

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка інформаційної системи обліку клієнтів стоматологічної клініки»

Студента 4 курсу, 7 групи,
напряму підготовки 6.050103
«Програмна інженерія»

підпис студента

Цьомка Олександра
Олександровича

Науковий керівник
кандидат технічних наук,
доцент

підпис керівника

Рассамакін
Володимир Якович

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ПІДГОТОВКА ДО ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	5
1.1 Організація роботи стоматологічної клініки	5
1.2 Технічне завдання на розробку інформаційної системи обліку клієнтів стоматологічної клініки	5
1.3 Висновки до розділу 1	8
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІС ОБЛІКУ КЛІЄНТІВ СТОМАТОЛОГІЧНОЇ КЛІНІКИ.....	9
2.1 Архітектура REST	9
2.2 Проектування бази даних	10
2.3 Проектування серверної частини ІС з використанням Spring Boot.....	12
2.4 Проектування клієнтської частини ІС з використанням Vue	16
2.5 Висновки до розділу 2.....	19
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІС ОБЛІКУ КЛІЄНТІВ СТОМАТОЛОГІЧНОЇ КЛІНІКИ	20
3.1 Побудова інтерфейсу ІС.....	20
3.2 Демонстрація роботи розробленої ІС.....	22
3.3 Висновки до розділу 3.....	34
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	36

					<i>КНТЕУ 6.050103 07-27.БР</i>			
Зм.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.		Криворучко О.В.			Розробка інформаційної системи обліку клієнтів стоматологічної клініки	Стадія	Аркуш	Аркушів
Керівник		Рассамакін В.Я.				Зміст	2	36
Гарант		Цензура М.О.			Зміст	Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив		Цьомка О.О.						

ВСТУП

Сьогодні сучасні технології - необхідне доповнення до майстерності стоматолога, що допомагає досягти найвищої якості робіт.

Кожен лікар стикається з проблемами ведення обліку в своїй клініці: потрібно записувати клієнтів, не забувати про розпорядок прийому клієнтів і вчасно повідомляти їх про ту чи іншу подію.

З кожним роком роботи потік пацієнтів в стоматологічній клініці збільшується. Настав момент, коли стало складно впізнавати кожного, намагатися згадати, що лікували, коли, який план лікування затвердили, які рахунки оплатили та інше. Для цього потрібно було б підняти журнал у адміністратора, заглянути в карту пацієнта, знайти рахунок за попереднє лікування. Це все займає багато часу. Для швидкої та злагодженої роботи співробітників важливо було систематизувати всі дані пацієнтів, створити загальну базу, де кожен працівник клініки зміг би легко знайти всю інформацію про пацієнта. Особливо це питання запису на прийом. Щоб інформація була не просто швидкою, але і «правильною». Кожен лікар хоч раз стикався з такою проблемою, коли на один і той же час було записано кілька пацієнтів або адміністратор залишив мало часу для певних медичних маніпуляцій.

Перед нами постало завдання - як вирішити всі питання, що накопичилися, як впоратися з усім: безліччю даних, потоком первинних пацієнтів, планами лікування, оплатою.

Мета: аналіз діяльності стоматологічної клініки та розробка інформаційної системи обліку клієнтів.

Об'єкт дослідження: організація обслуговування клієнтів.

Предмет дослідження: розробка інформаційної системи обліку клієнтів стоматологічної клініки

					КНТЕУ 6.050103 07-27.БР			
Зм.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.	Криворучко О.В.				Розробка інформаційної системи обліку клієнтів стоматологічної клініки	Стадія	Аркуш	Аркушів
Керівник	Рассамакін В.Я.					В	3	36
Гарант	Цензура М.О				Вступ	Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив	Цьомка О.О.							

Завдання:

- визначення діяльності стоматологічної клініки;
- створення технічного завдання на розробку ІС;
- розробка БД для ІС;
- розробка серверної частини мовою програмування Java (створення REST API);
- проектування зрозумілого та зручного для користувача інтерфейсу;
- розробка клієнтської частини за допомогою сучасного інструменту Vue.

					<i>КНТЕУ 6.050103 07-27.БР</i>	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		4

РОЗДІЛ 1. ПІДГОТОВКА ДО ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Організація роботи стоматологічної клініки

Стоматологічна клініка надає стоматологічні послуги а саме профілактику і лікування захворювань ротової порожнини. Штат стоматологічної клініки складається з директора, адміністратора, лікарів та медичного персоналу. Клініка працює за встановленим директором розкладом роботи. Лікарі приймають пацієнтів за попереднім записом. Запис та час прийомів контролює адміністратор. До початку прийому новий пацієнт заповнює медичну картку. Під час прийому лікар проводить консультацію, складає план лікування та за згодою пацієнта надає йому певні стоматологічні послуги. Після закінчення лікування та оплати наданих послуг прийом вважається завершеним. При необхідності лікар може призначити ще декілька прийомів узгодивши розклад з адміністратором в зручний для пацієнта час.

1.2 Технічне завдання на розробку інформаційної системи обліку клієнтів стоматологічної клініки

Розробка інформаційної системи передбачає наступне технічне завдання:

1) Загальні відомості.

Назва інформаційної системи: DentalClinic.

Терміни початку та кінця робіт: Листопад 2018 – Квітень 2019.

					КНТЕУ 6.050103 07-27.БР			
Зм.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.	Криворучко О.В.				Розробка інформаційної системи обліку клієнтів стоматологічної клініки	Стадія	Аркуш	Аркушів
Керівник	Рассамакін В.Я.					Р1	5	36
Гарант	Цензура М.О				Розділ 1 Підготовка до проектування інформаційної системи	Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив	Цьомка О.О.							

Потенційні користувачі:

- Індивідуальні лікарі-стоматологи
- Приватні стоматологічні клініки
- Державні стоматологічні заклади

Мета: організація роботи з клієнтами стоматологічної клініки.

Призначення: поліпшення якості обслуговування клієнтів.

2) Функціональні вимоги.

Інформаційна система повинна надати користувачу:

- Роботу з клієнтами клініки, а саме: збереження їх імен, контактних даних та віку; редагування даних клієнтів.
- Можливість створення запису прийому клієнта в певний час календарного дня та його редагування для кожного з чотирьох кабінетів.
- Роботу з поділенням на розділи прејскурантом за видом стоматологічних послуг: створення нових, редагування та видалення існуючих пунктів.
- Роботу зі штатом лікарів: створення та редагування даних; можливість виключення зі штату клініки.

Інші функціональні вимоги:

- ІС повинна мати візуалізацію розпорядку календарного дня для кожного кабінету.
- Візуалізація розпорядку повинна мати: швидкий доступ до контактних даних клієнта, списку послуг; можливість виклику форм редагування запису та списку послуг по ньому.
- Записи повинні мати наступні статуси: «Виконаний», «Заплановано», «Не додзвонилися», «Не з'явився» та відрізнятися кольором один від одного в візуалізації розпорядку.
- При створенні нового або редагуванні існуючого запису потрібно створити перевірку наявності конфліктів у часі з вже існуючими

					КНТЕУ 6.050103 07-27.БР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		6

записами, та оповістити про це користувача. Також виключити можливість створення конфліктуючого по часу запису.

- При створенні та редагуванні записів повинен бути реалізований пошук по прізвищу клієнта та вибір лікаря зі штату клініки.
- При редагуванні існуючого пункту прейскуранту, його зміни не повинні торкнутися списку послуг існуючих прийомів.
- При виключенні лікаря зі штату, закріплені за ним дані в існуючих записах повинні зберегтися.
- Реєстр клієнтів повинен: відображати 20 останніх зареєстрованих клієнтів; мати сортування по прізвищу; пошук по прізвищу або його фрагменту; розрахувати та відображати вік клієнтів
- Пункт прейскуранту складається з: коду, назви, ціни.
- Код пункту прейскуранту повинен містити три цифри. Перша цифра відповідає номеру розділу, дві інші обираються користувачем.
- ІС повинна фіксувати дату та час зміни або створення пункту прейскуранту.
- Реєстр штату клініки повинен складатися з лікарів які знаходяться у штаті в даний момент та з тих, яких було виключено.
- Усі поля форм повинні перевірятися в залежності від типу.

					КНТЕУ 6.050103 07-27.БР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		7

3) Вимоги до апаратного забезпечення.

Таблиця 1.1.

Назва	Клієнт	Сервер
Центральний процесор	2 GHz+	2 x 2 GHz+
Платформа	x86-64, ARM	x86-64
Оперативна пам'ять	2 Gb+	4 Gb+
Вільне місце на диску	350 Mb	1 Gb
Мережева карта	10/100/1000 Mbps	100/1000 Mbps

4) Вимоги до програмного забезпечення.

На серверному обладнанні повинно бути попередньо встановлено наступне ПЗ:

- Операційна система Windows 7+ або Ubuntu 10.04+
- JRE 1.8 202
- MySQL 8.0

Клієнтське обладнання повинне мати операційну систему та один із сучасних браузерів: Google Chrome, Mozilla Firefox, Brave.

1.3 Висновки до розділу 1

Проаналізовано організацію роботи стоматологічної клініки. Розроблено технічне завдання на створення інформаційної системи обліку клієнтів стоматологічної клініки за допомогою якого визначено основні вимоги до ІС, призначення та ціль ІС, апаратні та вимоги до ПЗ окремо для серверної та клієнтської частини.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІС ОБЛІКУ КЛІЄНТІВ СТОМАТОЛОГІЧНОЇ КЛІНІКИ

2.1 Архітектура REST

REST (англ. Representational State Transfer - «передача стану уявлення») - архітектурний стиль взаємодії компонентів розподіленого додатка в мережі.

В мережі Інтернет виклик віддаленої процедури може являти собою звичайний HTTP-запит (зазвичай «GET» або «POST»; такий запит називають «REST-запит»), а необхідні дані передаються в якості параметрів запиту (Рис 2.1 Діаграма архітектури REST додатку).

Одна транзакція RESTful сервісу складається як мінімум з:

- методу запиту (GET)
- шляху запиту (/api/entry)
- тіла запиту (params)
- коду відповіді (200 OK)
- тіла відповіді (JSON)

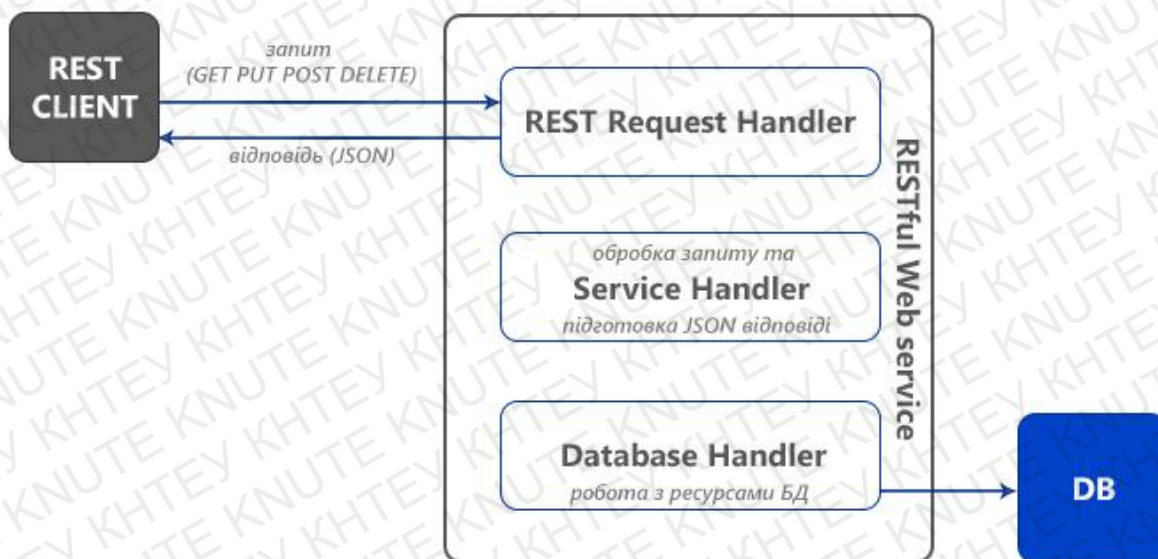


Рис 2.1 Діаграма архітектури REST додатку

					КНТЕУ 6.050103 07-27.БР			
Зм.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.	Криворучко О.В.				Розробка інформаційної системи обліку клієнтів стоматологічної клініки	Стадія	Аркуш	Аркушів
Керівник	Рассамакін В.Я.					Р2	9	36
Гарант	Цензура М.О				Розділ 2 Проектування ІС обліку клієнтів стоматологічної клініки	Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив	Цьомка О.О.							

