

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ФРАНКА**

**Факультет прикладної математики та інформатики**

**Кафедра дискретного аналізу та інтелектуальних систем**

**ДИПЛОМНА РОБОТА**

**Розробка застосунку та телеграм-боту для проведення оцінки  
вживаних автомобілів**

Виконав студент 4 курсу, групи ПМі-45с ,  
спеціальності «122 Комп'ютерні науки»  
Алхажж Шарбель

Керівник \_\_\_\_\_  
професор, д.ф.-м.н. Притула М.М.

Рецензент \_\_\_\_\_

Львів – 2023

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Кафедра дискретного аналізу та інтелектуальних систем

Спеціальність 122 - “Комп’ютерні науки”

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

професор Притула М.М.

" \_\_\_\_ " \_\_\_\_\_ 20 \_\_\_\_ року

## ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Алхажжу Шарбелю

1. Тема роботи “Розробка застосунку та телеграм-боту для проведення оцінки вживаних автомобілів”

керівник роботи д.ф. - м.н., професор Притула М.М.

затверджені Вченою радою факультету від "13" вересня 2022 року № 15

2. Строк подання студентом роботи 13 червня 2023 року

3. Вихідні дані до роботи Постановка задачі та її формалізація, створення програмної платформи, якою можуть користуватись автомайданчики для зручної оцінки вживаних автомобілів. В результаті отримати комплексний програмний продукт( веб застосунок та телеграм бот)

4. Зміст дипломної роботи (перелік питань, які потрібно розробити)

Аналіз доступних сервісів оцінки автомобілів;

Проектування та побудова веб-сервісу

Проектування та побудова програмного забезпечення;

### **5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

Таблиці та схеми, рисунки

### **6. Консультанти розділів роботи**

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	<u>д.ф.-м.н., професор Притула М.М.</u>		

**7. Дата видачі завдання 27 вересня 2022 року**

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітки
1.	Доступні сервіси оцінки автомобілів	27.09.2022 - 21.10.2022	
2.	Проектування та побудова веб-сервісу	21.10.2022 - 28.10.2022	
3.	Програмна реалізація	28.10.2022 - 13.12.2022	
4.	Створення телеграм бота	13.12.2022 - 27.03.2023	
5.	Функціонал	27.03.2023 - 10.05.2023	
6.	Оформлення звіту дипломної роботи	11.05.2023 - 10.06.2023	

Студент \_\_\_\_\_ **Алхажж Шарбель**  
( підпис )

Керівник роботи \_\_\_\_\_ **д.ф. - м.н., професор Притула М.М.**  
( підпис )

## Зміст

<b>ВСТУП</b>	6
Мета роботи	6
Структура роботи	6
Постановка задачі	6
<b>1. ДОСТУПНІ СЕРВІСИ ОЦІНКИ АВТОМОБІЛІВ</b>	8
1.1 Поняття про оцінку вживаних автомобілів	8
1.2 Платформи оцінки авто	8
1.3 Формування концепції системи, що розробляється	9
<b>2. ПРОЕКТУВАННЯ ТА ПОБУДОВА ВЕБ-СЕРВІСУ</b>	10
2.1 Архітектура веб-сервісу	10
2.1.1 Загальна структура веб-сервісу	10
2.1.2 Вибір архітектури протоколу обміну даними	10
2.1.3 Вибір архітектури інтерфейсу користувача	11
2.1.4 Вибір архітектурного шаблону організації сервісу	12
2.2 Вибір інструментів для розробки	13
2.2.1 Мова програмування	13
2.2.2 Програмний каркас	15
2.3 Алгоритм функціонування	18
2.3.1 Типи користувачів	18
2.3.2 Алгоритм функціонування веб-сервісу зі сторони спеціаліста	18
2.3.3 Алгоритм функціонування веб-сервісу зі сторони адміністратора	18
<b>3. ПРОГРАМНА РЕАЛІЗАЦІЯ</b>	20
3.1 Програмна реалізація веб сервісу	20
3.2. Створення телеграм бота	24
<b>ВИСНОВКИ</b>	31
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</b>	32

## ВСТУП

З ростом сучасних технологій активно набирає оборотів розвиток інтернет магазинів з продажу авто. На даний час автобазари та площадки повільно відходять на задній план. Ріст технологій відкриває можливість спростити купівлю та продаж авто. На мій погляд розробка цієї платформи є кроком вперед. Розроблена платформа може широко застосовуватись авто майданчиками. Оскільки дана платформа дозволить суттєво знизити ціну автомобіля, зробити оцінку авто. Також до даної платформи розроблено телеграм бота для простішого та більш зручного користування.

**Мета роботи:** запроектувати та створити програмну платформу для оцінки автомобілів.

Для цього необхідно:

- Навчитися роботі з Ruby on Rails.
- Створити зручну за інтерфейсом програму для користувача.;
- Провести оптимізацію над програмою для швидкої взаємодії із БД.;
- Створити телеграм бота до даної платформи.

### Структура роботи

Загалом її можна поділити на такі етапи:

1. Створення макету;
2. Моделювання бази даних;
3. Кодова реалізація програми.
4. Створення телеграм бота

### Постановка задачі

Розробити веб сервіс та телеграм бота до нього, який би давав змогу зберігати у базі даних дані про авто та його експертну оцінку.

А саме:

- Модель автомобіля (CarModels)

- Звіт з огляду авто (Review)
- Індивідуально підібрані питання до конкретної моделі (Questions) та відповіді на них після огляду (Answers)
- Забезпечити можливість додавати дані та їх видалення. Вивести список всіх оглядів авто або певний огляд авто.

# 1. ДОСТУПНІ СЕРВІСИ ОЦІНКИ АВТОМОБІЛІВ

## 1.1 Поняття про оцінку вживаних автомобілів

Одним з основних підходів для визначення оцінки вживаних авто є порівняльний принцип. Порівняльний принцип ґрунтується на аналізі продажу (пропозиції) аналогічних або подібних авто. Під первинним ринком слід розуміти, що це нові авто, під вторинним - вживані автомобілі, які вже були в користуванні. За цим методом ми можемо визначити середню ціну на авто, враховуючи базову ціну на ідентичні транспортні засоби з відповідними роками експлуатації. Подальше коригування ціни може залежати від комплектації авто, пробігу, укомплектованості та технічного стану об'єкта порівняння та об'єкта оцінки. Відповідно до цих параметрів потрібно проводити оцінку автомобіля.

