

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

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КУРСОВОЇ РОБОТИ
з дисципліни «Бази даних та інформаційні системи»
на тему:
Система бронювання квитків на відпочинок

Виконав:
студент групи КМ-41
Козик В.В.

Перевірів:
старший викладач кафедри ПМА
Терещенко І. О.

Київ
2017

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Завдання курсової роботи полягає у створенні системи, яка б дозволяла спростити та пришвидшити процес бронювання квитків на відпочинок, а саме: житла та розваг. Наразі не існує систем в Україні, які б вміщували в собі ці дві якості. Аудиторія на яку розрахований проект: люди, що досягнули повноліття, українського громадянства.

Система має дозволяти користувачам підбирати оптимальне місце для відпочинку, враховуючи різні фактори, так, як час, локація, ціна.

Система має працювати з такими даними:

- а) Інформація про користувача (електронна пошта, ім'я, вік, номер мобільного телефону).
- б) Дані про житло (місто, вулиця, кількість спальних місць, ціна, орендодавець).
- в) Дані про заходи (місто, вулиця, час проведення, ціна, організатор).
- г) Дані про бронювання (моніторинг дат, у випадку з житлом, та моніторинг кількості квитків у випадку з заходами).

Було проаналізовано способи реалізації схожих системи і прийнято рішення розробляти застосунок у вигляді веб-сайту.

АНОТАЦІЯ

Для побудови системи купівлі квитків на відпочинок було проаналізовано, що найбільш актуальним, з сторони потреби користувачів, є такі складові відпочинку, як житло та заходи, які можна відвідати в даній локації. Отож, система має акумулювати і обробляти інформацію від різних орендодавців та організаторів України. Крім того, інформація про бронювання також залишається в системі, що підвищує її складність, а тому й вимоги до зберігання даних.

Передпроектне дослідження – було досліджено, що наразі в Україні немає аналогів таким централізованим системам. Більше того, навіть немає колаборацій між популярними системами бронювання житла та заходів (таких як надавання відкритого API один одному). Крім того, останній рік для України ознаменувався взлеском активності молоді у відвідуванні заходів, відроджується культура музичних фестивалів.

Також, проект було розбито на спринти, в яких вказувались потрібні задачі для реалізації такої громіздкої системи. До кожного спринта існує свій список підзадач та вимог до розроблюваних частин продукту.

Розробка проекту – провівши передпроектне дослідження було побудовано основні діаграми для проекту, так, як: ERD, компонентна діаграма, діаграма послідовностей та Use-case діаграма. Так, ERD та компонентна діаграма є основою для створення сутностей та зв'язків між ними в базі даних. Крім того, компонентна діаграма, по суті, показує нам, які інтерфейси нам потрібні для взаємодії між частинами проекту. Діаграма послідовностей та Use-case показує логіку взаємодії користувачів з системою. Враховано психологічні фактори, що змушують людину рухатись в певному напрямку (мисленнєво) та побудовано UI застосунку.

Розробку програмного забезпечення було здійснено за допомогою наступних технологій: Python(microframework Flask), web-server Nginx, PowerDesigner, HTML/CSS/JS, Oracle DB 11g.

ЗМІСТ

АНОТАЦІЯ	3
ВСТУП.....	5
ОСНОВНА ЧАСТИНА	6
1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ.....	6
2 ПОСТАНОВКА ЗАДАЧІ	7
2.1 Категорії користувачів	7
2.2 Класи даних.....	8
2.3 Бізнес правила.....	8
2.4 Функції системи.....	9
3 МОДЕЛЮВАННЯ БІЗНЕС ПРОЦЕСІВ	12
3.1 USE CASE діаграма	12
3.2 Компонентна діаграма	12
3.3 Scrum	13
4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ	16
ВИСНОВКИ.....	19
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	20
ДОДАТОК А.....	21

ВСТУП

Процеси в світі стрімко автоматизуються. Якщо колись черга по запису в декілька днів нікого не дивувала, то тепер очікування більше години викликає обурення.

В світовій мережі існує багато ресурсів, що дозволяють підібрати житло, транспорт, розваги, купити товари, проте виникає парадокс: кожен хоче бути найкращим в своєму напрямку, тому переважають вузькоспеціалізовані системи.

Проте все ще витрачається час на пошук різних систем для задоволення власних потреб. Так, якщо ми вирішили подорожувати, то нам прийдеться зайти на декілька сайтів, що замовити житло, транспорт та знайти заходи на яких можна провести свій відпочинок.

Тому, система, яка акумулює в собі бронювання житла та заходів є дуже актуальною тому, що економить велику частину часу, нашого найціннішого ресурсу.

Хоча система є актуальною та потрібною, стає зрозуміло чому таких систем не існує серед найбільш популярних. Чим більше в собі вміщує система, тим зростає її складність, а тому й витрачені ресурси: час та гроші.

ОСНОВНА ЧАСТИНА

1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ

Для виявлення кращого варіанту проекту потрібне глибоке дослідження цільової аудиторії і наявних конкурентів. Був проведений аналіз ринку розваг і найбільш оптимальним варіантом є системи бронювання, які мають обмеження.

Під часу аналізу брались до уваги такі характеристики, як:

- 1) цільова аудиторія;
- 2) платіжна спроможність аудиторії;
- 3) популярність напрямку.

В даному напрямку було помічено, що системи даного типу користуються популярністю у ріелторів, а тому відходять від своєї цілі.

Тому потрібна система, яка б накладала обмеження на реєстрацію та активацію акаунтів, яка б поєднувала кращі риси з існуючих аналогів.

2 ПОСТАНОВКА ЗАДАЧІ

Основною метою проекту є полегшення і пришвидшення процесів купівлі-бронювання квитків на відпочинок.

Ця задача вміщує:

- а) оригінальний і зручний дизайн;
- б) створення зручних інструментів для керування сайтом;
- в) система бронювання квитків та житла;
- г) система додавання та керування житлом і заходами;
- д) розгалуження ролей;
- е) забезпечення захисту даних;

Граничні умови: Надання можливості бронювання житла та заходів доступних на території України, в межах 2 місяців та для користувачів, що досягнули 18 років.

Інформація про житло від орендодавця має містити: адресу, фото, ціну, кількість місць житла, доступність житла на найближчі пів року (дати), контакти для зв'язку, додаткову інформацію (наявність праски, фену, телевізора і т.п.).

Інформація про захід від організатора має містити: адресу проведення, дату проведення, ціну, кількість квитків, обмеження на вхід (вік, ріст, вага), опис заходу (загальна інформація про захід по бажанню організатора). Інформація про користувача має містити: ПІБ, електронна пошта, пароль, номер контактного телефону, ролі (відпочиваючий, організатор заходу, орендодавець), фото чи скан документу, що підтверджує особу.