Розроблена ІС використовує наступні методи запиту:

- GET – отримання даних об'єкту
- POST – створення об'єкту з переданих даних
- PUT – оновлення існуючого об'єкту
- DELETE – видалення існуючого об'єкту

Для тестування та створення колекцій веб сервісу використовується ПЗ під назвою Postman (Рис 2.2 Приклад GET запиту в Postman). Будь-який розробник або тестувальник відкривши колекцію зможе з легкістю розібратися в роботі сервісу. Тестувальники можуть писати тести і проводити автоматизоване тестування прямо з Postman. А інструменти для автоматичного документування за описами з створених колекцій заощадять час.

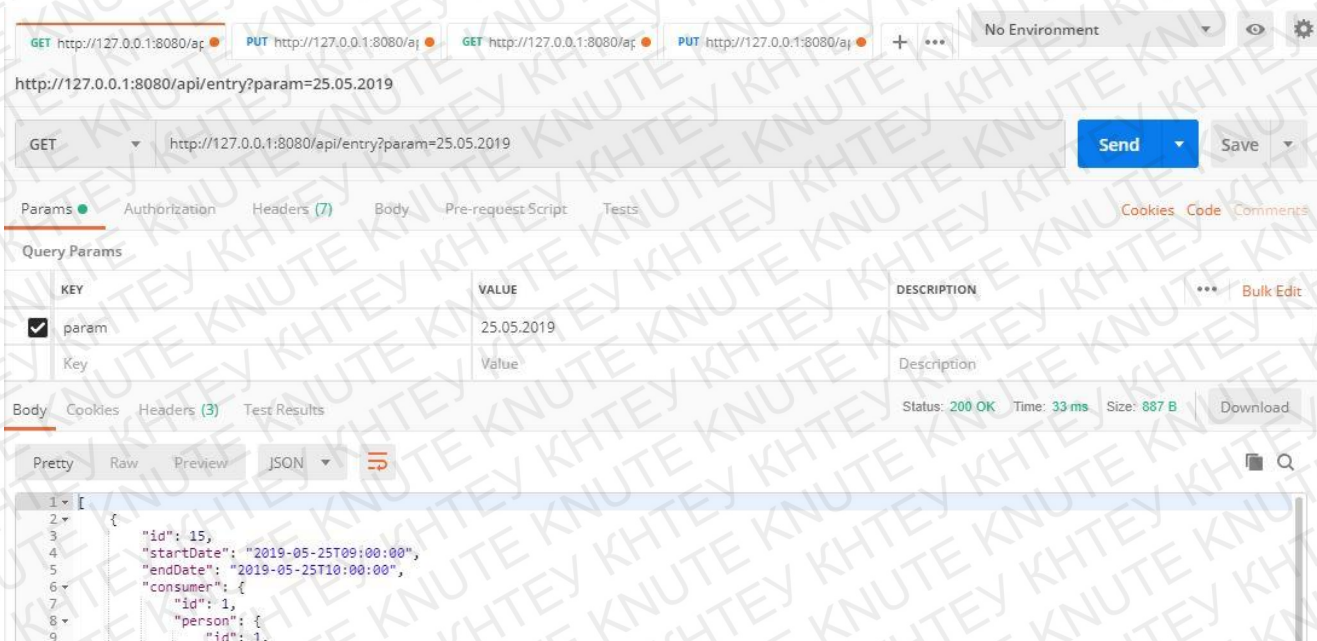


Рис 2.2 Приклад GET запиту в Postman

2.2 Проектування бази даних

Для розробки ІС було обрано СУБД MySQL. Всього створено десять таблиць. Чотири таблиці: «consumer», «doc», «person», «city» використовуються для зберігання даних клієнтів та лікарів. Дані преїскуранту зберігаються в двох таблицях: «catalog», «catalog_type». Для роботи з записом клієнтів на прийом – чотири таблиці: «entry», «entry_state», «cabinet», «service».

					КНТЕУ 6.050103 07-27.БР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		10

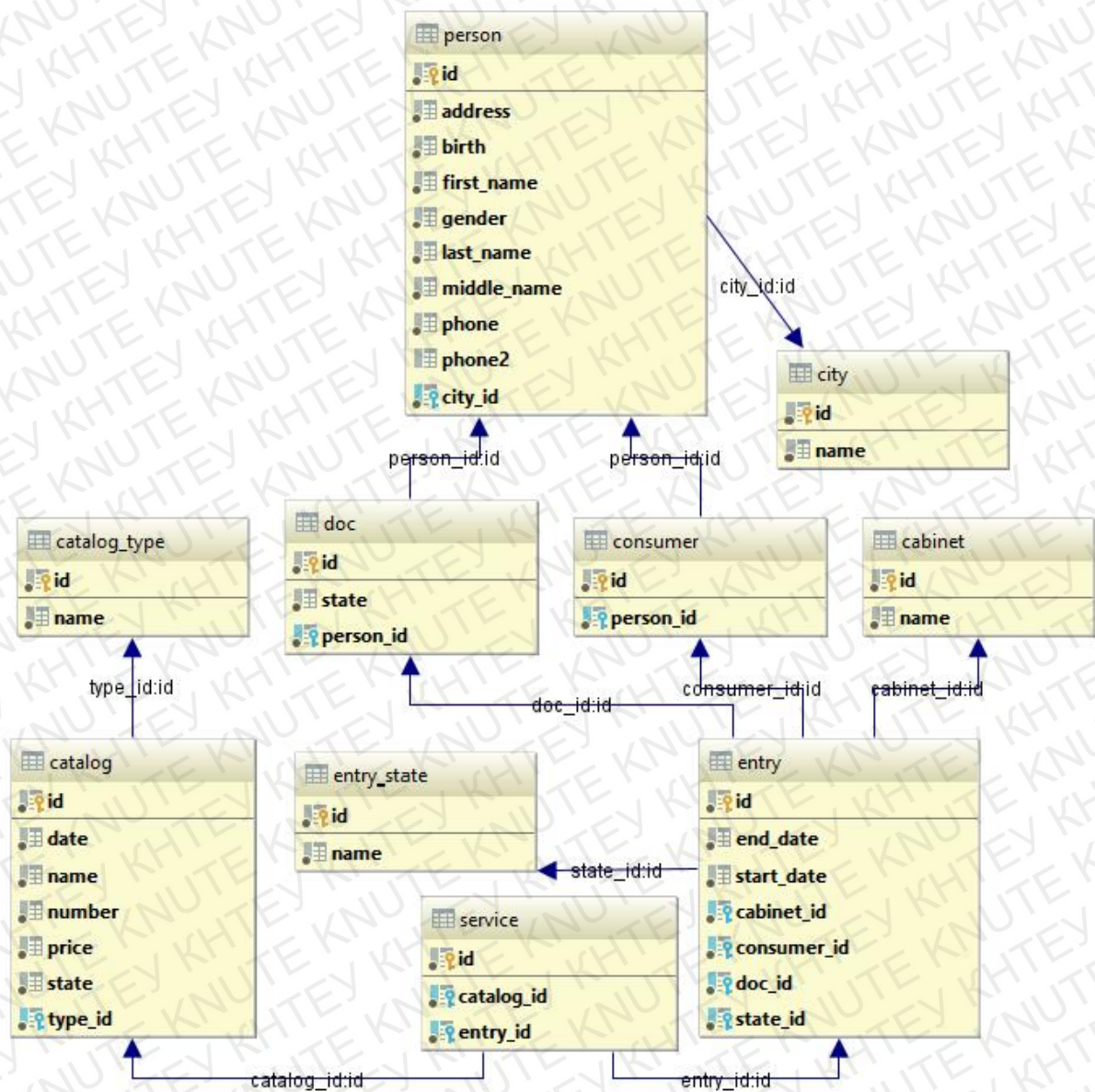


Рис 2.3 Логічна модель БД

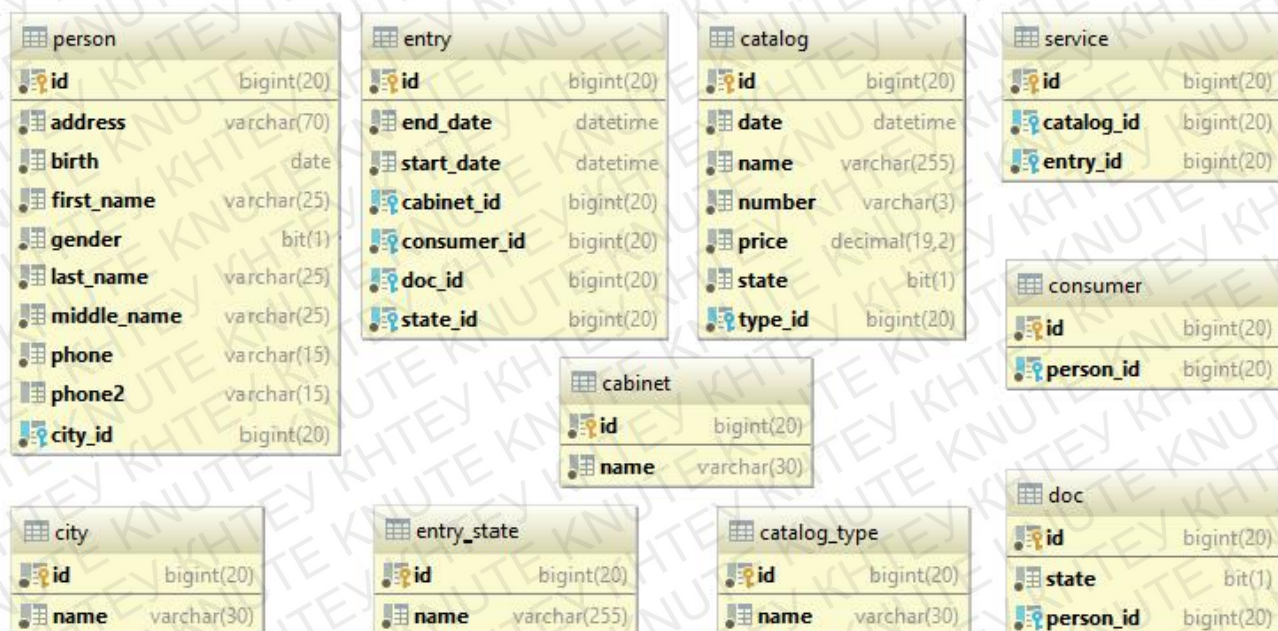


Рис 2.4 Фізична модель БД

2.3 Проектування серверної частини ІС з використанням Spring Boot

Серверну частину вирішено розробляти на мові програмування Java разом з Spring Boot та системою залежностей Maven. Spring Boot - платформа для розробки веб сервісів на основі Spring Framework з мінімальною потребою в конфігурації. Вона забезпечує вбудований HTTP сервер (*Tomcat, Jetty*). Spring Boot автоматично налаштовує додаток на основі залежностей, доданих до проекту за допомогою анотації *@EnableAutoConfiguration*. Точкою входу додатку є клас, що містить анотацію *@SpringBootApplication* і основний метод.

Головна перевага Maven - це управління залежностями. Рідко які проекти пишуться без використання сторонніх бібліотек. Ці сторонні бібліотеки теж в свою чергу використовують бібліотеки різних версій. Maven дозволяє керувати такими складними залежностями, що дозволяє вирішувати конфлікти версій і в разі потреби легко переходити на нові версії бібліотек. Залежності підключаються до проекту у файлі «*pom.xml*» (Рис 2.5 Залежності розроблюваної ІС).

```

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.15</version>
  </dependency>
</dependencies>

```

Рис 2.5 Залежності розроблюваної ІС

Структура проекту складається з пакетів що містять класи, конфігураційного файлу Spring, файлу залежностей Maven та пакету статичних ресурсів, що містить клієнтську частину (Рис 2.6 Структура проекту).

