2) має наступну структуру:

Вузли:

- Client – пристрій користувача
- Web Server – сервер, що обробляє запити користувачів

MySQL Server – сервер бази даних

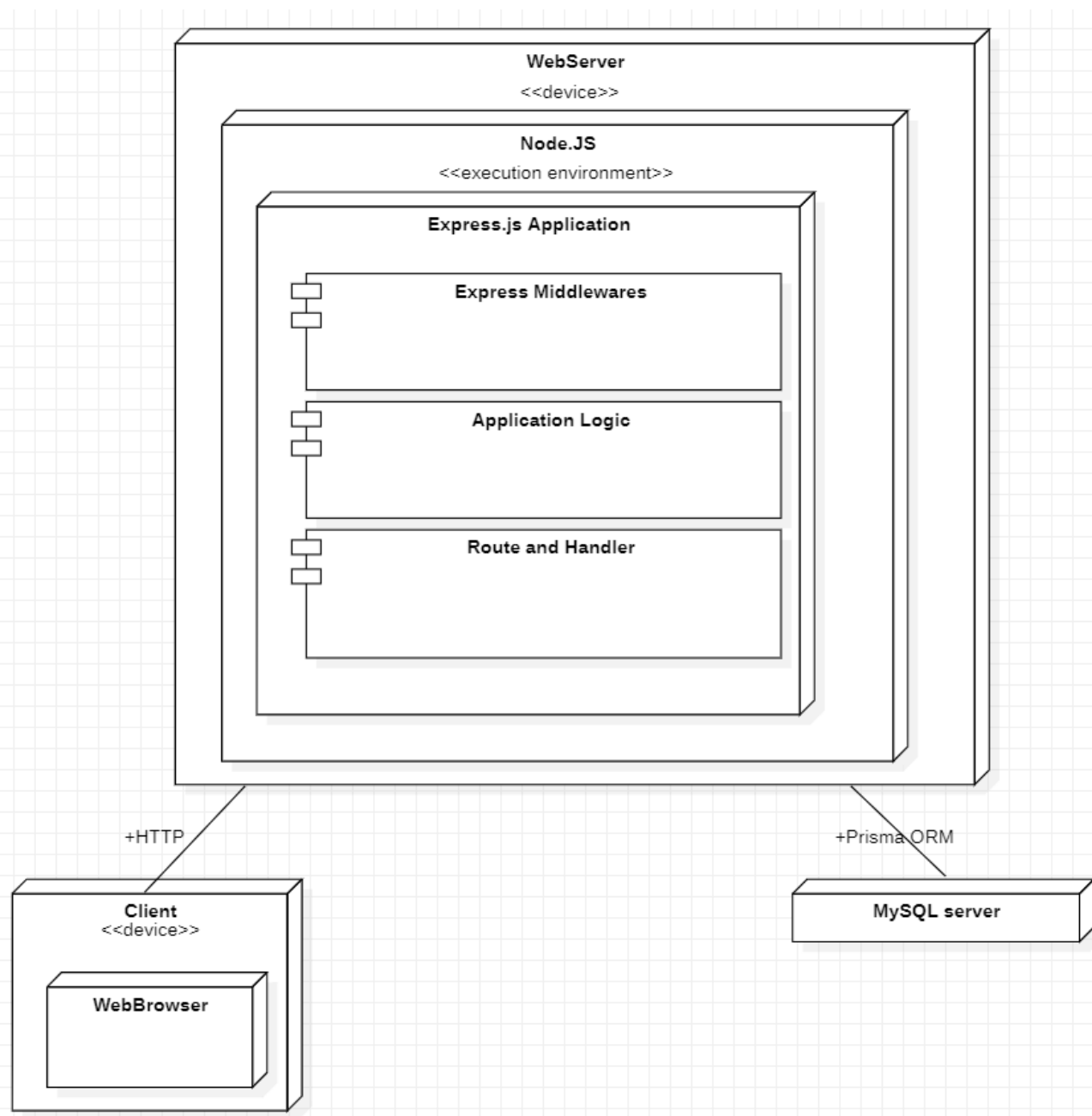


Рис. 3.2. Діаграма розгортання

3.3 Генерування програмного коду для прототипу програми

Під час вибору засобів моделювання було акцентовано на наявності кодогенерації. Зазначимо, що кодогенерація – це процес автоматичного перетворення діаграм UML у код програми.

Для генерації коду необхідна наявність діаграми класів та розширення для необхідної мови програмування, в нашому випадку JavaScript.