2.1 Категорії користувачів

З'являються ролі користувачів: адміністратор, орендодавець, організатор заходу, відпочиваючий. Відпочиваючий може обрати бажане житло в межах певної локації та часових рамок та заходи, які він хоче відвідати, залишати відгуки до житла чи дійства. Орендодавець може надіслати заявку на додавання

його житла в систему, змінювати інформацію про житло після підтвердження (ціна та доступність його на певну дату) та видаляти житло з системи, підтверджувати чи відхиляти бронювання житла, спілкуватись користувачами. Організатор заходу має права аналогічні з орендодавцем, тільки відносно заходів. Адміністратор модерує сайт, тобто розглядає заяви орендодавців та організаторів заходів, надає доступ до сайту всім користувачам системи та забороняє у випадку дій, що порушують правила системи.

2.2 Класи даних

Розроблюючи дану інформаційну було створено об'єкти, які містять наступні дані:

- а) Користувач (заповнюється інформація при реєстрації) – роль, електронна пошта, ім'я користувача, номер телефону, скан чи фото документа, що підтверджує вік особи.
- б) Житло – місто, вулиця, ціна, кількість спальних місць.
- в) Захід – місто, вулиця, назва, ціна, кількість квитків.
- г) Заброньовані дати житла – дата початку бронювання, дата кінця бронювання, житло та орендатор.
- д) Заброньовані квитки заходів – дата бронювання, покупець, назва заходу.

2.3 Бізнес правила

Проектуючи дану інформаційну систему, було встановлено наступні бізнес правила:

- а) доступ до функціоналу сайту має тільки зареєстрований користувач, заявка якого прийнята адміністратором;
- б) користувач отримує інформацію про доступність житла чи заходу;
- в) додавати житло чи заходи можуть тільки орендодавці чи організатори відповідно;
- г) фільтрація житла та апартаментів по локації, даті, ціні;
- д) уникнення ситуації накладання дат (*overlapping*);

- е) забезпечення адекватної діяльності системи бронювання, ліквідація конфліктних ситуацій;

2.4 Функції системи

- Реєстрація користувача

- а) менше 1 хвилини;
- б) інше житло з бази даних, що відповідає вимогам якнайбільше;
- в) структура даних: інформація про користувача у вигляді електронної форми (інформація вказана в граничних умовах);
- г) збереження в БД.

- Авторизація користувача

5

- а) менше 1 хвилини;
- б) інформація береться від користувача системи у вигляді електронної пошти та паролю;
- в) структура даних: POST запит;
- г) зчитування з БД.

- Подання заявки на додавання житла/заходу:

- а) менше 2 хвилин;
- б) дані про житло/захід у вигляді електронної заяви;
- в) структура даних: інформація про житло/захід у вигляді електронної заяви (інформація вказана в граничних умовах);
- г) збереження даних в БД.

- Додавання житла та заходів:

- а) менше 4 хвилин;
- б) інформація береться з заявок організаторів, орендодавців та власних угод адміністраторів;
- в) структура даних: електронна заява (у вигляді json) від орендодавця/організатора чи роздукована заява;
- г) збереження даних в БД.

- Видалення житла та заходів:

- а) менше 2 хвилин;
- б) інформація береться від орендодавця чи організатора у вигляді обраного

елементу для видалення;

в) структура даних: запит з інформацією про житло/захід (див. граничні умови);

г) зміна даних в БД.

б

- Перегляд доступного житла та заходів для заданої локації:

а) менше 2 хвилин;

б) інформація береться з заявок організаторів, орендодавців та власних угод адміністраторів;

в) структура даних: електронна заява (у вигляді json) від орендодавця/організатора чи роздукована заява;

г) завантаження даних з БД.

- Бронювання житла та/чи заходів:

а) менше 3 хвилин;

б) обране користувачем житло та заходи;

в) структура даних: запит про бронювання від користувача;

г) збереження даних з БД.

- Зміна доступності чи ціни житла/заходу

а) менше 1 хвилини;

б) житло чи захід, що стало недоступним чи змінилось в ціні, орендодавець/організатор змінює без підтвердження адміністратора;

в) структура даних: інформація від організатора чи орендодавця;

г) зміна даних в БД.

- Оптимізація пошуку житла, коли заброньоване бажане:

а) менше 1 хвилини;

б) інше житло з бази даних, що відповідає вимогам якнайбільше;

в) структура даних: елементи бази даних;

г) зчитування даних з БД.

- Оплата бронювання житла та заходів:

а) менше 4 хвилин;

7 б) інформація користувача для системи оплати (див. граничні умови) та інформація про обране житло чи захід;

- в) структура даних: запит з інформацією;
- г) зміни до БД

3 МОДЕЛЮВАННЯ БІЗНЕС ПРОЦЕСІВ

3.1 USE CASE діаграма

Під час проектування роботи інформаційної системи, було створено USE CASE UML діаграми для кожної з ролей:

- а) незареєстрований користувач;
- б) відпочиваючий;
- в) орендодавець;
- г) організатор;
- д) адміністратор.

І кожна діаграма складається з наступних елементів:

- а) актор;
- б) перехід
- в) елемент;

Діаграми були виконані в програмі PowerDesigner і зображені в Додатку А.

3.2 Компонентна діаграма

Компонентна діаграма показує основні зв'язки між функціональними частинами проекту, а саме: блок користувача, заходів, житла , системи бронювання та безпосередньо серверу з базою даних. На компонентній діаграмі ми можемо бачити, який інтерфейс та реалізацію надає кожна з частин. По компонентній діаграмі будується інтерфейс для взаємодії з сервером та базою даних.

Компонентна діаграма проекту зображена в Додатку Б.

3.3 Scrum

Для планування було використано методологію Scrum[1]. Її гнучкість і, водночас, точність дозволяє нам побудувати доволі наближений план дій щодо проекту.

Основою методології є спринти (*sprint*) – проміжки часу за які потрібно реалізувати даний план (також називають To Do list)

На рисунку 3.3.1 зображено спринти виконання роботи по створенню проекту:

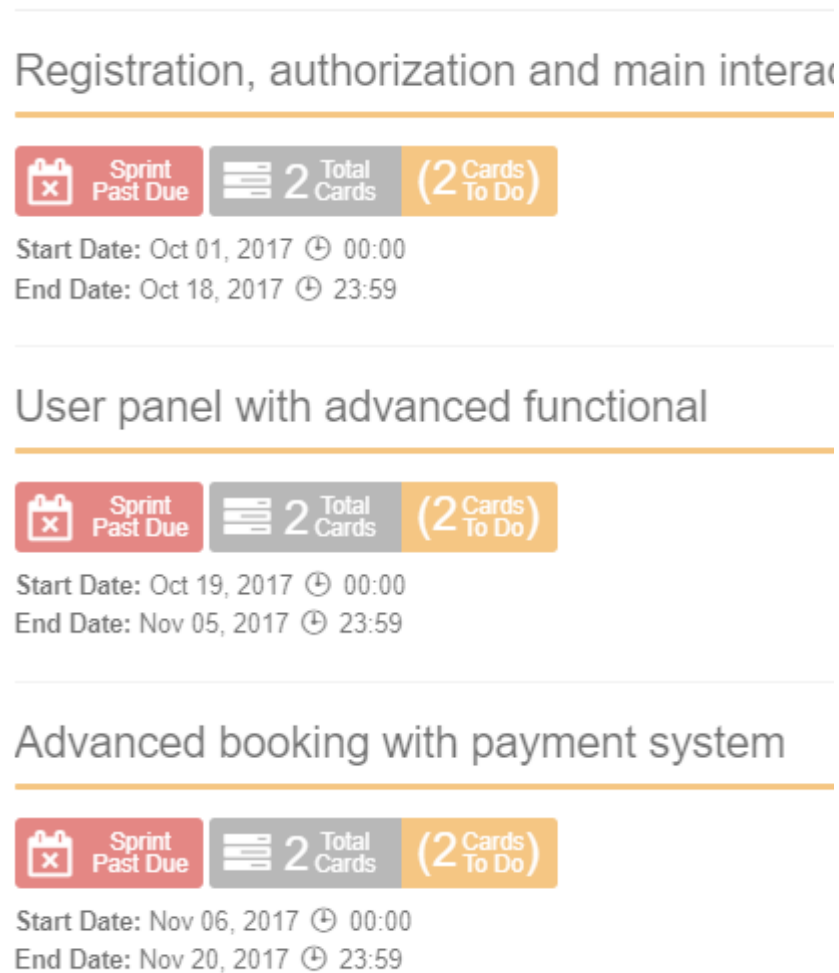


Рисунок 3.3.1 – спринти Scrum

Розглянемо тепер завдання для кожного з спринтів.

Таблиця 3.3.1 – Функції спринта «Registration, authorization and main interaction with apartments and facilities»

Завдання	Пояснення
Реєстрація та вхід	Реалізація механізму реєстрації користувачів.
	Система входу. Вибір та реалізація системи збереження даних про авторизованого користувача (тип сесії, сховища даних)
Взаємодія з житлом та заходами	Створення моделей (класів) для житла та заходів
	Реалізація функцій додавання, редагування та видалення житла.
	Реалізація функцій додавання редагування та видалення заходів.

Таблиця 3.3.2 – Функції спринта «User panel with advanced functional»

Завдання	Пояснення
Просунуте відображення житла та заходу	Фільтр для житла по локації, даті і ціні
	Фільтр для заходу по локації, даті і ціні
	Спільне відображення відфільтрованого житла та заходів
Бронювання житла та заходів	Бронювання житла
	Бронювання заходів
	Відфільтровування недоступних заходів (захід вже закінчився чи білетів немає)
	Відфільтровування недоступного житла (якщо в межах 2 місяців немає доступних дат)

Таблиця 3.3.3 – Функції спринта «Advanced booking system»

Завдання	Пояснення
Історія бронювань для користувачів	Історія поточних бронювань
	Відмова від бронювання
Список бронювань для орендодавців та організаторів	Список бронювань для орендодавців
	Список бронювань для заходів
	Керування бронюваннями заходів
	Керування бронюваннями житла

4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

На рисунку 4.1 зображена концептуальна діаграма.

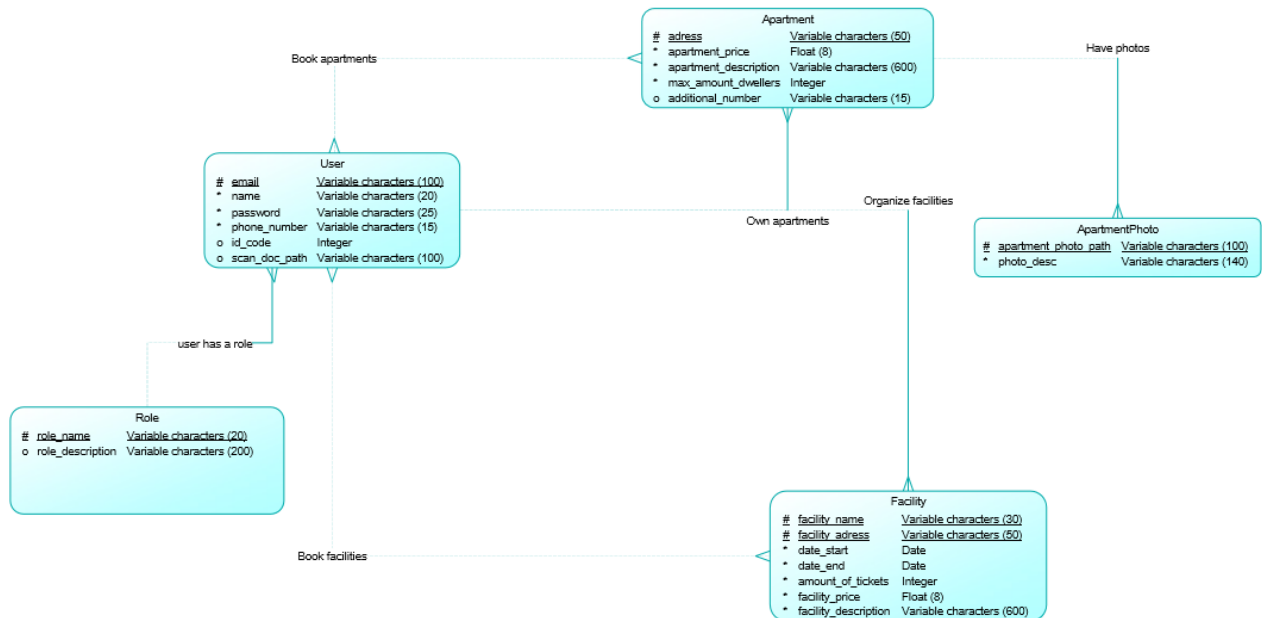


Рисунок 4.1 – Концептуальна діаграма

Концептуальна діаграма для проектування бази даних складається з наступних сутностей:

а) **User** – це сутність, яка є моделлю для користувачів, що зареєструвались в системі.

Вона містить наступні атрибути:

1. email - електронна адреса користувача, обов'язкове поле, первинний ключ;
2. name – ім'я користувача, обов'язкове поле;
3. password – прізвище користувача, обов'язкове поле;
4. phone_number – номер телефону, обов'язкове поле;
5. scan_doc_path – скан документу, що підтверджує особу, обов'язкове поле;

б) **Apartment** – це сутність, що моделює додане в систему житло.

Атрибути:

1. city - місто, де знаходиться житло, обов'язкове поле, первинний ключ;

2. street – вулиця, на якій знаходиться житло, обов’язкове поле, первинний ключ;
3. apartment_price – ціна за добу, обов’язкове поле;
4. apartment_description – опис житла, обов’язкове поле;
5. max_amount_dwellers - максимальна кількість жильців, обов’язкове поле;
6. additional_number – додатковий номер для контакту, необов’язкове поле.