## 1.2 Платформи оцінки авто

eTachki (eTachki.com.ua)- платформа для оцінки та допомоги продажу вживаних автомобілів. Система оцінює автомобіль, після чого пропонують найбільшу можливу ціну, за яку компанія готова купити автомобіль. Також в них є можливість після оцінки авто виставити її на аукціон, де можливо запропонують кращу ціну.

Winnerauto (Віннер Рітейл) - це також платформа для оцінки вживаних автомобілів. Дана система лише оцінює авто, вказуючи вам найвищу ціну, за якою можна продати автомобіль. Проте ця платформа не пропонує викупити машину, лише оцінити.

Vidi (vidi-automarket.com.ua)- платформа для оцінки вживаних автомобілів, вони оцінюють ваше авто, після чого пропонують максимально вигідні умови, щоб купити або обміняти ваш автомобіль на інше авто.



### **1.3 Формування концепції системи, що розробляється**

В останні декілька років великої популярності набрали майданчики авто-викупу, суть яких полягає в оренді майданчика на якому здійснюється оцінка, купівля та продаж вживаних авто. Проблемою таких площадок є достатньо велика собівартість утримання площадки та її працівників.

У зв'язку із цим я вирішив створити веб-сервіс, який, базуючись на даних про середню ринкову вартість та ряді персональних налаштувань, надавав би такі можливості:

- отримання інформації про реальну вартість автомобіля;
- автоматизований спосіб поверхневої оцінки авто ;
- перегляд звіту після поверхневої оцінки авто;

## **2. ПРОЕКТУВАННЯ ТА ПОБУДОВА ВЕБ-СЕРВІСУ**

### **2.1 Архітектура веб-сервісу**

#### **2.1.1 Загальна структура веб-сервісу**

При розробці веб-сервісів слід пам'ятати про те, що у перспективі до нього може звертатись велика кількість користувачів і враховувати можливі шляхи масштабування проекту.

На даний момент система є монолітною, але в майбутньому з масштабуванням програми, ми можемо розділити її на дві частини.

Таким чином при збільшенні навантаження на сервіс ці 2 частини можуть бути рознесені фізично, що дозволить підвищити максимальну кількість конкурентних запитів.

#### **2.1.2 Вибір архітектури протоколу обміну даними**

На сьогоднішній день найбільш популярним підходом доступу до інформаційних ресурсів є REST-підхід.

REST (скор. англ. Representational State Transfer, «передача стану подання») — підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів. Був описаний і популяризований у 2000 році Роєм Філдінгом (Roy Fielding), одним із творців протоколу HTTP. Найвідомішою системою, побудованою переважно за архітектурою REST, є сучасна Всесвітня павутина.

Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад HTML, XML, JSON). Мережевий протокол (як і HTTP) повинен підтримувати кешування, не повинен залежати від мережевого прошарку, не повинен зберігати інформацію про стан між парами «запит-відповідь». Стверджується, що такий підхід забезпечує масштабування системи і дозволяє їй еволюціонувати з новими вимогами.

Антиподом REST є підхід, заснований на виклику віддалених процедур (Remote Procedure Call, RPC). Підхід RPC дозволяє використовувати невелику кількість мережевих ресурсів з великою кількістю методів і складним протоколом. При підході REST кількість методів і складність протоколу суворо обмежені, що призводить до того, що кількість окремих ресурсів має бути великою.

При розробці даного веб-сервісу використовувався REST-підхід у зв'язку з простотою у реалізації, легкістю у підтримці та розширенні, а також з метою підтримки найбільш сучасних стандартів WEB.

### **2.1.3 Вибір архітектури інтерфейсу користувача**

В рамках REST-архітектури є зміст використати CRUD-підхід до архітектури інтерфейсу користувача, оскільки основні принципи CRUD-підходу відповідають основним принципам використання REST-архітектури.

CRUD-підхід до архітектури інтерфейсу користувача - це підхід до розробки програмного забезпечення, який базується на чотирьох основних операціях: створення (Create), читання (Read), оновлення (Update) та видалення (Delete). Ці операції становлять основу для будь-якої системи управління базами даних (СУБД), а також багатьох веб-додатків, що працюють з даними.

CRUD-підхід дозволяє спростувати розробку програмного забезпечення та зменшувати кількість коду, який потрібно написати, оскільки він забезпечує стандартні методи взаємодії з базою даних. Крім того, він дозволяє забезпечити більшу безпеку взаємодії з базою даних, оскільки ви не маєте прямого доступу до даних.

Наприклад, веб-додаток, який дозволяє користувачам створювати, переглядати, оновлювати та видаляти записи в базі даних, може використовувати CRUD-операції для забезпечення цих функцій. Кнопки "Створити", "Читати", "Оновити" та "Видалити" можуть бути розміщені на

сторінці веб-додатку, і вони будуть взаємодіяти з базою даних, використовуючи відповідні CRUD-операції.

Для прикладу, в програмі адресної книги, базовий об'єкт — це запис з контактними даними. Згідно CRUD-підходу, програма повинна надавати користувачу функції для:

- Додавання записів
- Пошук і зчитування записів
- Редагування записів
- Видалення існуючих записів

#### **2.1.4 Вибір архітектурного шаблону організації сервісу**

В якості шаблону для організації сервісу я вибрав модифікований архітектурний шаблон Модель-Вид-Контролер.

Модель-вид-контролер (або Модель-Вигляд-Контролер, англ. Model-view-controller, MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілими завдяки зменшенню складності.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було зазначено, інкапсулює ядро даних і основний функціонал з їх обробки. Також

компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

## **2.2 Вибір інструментів для розробки**

### **2.2.1 Мова програмування**

При виборі мови програмування я керувався наступними принципами:

- Наявність широкого кола користувачів
- Відкритість платформи
- Використовуваність у комерційних проектах
- Простота у засвоєнні
- Наявність необхідного інструментарію
- Висока швидкість розробки

Після детального порівняння у якості основних варіантів залишилися такі мови як Python, Ruby та Java.