Для генерації JavaScript у StarUML необхідно виконати такий порядок дій:

1. Відкрити файл з створеною раніше діаграмою класів;
2. Перейти у StarUML – Tools – Extension Manager та встановити розширення JavaScript;
3. Презавантажити StarUML;
4. Перейти у StarUML – Tools – JavaScript – Generate code та обрати класи, для яких потрібно згенерувати код;
5. Згенерований код з'явиться у обраній вами папці, яку запропонує обрати редактор.

Згенеровані класи спроектованої системи зображені на рисунку 3.3, а лістинг коду кожного класу наведено в додатку Б.

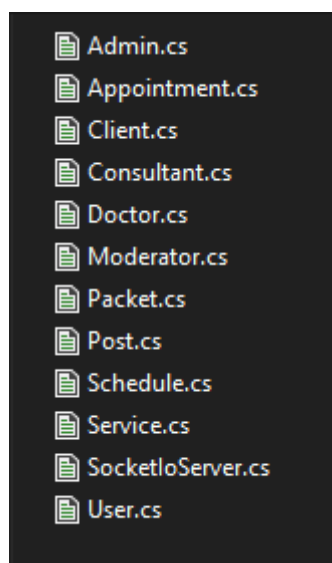


Рисунок 3.3 – Файли згенерованих класів. JavaScript не підтримується StarUML, тому довелося генерувати мовою C#

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Отже, під час виконання даної курсової роботи було закріплено пройдений матеріал, який вивчався протягом курсу, а саме: моделювання та аналіз програмного забезпечення за допомогою мови UML. Виконання курсового проекту було розділено на 3 етапи.

У першому розділі було отримано технічне завдання на розробку системи. Було обрано засіб моделювання для вирішення поставленого завдання. Також, було виокремлено та проаналізовано вимоги до програмного продукту.

У другому розділі було розроблено алгоритм роботи програми. Були розглянуті та побудовані діаграма діяльності, діаграма класів. Імплементовано патерни проектування. Також, було спроектовано діаграму послідовності для зображення взаємодії об'єктів системи.

У третьому розділі було створено фізичну модель та прототип програмного комплексу. Було продемонстровано взаємодію компонентів системи за допомогою побудови діаграми компонентів. Було обґрунтовано вибір архітектури програмного комплексу. Також, було побудовано діаграму розгортання. І було згенеровано програмний код з діаграми класів

Отже, виконана робота показала, що за допомогою вивченого матеріалу за курс можливо успішно моделювати програмні комплекси, використовуючи мову UML.

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ

ДЖЕРЕЛА ІНФОРМАЦІЇ

1. StarUML documentation: Introduction. URL: <https://docs.staruml.io>.
2. UML для бізнес-моделювання: для чого потрібні діаграми процесів [Електронний ресурс] – <https://evergreens.com.ua/ua/articles/uml-diagrams.html>.
3. Графічний редактор DRAW.IO [Електронний ресурс] – <https://app.diagrams.net/>.
4. Use Case Diagram [Електронний ресурс] – <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>.
5. Activity Diagram [Електронний ресурс] – <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>.
6. Class Diagram [Електронний ресурс] – <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>.
7. Factory Pattern [Електронний ресурс] <https://refactoring.guru/uk/design-patterns/factory-method>
8. Sequence Diagram [Електронний ресурс] – <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>.
9. Deployment Diagram [Електронний ресурс] – <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/>.
10. Unified Modeling Language [Електронний ресурс] – https://uk.wikipedia.org/wiki/Unified_Modeling_Language.

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

- | | | | | |
|-----|---|---------------|--------------|---------|
| 11. | Express.js | Documentation | [Електронний | ресурс] |
| | https://expressjs.com/en/guide/routing.html | | | |
| 12. | PrismaORM | Documentation | [Електронний | ресурс] |
| | https://www.prisma.io/docs | | | |
| 13. | MySQL | Documentation | [Електронний | ресурс] |
| | https://dev.mysql.com/doc/ | | | |
| 14. | JavaScript | Documentation | [Електронний | ресурс] |
| | https://developer.mozilla.org/en-US/docs/Web/JavaScript | | | |
| 15. | React.js | Documention | [Електронний | ресурс] |
| | https://reactjs.org/docs/getting-started.html | | | |

ДОДАТКИ

		Гопчук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Admin.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Admin : User {

    public Admin() {
    }

    public void setRole() {
        // TODO implement here
    }

}
```

Appointment.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Appointment {

    public Appointment() {
    }

    public int id;

    public datetime date;

    public int doctorId;

    public int serviceId;

    public void deleteAppointment() {
        // TODO implement here
    }

    public void getAppointment() {
        // TODO implement here
    }

    public void createAppointment() {
        // TODO implement here
    }

}
```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public void updateAppointment() {
        // TODO implement here
    }

    public void getAppointmentByClient() {
        // TODO implement here
    }

    public void getAppointmentByDoctor() {
        // TODO implement here
    }
}

```

Client.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Client : User {

    public Client() {
    }

    public int id;

    public string name;

    public string surname;

    public string phone;

    public void getClient() {
        // TODO implement here
    }

    public void deleteClient() {
        // TODO implement here
    }

    public void updateClient() {
        // TODO implement here
    }
}