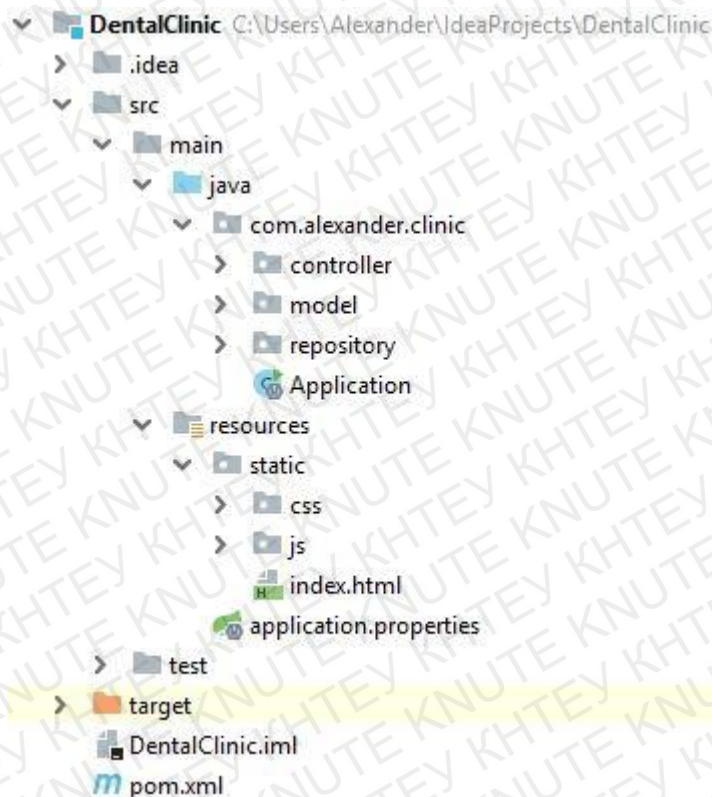


Рис 2.6 Структура проекту

Конфігураційний файл Spring (*application.properties*) використовується для налаштування підключення до бази даних, в якому містяться: адреса з параметрами часового поясу, логін та пароль (Рис 2.7 Конфігурація підключення до БД).

```

spring.datasource.url=jdbc:mysql://localhost:3306/demo?useUnicode=true&useJDBCCompliantTimezoneShift=
true&useLegacyDatetimeCode=false&serverTimezone=Europe/Kiev
spring.datasource.username=root
spring.datasource.password=12345678

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect

# Hibernate ddl auto (create, create-drop, validate, update)
spring.jpa.hibernate.ddl-auto = update

logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type=TRACE

```

Рис 2.7 Конфігурація підключення до БД

JpaRepository передбачає наявність сутностей (Рис 2.8 Фрагмент сутності «Entry») у виді класів які використовуються для роботи з таблицею БД. Сутність містить конструктори, геттери та сеттери. Використовуються наступні анотації:

- `@Entity` – оголошення сутності
- `@Table(name = "entry")` – дозволяє уточнити деталі таблиці
- `@Column` – оголошення стовпця
- `@Id` – ідентифікує стовпець первинним ключем
- `@GeneratedValue(strategy = GenerationType.IDENTITY)` – конфігурує спосіб інкрементації ключа
- `@NotNull` – не дозволяє приймати значення NULL
- `@Size(max = 70)` – оголошує максимальну кількість символів стовпця

```

@Entity
@Table(name = "entry")
public class Entry {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    ...
}

```

Рис 2.8 Фрагмент сутності «Entry»

Щоб додати до таблиці зовнішні ключі використовуються спеціальні анотації (Рис 2.9 Оголошення зовнішнього ключа): `@ManyToOne`, `@JoinColumn`, `@OnDelete`.

```

@ManyToOne(fetch = FetchType.EAGER, optional = false)
@JoinColumn(name = "consumer_id", nullable = false)
@OnDelete(action = OnDeleteAction.NO_ACTION)
private Consumer consumer;

```

Рис 2.9 Оголошення зовнішнього ключа

Окрім сутностей для них створено інтерфейси репозиторіїв. JpaRepository - це інтерфейс фреймворка Spring Data що надає набір стандартних методів JPA для роботи з БД. Він оголошується анотацією *@Repository*. При відсутності потрібних методів є можливість додати свій. За допомогою анотації *@Query* до створеного методу вказано потрібний запит до БД (Рис 2.10 Метод для перевірки конфліктів запису прийому).

```

@Query("select e from Entry e " +
      "where e.cabinet.id = :cabinet " +
      "and NOT (e.id = :id) " +
      "and ((:startTime >= e.startDate and :startTime < e.endDate) " +
      "or (:endTime > e.startDate and :endTime <= e.endDate) " +
      "or (:startTime <= e.startDate and :endTime > e.startDate) " +
      "or (:startTime < e.endDate and :endTime >= e.endDate))")
List<Entry> findBothBetweenDate(@Param("startTime") LocalDateTime startTime,
                              @Param("endTime") LocalDateTime endTime,
                              @Param("cabinet") long cabinet,
                              @Param("id") long id);

```

Рис 2.10 Метод для перевірки конфліктів запису прийому

Контролери визначають адресу та методи передачі даних (Рис 2.11 Контролер створення нового прийому). Для оголошення класу контролером існує спеціальна анотація *@RestController*. Анотація *@Autowired* дозволяє підключити потрібний репозиторій оголосивши його. API веб сервісу має наступну колекцію:

- /api/worktime (GET) – час роботи клініки
- /api/cabinet (GET) – список кабінетів
- /api/catalog (GET) – актуальні пункти прейскуранту
- /api/catalog/section (GET) – актуальні пункти по розділу прейскуранту
- /api/catalog/section/list (GET) – список розділів прейскуранту
- /api/catalog/suspend (PUT) – видалення пункту прейскуранту зі збереженням даних для існуючих прийомів
- /api/catalog/new (POST) – створення нового пункту прейскуранту
- /api/cities (GET) – список міст України

					КНТЕУ 6.050103 07-27.БР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		15

- `/api/consumerbysurname (GET)` – дані потрібного клієнта
- `/api/consumers (GET)` – дані останніх 20-ти зареєстрованих клієнтів
- `/api/consumers/new (POST)` – створення нового клієнту
- `/api/consumers/edit (PUT)` – редагування даних клієнта
- `/api/workingdocs (GET)` – список лікарів у штаті клініки
- `/api/docs/new (POST)` – створення нового лікаря
- `/api/docs/edit (PUT)` – редагування даних лікаря
- `/api/entrystates (GET)` – список статусів прийомів
- `/api/entry (GET)` – список прийомів потрібного календарного дня
- `/api/entry/new (POST)` – створення прийому
- `/api/entry/edit (PUT)` – редагування прийому
- `/api/service (GET)` – список наданих послуг існуючого прийому
- `/api/service/new (POST)` – створення списку наданих послуг прийому
- `/api/service/remove (DELETE)` – видалення списку наданих послуг прийому

```
@PostMapping("/entry/new")
public List<Entry> createEntry(@RequestParam String date,
                               @RequestParam String startTime,
                               @RequestParam String endTime,
                               @RequestParam String consumer,
                               @RequestParam String doc,
                               @RequestParam String cabinet,
                               @RequestParam String state) {
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm:ss");
    LocalDateTime sd = LocalDateTime.parse(text: date + " " + startTime + ":00", formatter);
    LocalDateTime ed = LocalDateTime.parse(text: date + " " + endTime + ":00", formatter);
    List<Entry> check = entryRepository.findBothBetweenDateWithoutId(sd, ed, Long.parseLong(cabinet));
    ...
}
```

Рис 2.11 Контролер створення нового прийому

2.4 Проектування клієнтської частини ІС з використанням Vue

Користувачами ІС є адміністратори клініки. Клієнтська частина являє собою веб додаток, реалізований за допомогою CSS, HTML, JS. Порівняння найпопулярніших JS фреймворків наведено у таблиці 2.1.

					КНТЕУ 6.050103 07-27.БР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		16

Таблиця 2.1. Порівняння JavaScript фреймворків

	Angular	React	Vue
Засновники	Google	Facebook	Колишній працівник Google
Перший випуск	Вересень 2016	Березень 2013	Лютий 2014
Підходить для	Широкомасштабні, багатофункціональні додатки	Сучасні веб розробки, native додатки під Android та iOS	Веб розробки та односторінкові додатки
Популярність	Широко поширений серед розробників	Понад 27 тисяч зірок додано за рік	Більше 40 тисяч зірок на GitHub протягом року
Компанії, які використовують	Google, Forbes, Wix та weather.com	Facebook, Uber, Netflix, Twitter, Reddit, Paypal, Walmart та інші	Alibaba, Baidu, GitLab та інші

Вирішено обрати фреймворк Vue для реалізації клієнтської частини тому, що:

- це найкращий вибір для односторінкових додатків;
- надає даним реактивності;
- набирає популярність серед розробників;
- простіший для освоєння на перших етапах фронтенд розробки.

React дуже схожий по перевагам, але показує трохи гірші результати у тестах на швидкодію.

Об'єктна модель документа (DOM) - це програмний інтерфейс (API) для HTML документів. DOM повністю підтримує об'єктно-орієнтоване уявлення веб-сторінки, роблячи можливим її зміну за допомогою мови JavaScript.

Всі шаблони Vue є дійсним HTML-кодом (Рис 2.12 Фрагмент шаблону Vue). Для роботи Vue компілює шаблони в `render`-функції віртуального DOM. У поєднанні з системою реактивності, Vue вміє визначати мінімальне число компонентів для повторного рендеру і застосовує мінімальну кількість маніпуляцій до DOM при зміні стану програми.

Щоб оголосити межі дії екземпляру на HTML сторінці треба вказати в його властивості «`el`» назву `id` існуючого DOM елементу.

```
<el-table :data="consumers_table" style="width: 100%">
  <el-table-column type="expand">
    <template slot-scope="props">
      <p>Дата народження: {{ props.row.birth }}</p>
      <p>Стать: {{ props.row.gender }}</p>
      <p>Адреса: {{ props.row.address }}</p>
      <p>Номер телефону: {{ props.row.phone }}</p>
    </template>
  </el-table-column>
  <el-table-column sortable label="ПІБ" prop="fullname">
  </el-table-column>
```

Рис 2.12 Фрагмент шаблону Vue

Для виводу змінних використовується `mustache` синтаксис: `{{ props.row.birth }}`. Двокрапка перед DOM атрибутом означає, що він реактивний та буде завантажений зі змінної екземпляру, у цьому випадку фігурні дужки не потрібні.

Змінні або масиви які потребують реактивності потрібно представити у виді об'єктів та розмістити в спеціальній властивості «`data`» потрібного екземпляру. Vue автоматично створить для них усі потрібні геттери та сеттери. Усі дані, які знаходяться поза властивістю «`data`», не можуть бути реактивними. Реактивність дозволяє відслідковувати зміни даних та створювати потрібні при змінах методи. Для використання функцій їх потрібно розмістити у властивості «`methods`».

Так як односторінковий додаток передбачає роботу з даними без потреби перезавантажувати сторінку потрібно якимсь чином додати йому динамічності. Для реалізації асинхронних запитів підключено JS бібліотеку `Axios` (Рис 2.13 Приклад `axios` запиту `GET`). На відміну від метода `fetch()`, при використанні якого для передачі даних в JSON, необхідно виконати метод в два етапи: спочатку зробити фактичний запит, а потім викликати метод `json()` для перетворення отриманих даних з сервера, `Axios` робить це автоматично.

```

axios
  .get('/api/entry', {
    params: {
      param: param_date
    }
  })
  .then(response => {
    data.entries = response.data;
  });

```

Рис 2.13 Приклад axios запиту GET

2.5 Висновки до розділу 2

Було розроблено фізичну та логічну модель бази даних для ІС. Проаналізовано існуючі JS фреймворки для клієнтської частини. Для розробки ІС вирішено використовувати такі методи та інструменти як: MySQL, REST, Postman, Vue, Axios, Spring Boot, Maven. Спроектовано та розроблено серверну частину, що використовує метод передачі даних REST. За допомогою фреймворку Vue та бібліотеки Axios створено односторінковий веб-додаток, котрий виступає клієнтською частиною.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІС ОБЛІКУ КЛІЄНТІВ СТОМАТОЛОГІЧНОЇ КЛІНІКИ

3.1 Побудова інтерфейсу ІС

Інтерфейс потребує сучасних, зручних та зрозумілих для користувачів UI елементів. Було обрано бібліотеку Element, яка створена спеціально для роботи з найпопулярнішими фреймворками. Вона містить у собі всі необхідні для розробки елементи. Розроблена ІС використовує такі UI елементи (Рис 3.1 Element UI): button, icon, switch, datepicker, timepicker, input, inputnumber, form, table, notification.

The screenshot displays a user interface for a dental clinic's appointment system. It includes a date picker set to 30.05.2019, a search bar labeled 'Пошук по прізвищу', a dropdown menu for 'Загальні послуги', a list of time slots from 08:00 to 09:00, a numeric input for a fee (currently 20.00), a 'Новий прийом' button, and a 'Початок прийому' button.