Серед них я вибрав Ruby, тому що у неї такий ряд переваг:

- Швидкість у розміщенні продукту
- Наявність інтерпретатора JRuby, який дозволяє використати інфраструктуру JVM, що має значний ефект на швидкодії
- Зручний синтаксис
- Активні користувачі, завжди готові відповісти на запитання

Ruby (англ. «Рубін», читається «Рубі») — це інтерпретована, повністю об'єктно-орієнтована мова програмування з чіткою динамічною типізацією і має такі особливості:

- Лаконічний і простий синтаксис, часто зустрічається вплив Ада, Eiffel і Python.
- Може обробляти винятки в стилі Java та Python.
- Дозволяє перевизначити оператори, які насправді є методами.
- Повністю об'єктно-орієнтована мова програмування. Всі дані в Ruby є об'єктами в розумінні Smalltalk. Єдиний виняток — керівні конструкції, які в Ruby на відміну від Smalltalk не є об'єктами.
- Не підтримується множинна спадковість, взамін може використовуватись концепція MixIn, основана в даній мові на основі механізмів модулів.
- Містить автоматичний прибиральник сміття. Він працює для всіх об'єктів Ruby, в тому числі для зовнішніх бібліотек.
- Підтримує цикли з повною прив'язкою до змінних.
- Цілі змінні в Ruby автоматично конвертуються між типами Fixnum (32-розрядні) і Bignum (більше 32 розрядів) в залежності від їхнього значення, що дозволяє виконувати цілочисельні математичні розрахунки з нескінченною точністю.
- Не вимагає попереднього оголошення змінних, хоча для інтерпретатора бажано, щоб змінній присвоювалось порожнє значення nil (тоді інтерпретатор знає, що ідентифікатор вказує на змінну, а не на ім'я

методу). Мова використовує прості домовленості для визначення області видимості. Приклад: просто `var` — локальна змінна, `@var` — змінна екземпляра (член або поле об'єкта класу), `@@var` — змінна класу, `$var` — глобальна змінна.

- Безпосередньо в мові Ruby реалізовано багато шаблонів програмування, так, наприклад, «одинак» (англ. singleton) може бути реалізований додаванням потрібних методів одному конкретному об'єктові.
- Має незалежну від ОС підтримку невитискальної багатопотоковості.
- Перенесена на багато платформ. Мова розроблялася на GNU/Linux, але працює на багатьох версіях Unix, Microsoft Windows (частково, Win32), Mac OS, BeOS, OS/2 і т. д. [1]

### 2.2.2 Програмний каркас

Для мови Ruby існує декілька основних програмних каркасів для створення веб-застосунків, а саме – Ruby on Rails, Merb та Sinatra.

Ruby on Rails — об'єктно-орієнтований програмний каркас (фреймворк) для створення веб-застосунків, написаний на мові програмування Ruby. Ruby on Rails надає каркас виду модель-вид-контролер (Model-View-Controller) для веб-застосунків, а також забезпечує їхню інтеграцію з веб-сервером і сервером бази даних. Ruby on Rails є відкритим програмним забезпеченням і розповсюджується за ліцензією MIT. [3][7]

Ruby on Rails керується такими принципами при розробці застосунків:

- Застосунки не повинні визначати власну архітектуру, оскільки вони використовують готовий каркас модель-вид-контролер.
- Мова Ruby дозволяє використовувати нотацію, що легко читається, для визначення семантики веб-застосунків (таких як відносини між таблицями в базі даних).

- Ruby on Rails надає механізми повторного використання, що дозволяють мінімізувати дублювання коду у веб-застосунку (принцип Don't Repeat Yourself - не повторюй себе).
- За замовчуванням використовуються угоди по конфігурації, типові для більшості веб-застосунків (принцип Convention over configuration - угоди над конфігурацією). Явна специфікація конфігурації потрібна тільки в нестандартних випадках.

Merb, скорочення від «Mongrel (HTTP сервер) + Erb» - це веб-фреймворк, що реалізує архітектуру Модель-Вид-Контролер написаний на Ruby. Розробники Merb зробили акцент на швидкому та легкому ядрі з базовою функціональністю, в той час як розширення її здійснюється за допомогою сторонніх плагінів.

Merb залишає за розробником вибір інструментів ORM (підтримуються ActiveRecord, Datamapper і Sequel), бібліотек JavaScript, і рушія шаблонів HTML (на вибір HAML або eRuby).

Merb зазнав значного впливу Ruby on Rails, і з таким же успіхом може бути використаний для створення складних веб-застосунків і REST-сервісів. При цьому Merb більш гнучкий і швидкий порівняно з ROR, і більшою мірою підходить для тих, хто збирається переписувати вже існуючі системи, а не створювати з нуля нові, а також у випадках, коли передбачаються дуже великі обсяги трафіку.

Sinatra — об'єктно-орієнтований програмний каркас(framework) для створення веб-застосунків, написана на мові програмування Ruby. На офіційному сайті написано, що Sinatra це навіть не фреймворк, а DSL для створення веб-додатків. Залежить від інтерфейсу веб-сервера Rack. Вона є альтернативою іншим Ruby фреймворкам, таким як Ruby on Rails, Merb, Nitro та Camping.



Sinatra - це мінімалістичний веб-фреймворк для розробки веб-додатків, написаний мовою Ruby. Sinatra пропонує простий і легковаговий спосіб створення веб-додатків, дозволяючи програмістам створювати веб-сторінки і API без надмірної складності.

DSL - це абревіатура від Domain-Specific Language, що перекладається як «мова, спеціально призначена для конкретної області застосування». Це означає, що DSL - це мова програмування, яка створена з метою розв'язання конкретної задачі або в обмеженій області застосування.

У випадку з Sinatra, DSL - це мова програмування, створена для спрощення процесу створення веб-додатків. DSL Sinatra надає простий і зрозумілий синтаксис для роботи з HTTP-маршрутами, зверненнями до баз даних і змінними URL. Завдяки DSL програмісти можуть писати код веб-додатків на більш зрозумілій мові, збільшуючи продуктивність і зменшуючи кількість помилок.

Sinatra є відкритим програмним забезпеченням і розповсюджується під ліцензією MIT.