```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Consultant.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Consultant : User {

    public Consultant() {
    }

    public int id;

    public string name;

    public string surname;

    public string patronymic;

    public string phone;

    public string email;

    public void getConsultant() {
        // TODO implement here
    }

    public void getConsultant() {
        // TODO implement here
    }

    public void deleteConsultant() {
        // TODO implement here
    }

    public void updateConsultant() {
        // TODO implement here
    }

    public void Operation1() {
        // TODO implement here
    }

}
```

Doctor.cs

```
using System;
using System.Collections.Generic;
```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System.Linq;
using System.Text;

public class Doctor : User {

    public Doctor() {
    }

    public int id;

    public string name;

    public string surname;

    public string patronymic;

    public <array> specialization;

    public int experience;

    public string phone;

    public string email;

    public string photo;

    public void getDoctor() {
        // TODO implement here
    }

    public void getDoctors() {
        // TODO implement here
    }

    public void deleteDoctor() {
        // TODO implement here
    }

    public void updateDoctor() {
        // TODO implement here
    }

    public void getDoctoryBy() {
        // TODO implement here
    }

    public void getAppointment() {
        // TODO implement here
    }
}

```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public void Operation2() {
        // TODO implement here
    }

    public void Operation3() {
        // TODO implement here
    }

    public void Operation4() {
        // TODO implement here
    }

    public void Operation6() {
        // TODO implement here
    }
}

```

Moderator.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Moderator : User {

    public Moderator() {
    }

    public void Attribute1;

    public void getModeratorByPost() {
        // TODO implement here
    }
}

```

Packet.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Packet {

    public Packet() {
    }
}

```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

```
public int id;

public string name;

public float price;

public void Attribute3;

public string description;

public void getPacket() {
    // TODO implement here
}

public void createPacket() {
    // TODO implement here
}

public void deletePacket() {
    // TODO implement here
}

public void updatePacket() {
    // TODO implement here
}

public void calculatePrice() {
    // TODO implement here
}

}
```

Post.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Post {

    public Post() {
    }

    public int id;

    public string title;

    public string body;

}
```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public void createPost() {
    // TODO implement here
}

public void getPost() {
    // TODO implement here
}

public void getPosts() {
    // TODO implement here
}

public void updatePost() {
    // TODO implement here
}

public void deletePost() {
    // TODO implement here
}

public void Operation1() {
    // TODO implement here
}
}

```

Schedule.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Schedule {

    public Schedule() {
    }

    public int id;

    public int doctorId;

    public void getSchedule() {
        // TODO implement here
    }

    public void createSchedule() {
        // TODO implement here
    }
}

```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				32
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        public void Operation1() {
            // TODO implement here
        }
    }
}

```

Service.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class Service {

    public Service() {
    }

    public int id;

    public string name;

    public float price;

    public void Attribute4;

    public void Attribute5;

    public string description;

    public void getService() {
        // TODO implement here
    }

    public void createService() {
        // TODO implement here
    }

    public void deleteService() {
        // TODO implement here
    }

    public void updateService() {
        // TODO implement here
    }

    public void getServicesByPacket() {
        // TODO implement here
    }

}

```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

User.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class User {

    public User() {
    }

    public int id;

    public string email;

    public string login;

    public string role;

    public string password;

    public void Attribute5;

    public void Attribute6;

    public void getUser() {
        // TODO implement here
    }

    public void createUser() {
        // TODO implement here
    }

    public void login() {
        // TODO implement here
    }

    public void logout() {
        // TODO implement here
    }

    public void register() {
        // TODO implement here
    }

    public void getCurrentUser() {
        // TODO implement here
    }
}
```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public void isAdmin() {
        // TODO implement here
    }

    public void isDoctor() {
        // TODO implement here
    }

    public void isConsultant() {
        // TODO implement here
    }

    public void isModerator() {
        // TODO implement here
    }

    public void deleteUser() {
        // TODO implement here
    }

    public void updateUser() {
        // TODO implement here
    }
}

```

SocketIoServer.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

public class SocketIoServer {

    public SocketIoServer() {
    }

}

```

		Гончук О. Я.			ДУ «Житомирська політехніка».22.121.03.000 - ПЗ	Арк.
		Сугоняк І.І.				35
Змн.	Арк.	№ докум.	Підпис	Дата		