

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

ПОЯСНЮВАЛЬНА ЗАПИСКА ДО КУРСОВОЇ РОБОТИ
із дисципліни «Бази даних та інформаційні системи»
на тему
Оренда автомобіля

Виконав:
студент групи КМ-42
Камінський Н.-М. А.

Керівник:

Терещенко І. О.

ЗМІСТ

Вступ	6
1 ПЕРЕДПРОЕКТНЕ ДОСЛІДЖЕННЯ	7
1.1 Мета	7
1.2 Граничні умови	7
1.3 Ролі користувачів	7
1.4 Функції	8
2 ПОСТАНОВКА ЗАДАЧІ	10
2.1 Визначення підгруп користувачів.....	10
2.1.1 Неавторизовані користувачі	10
2.1.2 Авторизований клієнт	11
2.1.3 Авторизований клієнт	11
2.1.4 Водій	12
2.1.5 Адміністратор інформаційної системи.....	12
2.2 Визначення підгрупи даних	12
2.3 Бізнес-правила	13
2.4 Діаграма послідовностей	14
3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ.....	18
3.1 Use-case	18
3.2 Component diagram	23
3.3 Scrum. Sprint Planning	26
4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ.....	34
4.1 Користувач.....	35
4.2 Роль.....	36

	3
4.3 Заявка	36
4.4 Стан	36
4.5 Фото	37
4.6 Коментарі	37
5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ.....	38
Висновки	40
Перелік посилань	41

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Основним завданням курсової роботи є проектування та розробка інформаційної системи "Оренда автомобіля".

У курсовій роботі потрібно правильно визначити сутності та їх атрибути. Щодо проектування потрібно розробити концептуальну, логічну та фізичну моделі, компонентну діаграму та діаграму послідовностей . Створити відповідну базу даних, що віддповідає моделям.

Система повинна спростити процес орендування автомобілів. Для цього потрібно розробити веб-сайт, що буде представляти дану інформаційну систему та реалізовувати "клієнт-серверну" організацію.

На розробку інформаційної системи не будуть витрачені ніякі кошти та не будуть залучені жодні інвестиції.

АНОТАЦІЯ

Тема курсової роботи - створення інформаційної системи для оренди автомобілів.

Для створення повноцінної системи було проведено декілька етапів передпроектне дослідження, побудова плану роботи за допомогою SCRUM (створення спринтів та завдань), визначення Use Cases, розроблення діаграми компонент та діаграми послідовстей для кожної ролі користувачів. проектування моделі системи (концептуальна, логічна та фізична діаграми), створення бази даних, розробка клієнтської частини.

Розробку програмного забезпечення було здійснено за допомогою наступних технологій: PowerDesigner, PHP, HTML, CSS, BootStrap, Apache, Oracle 11g.

Оскільки система призначена для оренди автомобілів, то ключові слова з предметної області: орендодавець, орендар, угода, заявка, веб-сайт, база даних, інформаційна система ІС.

ВСТУП

Все більше і більше часу люди проводять в інтернеті, щоб знайти відповіді на різні питання. Вже стало звичайним здійснювати покупки та укладати угоди через інтернет. Тому було б дивним не використати цей потужний потенціал, для вирішення однієї з багатьох людських потреб, а саме оренда автомобіля через інтернет.

Перш за все, що цікавить користувачів це простий та доступний функціонал системи, що надає повний набір можливостей по виборі найкращого автомобіля.

Темою курсового проекту є розробка інформаційної системи, що спрощує процес оренди автомобіля. Далі розглянуто основні етапи створення інформаційної системи за цією темою.

1 ПЕРЕДПРОЕКТНЕ ДОСЛІДЖЕННЯ

1.1 Мета

Спрощення та автоматизація процесу оренди автомобіля. Створити можливість оптимального підбору найкращого автомобіля для користування за вказаними параметрами користувача: ціна, час користування, призначення поїздки, марка та модель автомобіля, послуги водія. Реалізувати зручний та інтуїтивно зрозумілий інтерфейс.

1.2 Граничні умови

Оренда доступного на даний момент автомобіля на території м.Києва особами, що старше 21 року та досвідом водіння більше 2 років на термін від 1 дня до 1 місяця.

1.3 Ролі користувачів

- Адміністратор – має права на додавання, редагування, видалення заявок про оренду автомобіля від орендодавців (власників автомобілів) та вже існуючих пропозицій;

- Орендодавець – має права реєстрації та авторизації в системі, надсилання заявок та поправок (до вже існуючих пропозицій) про оренду автомобілю, отримання сповіщення про бронювання автомобілю, підтвердження оренди автомобіля (укладання електронного договору з користувачем), додавання скарги на користувача;

- Користувач – має права реєстрації та авторизації в системі, перегляд

пропозицій про оренду автомобілів, бронювання автомобіля, підтвердження оренди (укладання електронного договору з орендодавцем), отримання підказок про оптимальний вибір автомобіля, голосування, додавання скарги на орендодавця;

1.4 Функції

а) Реєстрація користувача (ПІБ, e-mail, пароль, номер телефону, тип (клієнт чи орендодавець))

- час виконання: менше 1 хв;
- джерело інформації: електронна форма-реєстрації з характеристиками;
- структура даних: json файл з відповідними параметрами (з форми реєстрації);
- збереження інформації в БД.

б) Авторизація користувача (e-mail, пароль)

- час виконання: менше 1 хв;
- джерело інформації: електронна форма-авторизації;
- структура даних: POST запит;
- зчитування інформації з БД.

в) Подання заявки про доступний автомобіль (id-орендодавця, характеристики автомобіля (марка, модель, рік випуску, колір, стан (по 10-бальній шкалі, дата останнього ТО, витрати пального, період доступності авто, ціна за оренду в день), наявність страхування, фото (до 2 Mb одне фото))

- час виконання: менше 2 хв;

- джерело інформації: електронна форма-заявка з характеристиками автомобіля (параметри даної функції);

- структура даних : json файл з вище зазначеними характеристиками;
- збереження даних в БД.

г) Затвердження (Додавання) нового автомобіля (id-заявки на додавання)

- час виконання: менше 1 хв;

- джерело інформації : електронні заявки орендодавців;
- структура даних: json файл з полями відповідно до заявки;
- збереження даних в БД.

д) Перегляд доступних пропозицій на оренду автомобілів (обмеження по характеристиках автомобіля, що задає користувач)

- час виконання: менше 1 хв;
- джерело інформації: БД ;
- структура даних: список доступних автомобілів;
- зчитування інформації з БД.

е) Бронювання доступного автомобіля (id-орендодавця, id – автомобіля, id – користувача, термін оренди, локація та час отримання-передачі автомобіля)

- час виконання: менше 2 хв;
- джерело інформації: БД та параметри оренди з електронної форми (термін оренди, локація та час отримання-передачі автомобіля);
- збереження інформації в БД, виклик функції сповіщення орендодавця про бронь.

є) Підтвердження оренди автомобіля (id-орендодавця, id – автомобіля, id – користувача)

- час виконання: менше 1 хв;
- джерело інформації: БД та параметри функції;
- структура даних: список [користувач, орендодавець, електронний договір];
- збереження інформації в БД, виклик функції сповіщення орендодавця та клієнта про укладання електронного договору.

2 ПОСТАНОВКА ЗАДАЧІ

Основна мета даного проекту є створення повноцінної інформаційної системи (веб-сайт , бізнес логіка та база даних), що спрощує, пришвидшує та робить зручнішим процес оренди автомобіля. Дана інформаційна система повинна бути зручна, зрозуміла та виконувати ряд функцій:

- а) реалізувати повний процес оренди автомобіля;
- б) розділення ролей користувачів (клієнт, орендодавець, адміністратор);
- в) відстеження історії як клієнтів так і орендодавців.

2.1 Визначення підгруп користувачів

Дана інформаційна система передбачає 5-ть типів користувачів:

- а) неавторизовані користувачі;
- б) авторизовані клієнти;
- в) авторизовані орендодавці;
- г) водій;
- д) адміністратор інформаційної системи.

Розглянемо детільніше кожен із них.

2.1.1 Неавторизовані користувачі

Відповідно до роботи системи даний тип користувачів не має багато можливостей і може виконати лише базові функції. А саме, неавторизованим користувачам буде надано можливість:

- а) зареєструватись (відповідно до намірів клієнт або орендодавець);
- б) переглядати інформацію про наявні автомобілі;
- в) здійснювати пошук за певними критеріями.

2.1.2 Авторизований клієнт

Для авторизованих користувачів відбувається розділення на конкретні ролі, зокрема в даному випадку ролі клієнта. Авторизований клієнт має наступний додатковий функціонал, у порівнянні з неавторизованим:

- а) бронювання автомобілів;
- б) перегляд історії бронювань;
- в) залишення відгуків.

2.1.3 Авторизований клієнт

Для авторизованих користувачів відбувається розділення на конкретні ролі, зокрема в даному випадку ролі орендодавця. Авторизований клієнт має наступний додатковий функціонал, у порівнянні з неавторизованим:

- а) додавання заявок на автомобіль;
- б) редагування наявних заявок;
- в) видалення заявок;
- г) перегляд всіх поданих автомобілів;

2.1.4 Водій

Дану роль виконують люди що вже занесені в базу і надають свої послуги користувачам. Тобто, клієнт при оренді автомобіля може замовити також і водія.

2.1.5 Адміністратор інформаційної системи

Адміністратор ІС обмежений по функціоналу правами, що йому надає розробник ІС. Основні функції адміністратора:

- а) перегляд нових заявок;
- б) перегляд всіх заявок;
- в) підтвердження нових заявок;
- г) видалення нових заявок;
- д) видалення існуючих заявок;

2.2 Визначення підгрупи даних

Розглянемо дані, які потребує функціонал системи для різних користувачів:

а) роль - кожному користувачеві відводиться певна роль, яка має назву ("user "lessor "admin "driver") ;

б) клієнт (заповнює дану інформацію при реєстрації) - ім'я, прізвище, день народження, номер телефону, електронну пошту, серію водійського посвідчення та дату видачі;

в) орендодавець (заповнює дану інформацію при реєстрації) - ім'я, прізвище, день народження, номер телефону, електронну пошту, серію водійського посвідчення

та дату видачі;

г) заявка (на новий автомобіль) - модель, марка, номер кузова та мотору, рік випуску, дата останнього ТО, ціна оренди за день, фото (необов'язково);

д) договір (створюється системою при оренді) - інформація про клієнта, інформація про власника, час оренди, місце та час зустрічі ;

2.3 Бізнес-правила

Розглянемо основні бізнес-правила, що наявні у ІС оренди автомобілів.

До основних фактів, що формують бізнес-правила належать наступні твердження:

а) в системі можуть зареєструватись користувачі, що старші 21 року (юридична сторона) та з досвідом керування автомобілем не менше 2-х років;

б) орендодавець може додавати будь-яку кількість заявок;

в) користувач може орендувати будь-який доступний автомобіль на термін від 1 дня до 31 дня;

г) видалити зареєстрованого користувача може лише адмін;

д) видалити зареєстрованого користувача може лише адмін.

е) орендодавець може змінювати інформацію по машині , але тільки тоді коли вона не знаходиться в замовденні;

є) користувач може замовити водія;

ж) оплата провидиться при зустрічі та заключається письмовий договір для гарантії.

Очевидно, що заявка на машину (можна розуміти як саму машину, що здається в оренду) є основною в життєвому циклі орендування автомобіля. Вона має спіральну модель життєвого циклу, аж поки її не видалять (певні технічні причини).

Заявка має 4 стани:

а) створена (1);

- б) підтверджена (2);
- в) в замовленні (3);
- г) видалена (4).

Інформаційна система відслідковує ці стани, щоб клієнт не міг орендувати неперевірену машину (стан 1) чи зайняту машину (стан 3).

2.4 Діаграма послідовностей

Розглянемо діаграми послідовностей для незареєстрованих (рис.2.1) та зареєстрованих (рис.2.2) користувачів.