Sinatra є крихітною (близько 1500 рядків коду, що майже 1/100-а розміру Rails) та надзвичайно гнучкою. Вона не слідує типовій архітектурі Модель-Вид-Контролер (Model-View-Controller, MVC), яка використовується в інших фреймворках, таких як Ruby On Rails. Натомість, Sinatra зосереджена на «швидкому створенню веб-додатків на Ruby із мінімальними зусиллями». Sinatra використовує всі можливості Ruby і є його оригінальним розширенням.

Серед вищеописаних програмних каркасів, я віддав перевагу Sinatra тому, що у нього є такі переваги над RoR та Merb:

- Відсутня жорстка прив'язка до конкретної архітектури
- Включає у себе лише мінімально необхідний функціонал
- Є менш вибагливим до ресурсів

## **2.3 Алгоритм функціонування**

### **2.3.1 Типи користувачів**

Створений веб-сервіс поділяє користувачів на 2 основні групи:

- Спеціаліст – користувач, який має можливість додавати інформацію про авто (виділені жовтим кольором) після їх огляду.
- Адміністратор – користувач, який володіє правами модератора, а також можливістю створювати нові заявки для огляду авто, додавати запитання, які слід ставити під час огляду авто, додавати нові моделі авто.

### **2.3.2 Алгоритм функціонування веб-сервісу зі сторони спеціаліста**

Зі сторони спеціаліста створений веб-сервіс при огляді авто повинен працювати за наступним алгоритмом:

1. Спеціаліст робить звіт з огляду авто, вказуючи такі параметри:
  - реальна ціна автомобіля
  - заповнює поля, які описують загальний стан авто
  - при потребі може додати коментар щодо стану авто
2. Веб додаток отримує інформацію надану користувачем, обробляє, відправляє її в базу даних, опрацьовану інформацію надсилає користувачеві.

### **2.3.3 Алгоритм функціонування веб-сервісу зі сторони адміністратора**

Зі сторони адміністратора створений веб-сервіс повинен працювати за наступним алгоритмом:

1. Адміністратор може додати заявку, після проведення огляду приймає рішення чи продовжити спілкуватись з продавцем авто або видалити цю пропозицію.

Може бути застосована автоматизація процесу прийняття рішення, що дозволить діяти без участі адміністратора.

## 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Програмна реалізація веб сервісу

База даних складається з чотирьох основних таблиць: Огляд (Review), Моделі авто (Car\_Models), Питання (Question), Відповіді (Answer), які пов'язані між собою вказаними нижче зв'язками.

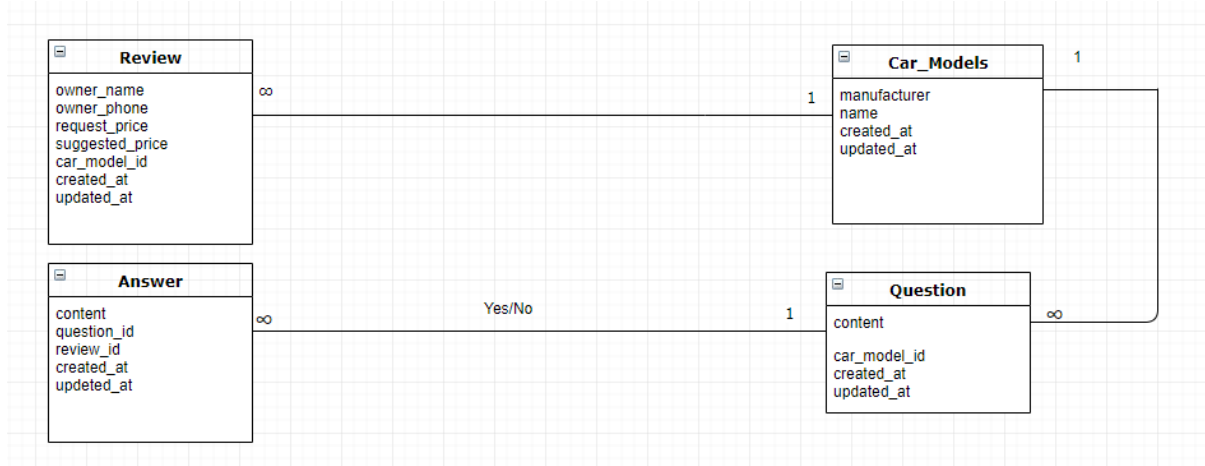


Рис.1 Діаграма зв'язків між таблицями у базі даних.

Огляди авто - це список автомобілів, які мають бути оглянуті спеціалістом або вже були оглянуті спеціалістом. Відповідно до цього стан автомобіля змінюється, зробив це виділенням кольору:

- жовтий - автомобіль, ще не був оглянутий експертом.
- зелений - автомобіль був оглянутий експертом, оцінка авто проведена, заявлена ціна власника нижча ніж ціна експерта - можна купляти.
- червоний - автомобіль був оглянутий експертом, але через певні проблеми з авто або ж завищену вартість авто не рекомендується до покупки.

### Огляди авто

Ідентифікатор	Модель автомобіля	Ціна продавця	Ім'я власника	Номер телефону власника	Додано	Дії
2	Audi - A6	800.0	Петро	+380961359945	Mon, 27 May 2023 16:46:35 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
3	Daewoo - Matiz	1900.0	Любов	+380934615523	Mon, 27 May 2023 16:47:09 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
4	Opel - Vectra B	2250.0	Данило	+380734916637	Mon, 27 May 2023 16:47:41 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
1	Skoda - Octavia	500.0	Вася	+380684613325	Mon, 27 May 2023 15:40:20 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
5	Volkswagen - Caddy	6800.0	Віталій	+380503645529	Mon, 27 May 2023 16:48:57 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
6	Volkswagen - B5	5900.0	Сяген	+380664513368	Mon, 27 May 2023 16:49:25 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
7	Peugeot - Partner	5900.0	Богдан	+380976727454	Mon, 27 May 2023 16:50:02 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
8	Skoda - Octavia	6900.0	Дмитро	+380986595516	Mon, 27 May 2023 16:50:43 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
9	Mercedes - Benz - C 220	1600.0	Богдан	+380995613315	Mon, 27 May 2023 16:51:25 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
10	Seat - Ibiza	6500.0	Нестор	+380684613579	Mon, 27 May 2023 16:52:17 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>

[Створити](#)

Рис.2 Огляди автомобілів

Ось так виглядає меню спеціаліста, яке він заповнює після огляду авто.