Рис 3.1 Element UI

Для сповіщення успішних дій користувача або помилок використовуються спеціальні повідомлення (Рис 3.2 Повідомлення для сповіщення користувача).

					КНТЕУ 6.050103 07-27.БР			
Зм.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.	Криворучко О.В.				Розробка інформаційної системи обліку клієнтів стоматологічної клініки	Стадія	Аркуш	Аркушів
Керівник	Рассамакін В.Я.					РЗ	20	36
Гарант	Цензура М.О				Розділ 3 Програмна реалізація ІС обліку клієнтів стоматологічної клініки	Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив	Цьомка О.О.							

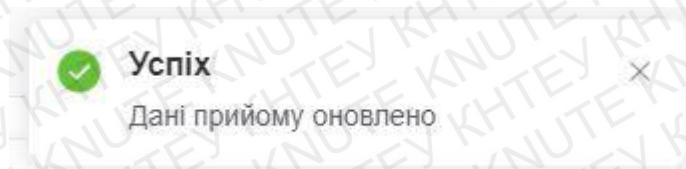


Рис 3.2 Повідомлення для сповіщення користувача

Інтерфейс складається з вертикального бокового меню та основного контейнеру. Для візуалізації розпорядку дня використовується JS бібліотека під назвою «*timetable*». Розклад спроектовано таким чином, щоб розпорядок дня перезбирався під потрібну користувачу дату, підгружаючи необхідні дані з серверу (Рис 3.3 Візуалізація розпорядку дня).

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
КАБІНЕТ 1		[09:00-10:00] Крилов Валерій									
КАБІНЕТ 2											
КАБІНЕТ 3											
КАБІНЕТ 4											

Рис 3.3 Візуалізація розпорядку дня

По горизонталі розміщені кабінети, а по вертикалі час роботи клініки, який надає сервер. Блоки запису розташовуються стосовно свого кабінету та часу, містять у собі: час запису, прізвище та ім'я клієнта. Залежно від статусу запису блоки відрізняються кольором (Рис 3.4 Візуальне відображення статусу прийому):

- Зелений (виконаний)
- Блакитний (заплановано)
- Помаранчевий (не додзвонилися)
- Сірий (не з'явився)

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
КАБІНЕТ 1	[08:00-09:00] Кудрявцева Вероніка	[09:00-10:00] Крилов Валерій									
КАБІНЕТ 2			[10:00-11:15] Шашков Павло								
КАБІНЕТ 3											
КАБІНЕТ 4						[12:15-13:45] Портнов Євгеній					

Рис 3.4 Візуальне відображення статусу прийому

Усі форми представлені у діалогових вікнах, котрі викликаються за допомогою кнопок (Рис 3.5 Форма створення нового запису).

Рис 3.5 Форма створення нового запису

Введені у поля дані перевіряються на валідність. При помилці вводу користувач отримає повідомлення.

3.2 Демонстрація роботи розробленої ІС

Для початку роботи з інформаційною системою потрібно скласти преїскурант клініки. У горизонтальному меню, що знаходиться в лівій частині сторінки, обрати пункт «Преїскурант». Він поділений на розділи за видом

стоматологічних послуг (Рис 3.6 Розділи сторінки прейскуранту). Пункти послуг подаються у вигляді таблиці, яка містить: код, назву послуги, ціну, кнопок для виклику діалогу редагування, дати та часу останніх змін.

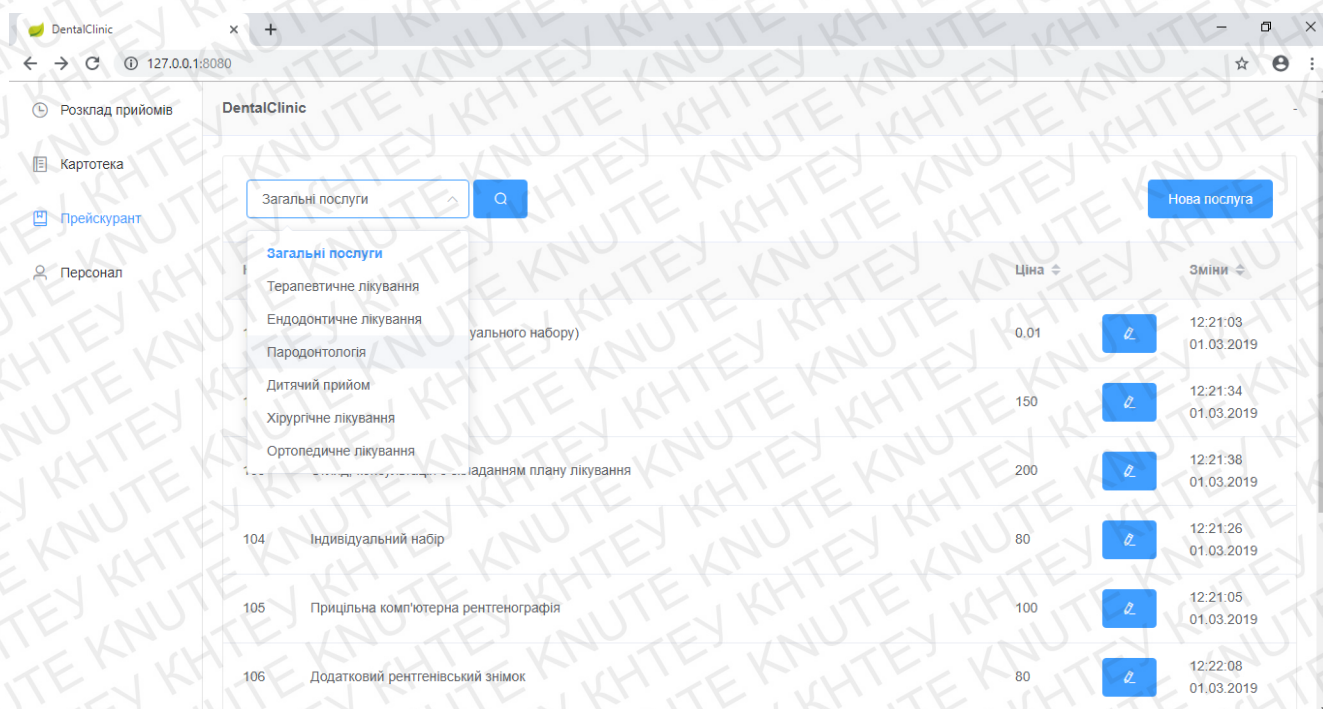


Рис 3.6 Розділи сторінки прейскуранту

Для перегляду пунктів прейскуранту потрібно обрати розділ у випадаючому списку після чого натиснути кнопку «Пошук».

У верхній правій частині сторінки прейскуранту розміщена кнопка для створення нового пункту. Натиснувши на неї відкриється діалогове вікно з формою для заповнення (Рис 3.7 Діалог створення нового пункту прейскуранту). Вона містить:

- випадаючий список з переліком розділів стоматологічних послуг
- поле для коду з трьох цифр в якому перша автоматично підставляється згідно вибраного розділу
- текстове поле для назви послуги
- поле для вибору ціни

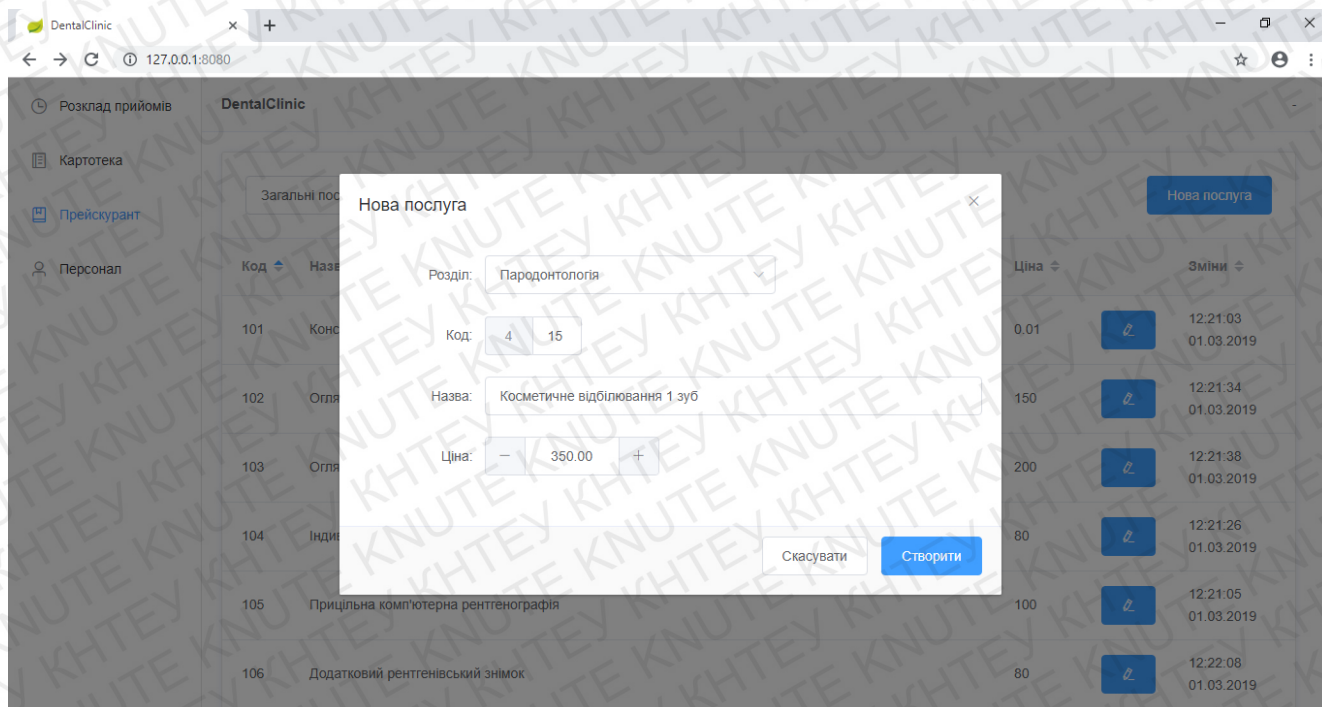


Рис 3.7 Діалог створення нового пункту прейскуранту

Для внесення змін пункту прейскуранту потрібно натиснути відповідну йому кнопку, на якій зображено олівець. Форма редагування містить червону кнопку для видалення з прейскуранту (Рис 3.8 Діалог редагування пункту прейскуранту). Редагування та видалення пункту з прейскуранту не внесе змін до списку наданих послуг існуючих записів на прийом.

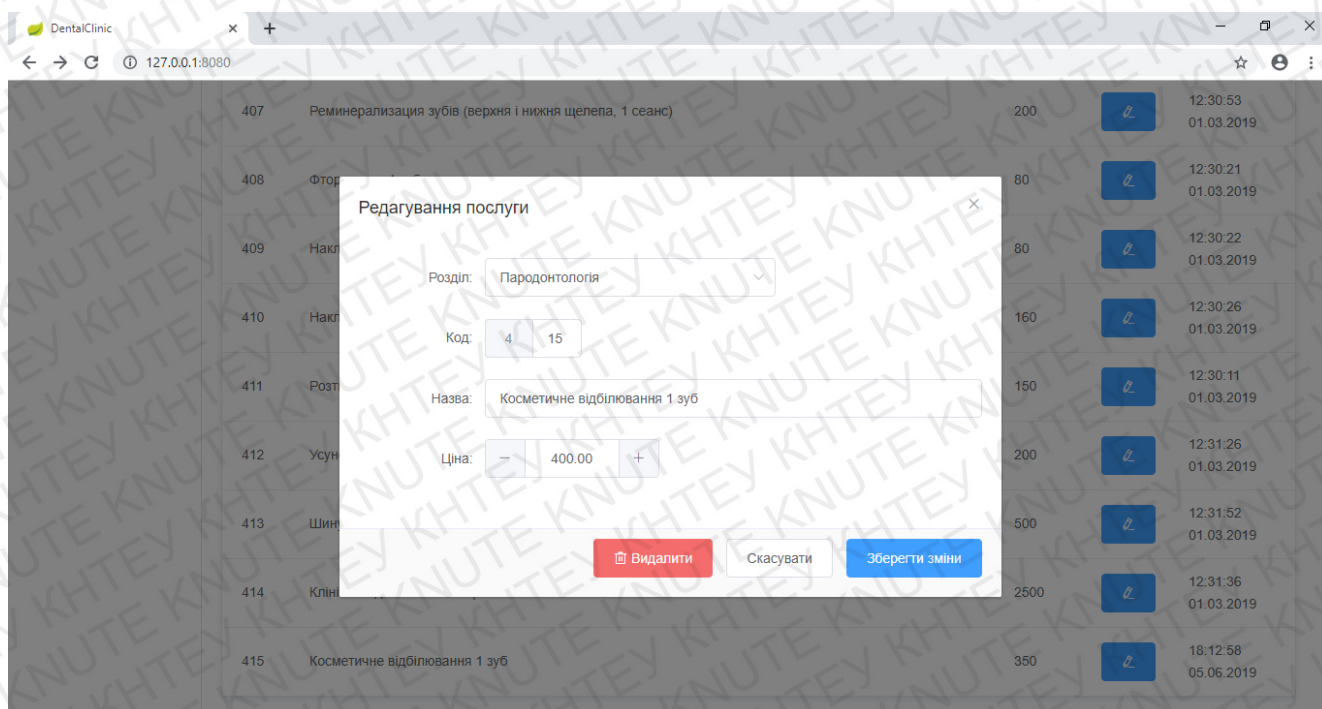


Рис 3.8 Діалог редагування пункту прейскуранту

Для управління даними лікарів клініки існує сторінка персоналу. Вони подані у вигляді таблиці з повним ім'ям лікаря, строки якої розгортаються для перегляду додаткової інформації: дати народження та віку, статі, адреси, номеру телефону (Рис 3.9 Сторінка «Персонал»).