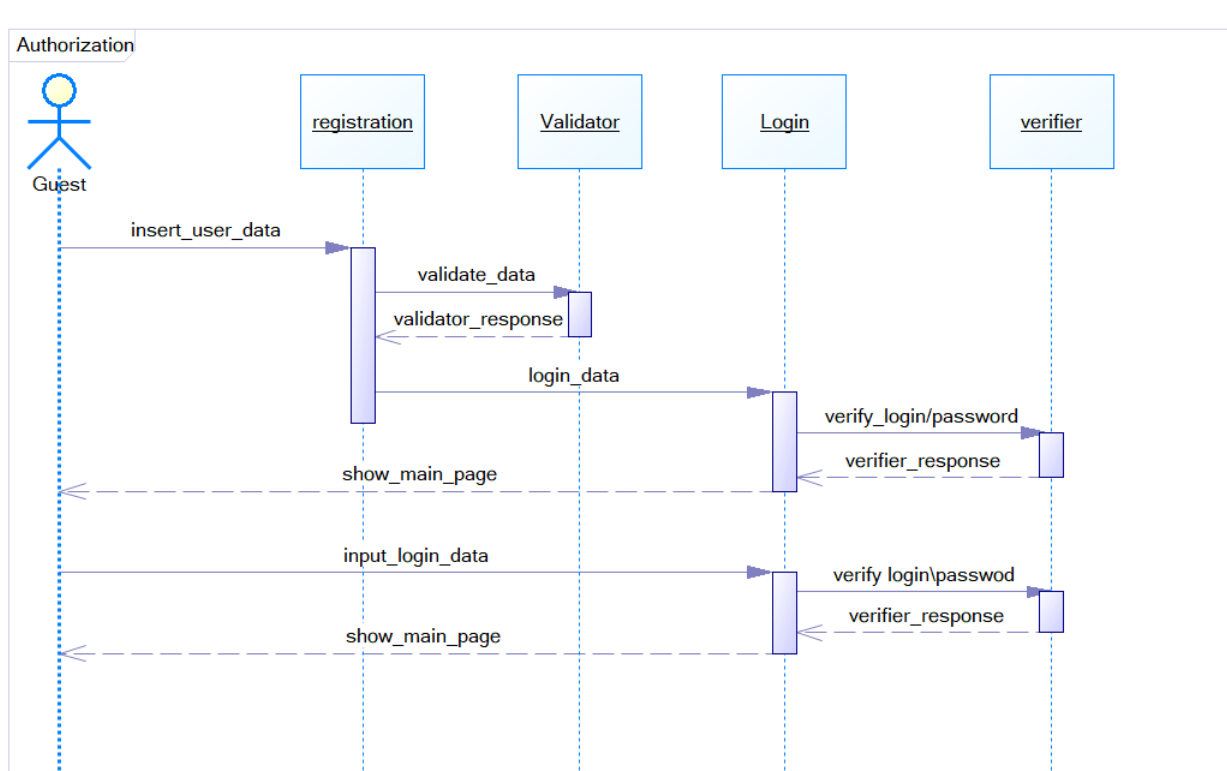


Рисунок 2.1 – Діаграма послідовностей для незареєстрованого користувача

Також розглянемо діаграми послідовностей для орендодавця (рис.2.3) та адміністратора (рис.2.4).