### Редагувати Огляд

Модель авто: Volkswagen - Caddy

Ціна від продавця: 6800.0

Ім'я власника: Віталій

Номер телефону власника: +380503645529

Ціна від спеціаліста: 6400

Коментар: Підткас масло, надрана подушка мотору, дві вмятини на лівому крилі.

Не битий

Не фарбована

Гаражне зберігання

Не розмитнений

Сервісна книжка

Авто в кредит

Потребує ремонту

Індивідуальна комплектація

[Update Review](#) [Відміна](#)

Рис.3 Меню спеціаліста, який проводить огляд авто

Після заповнення огляду авто, статус автомобіля змінюється.

### Огляди авто

Ідентифікатор	Модель автомобіля	Ціна продавця	Ім'я власника	Номер телефону власника	Додано	Дії
2	Audi - A6	800.0	Петро	+380961359945	Mon, 27 May 2023 16:46:35 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
3	Daewoo - Matiz	1900.0	Любов	+380934615523	Mon, 27 May 2023 16:47:09 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
4	Opel - Vectra B	2250.0	Данило	+380734916637	Mon, 27 May 2023 16:47:41 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
1	Skoda - Octavia	500.0	Вася	+380684613325	Mon, 27 May 2023 15:40:20 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
6	Volkswagen - B5	5900.0	Сяген	+380664513368	Mon, 27 May 2023 16:49:25 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
7	Peugeot - Partner	5900.0	Богдан	+380976727454	Mon, 27 May 2023 16:50:02 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
8	Skoda - Octavia	6900.0	Дмитро	+380986595516	Mon, 27 May 2023 16:50:43 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
9	Mercedes - Benz - C 220	1600.0	Богдан	+380995613315	Mon, 27 May 2023 16:51:25 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
10	Seat - Ibiza	6500.0	Нестор	+380684613579	Mon, 27 May 2023 16:52:17 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
5	Volkswagen - Caddy	6800.0	Віталій	+380503645529	Mon, 27 May 2023 16:48:57 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>

[Створити](#)

Рис.4 Зміна статусу автомобіля після огляду

Ось так виглядає меню до та після перегляду звіту спеціаліста після огляду авто.

## Огляд

Ідентифікатор автом... 3  
 Ціна продавця: 5900.0  
 Ціна від спеціаліста:  
 Коментар:  
 Ім'я власника: Євген  
 Номер телефону вл... +380664513368

[Назад](#)
[Редагувати](#)
[Видалити](#)

Рис.5 Звіт до огляду спеціаліста

## Огляд

Ідентифікатор автом... 11  
 Ціна продавця: 6500.0  
 Ціна від спеціаліста: 6800.0  
 Коментар: Хороший стан автомобіля. Зауважень до ходової частини, мотору, коробки та кузова немає. Потрібна хімічистка салону.  
 Ім'я власника: Нестор  
 Номер телефону вл... +380664613579  
 Не битий: Так  
 Не фарбована: Так  
 Гаражне зберігання: Так  
 Розмитнений: Так  
 Сервісна книжка: Так  
 Авто в кредиті: Ні  
 Потребує ремонту: Ні  
 Індивідуальна комп... Так

[Назад](#)
[Редагувати](#)
[Видалити](#)

Рис.6 Звіт після огляду спеціаліста

Тут ми бачимо список автомобілів, які ми вже оглядали, або маємо оглянути.

## Моделі авто

Ідентифікатор	Назва	Виробник	Додано	Дії
1	Octavia	Skoda	Mon, 27 May 2023 15:39:24 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
2	A6	Audi	Mon, 27 May 2023 16:34:30 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
3	B5	Volkswagen	Mon, 27 May 2023 16:34:48 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
4	Lanos	Daewoo	Mon, 27 May 2023 16:35:08 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
5	Partner	Peugeot	Mon, 27 May 2023 16:37:35 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
6	Fabia	Skoda	Mon, 27 May 2023 16:37:54 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
7	C 220	Mercedes - Benz	Mon, 27 May 2023 16:38:14 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
8	Caddy	Volkswagen	Mon, 27 May 2023 16:38:36 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
9	Matiz	Daewoo	Mon, 27 May 2023 16:40:03 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
10	Vectra B	Opel	Mon, 27 May 2023 16:40:16 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
11	Ibiza	Seat	Mon, 27 May 2023 16:51:49 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>

[Створити](#)

**Рис.7** Список автомобілів, які вже пройшли огляд або мають пройти огляд  
Нижче наведено приклад, як можна додати нову марку автомобіля та редагувати її.

#### Створити Модель Авто

Назва моделі

Виробник

[Create Car model](#) [Відміна](#)

**Рис.8** Меню додавання нових марок та моделей автомобілів

#### Редагувати Модель Авто

Назва моделі

Виробник

[Update Car model](#) [Відміна](#)

**Рис.9** Меню редагування марок та моделей авто

Нижче ми бачимо список запитань, які є за замовчуванням або які слід запитувати при огляді певної моделі авто.

#### Запитання

Ідентифікатор	Питання	Модель авто	Додано	Дії
3	Не фарбована		Mon, 27 May 2023 16:41:08 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
4	Гаражне зберігання		Mon, 27 May 2023 16:41:38 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
5	Проблеми з електрикою?	Volkswagen - B5	Mon, 27 May 2023 16:42:04 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
7	Сервісна книжка		Mon, 27 May 2023 16:43:40 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
8	Авто в кредиті		Mon, 27 May 2023 16:43:55 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
9	Потребує ремонту		Mon, 27 May 2023 16:44:13 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
2	Проблеми з втягуючим?	Skoda - Octavia	Mon, 27 May 2023 15:39:55 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
10	Гнилий кузов?	Mercedes - Benz - C 220	Mon, 27 May 2023 16:45:03 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
11	Індивідуальна комплектація		Mon, 27 May 2023 16:45:29 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
1	Не битий		Mon, 27 May 2023 15:39:44 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
6	Розмитнений		Mon, 27 May 2023 16:42:50 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>

[Створити](#)

**Рис.10** Список запитань при огляді певної моделі авто

Так виглядає меню додавання запитань. Слід написати бажане запитання та обрати чи це питання за замовчуванням чи до певної моделі авто.

