Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

до лабораторної роботи №3

з дисципліни: “Інженерія програмного забезпечення”

на тему: «Розробка серверної частини. Розробка комунікації за протоколом TCP. Підключення серверного модуля до БД.»

Варіант №15

Виконав:

ст.гр. КІ-34

Ольховик О.С.

Прийняв:

ст. викл. каф.

Цигилик Л.О.

**Львів 2022**

**Мета:** Розробити консольну аплікацію що буде підтримувати зв’язок по протоколу TCP/IP, отримувати дані та записувати у БД. Також, згідно деякої команди, вичитувати з БД необхідну інформацію та передавати по TCP протоколу на клієнтську частину.

.

**Варіант №15:** Система бронювання та купівлі авіаквитків.

**Виконання**

**Вікно авторизації**

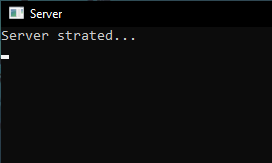


Рис.1 Вікно програми - сервера

Функція **IMongoCollection<T> ConnectToMongo<T>(in string collection)** виконує підключення до бази даних.

Функція **string AddNewFlight(Flight flight)** виконує додавання нового рейсу до бази даних.

Функція **void GeneretePlaces(Flight flight)** під час створення нового рейсу, генерує дані про місця заданого рейсу.

Функція **void UpdateFlightCollection(IMongoCollection<Flight> flightCol)** оновлює дані рейсів, змінюючи застарілі дати польотів на нові.

Функція **void UpdatePlaceCollection(IMongoCollection<Place> placeCol, Flight flight)** оноалює дані про політ, який зберігається в цій колекції.

Функція **string UpdateFlight(Flight flight)** оновлює дані, введені адміністатором, щодо заданого рейсу.

Функція **string UpdateUserInformation(User user)** оновлює дані користувача в базі даних.

Функція **(Flight[], string) DeleteFlight(Flight flight)** видаляє рейс з бази даних, вибраний адміністратором.

Функція **(User[], string) DeleteUser(User user)** видаляє користувача з бази даних, вибраний адміністратором.

Функція **(Flight[], string) GetAllFlights()** повертає список всіх рейсів адміністратору.

Функція **(Order[], string) GetAllOrders()** повертає список всіх замовлень адміністатору.

Функція **(User[], string) GetAllUsers()** повертає список всіх користувачів адміністратору.

Функція **(City[], string) GetAllCities()** повертає список всіх міст в базі даних, коли користувач хоче знайти рейси між потрібними йому пунктами.

Функція **(Flight[], string) GetFlights(Flight flight)** після отримання від користувача початкового та кінцевого пункту призначення і дати польоту, надсилає йому відповідний список рейсів.

Функція **(Place[], string) GetPlaces(Flight flight)** повертає список місць рейсу, який обрав користувач для бронювання.

Функція **(Order[], string) GetUserReservation(User user)** повертає список бронювань та куплених білетів користувачу.

Функція **void WriteAdminRequest(string request)** виводить в консоль програми-сервера запити, надіслані адміністратором на сервер.

Функція **void WriteUserRequest(User user, string str)** виводить в консоль програми-сервера запити, надіслані користувачем на сервер.

Функція **PrintError(string ex)** при виникненні виключних ситуацій на сервері, виводить помилку в консоль сервера.

Функція **User Autorization(User user)** перевіряє, чи існує такий користувач та здійснює вхід в систему.

Функція **void Connect(User user)** викликається при успішній авторизації коритстувача і записує до бази даних що цей користувач авторизований.

Функція **string Registration(User newUser)** перевіряє, чи такий користувач вже існує в базі даних, якщо ні, вона його реєструє.

Функція **void Disconnect(User user)** викликається, коли користувач хоче вийти із системи і записує до бази даних що цей користувач не аторизований.

Функція **(Order[], string) CancelReservation(Order order)** видаляє інформацію про бронювання з бази даних.

Функція **(Order[], string) BuyTicket(Order order)** записує до бази даних, що заброньоване місце вже куплене.

Функція **(Place[],string) ReservePlace(User user, Place place)** бронює вибране місце користувачем, записуючи це до бази даних.

Функція **void SetPlaceState(Place place, string state)** записує новий стан для заданого місця.

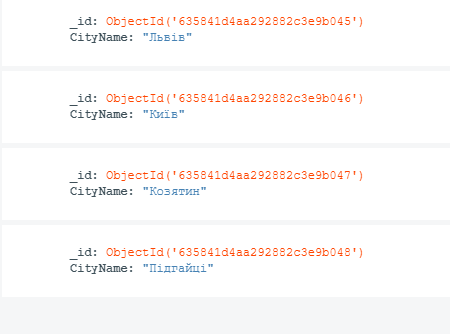


Рис. 2. База даних зареєстрованих користувачів

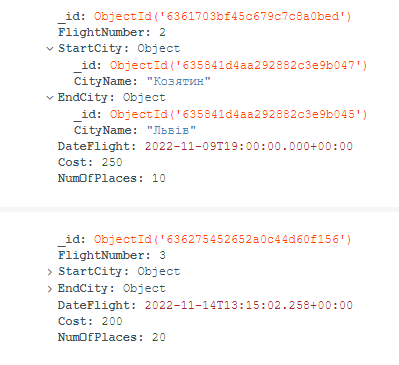


Рис. 3. База даних рейсів

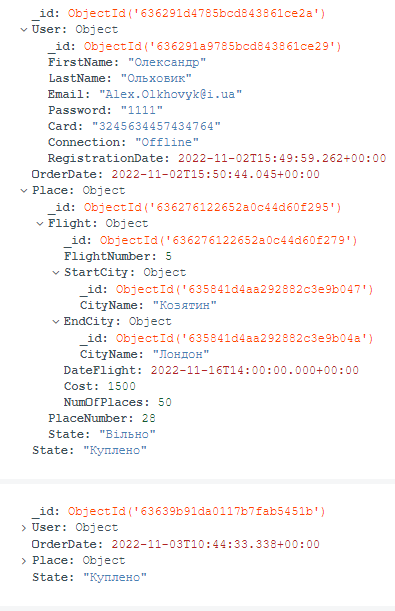


Рис. 4. База даних замовлень



Рис. 5. База даних місць



Рис. 5. База даних зареєстрованих користувачів

**Висновок:** На даній лабораторній роботі я освоїв основи розробки консольної аплікації що буде підтримувати зв’язок по протоколу TCP/IP, отримувати дані та записувати у БД. Також, згідно деякої команди, вичитувати з БД необхідну інформацію та передавати по TCP протоколу на клієнтську частину.