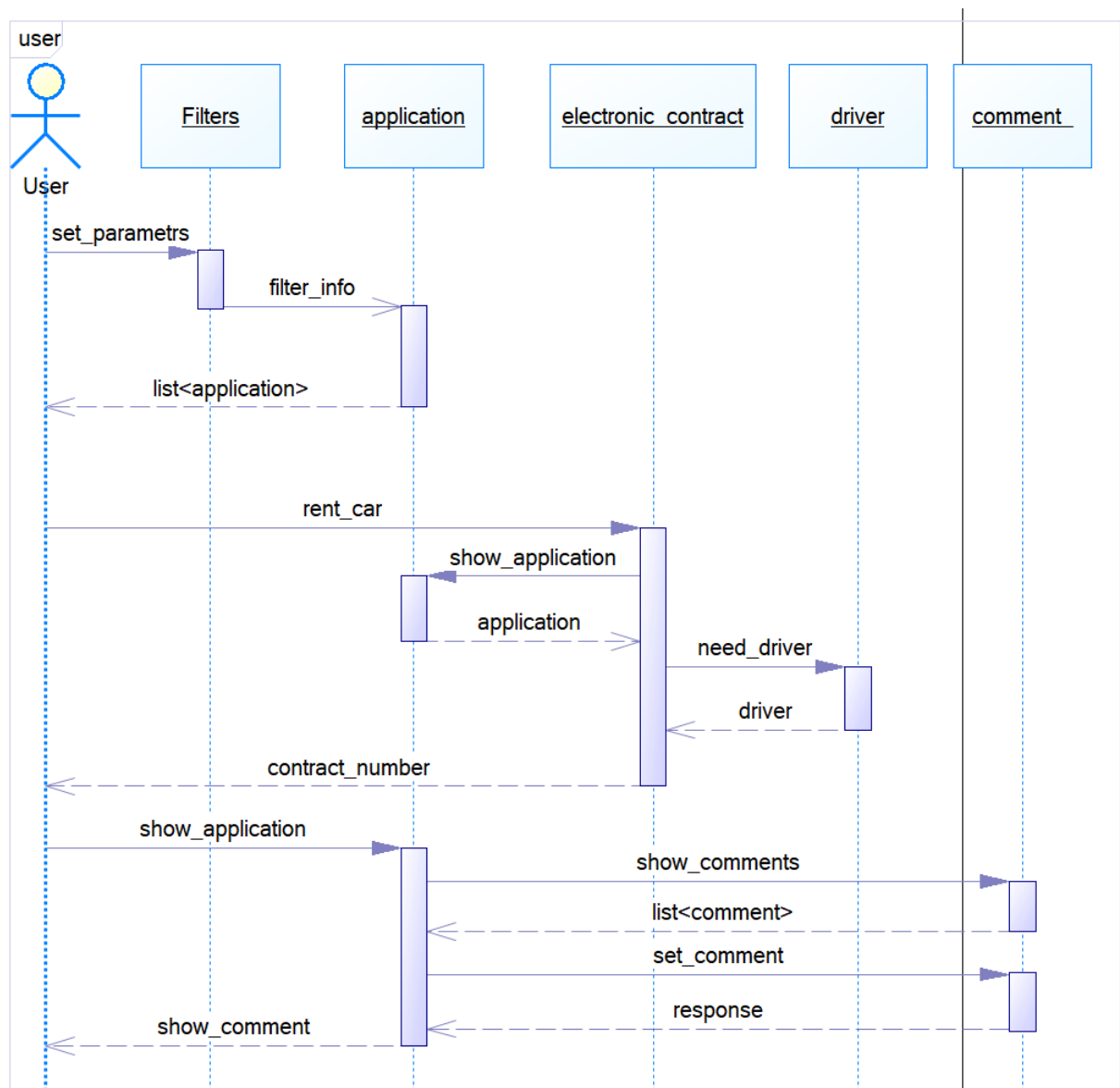


Рисунок 2.2 – Діаграма послідовностей для зареєстрованих користувачів

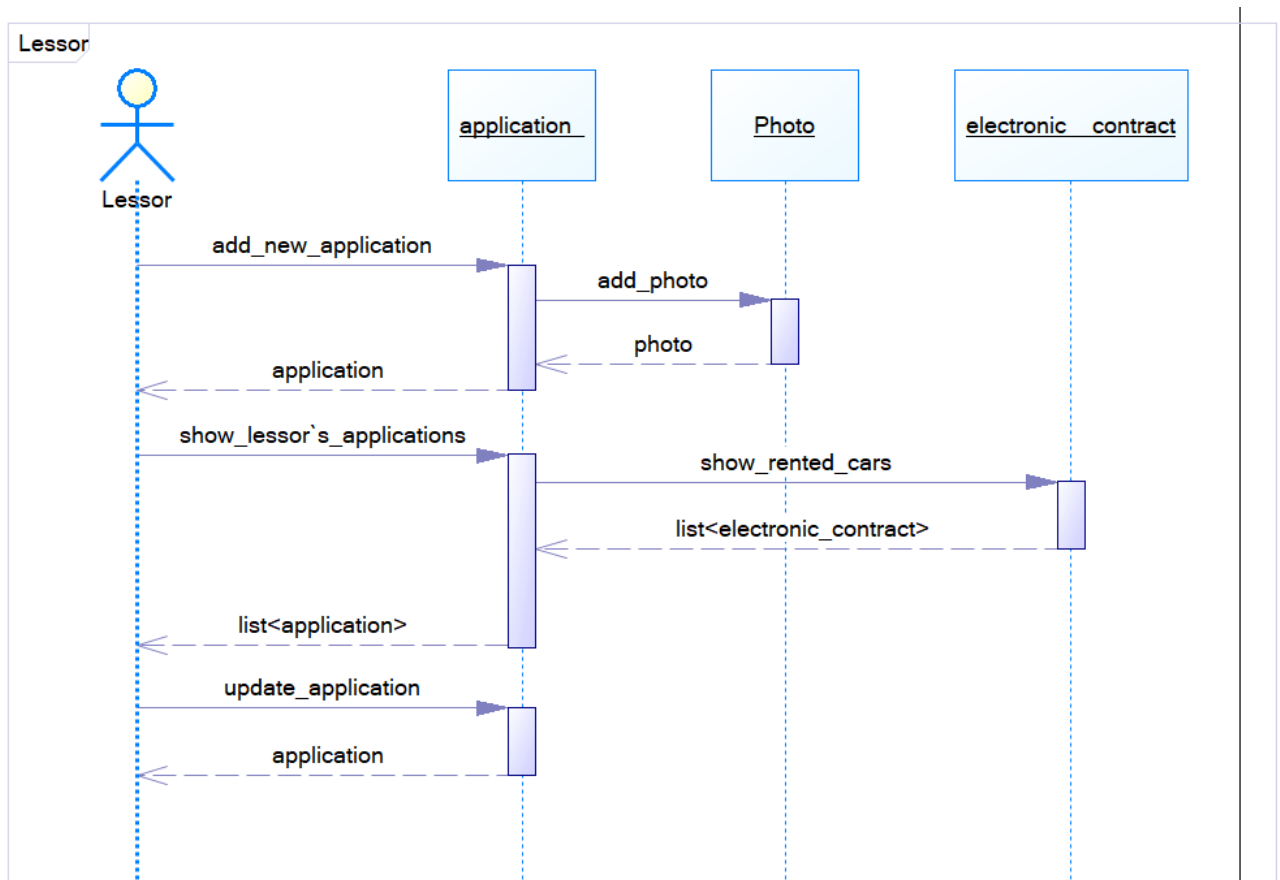


Рисунок 2.3 – Діаграма послідовностей для орендодавця

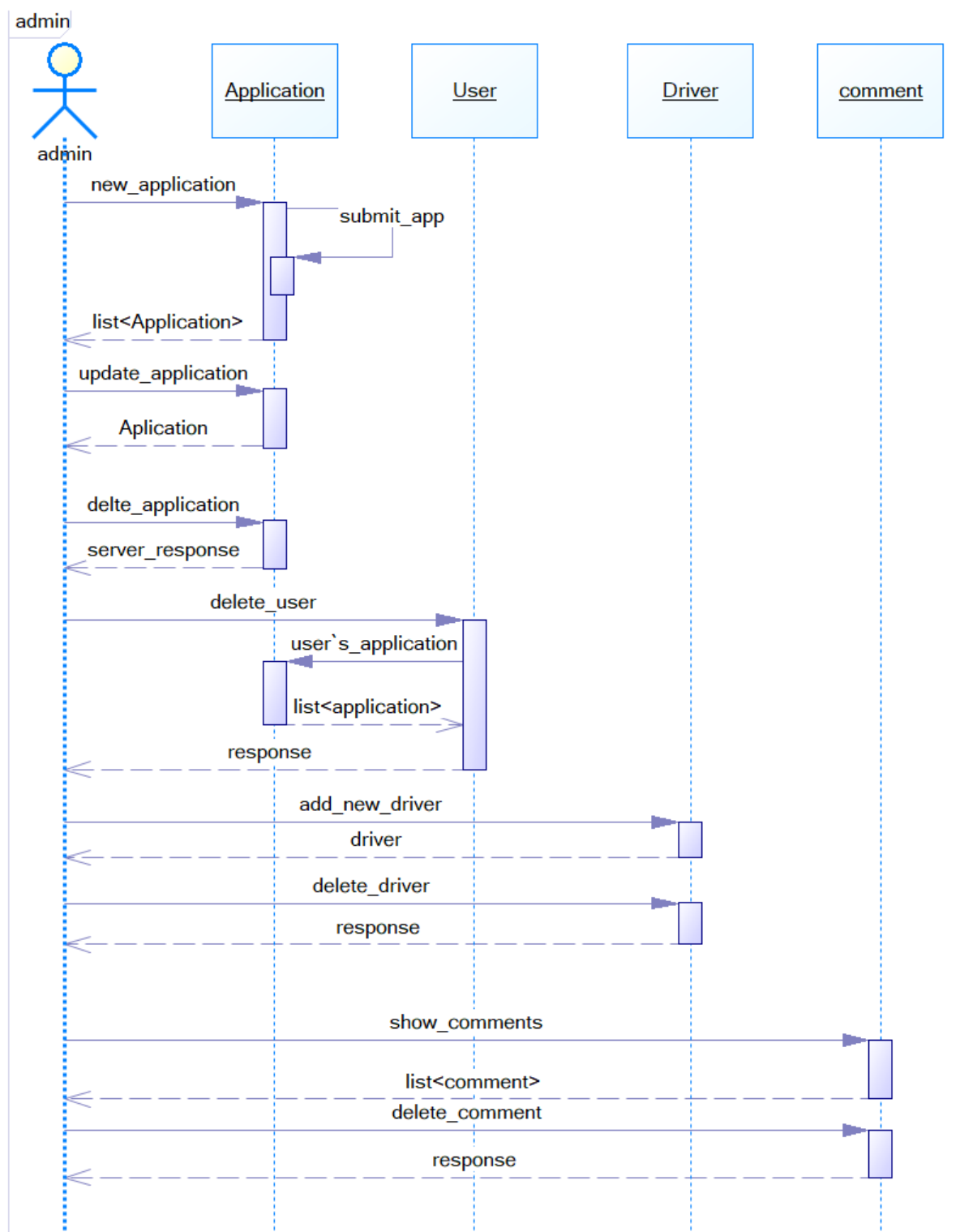


Рисунок 2.4 – Діаграма послідовностей для адміністратора

3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

3.1 Use-case

У системі передбачена робота трьох типів користувачів, що виконують різні ролі. Це клієнт, орендодавець та адміністратор (рис.3.1, рис.3.2, рис.3.3). Так як кожен з них має різні можливості, то для кожного з них існує свій use case[1] (рис.3.4, рис.3.5, рис.3.6, рис.3.7, рис.3.8, рис.3.9).

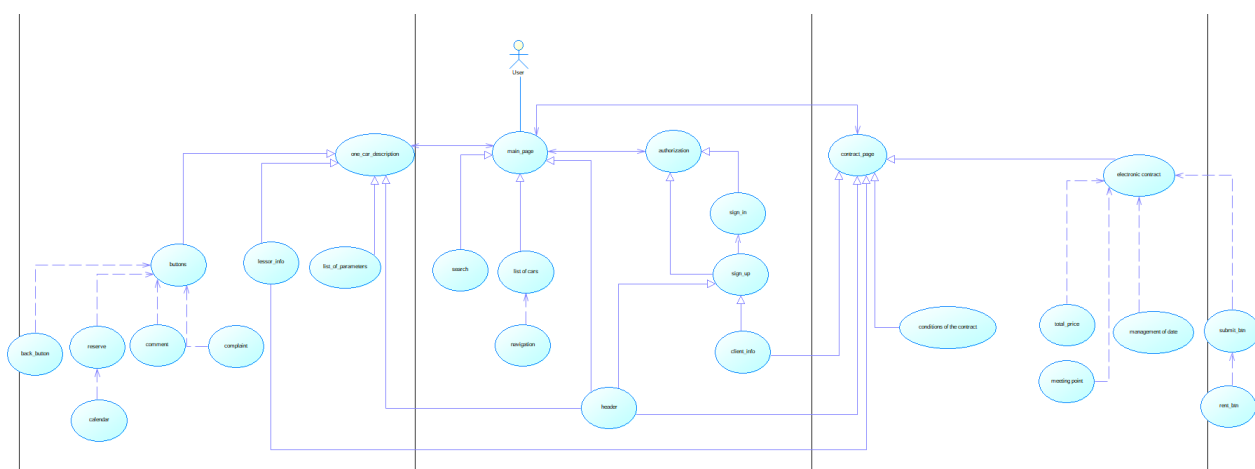


Рисунок 3.1 – Use case для користувача.

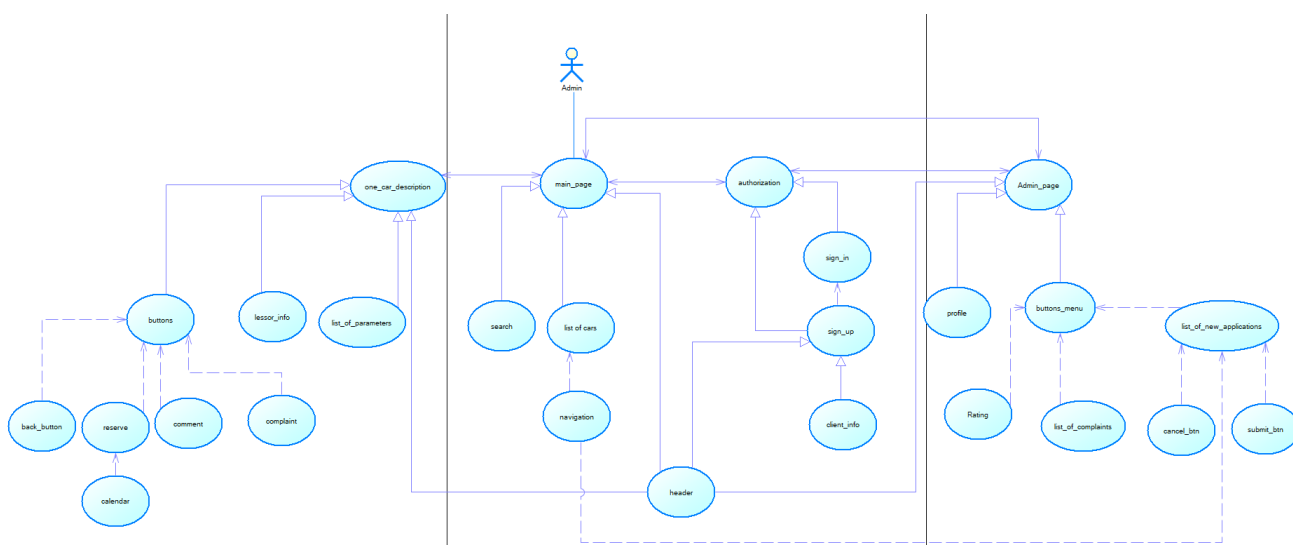


Рисунок 3.2 – Use case для адміністратора.

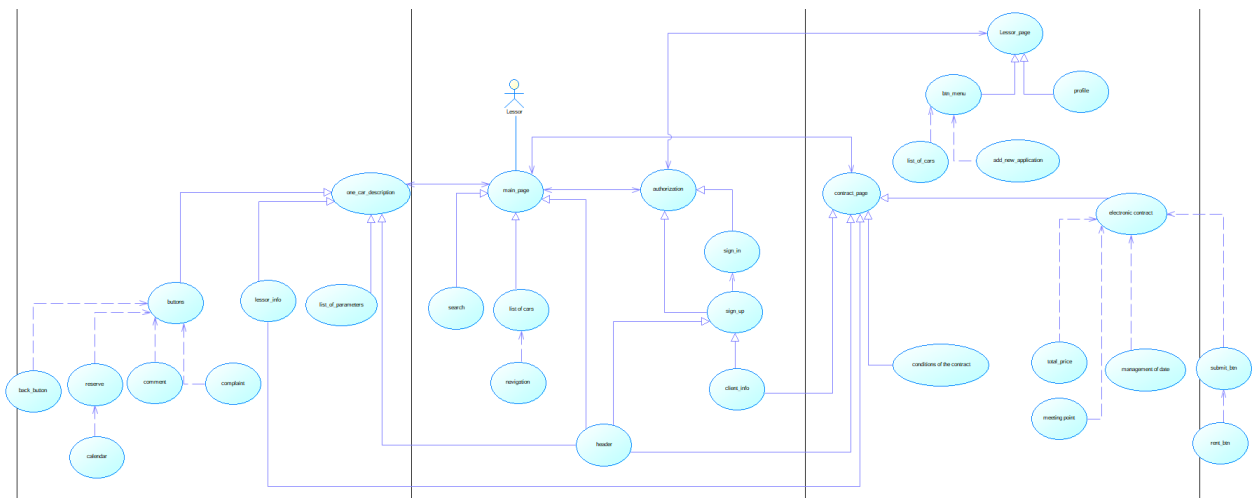


Рисунок 3.3 – Use case для орендодавця.

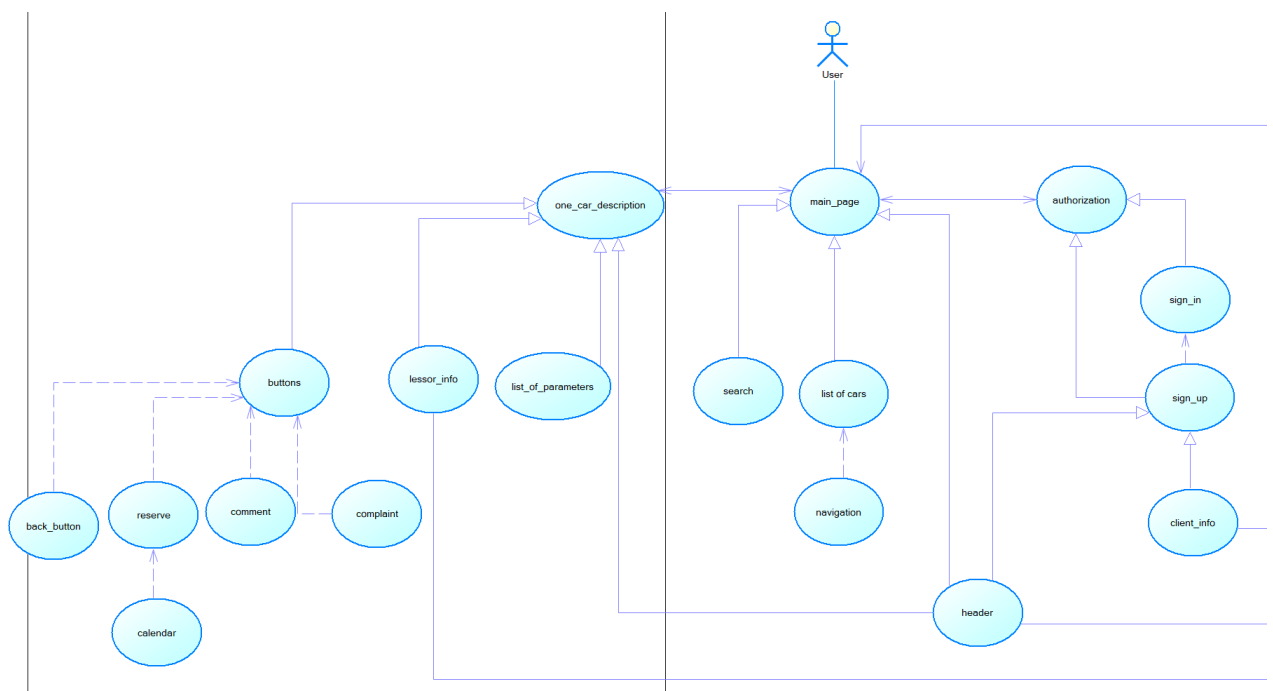


Рисунок 3.4 – Use case для користувача (збільшено).