## Створити Question

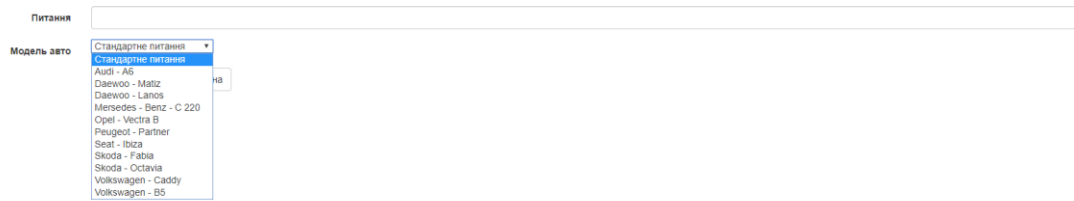


Рис.11 Меню додавання запитань

### 3.2. Створення телеграм бота

Нам потрібно створити телеграм-бота, який буде слугувати в якості користувацького інтерфейсу для нашої програми.

Боти - це спеціальні акаунти в Telegram, які можуть автоматично обробляти і відправляти повідомлення. Вони створюються програмістами і працюють у них на сервері. Боти можуть виконувати практично будь-які завдання, які може робити кожен користувач акаунта Telegram з онлайн-сервісами. По суті, боти – це такий собі зручний для людини інтерфейс роботи з різноманітними веб-службами. [2]

BotFather - це найпростіший спосіб для реєстрації, настройки та управління іншими Telegram ботами. З допомогою BotFather можна зареєструвати не обмежену кількість нових ботів. Єдиною умовою реєстрації нового бота є його унікальне ім'я.

Токен (*ключ доступу*) — унікальний випадково згенерований ключ, призначений для забезпечення безпечного віддаленого доступу до інформаційних ресурсів Telegram для нашої програми.

Для створення бота в Telegram, знайдемо і додамо до своїх контактів @BotFather і введемо команду /start. [4]



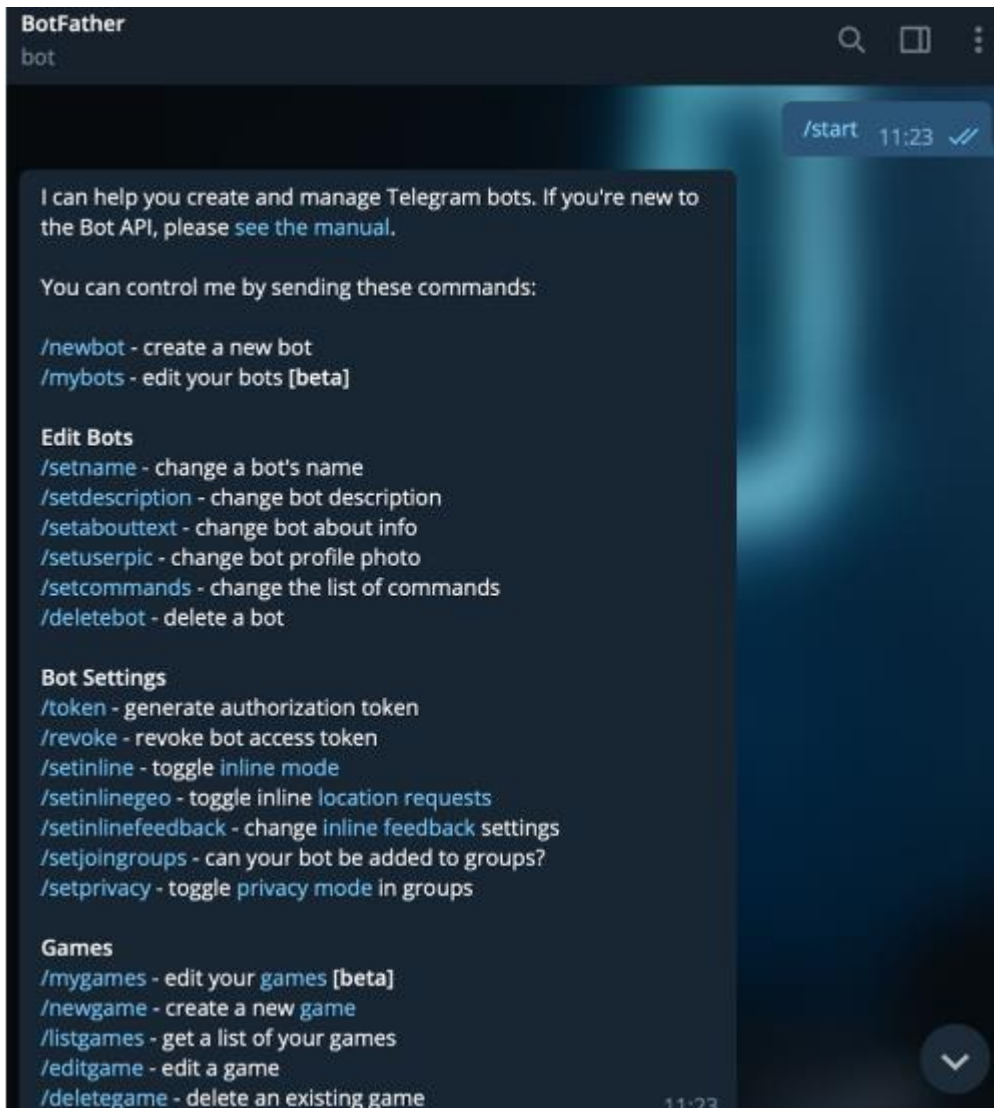


Рис.12 Початкове меню BotFather

Тепер напишемо команду `/newbot` для створення нового бота.

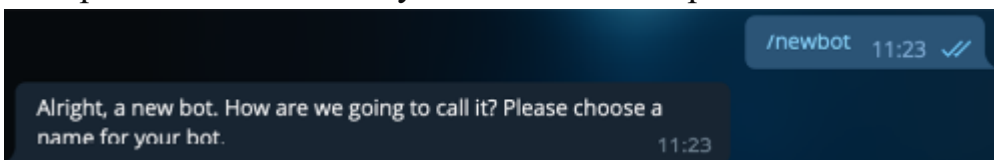


Рис.13 Створення ім'я для бота

Нашого бота назвемо - "CarChecker27\_bot". За цією назвою ми і знайдемо його в Telegram.