в) Facility – це сутність, що моделює додані в систему заходи.

Атрибути:

1. city - місто, де проводиться захід, обов’язкове поле, первинний ключ;
2. street – вулиця, на якій проводиться захід, обов’язкове поле, первинний ключ;
3. date_start – дата початку заходу, обов’язкове поле;
4. date_end – дата кінця заходу, обов’язкове поле;
5. facility_price - ціна входу, обов’язкове поле;
6. amount_of_tickets – загальна кількість білетів, обов’язкове поле;
7. facility_description – опис заходу, обов’язкове поле.

г) ApartmentPhoto – це сутність, яка моделює та вміщує фото для житла.

Атрибути:

1. apartment_photo_path - шлях до статичного файлу (фото) апартаменту, обов’язкове поле, первинний ключ;
2. photo_desc – опис фото, необов’язкове поле.

д) Role – це сутність, яка моделює роль користувача.

Атрибути:

1. role_name – назва ролі, обов’язкове поле, первинний ключ;
2. role_description – опис ролі, необов’язкове поле.

Загалом, між таблицями можна виділити зв’язки. Кожен користувач може мати одну роль, тобто може бути або відпочиваючим або орендодавцем або організатором. Це зв’язок «багато-до-одного».

Від користувача також йдуть по два зв’язки до сутності Apartment та до сутності Facility. Ці зв’язки описують такий зв’язок як: «користувач може надати житло», «користувач може організувати захід». Тут зв’язок «один-до-багато», візьмемо, що у заході є тільки один основний організатор. Крім цього, з сутності User ще є зв’язки «багато-до-багато» до сутностей Facility та Apartment – це описує зв’язок «користувач може бронювати житло, але житло

можуть бронювати багато користувачів» та «користувач може бронювати захід, але квитки на захід бронюють багато користувачів».

Сутність Apartment, в свою чергу, має ще зв'язок «один до багато» з сутністю ApartmentPhoto – це описує «кожне житло може мати декілька оригінальних фото».

ВИСНОВКИ

В ході курсової роботи було розроблену систему купівлі квитків на відпочинок. Система орієнтована на українців віком від 18 років. Вона дозволяє шукати та бронювати житло по Україні та квитки на різноманітні заходи.

Для побудови стійкої архітектури було проведено передпроектний аналіз проблеми, який прояснив способи реалізації вирішення проблеми. Розробка програмного продукту відбувалась згідно методології Scrum, використовуючи спринти для кожного етапу.

Архітектура застосунку напряду залежить від побудованих діаграм, які допомогли зрозуміти краще взаємодію з користувачем, взаємодію всередині застосунку та побудувати оптимальний інтерфейс.

Основою системи є база даних, тому її цілісність є найпріоритетнішою задачею. Тому в якості бази обрано одну з найбільш просунутих – Oracle 11g. Тому більшість операцій реалізовано процедурно чи функціонально в самій базі даних. Крім того, реалізовані два способи фільтрування інформації: програмно за допомогою Python та за допомогою PIPELINED таблиць Oracle.

Сама програма, що забезпечує роботу системи (веб-сайту) реалізована мовою програмування Python з використанням деяких можливостей мікрофреймворку Flask[3]. В якості веб-сервера застосунку обрано Nginx[4] через його швидкість обробки статичних файлів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Learn About Scrum [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.scrumalliance.org/why-scrum>.
2. UML - Use Case Diagrams [Електронний ресурс] – Режим доступу до ресурсу:
https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm.
3. -. Flask documentation [Електронний ресурс] / - – Режим доступу до ресурсу:
<http://flask.pocoo.org/docs/0.12/>.
4. Nginx vs Apache [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>.

ДОДАТОК А

Use case діаграми для незареєстрованого користувача (рисунок А.1)

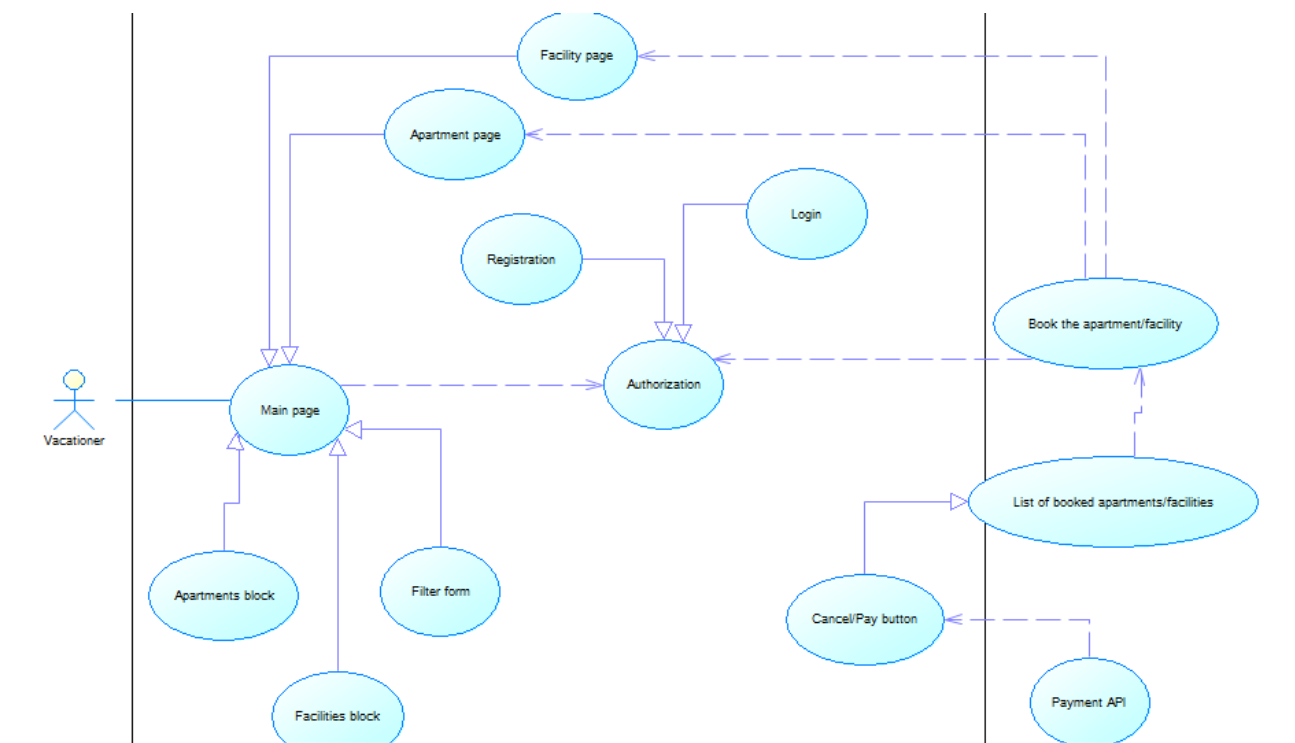


Рисунок А.1 – Use case для відпочиваючого

Use case для орендодавця/організатора (рисунок А.2)