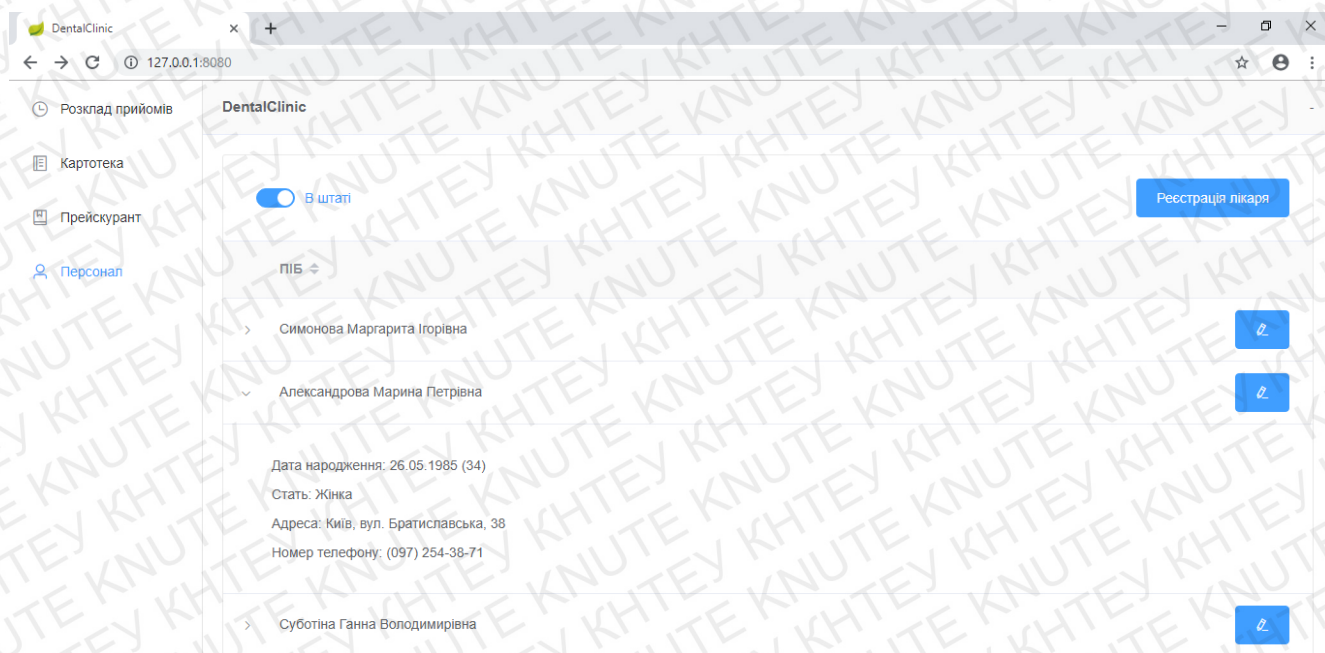


Рис 3.9 Сторінка «Персонал»

Тумблер, що знаходиться над таблицею з даними відповідає за перегляд лікарів, що знаходяться у штаті клініки на даний момент. Якщо його вимкнути, у таблиці будуть відображатися виключені зі штату лікарі (Рис 3.10 Відображення виключених зі штату клініки лікарів).

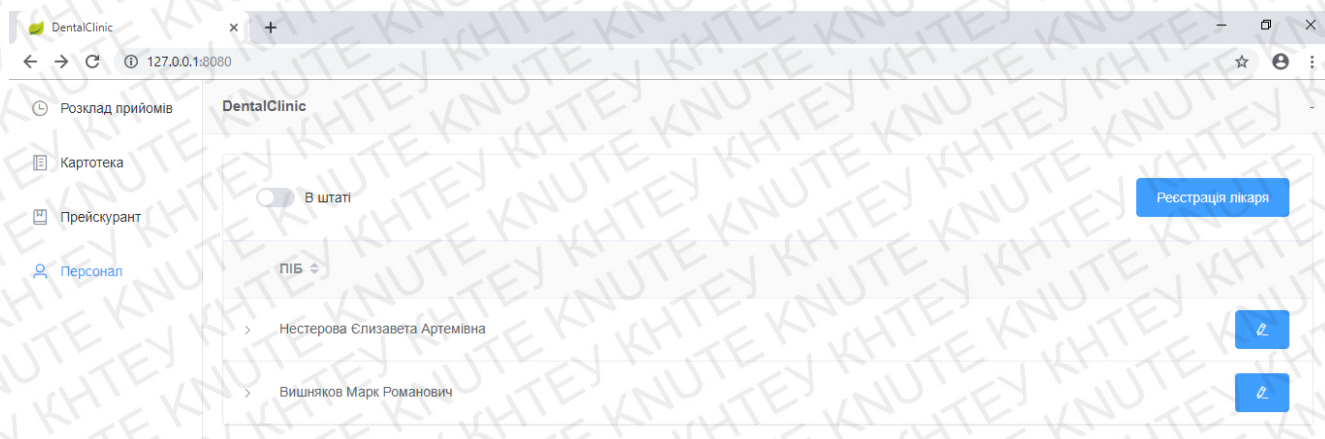


Рис 3.10 Відображення виключених зі штату клініки лікарів

Натиснувши на кнопку «Реєстрація лікаря», відкриється діалогове вікно з формою для введення наступних даних:

- Прізвище
- Ім'я
- По-батькові
- Стать
- Дата народження
- Контактний телефон
- Місто
- Адреса

При реєстрації нового лікаря його одразу буде включено до штату клініки. Для редагування даних лікарів використовується спеціальна форма, яка відкривається при натисканні на відповідну йому кнопку з малюнком олівця (Рис 3.11 Діалог для редагування даних лікаря). При виключенні лікаря зі штату, дані прийомів буде збережено.

The screenshot displays the 'DentalClinic' web application. A modal window titled 'Редагування даних лікаря' is open, allowing for the editing of a doctor's profile. The form includes the following fields and values:

- Прізвище: Симонова
- Ім'я: Маргарита
- По-батькові: Ігорівна
- Стать: Жінка
- Дата народження: 14.09.1991
- Номер телефону: 0952445745
- Місто: Київ
- Адреса: вул. Юності, 9/6
- Стан: Знаходиться у штаті

At the bottom of the modal, there are two buttons: 'Скасувати' (Cancel) and 'Зберегти зміни' (Save changes). The background interface shows a sidebar with navigation options like 'Розклад прийомів', 'Картотека', 'Прейскурант', and 'Персонал'. The 'Персонал' section is active, showing a list of doctors with a toggle switch for 'В штаті' (In staff) and a list of names: Симонова, Александр, and Суботина.

Рис 3.11 Діалог для редагування даних лікаря

Перед тим, як записати нового клієнта на прийом, потрібно провести його реєстрацію на сторінці під назвою «Картотека». Для зручності користувача вивід даних клієнтів реалізовано в такому ж вигляді як лікарів на сторінці «Персонал» (Рис 3.12 Сторінка «Картотека»).

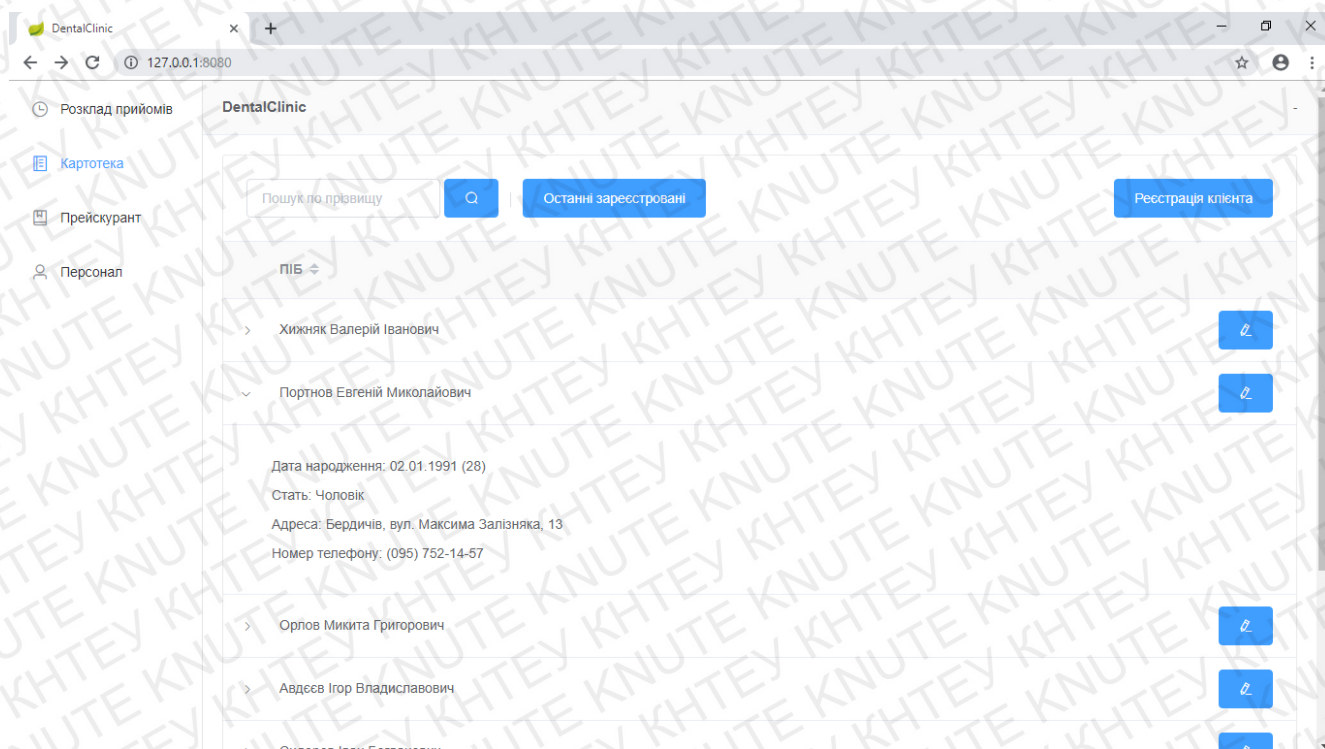


Рис 3.12 Сторінка «Картотека»

В таблиці відображаються 20 останніх зареєстрованих клієнтів. Над нею знаходиться поле для пошуку клієнта. Щоб скористатися пошуком, потрібно ввести прізвище або його сегмент у текстове поле, після чого натиснути кнопку «Пошук» (Рис 3.13 Пошук клієнта по сегменту його прізвища). Форма редагування та реєстрації клієнта до картотеки ідентична тій, що використовується для управління даними лікаря.