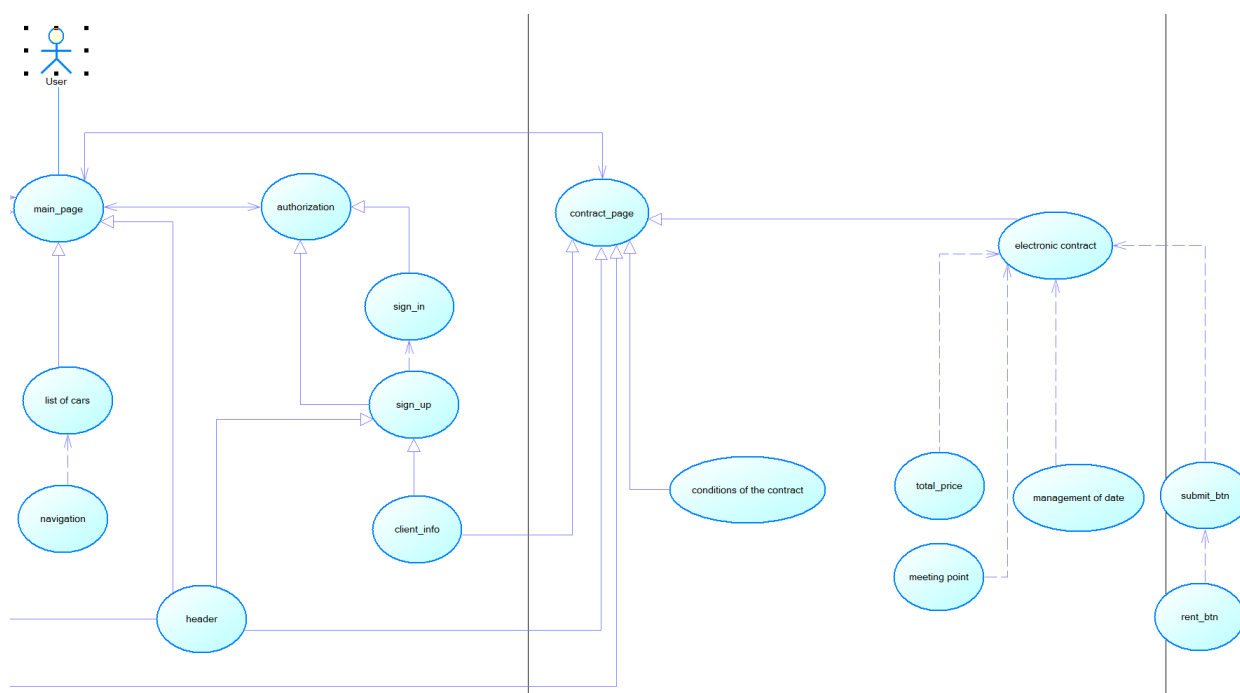


Рисунок 3.5 – Use case для користувача (збільшено).

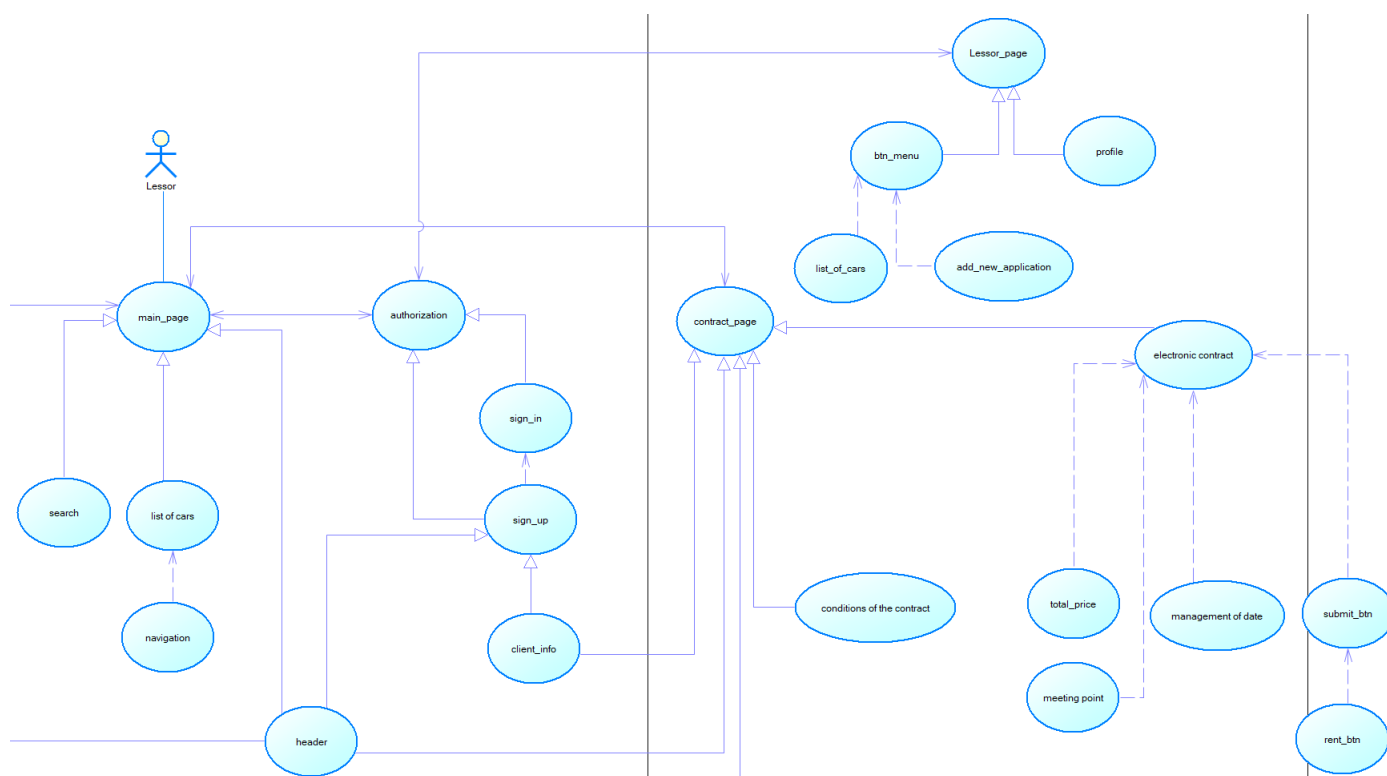


Рисунок 3.6 – Use case для орендодавця (збільшено).

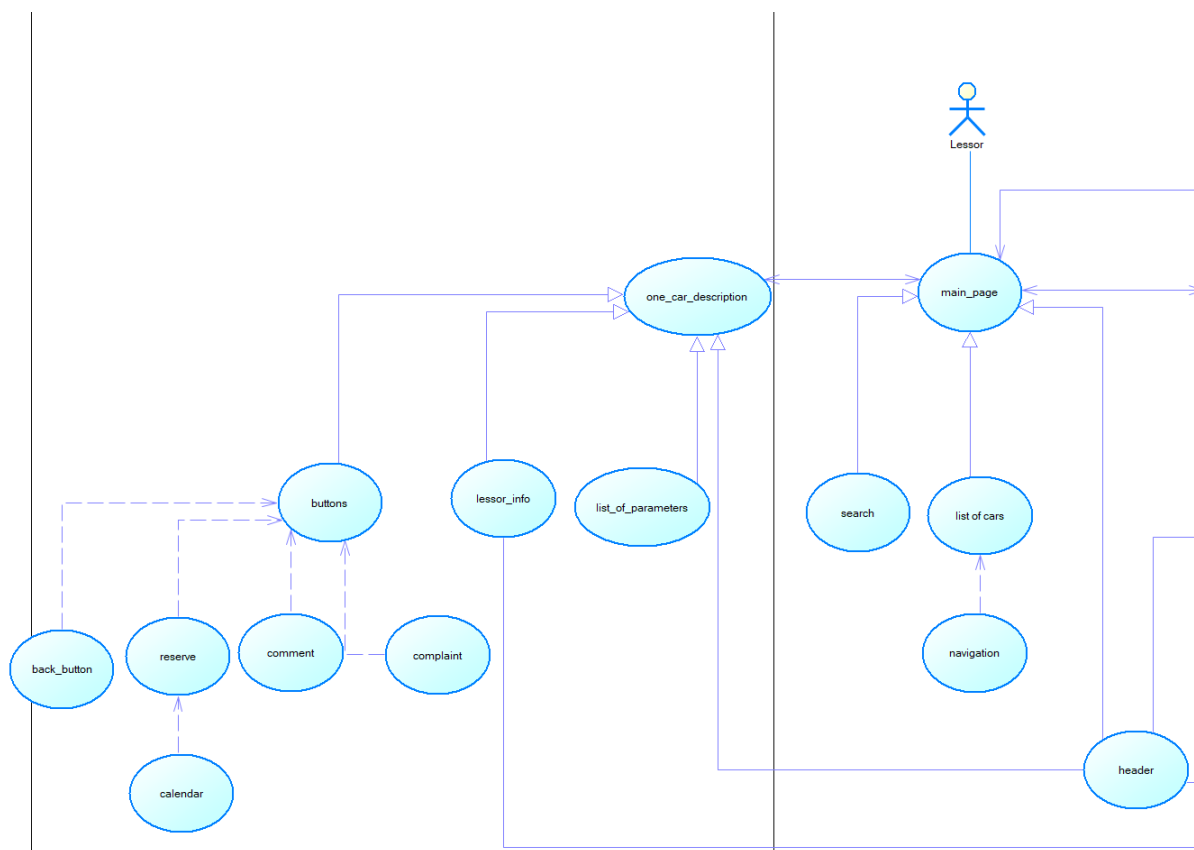


Рисунок 3.7 – Use case для орендодавця (збільшено).

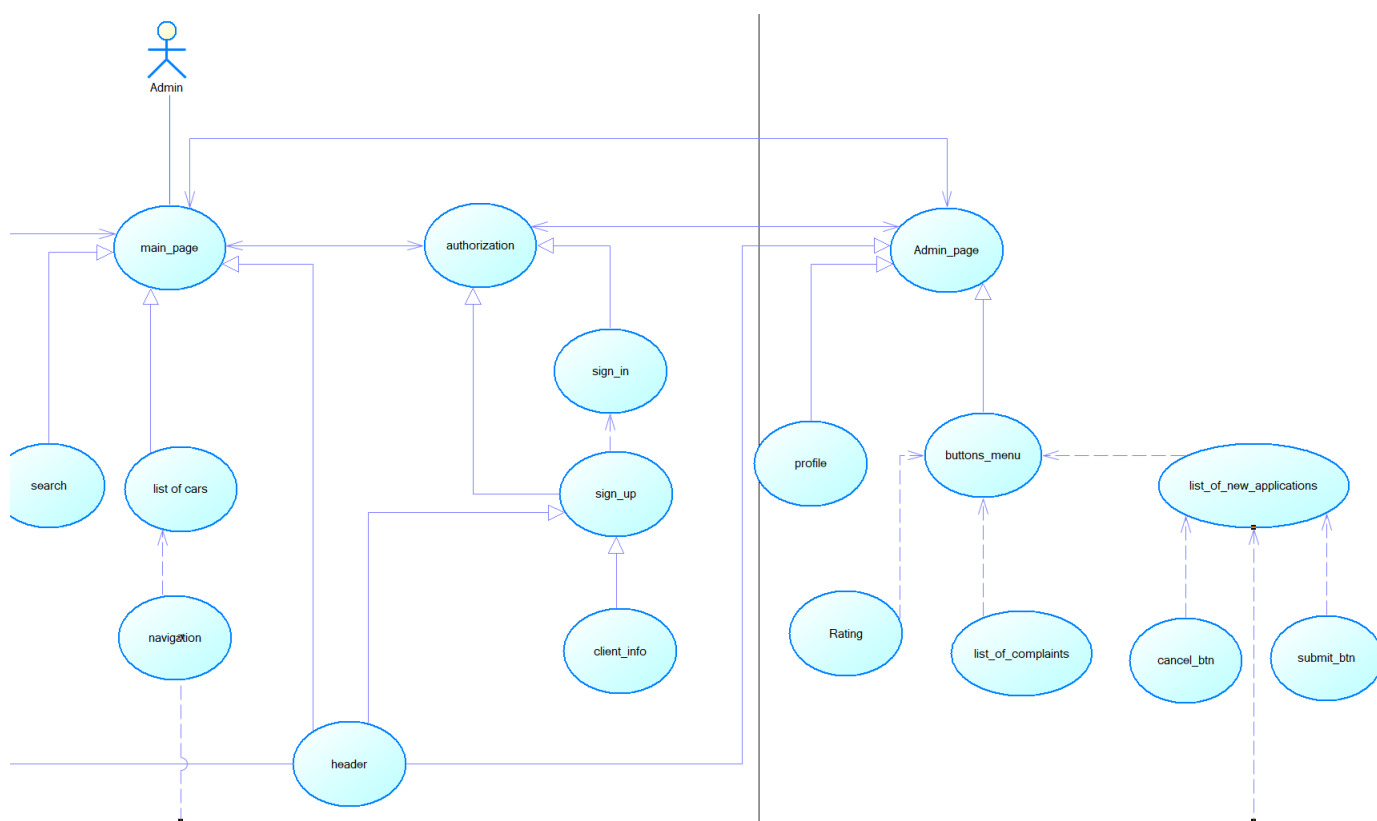


Рисунок 3.8 – Use case для адміністратора (збільшено).