Тут ми можемо бачити принцип роботи Telegram бота:



Рис.14 Принцип роботи Telegram бота

Перш ніж наша програма зможе опрацьовувати повідомлення отримані нашим ботом, ми повинні зареєструвати нашу програму як обробник на стороні Telegram (п.1 з рис. 14). Реєструємо в Telegram сервері обробника для повідомлень з допомогою виконання наступного скрипта:

```
bin/rake telegram:bot:set_webhook RAILS_ENV=production
```

Цей скрипт виконує запит до Telegram API з використанням токена отриманого при реєстрації бота і встановлює наш веб-сервер в якості обробника повідомлень для нашого бота.

Після реєстрації нашого обробника повідомлень з мобільного телефону заходимо в Telegram бот та вводимо команду “/list” (п.2 з рис.14), після чого телеграм сервер відправляє дані на сервер Rails (п.3 з рис.14), опрацьовує команду та повертає список автомобілів для огляду спочатку на телеграм сервер (п4. з рис.14) , а потім телеграм повертає цей список на мобільний додаток (п.4 з рис.14).

Розпочинаємо роботу з нашим ботом командою “/start”



**Sharbel**  
/start



**CarChecker**  
Дипломний проект  
студента групи ПМО-41  
Алхажж Шарбея

Для початку роботи використайте команду `/list`

До огляду наш веб сервіс показує машини які потрібно оглянути спеціалісту.

Огляди авто

Ідентифікатор	Модель автомобіля	Ціна продавця	Ім'я власника	Номер телефону власника	Додано	Дії
14	Audi - A6	11300.0	Петро	+380961359945	Tue, 09 Jul 2023 12:12:21 +0000	<a href="#">Перегляд</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>
15	Daewoo - Lanos	2400.0	Данило	+380934615523	Tue, 09 Jul 2023 12:12:46 +0000	<a href="#">Перегляд</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>
17	Mercedes - Benz - C 220	5600.0	Микола	+380503645529	Tue, 09 Jul 2023 12:13:30 +0000	<a href="#">Перегляд</a>   <a href="#">Редагувати</a>   <a href="#">Видалити</a>

[Створити](#)

Рис.15 Вигляд веб сервісу до огляду автомобіля спеціалістом

## Огляд

Ідентифікатор автом... 2  
 Ціна продавця: 11300.0  
 Ціна від спеціаліста:  
 Коментар:  
 Ім'я власника: Петро  
 Номер телефону вл... +380961359945

[Назад](#)

[Редагувати](#)

[Видалити](#)

Рис.16 Вигляд меню до огляду спеціалістом

Робимо виклик команди “/list”



**Sharbel**

/list



**CarChecker**

Виберіть авто для огляду

Mercedes - Benz - C 220 | 5600.0 | Микола | +380503645529

Audi - A6 | 11300.0 | Петро | +380961359945

Seat - Ibiza | 5900.0 | Віталій | +380734916637

Рис.17 Виклик команди “/list”

Бот повертає нам список машин, які потрібно оглянути. Вибираємо наприклад “А6” від продавця Петра та проводимо огляд машини.

Не битий

ТАК

НІ

Не фарбована

ТАК

НІ

Гаражне зберігання

ТАК

НІ

Коментар після огляду

ВН

**Sharbel**

Авто в гарному стані, єдиний мінус потребує заміни гальмівних колодок. Орієнтовна ціна запчастин з ремонтом 200\$.

СА

**CarChecker**

Рекомендована ціна після огляду

ВН

**Sharbel**

11000

Рис.18 Огляд авто спеціалістом

Після огляду автомобіля спеціалістом бот виводить нам звіт з огляду автомобіля.



## CarChecker

Audi - A6 | 11300.0 | Петро | +380961359945

Рекомендована ціна після огляду: 11000.0

Не битий: ТАК

Не фарбована: ТАК

Гаражне зберігання: ТАК

Розмитнений: ТАК

Сервісна книжка: ТАК

Авто в кредиті: НІ

Потребує ремонту: НІ

Індивідуальна комплектація: ТАК

Авто в гарному стані, єдиний мінус потребує заміни гальмівних колодок. Орієнтовна ціна запчастин з ремонтом 200\$.

Рис.19 Звіт після завершення огляду автомобіля

Після завершення огляду на нашому сервері появились збережені зміни.

Огляди авто

Ідентифікатор	Модель автомобіля	Ціна продавця	Ім'я власника	Номер телефону власника	Додано	Дії
14	Audi - A6	11300.0	Петро	+380961359945	Tue, 09 Jun 2023 12:12:21 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
15	Daewoo - Lanos	2400.0	Данило	+380934615523	Tue, 09 Jun 2023 12:12:46 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
17	Mercedes - Benz - C 220	9600.0	Микола	+380503645529	Tue, 09 Jun 2023 12:13:30 +0000	<a href="#">Перегляд</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>

[Створити](#)

Рис.20 Вигляд веб сервісу після огляду автомобіля спеціалістом

## Огляд

Ідентифікатор автом...	2
Ціна продавця:	11300.0
Ціна від спеціаліста:	11000.0
Коментар:	Авто в гарному стані, єдиний мінус потребує заміни гальмівних колодок. Орієнтовна ціна запчастин з ремонтом 200\$.
Ім'я власника:	Петро
Номер телефону вл...	+380961359945
Не битий:	Так
Не фарбована:	Так
Гаражне зберігання:	Так
Розмитнений:	Так
Сервісна книжка:	Так
Авто в кредиті:	Ні
Потребує ремонту:	Ні
Індивідуальна комп...	Так

[Назад](#) [Редагувати](#) [Видалити](#)

Рис.21 Вигляд меню після огляду спеціалістом

Нижче наведено діаграму, як працює логіка обробника повідомлень.