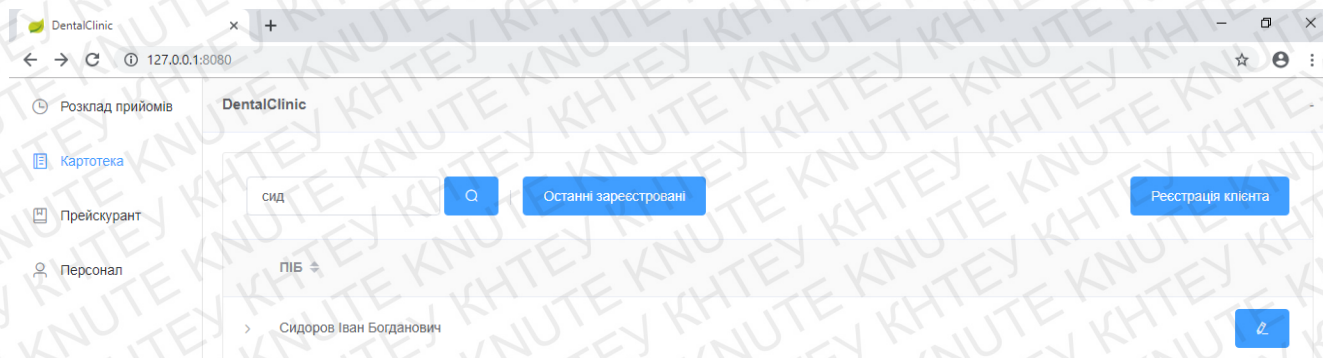


Рис 3.13 Пошук клієнта по сегменту його прізвища

Розклад записів на прийом доступний в першому пункті меню. За замовчуванням розклад відображається на сьогоднішній день (Рис 3.14 Сторінка «Розклад прийомів»). При необхідності є можливість перегляду розкладу будь-якого календарного дня. Для цього треба обрати потрібну дату та натиснути кнопку «Пошук» (Рис 3.14 Вибір довільної дати для перегляду розкладу).

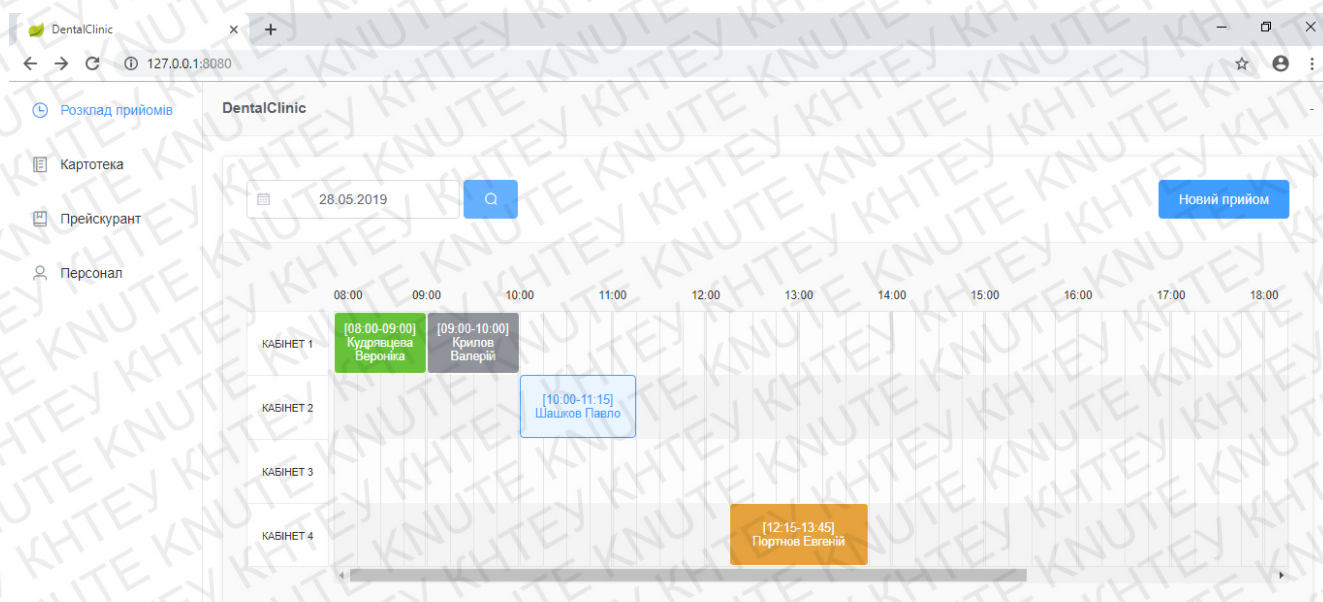


Рис 3.14 Сторінка «Розклад прийомів»

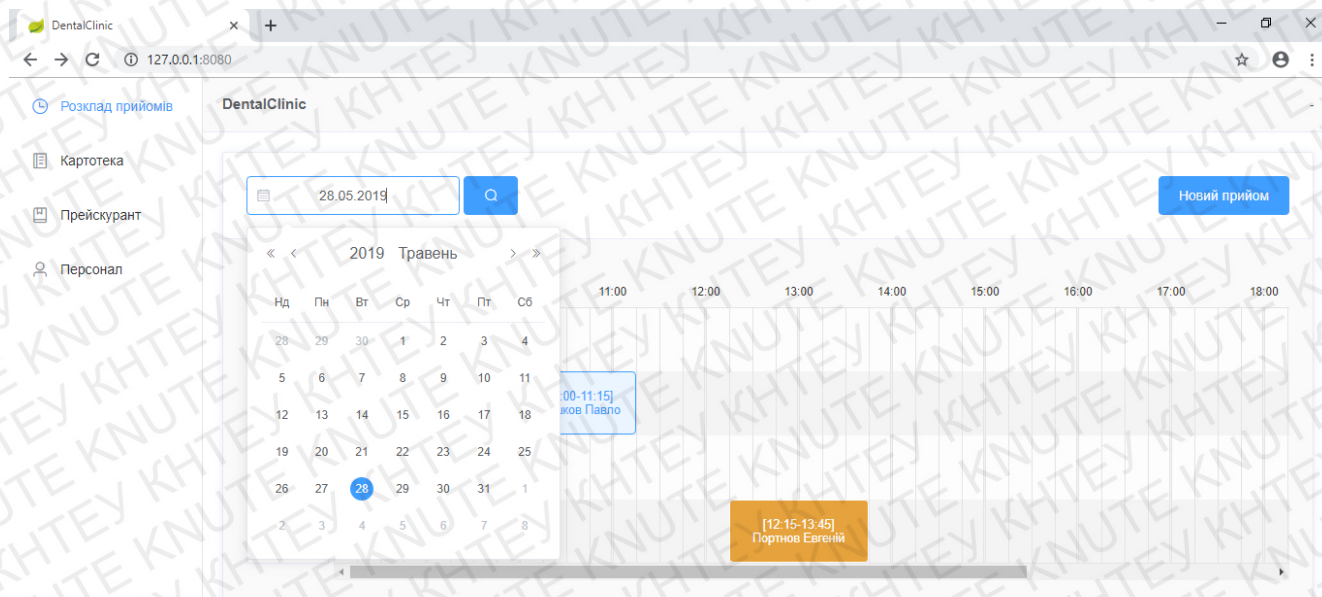


Рис 3.14 Вибір довільної дати для перегляду розкладу

Щоб створити запис, спочатку потрібно відкрити діалогове вікно за допомогою кнопки «Новий прийом», яка розташована у верхній правій частині сторінки. Далі ввести наступні дані форми:

- Дата
- Клієнт
- Лікар
- Кабінет
- Час початку та кінця прийому
- Стан прийому (статус)

Для вибору потрібного пацієнта використовується пошук по картотеці за прізвищем або його сегментом (Рис 3.15 Діалог створення нового запису на прийом). У випадаючому списку вибору лікаря відображаються лише лікарі, що знаходяться у штаті клініки.

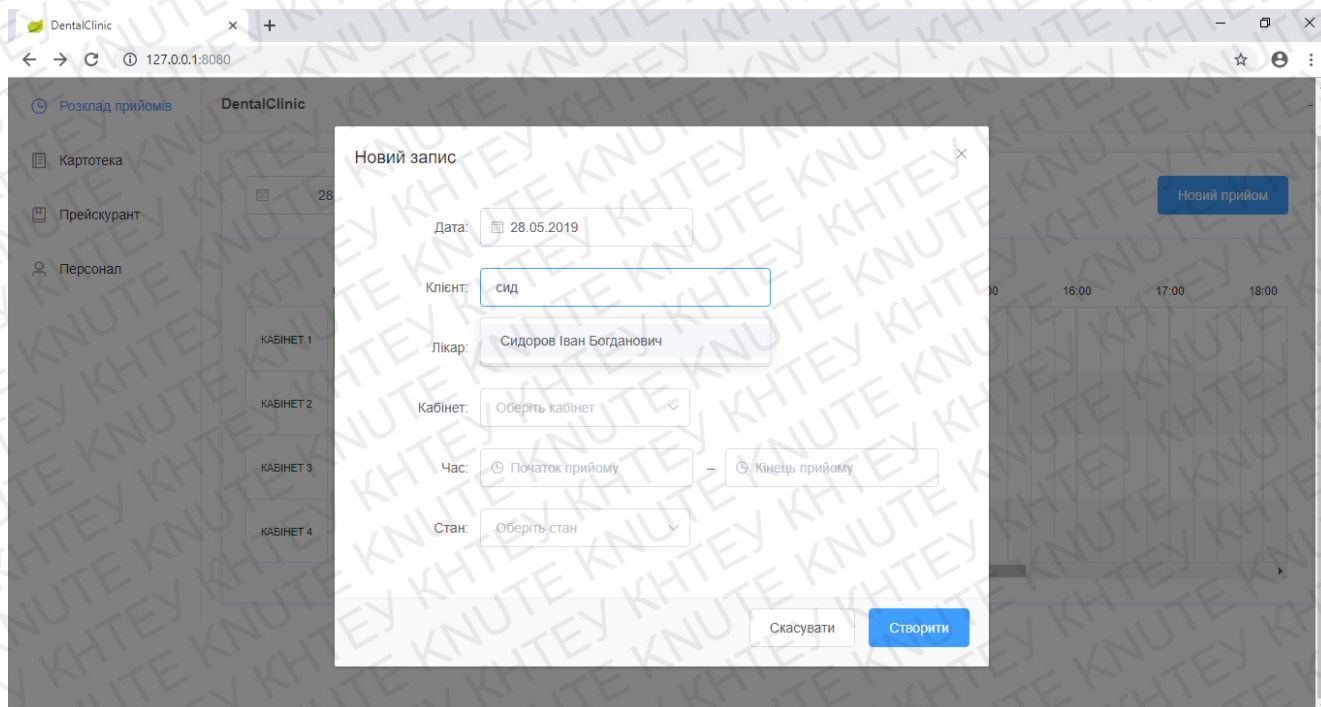


Рис 3.15 Діалог створення нового запису на прийом

При спробі створити запис, система спочатку перевірить наявність вільного часу та у разі конфлікту з існуючими записами сповістить про це користувача. У цьому випадку для створення запису потрібно відредагувати час опираючись на повідомлення (Рис 3.16 Спроба створення запису на період часу зайнятий іншими прийомами).