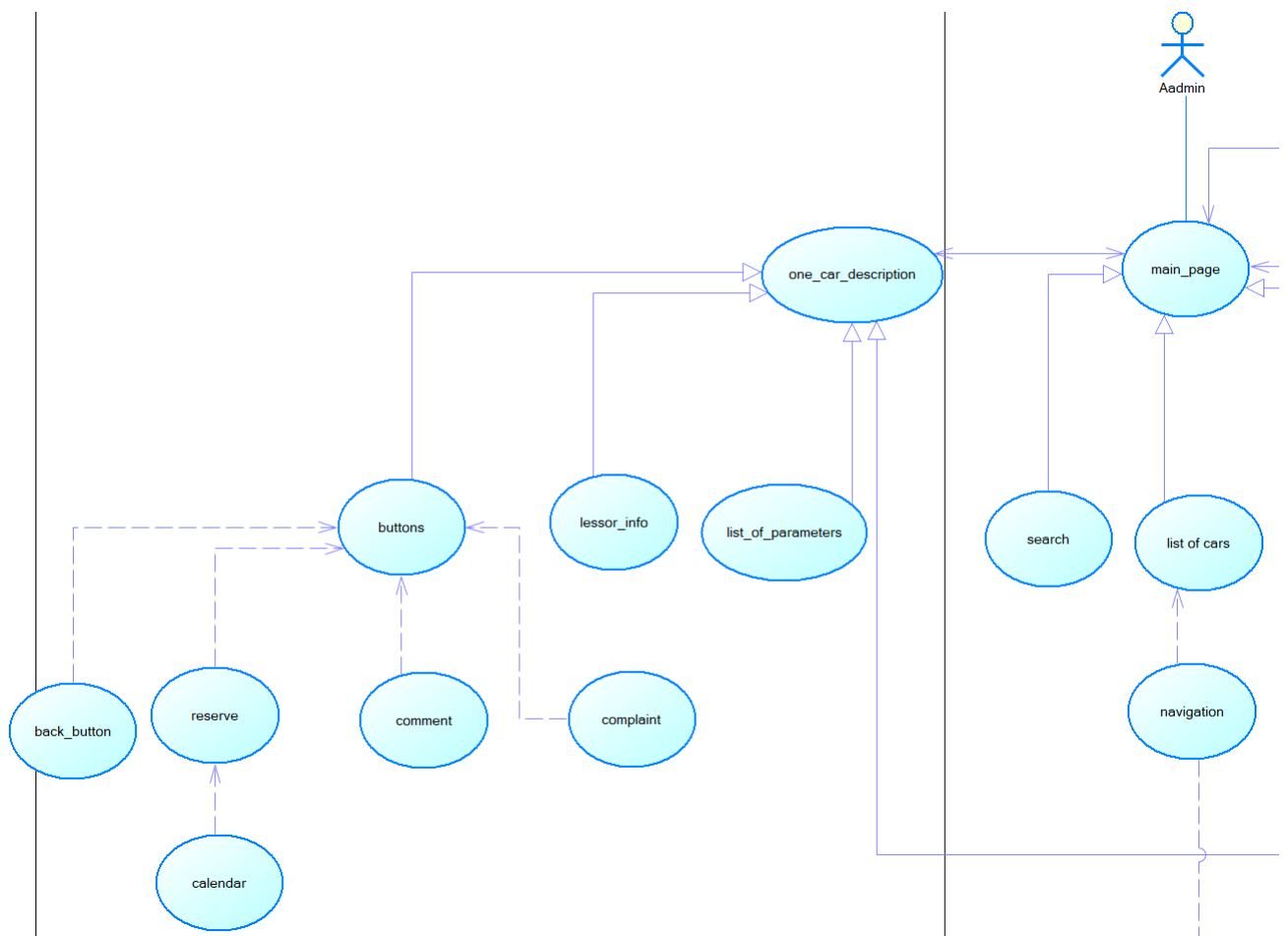


Рисунок 3.9 – Use case для адміністратора (збільшено).

3.2 Component diagram

Розглянемо компоненту діаграму[1], реалізовану для інформаційної системи. Загальна схема діаграми зображена на рисунку 3.10.

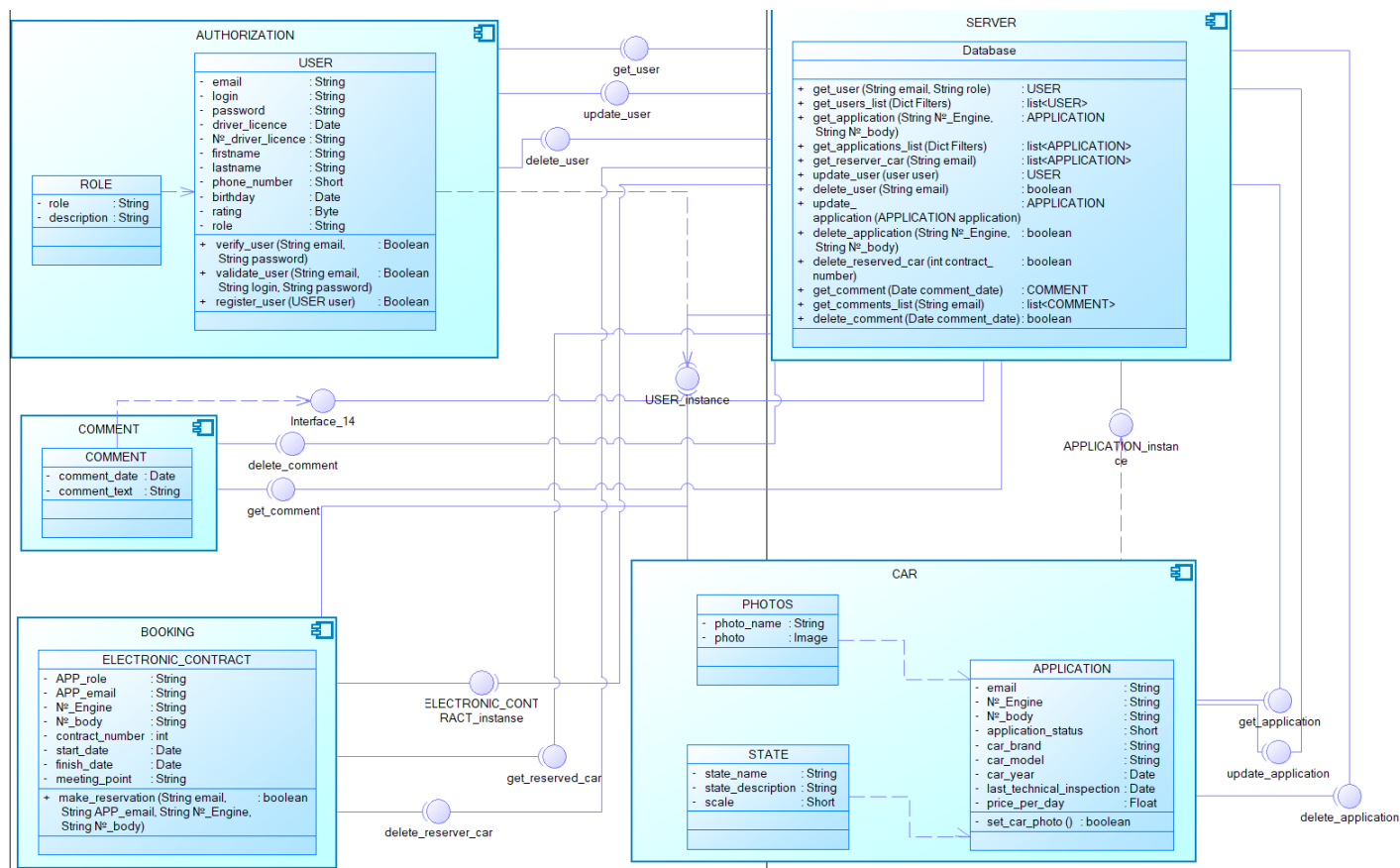


Рисунок 3.10 – Компонентна діаграма

Більш детальна структура даної діаграми на рисунках 3.11, 3.12, 3.13, 3.14

Діаграма компонентів дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними. Компоненти об'єднуються, разом використовуючи структурні зв'язки. Це ілюструє зв'язок типу «клієнт-сервер». Структурна взаємодія — «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

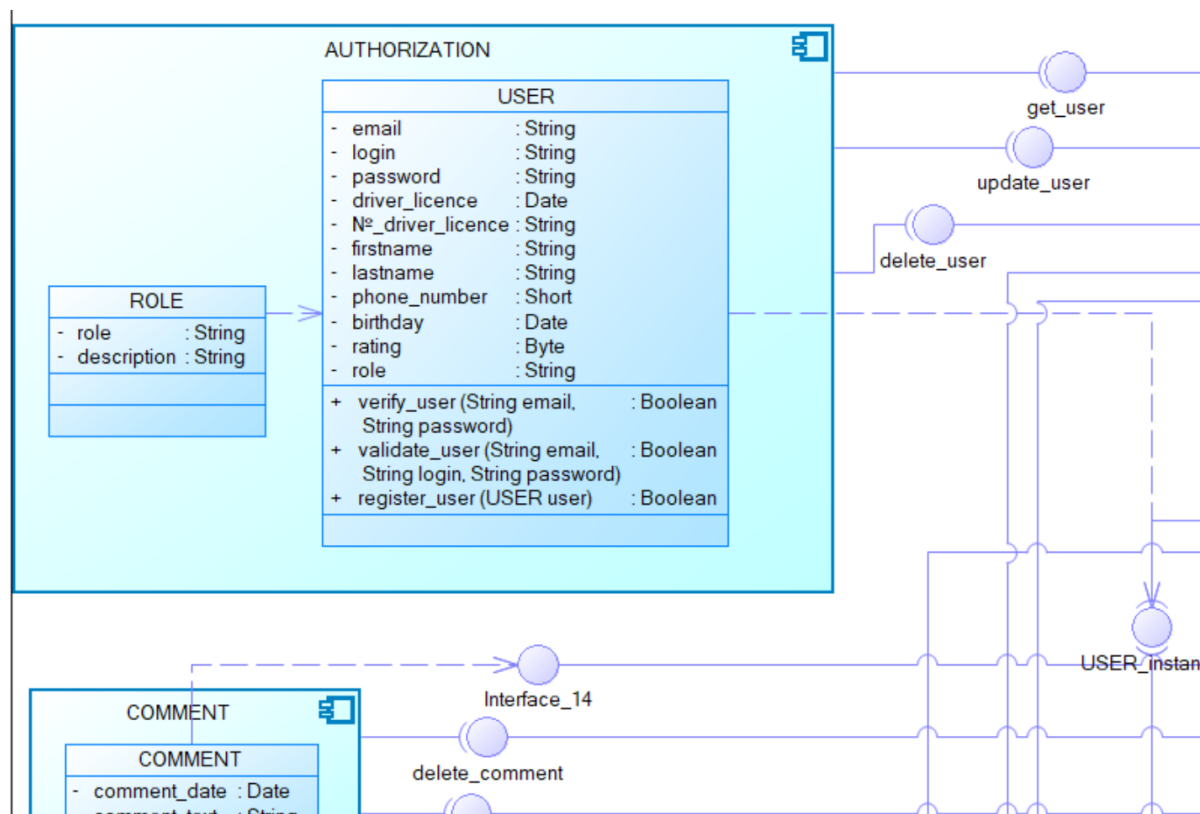


Рисунок 3.11 – Компонентна діаграма(частина 1)