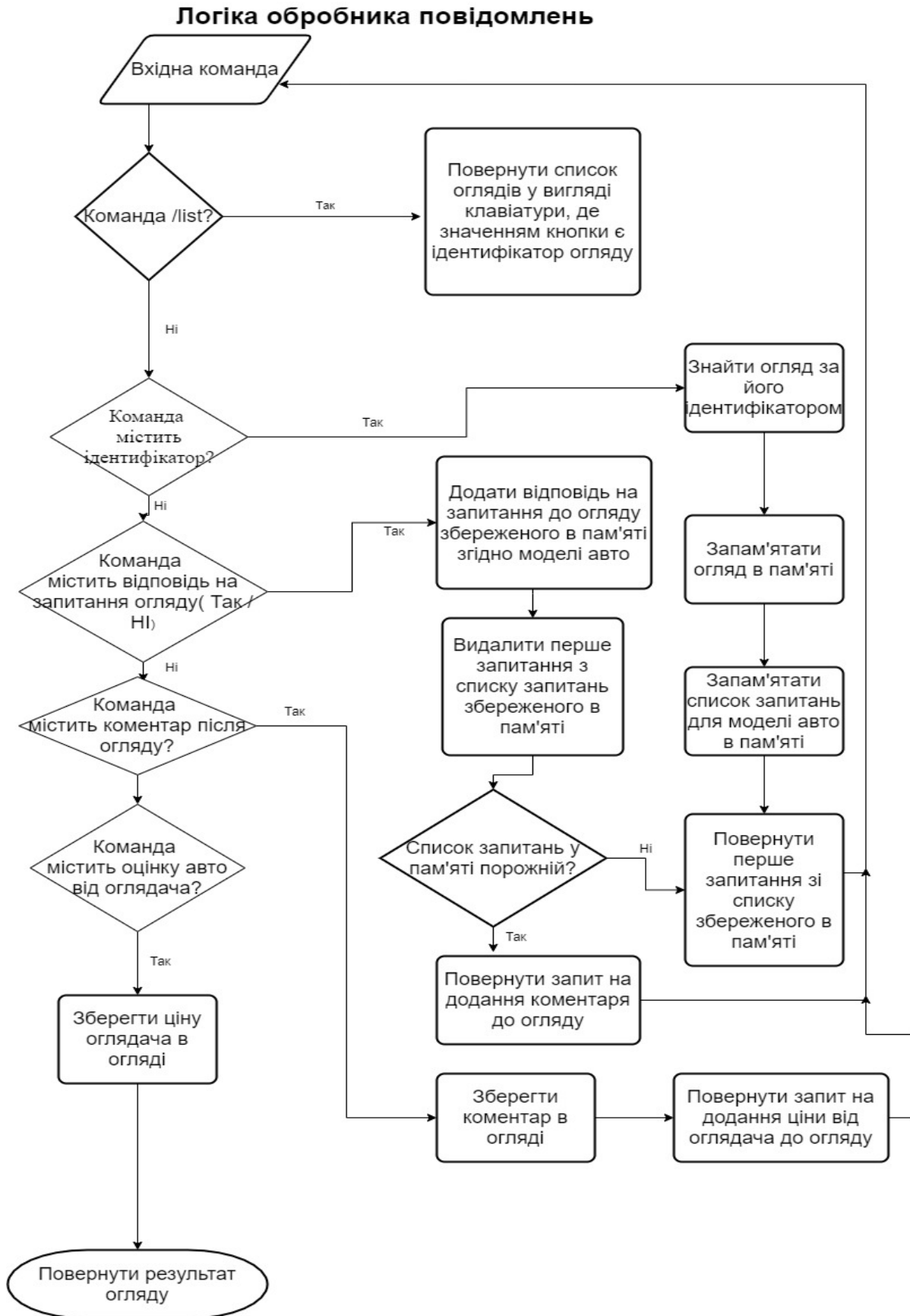


Рис.22 Логіка обробника повідомлень

## ВИСНОВКИ

1. У ході дослідження та розробки програмного застосунку та телеграм чат-бота для оцінки стану автомобілів, було виявлено, що Ruby on Rails є потужним фреймворком для швидкої та ефективної розробки веб-додатків. Він надавав зручні інструменти для реалізації функціональності, забезпечував безпеку даних та сприяв забезпеченню високої продуктивності.

2. Розроблений програмний застосунок дозволяє користувачам оцінювати стан автомобілів, надаючи зручний інтерфейс для введення необхідних даних та отримання результатів. Застосунок забезпечує широкі можливості для персоналізації та налаштування оцінювальних критеріїв залежно від потреб користувача.

3. Телеграм чат-бот, розроблений у рамках дипломної роботи, дозволяє користувачам отримувати оцінку стану автомобілів безпосередньо у месенджері. Він використовує Ruby on Rails для зв'язку з програмним застосунком та обробки запитів користувачів. Чат-бот є зручним та доступним інструментом для швидкого отримання інформації про стан автомобіля.

4. Під час розробки були використані сучасні методи та практики програмування, зокрема використання патерну MVC (Model-View-Controller) для відокремлення логіки, дизайну та бази даних. Це сприяло зрозумілості та легкості розробки, підтримки та розширення проекту.

5. Результати тестування програмного застосунку та телеграм чат-бота показали їхню стабільність, надійність та відповідність функціональним вимогам. Виявлені помилки були виправлені, а ефективність роботи була підтверджена при великому обсязі введених даних та паралельних запитах користувачів.

6. Застосування Ruby on Rails для розробки програмного застосунку та телеграм чат-бота для оцінки стану автомобілів підтвердило його переваги, такі як швидкість розробки, легкість у використанні та забезпечення безпеки. Це робить його відмінним вибором для подібних проектів.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ruby on Rails Guides. Available from : <https://guides.rubyonrails.org>
2. Help and documentation for the Ruby programming language. Available from : <https://ruby-doc.org/>
3. Чат боти. Available from : <https://uk.wikipedia.org/wiki/%D0%A7%D0%B0%D1%82-%D0%B1%D0%BE%D1%82>
4. Learn to program on Ruby. Режим доступу: <https://www.codecademy.com/learn/learn-ruby>
5. Quickly create a Telegram Bot in Ruby. Available from : <https://www.sitepoint.com/quickly-create-a-telegram-bot-in-ruby/>
6. Гарсиа-Молина Г. Ульман Д. Уидом Д., “Системы баз данных. Полный курс” 2003 - 1088с. Available from: <http://www.sao.ru/hq/zhe/DDBMS/book/GarsiaMolinaDBMS.pdf>
7. Michael Hartl, “Ruby on Rails Tutorial” , - 2500 p. Available from : <https://www.railstutorial.org/book>