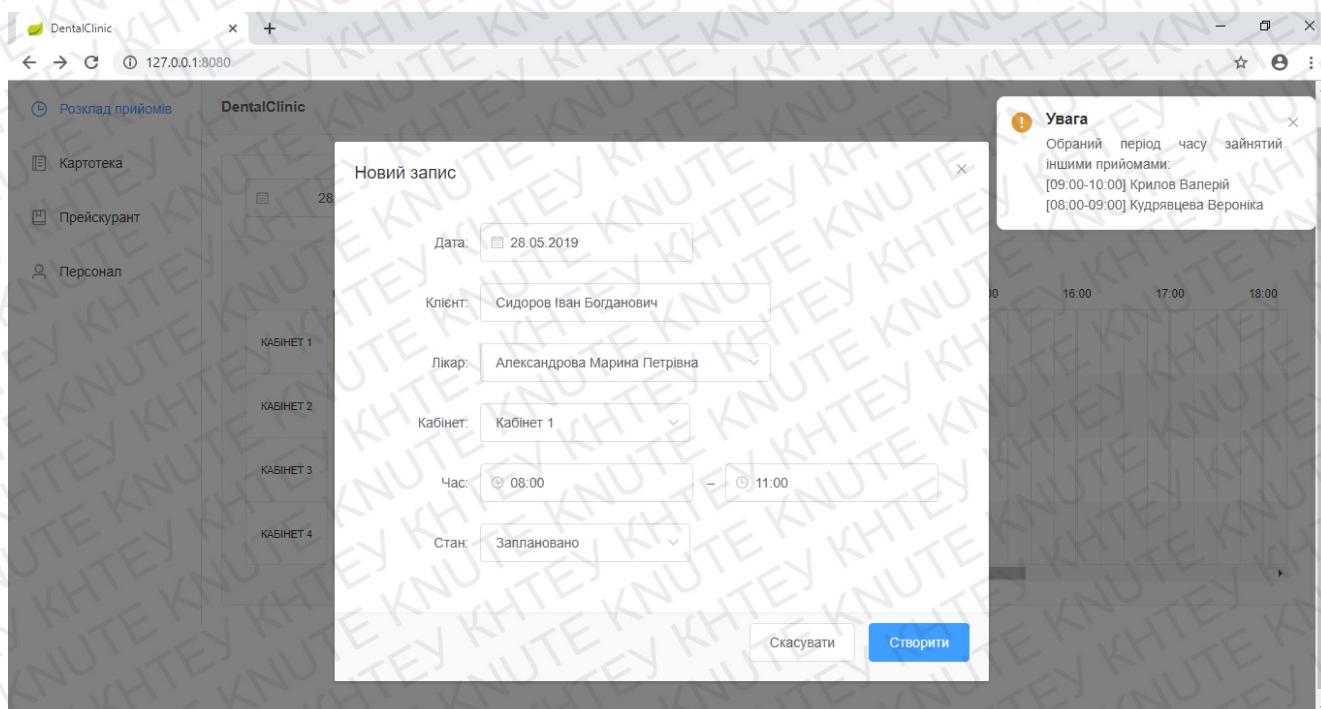


Рис 3.16 Спроба створення запису на період часу зайнятий іншими прийомами

Якщо усі дані форми було заповнено правильно та немає конфліктів у часі, запис на прийом буде створено. Після цього буде завантажений розклад дня який відповідає створеному прийому (Рис 3.17 Успішне створення запису на прийом)

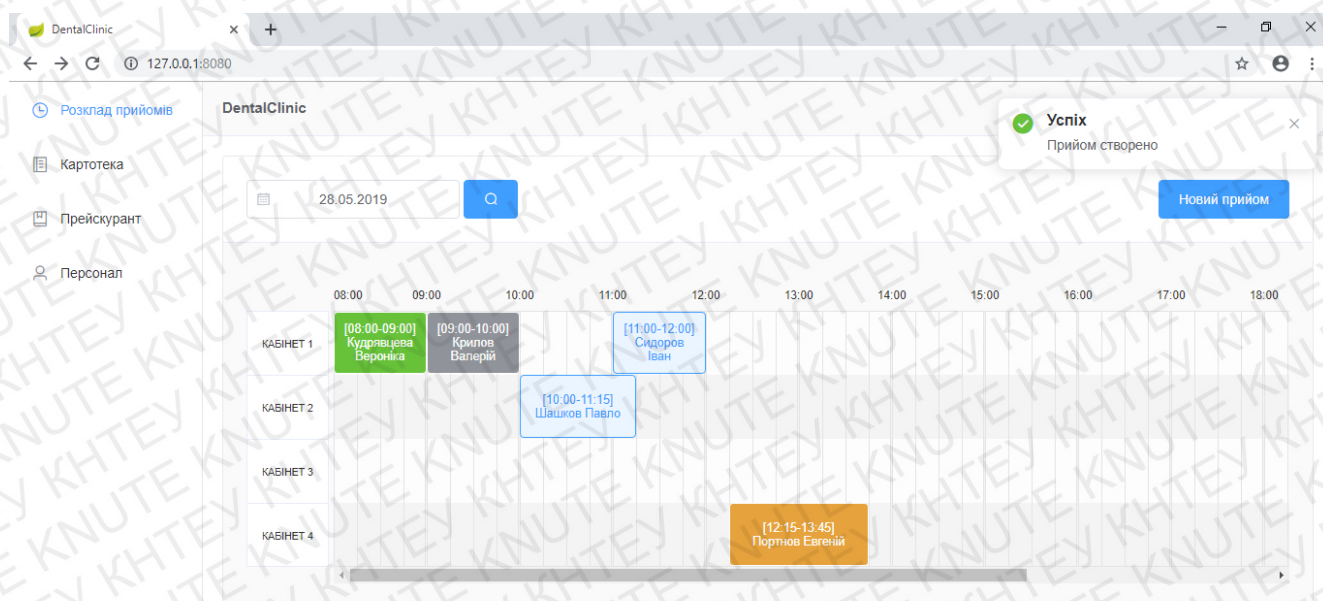


Рис 3.17 Успішне створення запису на прийом

Блоки прийому містять час, прізвище, ім'я клієнта та мають відповідний до статусу запису колір. Для подальшої роботи з прийомом потрібно натиснути на нього. Відкриється вікно з більш детальною інформацією про клієнта та двома кнопками: «Редагувати прийом», «Перелік послуг» (Рис 3.18 Діалогове вікно при натисканні на блок запису).

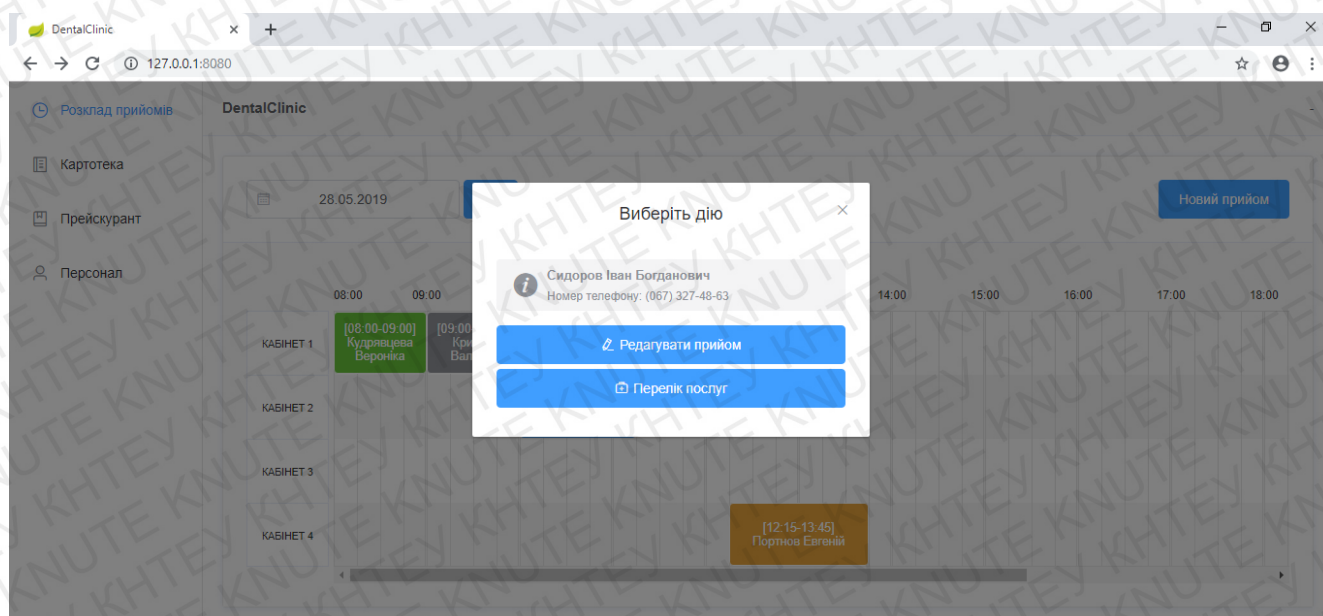


Рис 3.18 Діалогове вікно при натисканні на блок запису

Остання кнопка служит для перегляду та редагування послуг наданих пацієнту по цьому прийому (Рис 3.19 Список наданих послуг прийому).

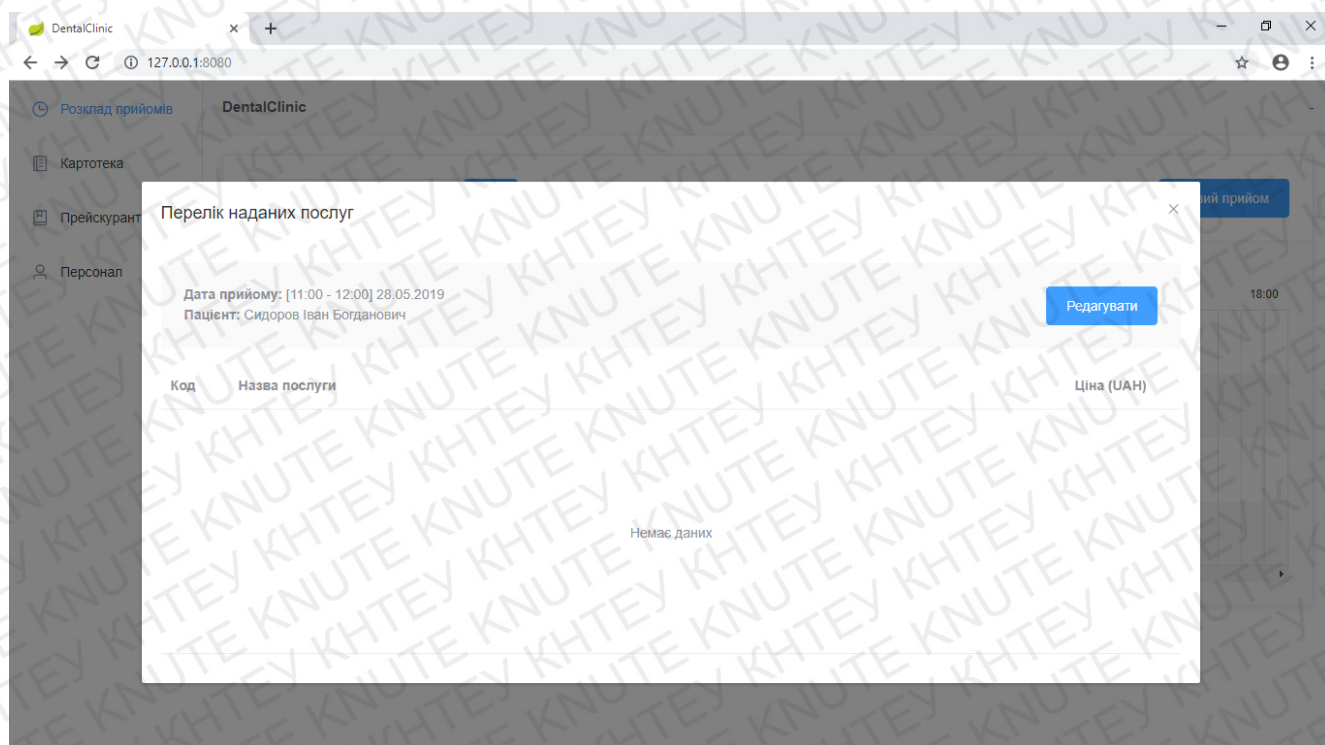


Рис 3.19 Список наданих послуг прийому

У верхній правій частині вікна є кнопка, що дозволяє редагувати цей список послуг. Вона не доступна, якщо прийом має статус «Виконаний». Щоб додати послугу, потрібно обрати її за допомогою випадаючого списку, що містить: код, назву та ціну пункту прейскуранту. Для зручності користувача реалізовано пошук по цьому списку. Потрібно ввести в нього код або сегмент назви послуги. Після чого обрати один з наданих результатів та натиснути кнопку «Додати» (Рис 3.20 Пошук потрібного пункту прейскуранту при редагуванні переліку наданих послуг).