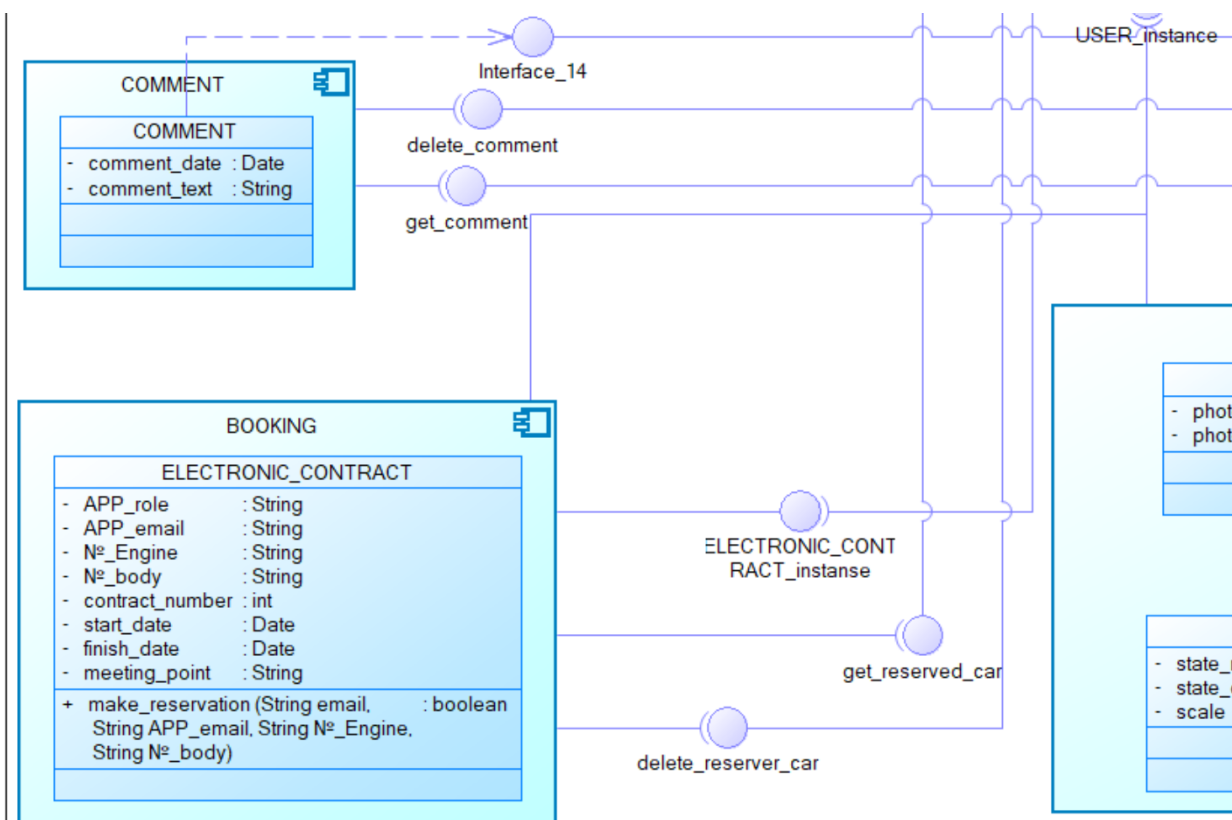


Рисунок 3.12 – Компонентна діаграма(частина 2)

3.3 Scrum. Sprint Planning

Для реалізації системи оренди автомобілю було сплановано 3 спринти: «Application», «Authorization and electronic contract» та «Review». В першому спринті «Application» повинні бути реалізовані основні функції системи по роботі з заявками на автомобілі, що будуть орендуватись. Це і подавання заявки орендодавцем (рис.3.15, рис.3.16), підтвердження (затвердження) автомобілів адміністратором та їх подальший перегляд користувачами (рис.3.17, рис.3.18, рис.3.19). В першому спринті також передбачені функції резервування автомобілів користувачами та видалення заявок адміністратором (рис.3.20, рис.3.21, рис.3.22).

В наступному спринті «Authorization and electronic contract» реалізовано функції авторизації в системі. Тут також запланована функція створення електронного договору між клієнтом та орендодавцем (рис.3.23, рис.3.24, рис.3.25, рис.3.26).

В останньому спринті «Review» буде реалізовано функціонал коментування та подання скарг як користувачів так і орендодавців (рис.3.27, рис.3.28)

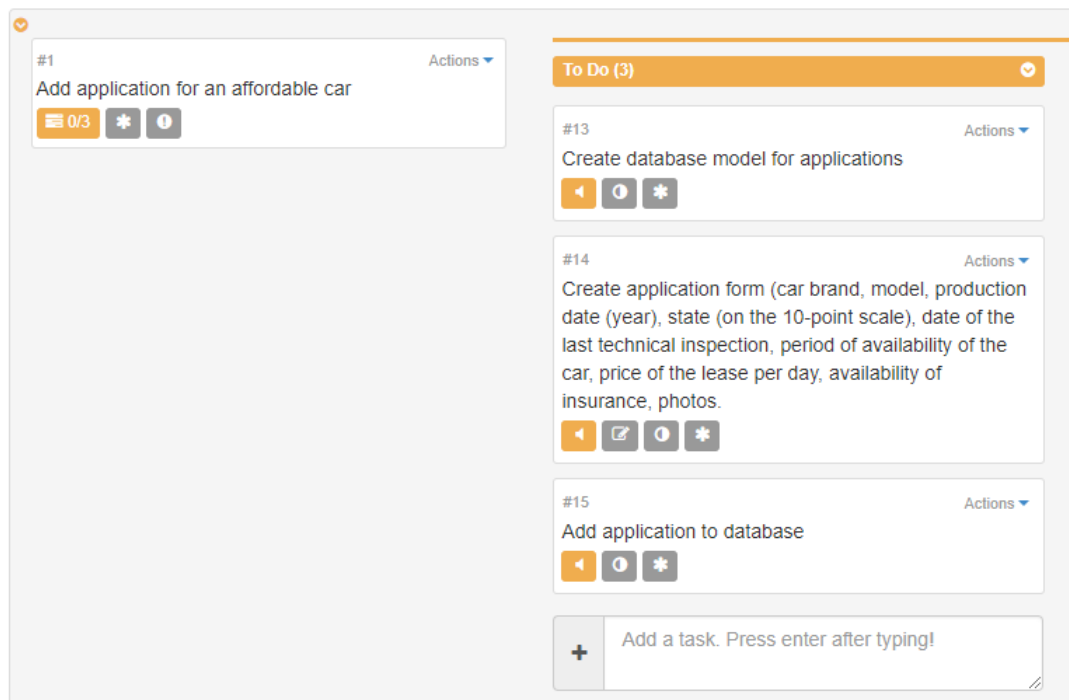


Рисунок 3.15 – Планування функції додавання заявок на автомобіль.

Task Description (#14)

- car brand, model, production date (year) - list, mask car_brand/model/years ,non editable.
- state - linear scale, 10 points (0,1, ... , 10), varied, non editable.
- date of the last technical inspection, - textbox(string), mask dd / mm / yyyy , acceptable characters : 0 - 9, editable.
- price of the lease per day - textbox(string), mask ddd.dd , editable.
- availability of insurance, - switch (boolean),varied , non editable
- photos - file (show only *.png, *.jpeg, *.jpg files which size < 2Mb), max - 4 photos, max size of one file 2Mb., editable.
- send_btn - button, click, text = "ПОДАТИ", non editable. feedback - add application to database with state "new"
- NN incorrect extension, non existent file, no connection with database.

Рисунок 3.16 – Опис полів для форми додавання заявок на автомобіль.

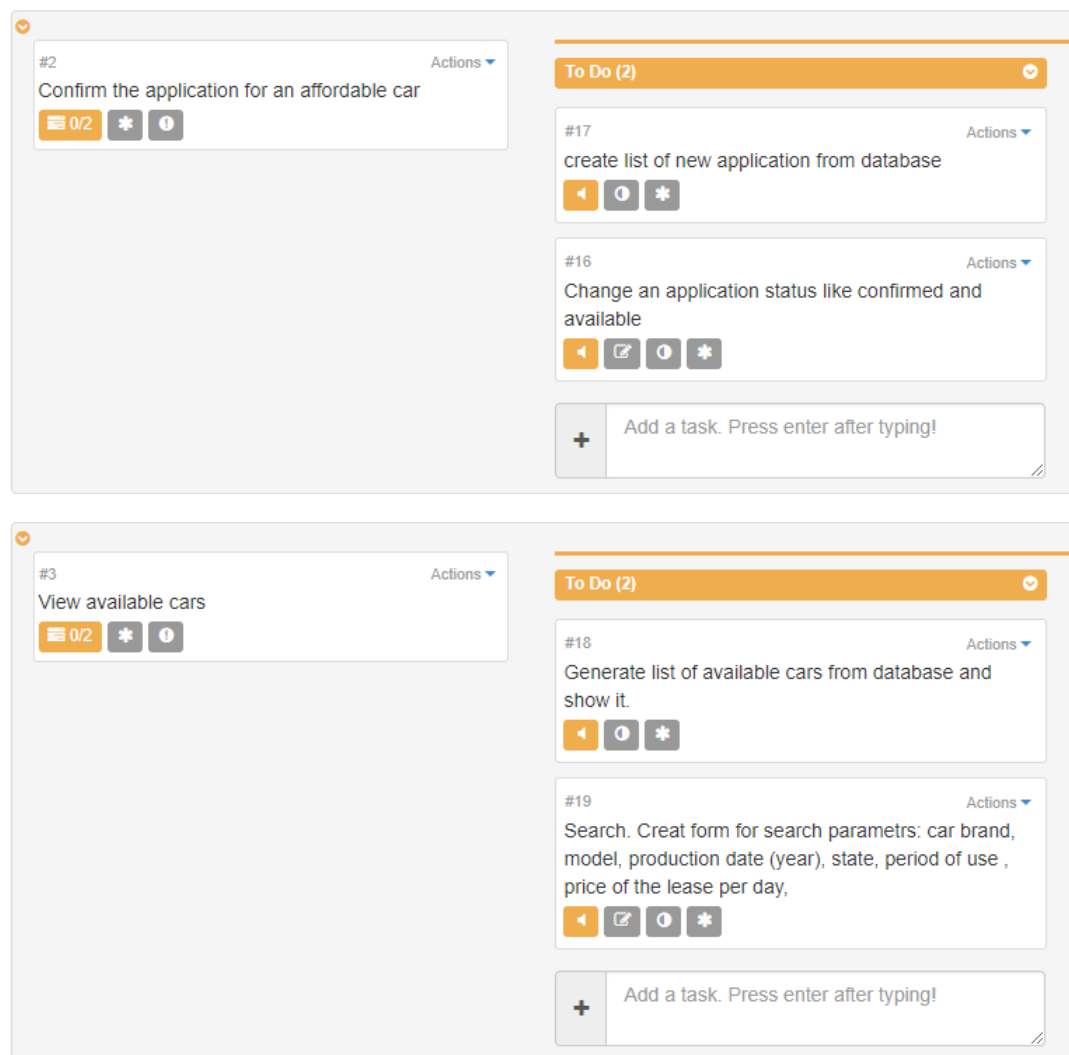


Рисунок 3.17 – Планування функцій підтвердження та перегляду автомобілів.

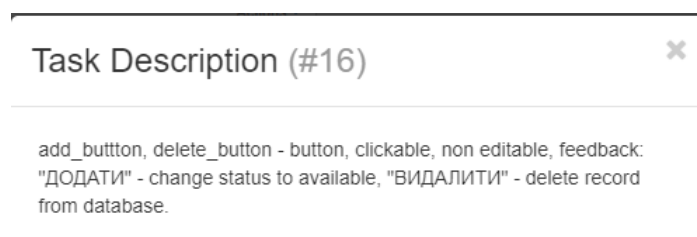


Рисунок 3.18 – Опис полів для форми підтвердження заявки.