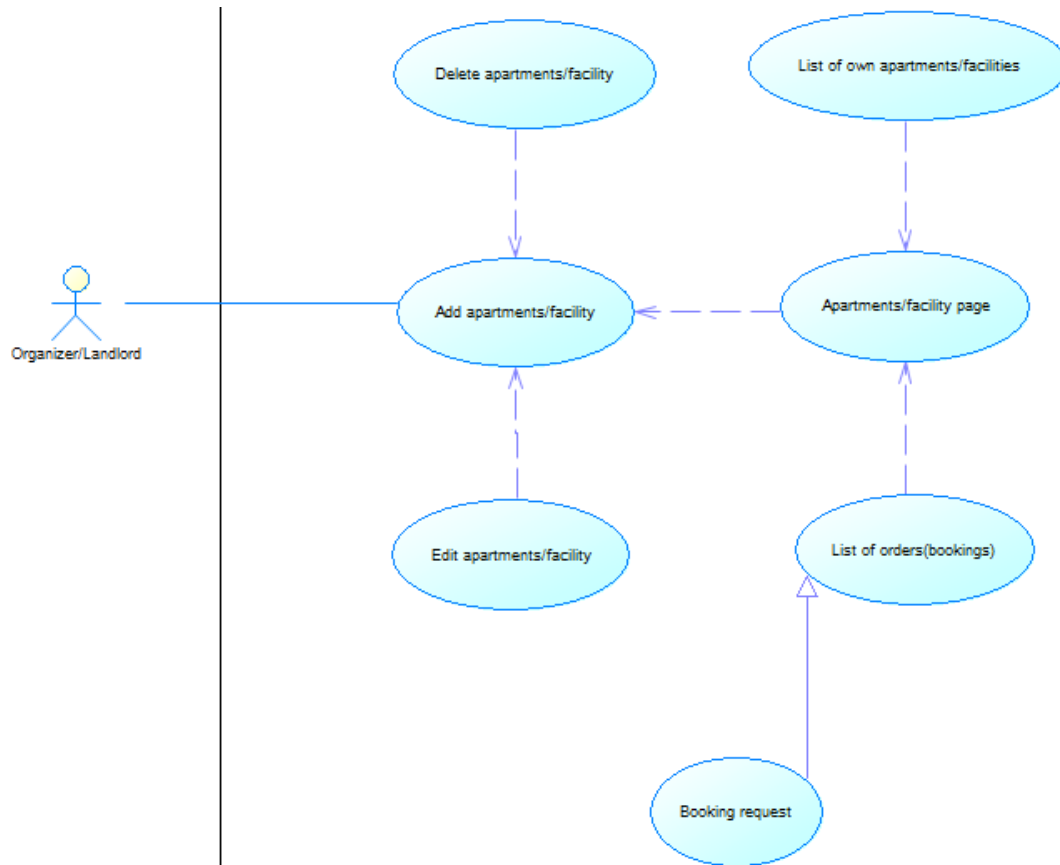


Рисунок А.2 – Use case для організатора/орендодавця

Діаграма компонент складається з частин:

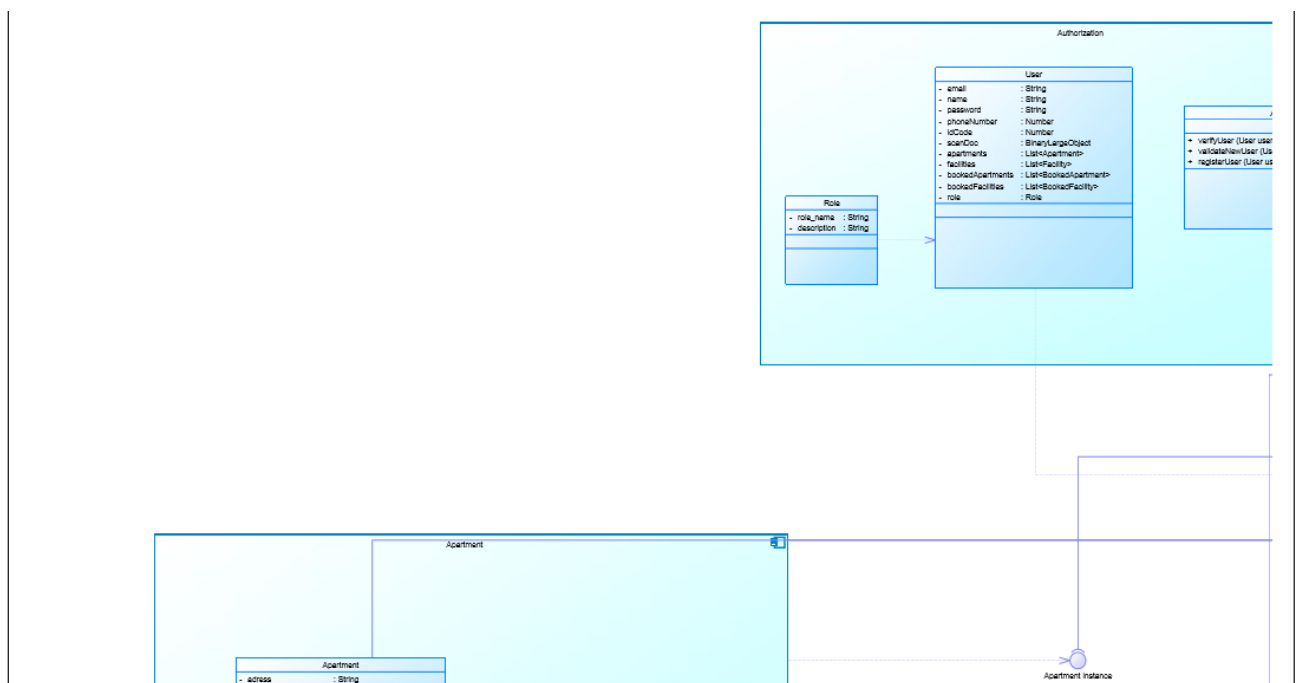


Рисунок А.3 – Компонента для авторизації

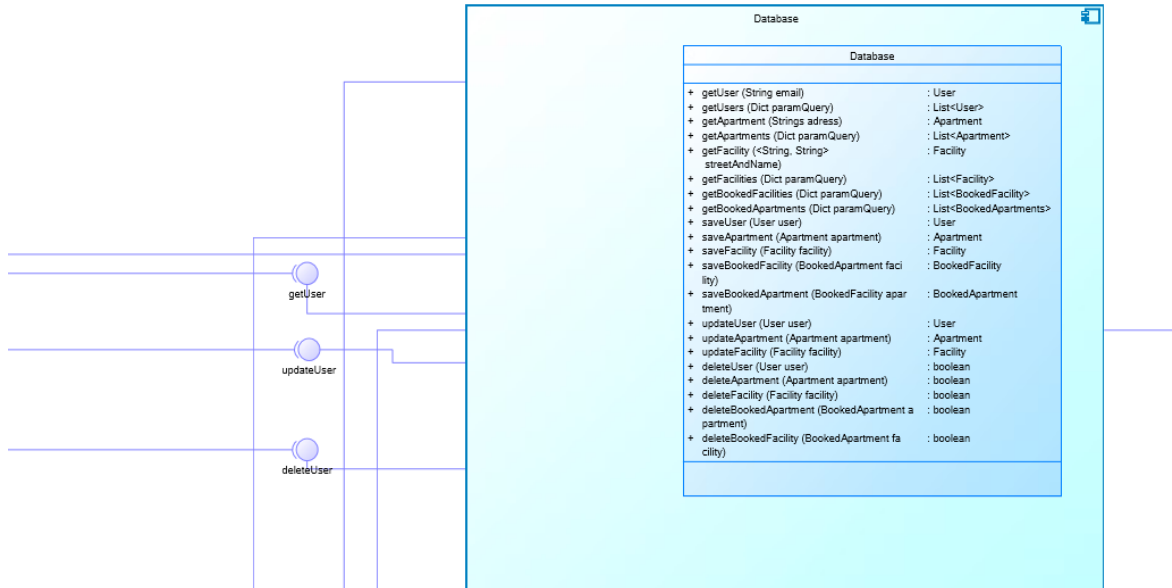


Рисунок А.3 – Компонента для сервера

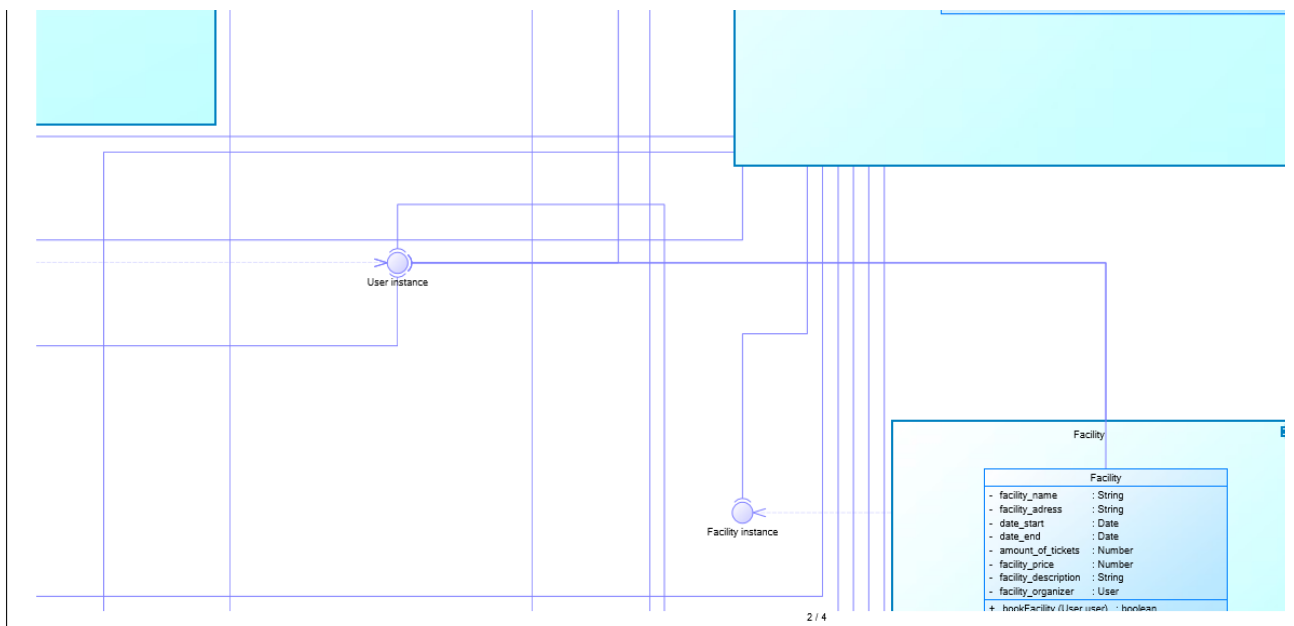


Рисунок А.4 – Компонента для заходів

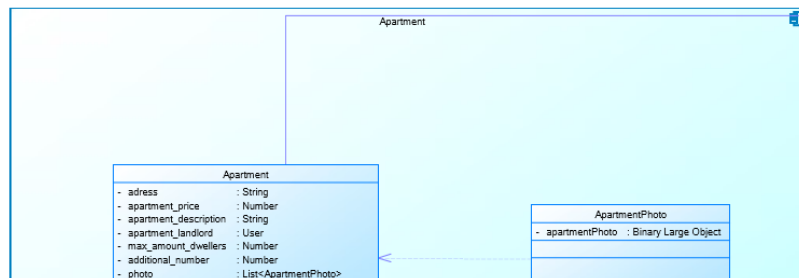


Рисунок А.5 – Компонента для житла

Діаграма послідовностей для організатора:

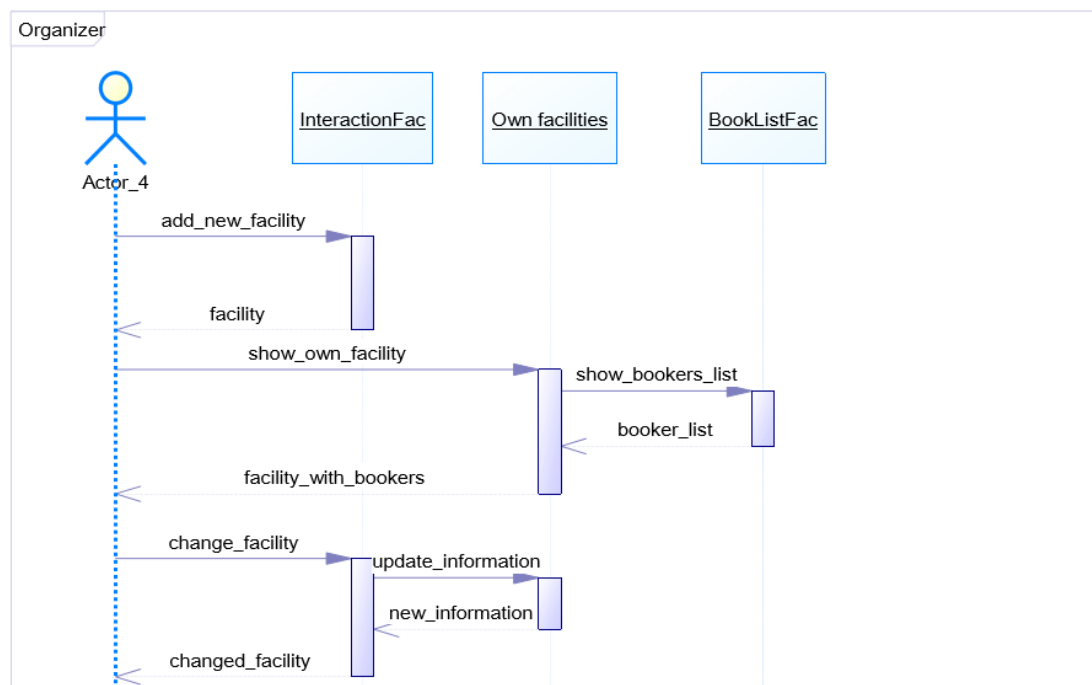


Рисунок А.6 – Діаграма послідовностей для організатора

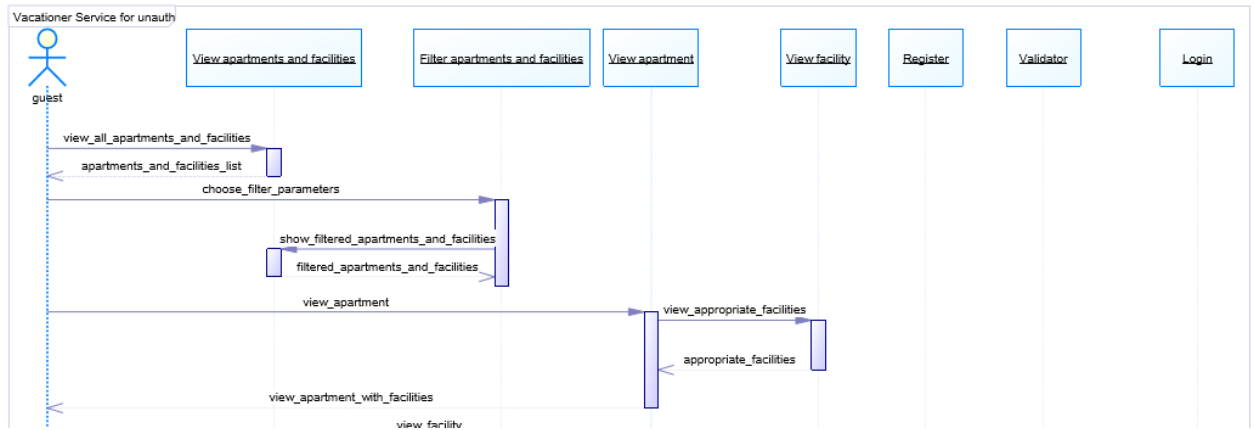


Рисунок А.7 – Діаграма послідовностей для незареєстрованого користувача

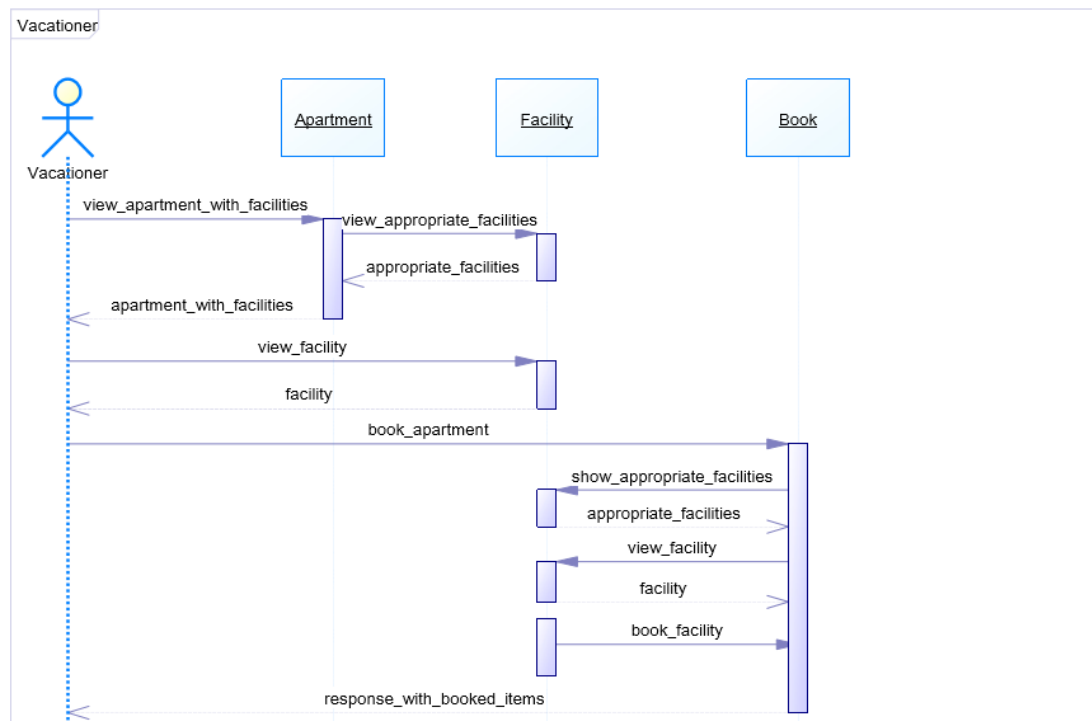