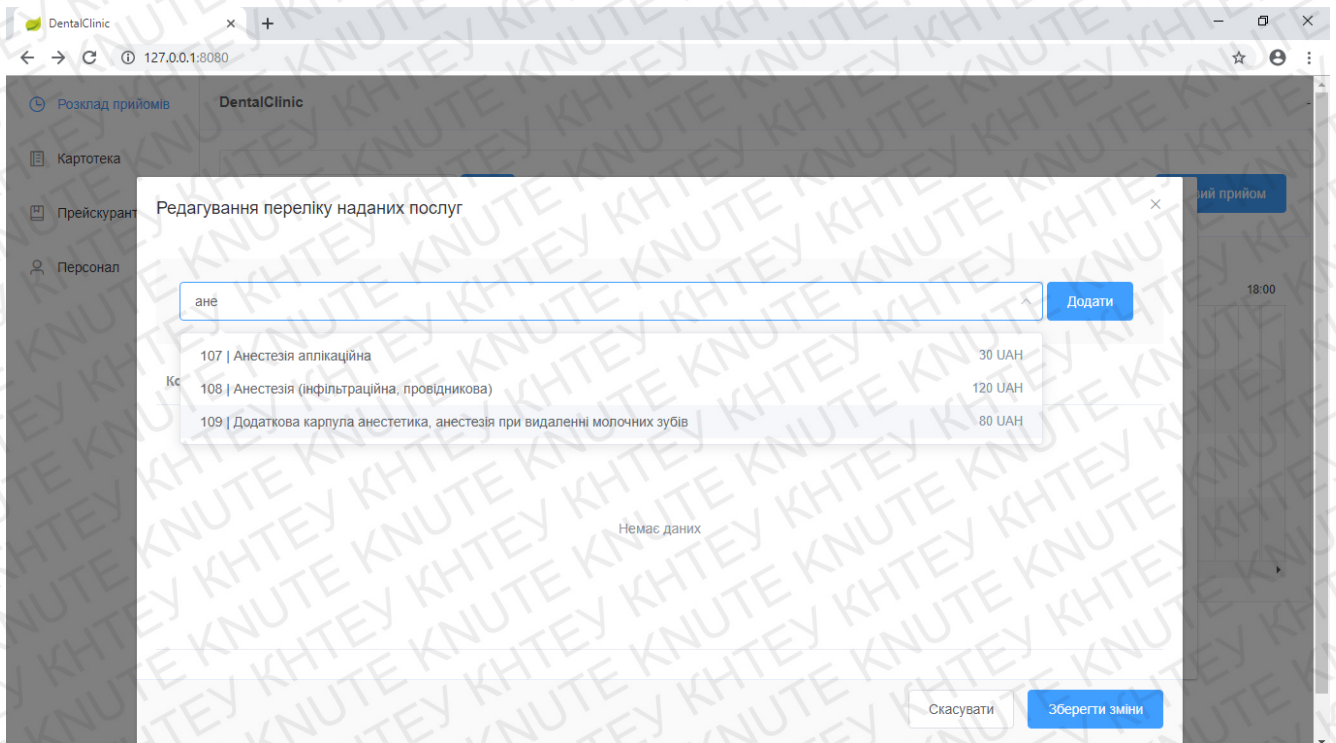


Рис 3.20 Пошук потрібного пункту прейскуранту при редагуванні переліку наданих послуг

При потребі видалення пункту з переліку наданих послуг передбачені відповідні ним червоні кнопки з малюнком кошика (Рис 3.21 Видалення пункту з переліку наданих послуг).

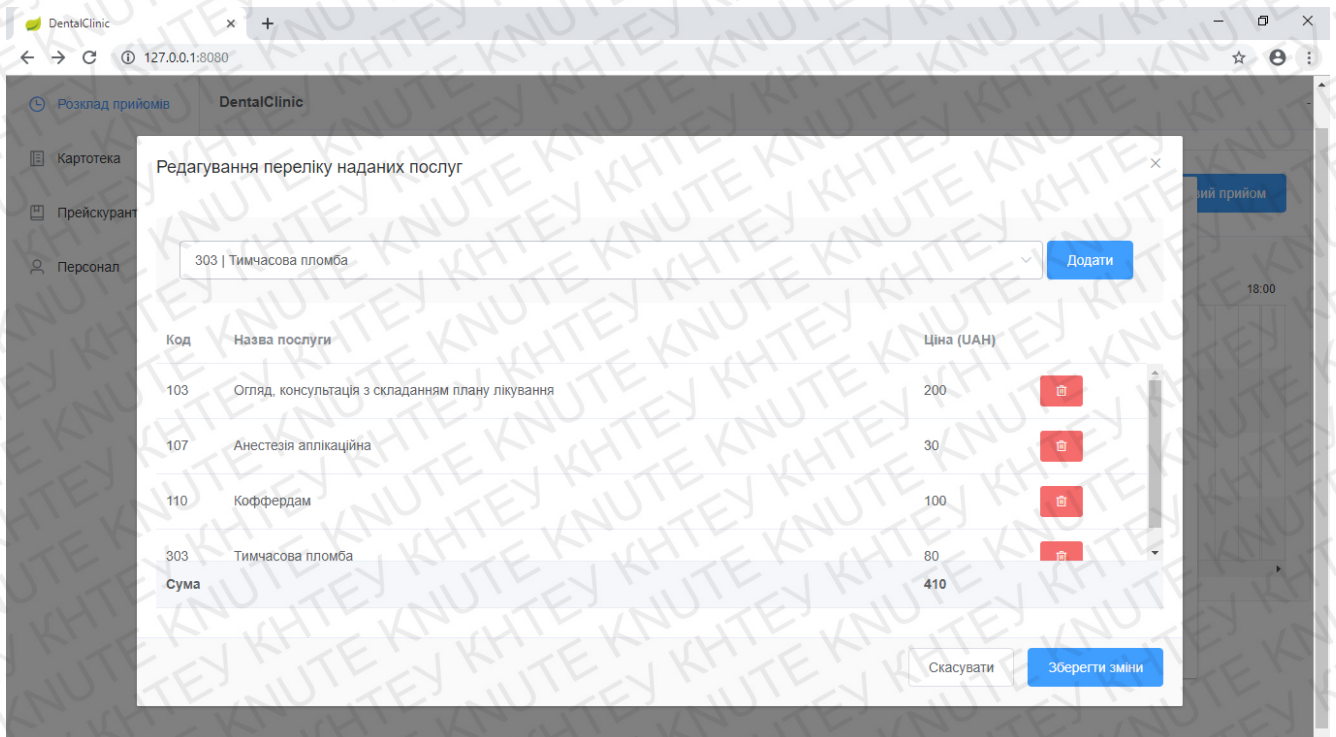


Рис 3.21 Видалення пункту з переліку наданих послуг

Після вибору потрібних пунктів, зміни потрібно зберегти натиснувши на кнопку у нижній частині діалогового вікна. Тепер перелік наданих послуг відредаговано та можна побачити загальну суму оплати по цьому прийому (Рис 3.22 Заповнений список наданих послуг).

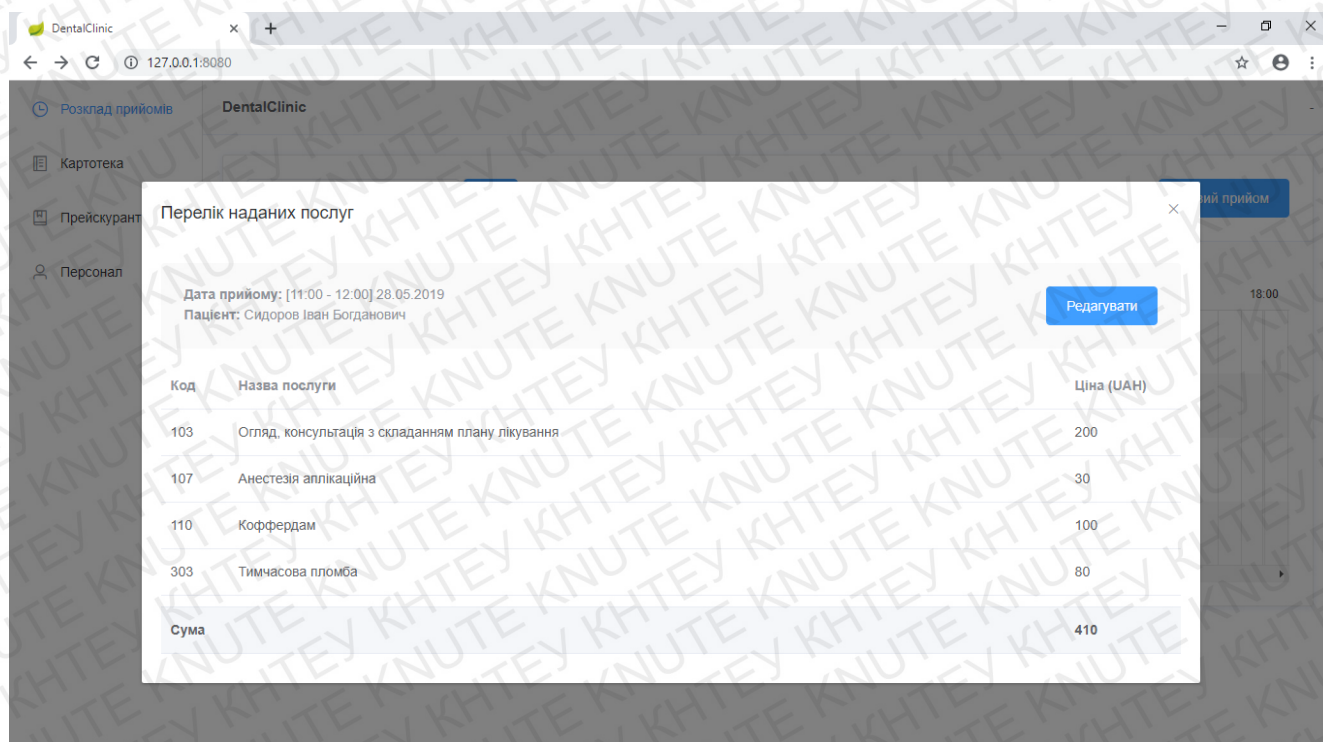


Рис 3.22 Заповнений список наданих послуг

3.3 Висновки до розділу 3

З використанням бібліотеки Element, було розроблено зручний та зрозумілий інтерфейс для користувача клієнтської частини ІС. Щоб візуалізувати розпорядок дня для кожного з кабінетів було підключено та налаштовано бібліотеку Timetable. Створено інструкцію користування з описом усіх функцій ІС.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Відповідно до поставлених завдань, можна зробити наступні висновки.

Було проаналізовано діяльність стоматологічної клініки, визначено основні функціональні вимоги та створено технічне завдання на розробку інформаційної системи обліку клієнтів. Згідно технічного завдання було розроблено базу даних, серверну частину з використанням мови програмування Java та методу передачі даних REST. Окрім серверної частини було спроектовано зрозумілий та зручний інтерфейс для клієнтської частини, яка розроблена з використанням сучасного інструменту Vue. Дана розробка мала перспективу впровадження у приватній стоматологічній клініці.

Щодо подальшого вдосконалення система може бути доповнена двофакторною автентифікацією. На дану тематику, автор приймав участь у Всеукраїнській науково-практичній конференції «Безпека соціально-економічних процесів в кіберпросторі», де були опубліковані тези «Двофакторна автентифікація апаратним токеном з підтримкою стандарту FIDO U2F».

					КНТЕУ 6.050103 07-27.БР			
Зм.	Арк.	№ докум.	Підпис	Дата				
Зав. Каф.		Криворучко О.В.			Розробка інформаційної системи обліку клієнтів стоматологічної клініки	Стадія	Аркуш	Аркушів
Керівник		Рассамакін В.Я.				ВП	35	36
Гарант		Цензура М.О			Висновки та пропозиції	Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Розробив		Цьомка О.О.						

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Herbert Schildt - Java: The Complete Reference, Ninth Edition. / Herbert Schildt - McGraw-Hill Education; 9 edition, 2014 – 1312 с.
2. Shelley Powers - JavaScript Cookbook: Programming the Web / Shelley Powers - O'Reilly Media; 2 edition, 2015 – 634 с.
3. Paul DuBois - MySQL Cookbook: Solutions for Database Developers and Administrators / Paul DuBois - O'Reilly Media; 3 edition, 2014 – 866 с.
4. Цьомка О.О. - Двофакторна автентифікація апаратним токеном з підтримкою стандарту FIDO U2F: журнал Безпека соціально-економічних процесів в кіберпросторі / Цьомка О.О., Рассамкін В.Я. – К.: Київ. Нац. Торг.-Екон. Ун-т, 2019 – 107-108 с.

Інтернет ресурси

1. Getting Started - Building a RESTful Web Service [Електронний ресурс]. – Режим доступу : <https://spring.io/guides/gs/rest-service>.
2. Spring Data JPA - Reference Documentation [Електронний ресурс]. – Режим доступу: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html>.
3. Guide - Vue.js [Електронний ресурс]. – Режим доступу: <https://vuejs.org/v2/guide>.
4. Java tutorial – Learn Java [Електронний ресурс]. – Режим доступу: <https://www.javatpoint.com/java-tutorial>.
5. Components - Element [Електронний ресурс]. – Режим доступу: <https://element.eleme.io/#/en-US/component>.

					<i>КНТЕУ 6.050103 07-27.БР</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.	Криворучко О.В.				Розробка інформаційної системи обліку клієнтів стоматологічної клініки	Стадія	Аркуш
Керівник	Рассамкін В.Я.					СД	36
Гарант	Цензура М.О.				Список використаних джерел	Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група	
Розробив	Цьомка О.О.						