Task Description (#19)

- car brand, model, production date (year) - list, mask car_brand/model/years ,non editable.
- state - linear scale, 10 points (0,1, ... , 10), varied, non editable.
- date of use - lists for day, mounth, year, mask dd / mm / yyyy, non editable.
- price of the lease per day - textbox(string), mask ddd.dd , editable.
- search_btn - button, click, text = "ПОШУК", non editable. Feedback - make select from database.
- NN have no access to the database.

Рисунок 3.19 – Опис фільтрів для пошуку доступних автомобілів.

#4

Reservation of available cars

0/1

*

!

To Do (1)

#20

Mark as reserved. Save reservation in database.

<

!

!

*

+

Add a task. Press enter after typing!

#6

Delete the application for an affordable car

0/1

*

!

To Do (1)

#21

Delete record from database.

<

!

!

*

+

Add a task. Press enter after typing!

Рисунок 3.20 – Планування функцій резервування та видалення.

Task Description (#20)

reservation_btn - button, text = 'ЗАБРОНЮВАТИ', clickable, non editable.
Feedback - change status as 'reserved', save changes in database.

Рисунок 3.21 – Опис полів для резервування автомобіля.

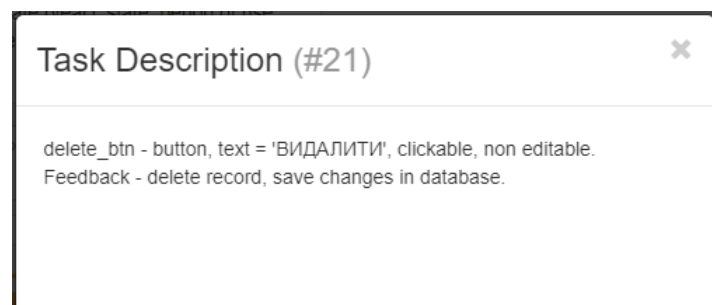


Рисунок 3.22 – Опис полів для видалення заявки.

Authorization&electronic contract

22 Days To Start

3 Total Cards

(3 Cards To Do)

1 Weeks Duration

#7

Sign up

0/2

✱

ⓘ

Actions

To Do (2)

#22

Create database model for users. Divide into three groups: client, administrator, lessor.

⏪

ⓘ

✱

Actions

#23

Create registration form (Last name , First name,e-mail, password, confirm password, phone number, date of birth , role)

⏪

✍

ⓘ

✱

Actions

+ Add a task. Press enter after typing!

#8

Sign in

0/1

✱

ⓘ

Actions

To Do (1)

#24

Create authorization form (e-mail, password)

⏪

✍

ⓘ

✱

Actions

+ Add a task. Press enter after typing!

#9

Conclusion of an electronic contract

0/1

✱

ⓘ

Actions

To Do (1)

#25

Create form of electronic contract with next data (client data, lessor data, period, meeting point, agreement)

⏪

✍

ⓘ

✱

Actions

Рисунок 3.23 – Планування функцій авторизації користувачів та укладання електронного контракту.

Task Description (#23)



- Last name, First name - textbox, max 15 symbols, acceptable characters: A-Z, a-z, '-', editable.
- e-mail - textbox,email-mask, max 20 symbols,acceptable characters : A-Z, a-z, 0-9, " . " _ " - " "\$#@!^&*()?"'"/" editable.
- password, confirm password - textbox,password mask, min - 6 symbols (letter and digit), max - 15 symbols (letter and digit) ,acceptable characters : A-Z, a-z, 0-9, " . " _ " - " "\$#@!^&*()?"'"/" , editable.
- date of birth - lists, mask dd / mm / yyyy , non editable.
- number - textbox, mask: +(380) dd ddd dddd, acceptable characters : 0 - 9, editable.
- role - checkbox, possible options: client, lessor, checkable, non editable.
- confirm - button, click, text = "ЗАПЕЄСТРУВАТИСЬ", non editable. Feedback : if correct data make add record to the database.
- NN - no connection with database, no internet.

Рисунок 3.24 – Опис форми реєстрації.

Task Description (#24)



- e-mail - textbox, max 20 symbols,acceptable characters : A-Z, a-z, 0-9, " . " _ " - " "\$#@!^&*()?"'"/" editable.
- password - textbox, min - 6 symbols (letter and digit), max - 15 symbols (letter and digit) ,acceptable characters : A-Z, a-z, 0-9, " . " _ " - " "\$#@!^&*()?"'"/" , editable.
- confirm - button, click, text = "log in", non editable. Feedback - user verification in database.

Рисунок 3.25 – Опис форми входу в систему.

Task Description (#25)



- text - text, auto-template, non editable.
- start date, end date - date, calendar, non editable.
- meeting point - textbpx (string), acceptable characters: A-Z, a-z, '-', 0 - 9, '№' , editable.
- agreement - checkbox, text = "ПОГОДЖУЮСЬ З УМОВАМИ ЕЛЕКТРОННОГО ДОГОВОРУ", selectable, non editable.
- contract_submit_btn - button , clickable, text = 'ОРЕНДУВАТИ', non editable. Feedback: make new record about rent, save changes in database.

Рисунок 3.26 – Опис функції укладання електронного контракту.

The image shows three screenshots of a task management interface. Each screenshot displays a task card with a title, a description, and a 'To Do' list. The task cards are numbered #10, #11, and #12. Each card has a 'To Do' list with a 'To Do' button and a 'Add a task' button. The task cards are arranged in a grid.

Task #10: Add a review about the car. To Do (2): #26 Create database model for review and comments, #27 Create form for review.

Task #11: Add a complaint to the lessor/client. To Do (1): #28 Create form for complaint.

Task #12: Processing complaints. To Do (1): #29 Delete application of car or lessor, if it is a reason. Or ignore complaint.

Рисунок 3.27 – Планування функцій коментування та подання скарг.

Task Description (#27)

- comment - textbox, max symbols - 300, acceptable characters : A-Z, a-z, 0-9, " . , " _ , " - , " \$ # @ ! ^ & * () ? / " , editable.
- send_btn - button, text = "ВІДПРАВИТИ", clickable, non editable,
- feedback - add new record to the database.

Рисунок 3.28 – Опис полів для введення скарг та коментарів.

4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

В даній інформаційній системі можна виділити наступні сутності:

- користувач (USER);
- роль користувача (ROLE);
- заявка на автомобіль (APPLICATION);
- стан машини (STATE);
- фото (PHOTOS);
- коментарі (COMMENT).

Концептуальна діаграма [3] інформаційної системи зображена на рисунку 4.1.

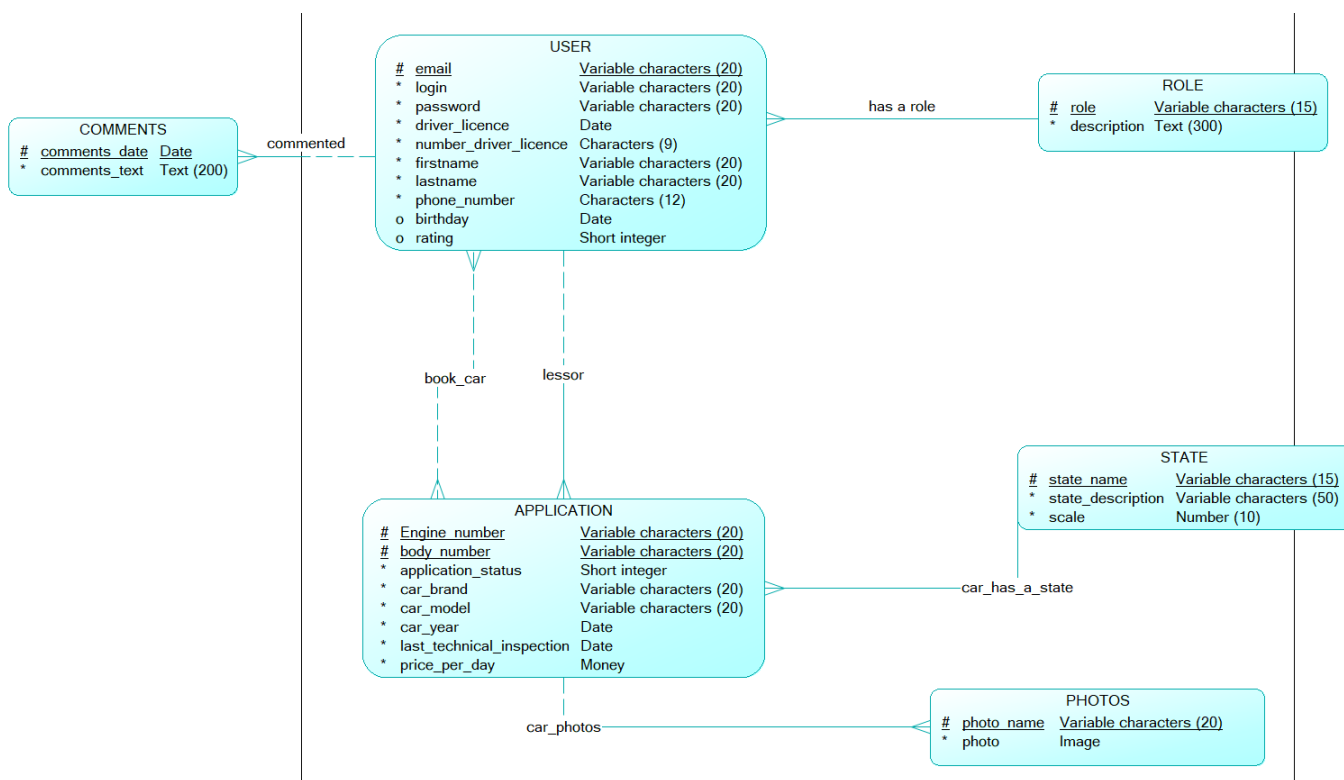


Рисунок 4.1 – Conceptual model

Всі ці сутності поєднані різними зв'язками:

- а) Користувач "подає" Заявку (один до багато);
- б) Заявка "подається" Користувачем (багато до одного);
- в) Користувач "має" Роль (багато до одного);

- г) Роль "надається"Користувачу (один до багато);
- д) Користувач "пише"Коментар (один до багато);
- е) Коментар "залишається"Користувачем (багато до одного);
- є) Користувач "орендує"Заявку (багато до багато);
- ж) Заявка "використовується"Користувачем (багато до багато);
- з) Заявка "має"Фото (один до багато);
- и) Фото "прикріплені до"Заявки (багато до одного);
- і) Стан "існує в"Заявки (один до багато);
- ї) Заявка "має"Стан (багато до одного).

Розглянемо детільніше кожен з сутностей.

4.1 Користувач

Користувач - зареєстрована людина, яка отримує свої права в залежності від ролі, яку вона обрала при реєстрації . Дана сутність є однією з головних, бо тоді ніхто не зможе ні орендувати машину , ні подати заявку на машину. Атрибути даної сутності:

- а) email - є ключем та однозначно визначає користувача;
- б) login - обов'язковий;
- в) password - обов'язковий;
- г) driver_licence - обов'язковий;
- д) number_driver_licence - обов'язковий;
- е) firstname - обов'язковий;
- є) lastname - обов'язковий;
- ж) phone_number - обов'язковий;
- з) birthday - обов'язковий;
- и) rating - не обов'язковий.

4.2 Роль

Роль - кожен користувач системи має певну роль, яка надає йому відповідні права для роботи з інформаційною системою. Атрибути даної сутності:

- а) role - є ключем та однозначно визначає назву ролі;
- б) description - обов'язковий.

4.3 Заявка

Заявка - сутність, що відповідає за автомобіль та містить всі його основні характеристики. Є дуже важливою сутністю, адже саме на ній і будується бізнес-логіка системи. Атрибути даної сутності:

- а) Engine_number- є ключем та є одним з ідентифікаторів автомобіля;
- б) body_number - є ключем та є одним з ідентифікаторів автомобіля;
- в) application_status - обов'язковий, визначає стан заявки;
- г) car_brand - обов'язковий;
- д) car_model - обов'язковий;
- е) car_year - обов'язковий;
- є) last_technical_inspection - обов'язковий;
- ж) price_per_day - обов'язковий.