Рисунок А.8 – Діаграма послідовностей для відпочиваючого

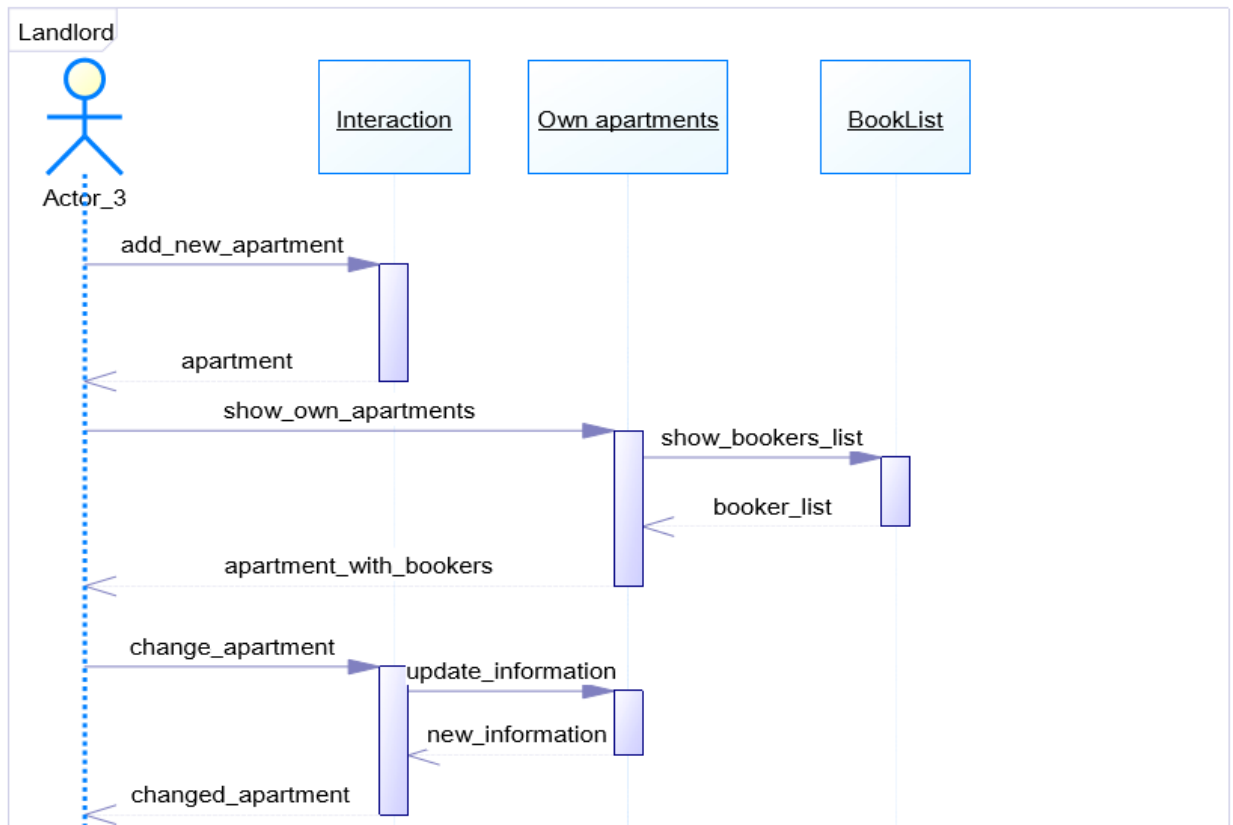


Рисунок А.9 – Діаграма послідовностей для орендодавця

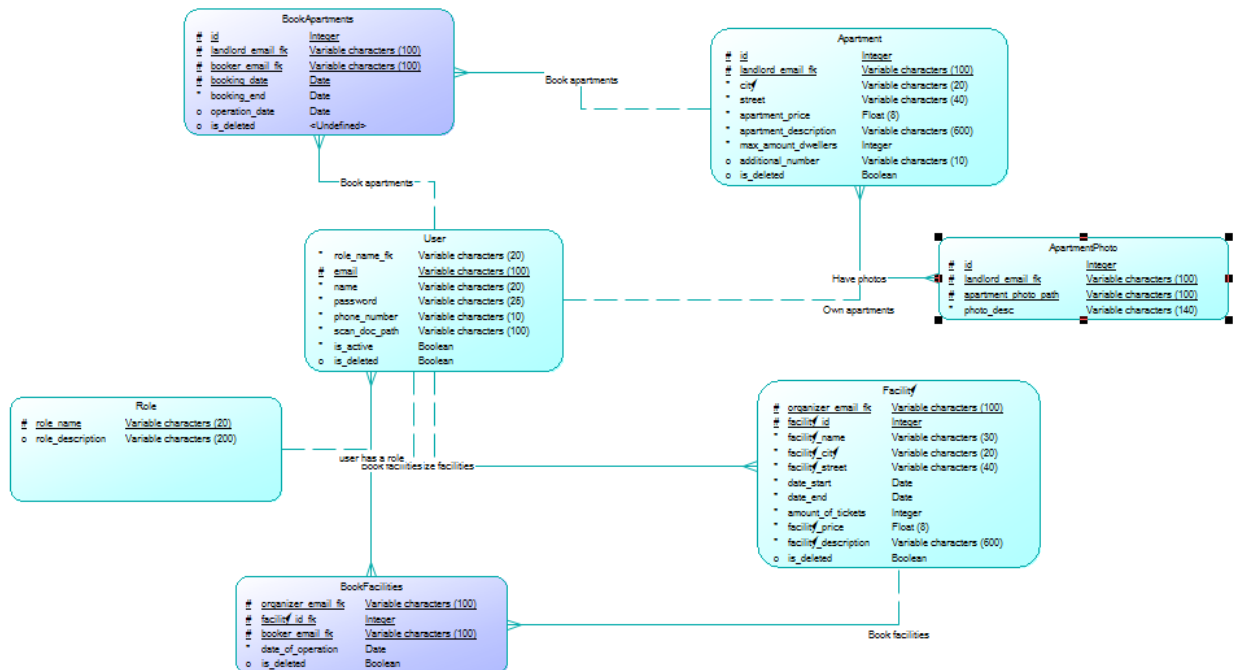


Рисунок А.10 – Оптимізована логічна модель

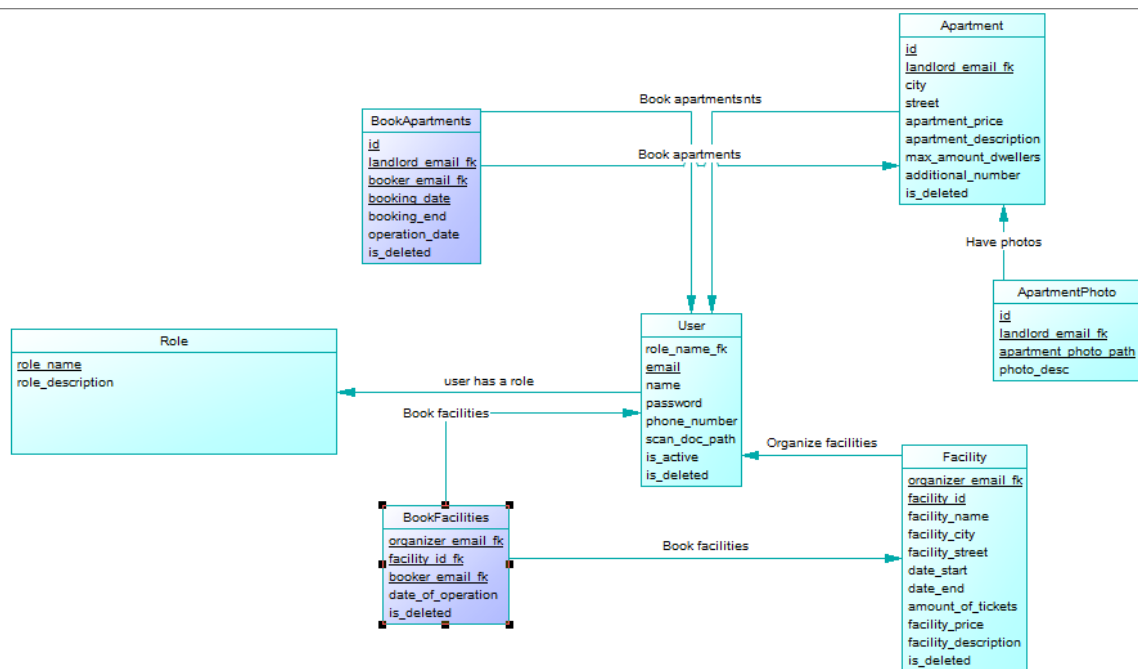


Рисунок А.11 – Фізична модель