4.4 Стан

Стан - кожен автомобіль має певний технічний стан, користувачу(орендодавцю) пропонується визначити його певним словом - станом (наприклад - добрий, відмінний

і т.д.). Атрибути даної сутності:

- а) state_name - є ключем та однозначно визначає стан;
- б) state_description - обов'язковий, опис стану;
- в) scale - обов'язковий, оцінка стану;

4.5 Фото

Фото - до кожної заявки може бути додано декілька фотографій. Атрибути даної сутності:

- а) photo_name - є ключем та однозначно визначає назву фотографії;
- б) photo - обов'язковий, містить шлях до фотографії;

4.6 Коментарі

Коментарі - користувач може створювати відгуки про оренду автомобіля та ділитись ними на сайті. Атрибути даної сутності:

- а) comments_date - є ключем та однозначно визначає час відгуку;
- б) comments_text - обов'язковий, тіло відгуку;

5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

Було зпроектовано логічну, логічну-оптимізовану та фізичну моделі [1] інформаційної системи оренди автомобілів.

Логічна модель інформаційної системи зображена на рисунку 5.1.

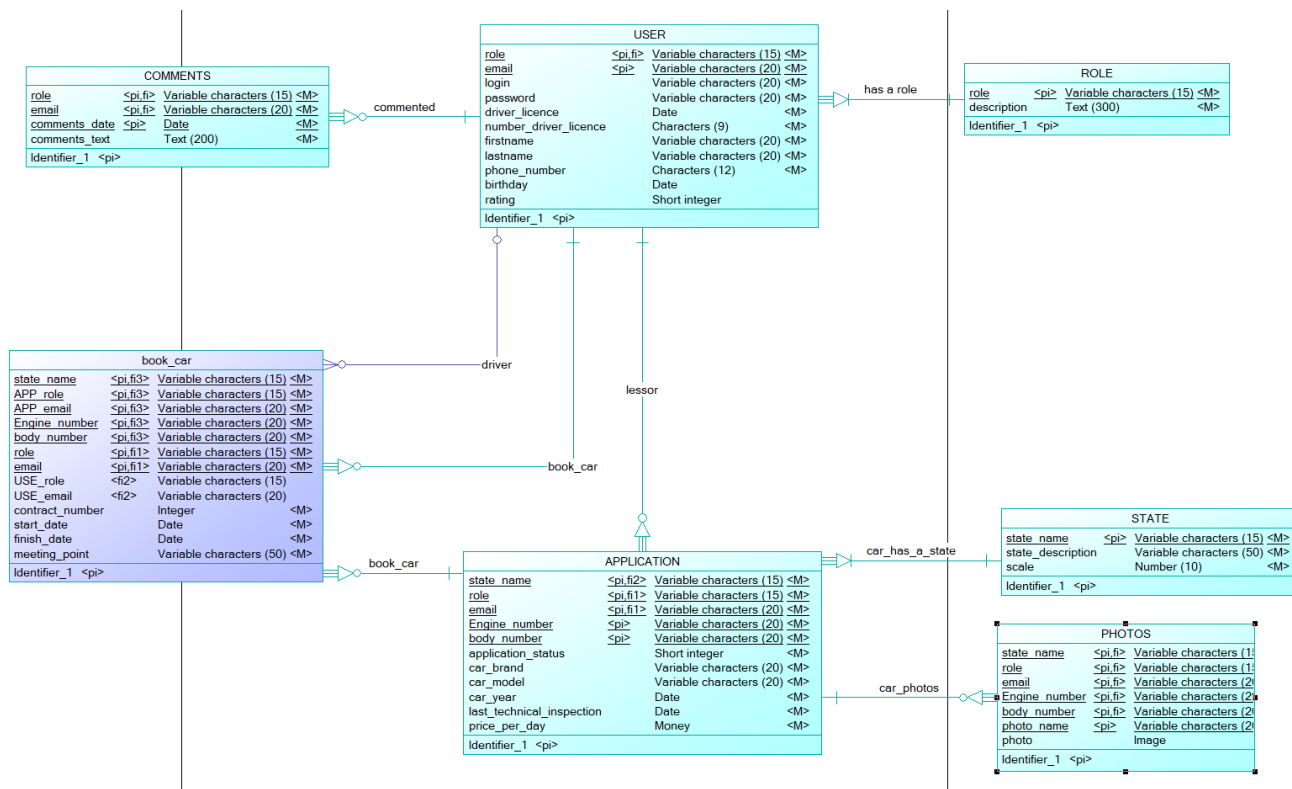


Рисунок 5.1 – Логічна модель

У порівнянні з концептуальною, відбулися наступні зміни:

- з'явилися зовнішні ключі;
 - з'явилась проміжна таблиця Контракт, що відповідає за зв'язок багато до багато;
 - з'явився зв'язок Водій, коли користувач може замовити водія з машиною;
- Логічно-оптимізована модель інформаційної системи зображена на рисунку 5.2.

У порівнянні з логічною, відбулися наступні зміни:

- Заявка має новий атрибут - application_id. Ввели фіктивні ключі, щоб

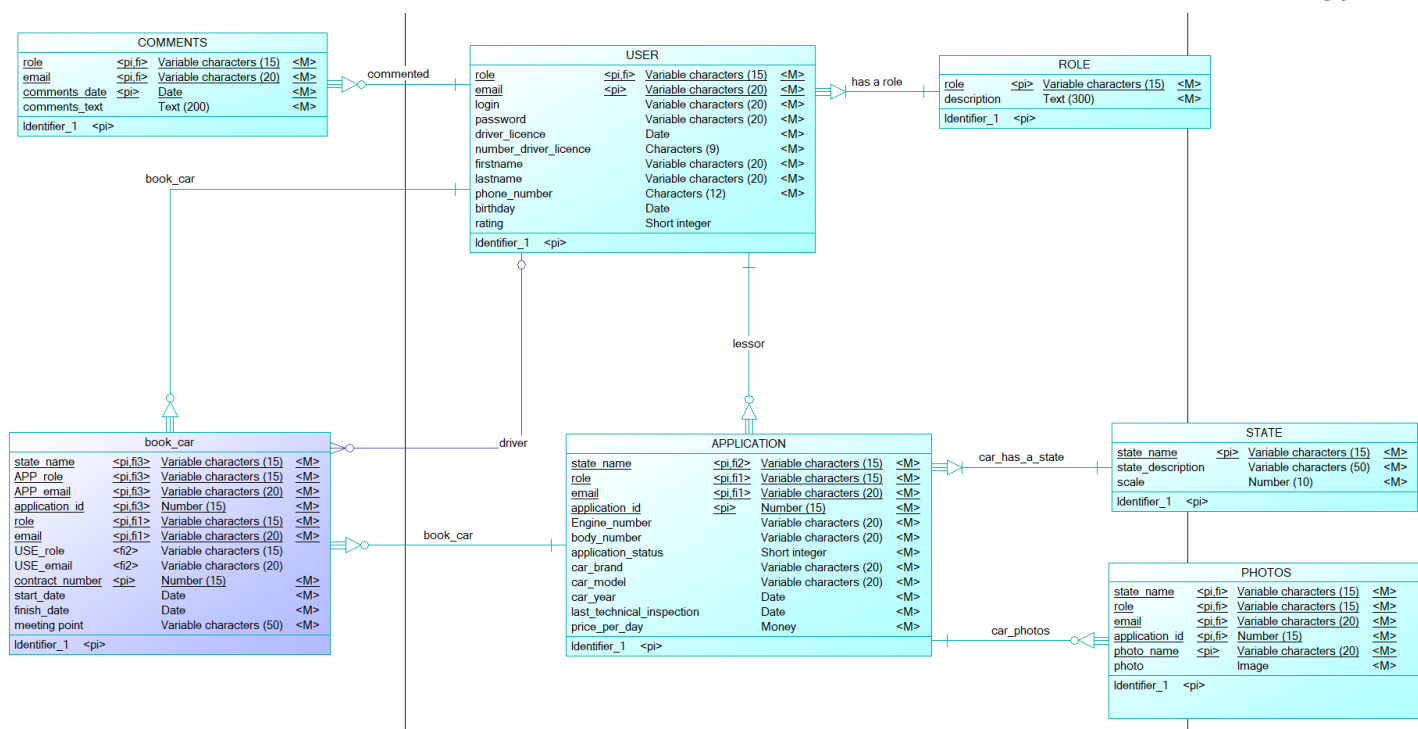


Рисунок 5.2 – Логічна оптимізована модель

ідентифікувати заявку по одному полю.

Фізична модель інформаційної системи зображена на рисунку 5.3.

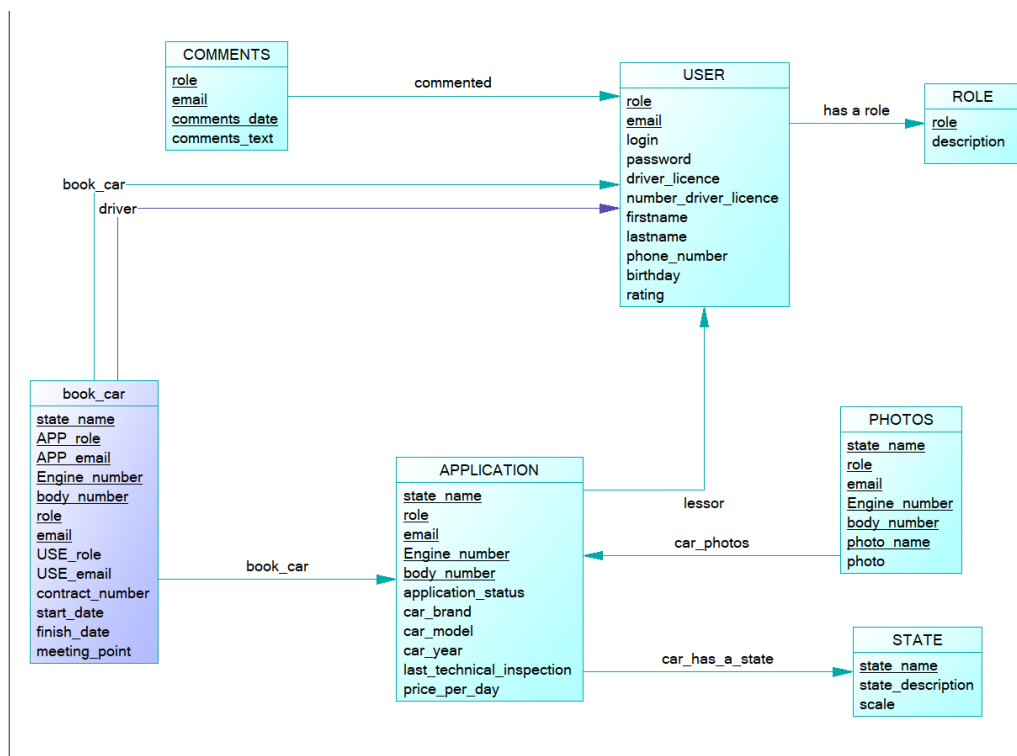


Рисунок 5.3 – Фізична модель

ВИСНОВКИ

Отже, було досліджено, зпроектовано та створенно інформаційну систему для оренди автомобілів. Під час виконання даних етапів було створено USE CASES для різних користувачів системи, побудовано компонентну діаграму та діаграму послідовностей. Модель бази даних було представлено за допомогою концептуальної, логічної та фізичної ERD.

Реалізована інформаційна система має наступний функціонал для користувачів:

- а) реєстрація та авторизація користувачів;
- б) додавання заявок та фото для них;
- в) бронювання автомобіля;
- г) пошук автомобіля за певними критеріями.

Отримано навички в розробці та проектуванні повних інформаційних систем, та в реалізації "клієнт-серверної" архітектури.

ПЕРЕЛІК ПОСИЛАНЬ

1. UML Diagrams [Електронний ресурс]. <https://www.smartdraw.com/uml-diagram/>
2. SCRUM [Електронний ресурс]. <https://www.flying-donut.com/>
3. Oracle. Help center [Електронний ресурс]. <https://docs.oracle.com/en/>
4. Форт Б. SQL. — [3th ed.] — Москва, 2006. — 169 с.