Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського" Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни

«Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET»

“**Проектування REST веб-API**”

Виконав(ла) *ІП-15 Костін В.А.* _____

(шифр, прізвище, ім'я, по батькові)

Перевірив *Бардін В.* _____

(прізвище, ім'я, по батькові)

Київ 2023

Лабораторна робота 3

Проектування REST веб-API

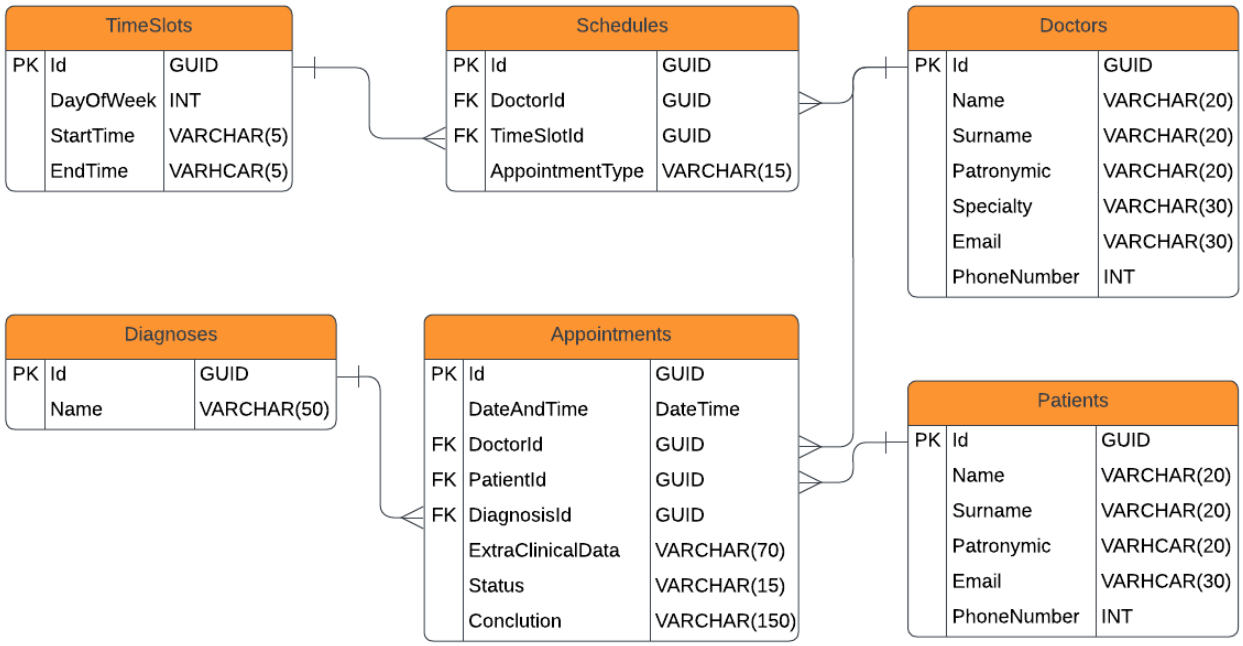
Завдання

1. З дотриманням вимог REST-у спроектувати веб-API для обраної(згідно варіанту) доменної області, використовуючи методологію C4 для створення діаграми архітектури системи.
2. Створити ER-діаграму для DAL (Data Access Layer), яка відображатиме структуру бази даних веб-API.
3. Оформити спроєктоване рішення у вигляді звіту до лабораторної роботи.

Доменна область:

9	Реєстратура лікарні. Запис до лікарів на прийом	1. Реєстратура надає дані стосовно наявності лікарів та розкладу прийому хворих. 2. Хворим можливо записатись на прийом до лікаря, якщо є вільний час у розкладі лікаря. 3. В реєстратурі ведуться картки відвідування хворими лікарні, в які записується час відвідання лікаря, діагноз та лікар, що його поставив. **Функціональні вимоги:** 1. Ведення картотеки лікарні; 2. Керування прийомами хворих у лікарів.

ER-діаграма



Посилання:

https://lucid.app/lucidchart/9ed9785f-d2af-42ed-ba34-6cc8900082e7/edit?viewport_loc=-29%2C25%2C2219%2C1057%2C0_0&invitationId=inv_ef3455d0-bdeb-4247-8dd7-c8e9277388b5

Таблиця: TimeSlots

Призначення: Проміжок часу для складання робочого графіку доктора.

Поля:

- Id: GUID – унікальний ідентифікатор проміжку часу.
- DayOfWeek:INT – день тижня.
- StartTime:VARCHAR(5) – початок проміжку часу.
- EndTime:VARCHAR(5) – кінець проміжку часу.

Таблиця: Schedules

Призначення: Часова одиниця робочого графіку доктора, опираючись на яку формуються записи.

Поля:

- Id: GUID – унікальний ідентифікатор часового графіку.
- DoctorId:GUID – унікальний ідентифікатор лякаря.
- TimeSlotId:GUID – унікальний ідентифікатор проміжку часу.
- AppointmentTime:VARCHAR(15) – тип запису, який припадає на лікаря та встановленні години(також може бути перервою).

Таблиця: Doctors

Призначення: Облікова таблиця докторів лікарні.

Поля:

- Id: GUID – унікальний ідентифікатор доктору.
- Name:VARCHAR(20) – ім’я доктору.
- Surname:VARCHAR(20) – прізвище доктору.
- Patronymic:VARHCAR(20) – по-батькові.
- Specialty:VARCHAR(30) – спеціальність доктора(хірург, травматолог, тощо).
- Email:VARCHAR(30) – поштова адреса доктора.
- PhoneNumber:INT – номер телефону доктора.

Таблиця: Patients

Призначення: Облікова таблиця пацієнтів лікарні.

Поля:

- Id: GUID – унікальний ідентифікатор пацієнта.
- Name:VARCHAR(20) – ім’я пацієнта.
- Surname:VARCHAR(20) – прізвище пацієнта.
- Patronymic:VARHCAR(20) – по-батькові.
- Email:VARCHAR(30) – поштова адреса пацієнта.
- PhoneNumber:INT – номер телефону пацієнта.

Таблиця: Diagnoses

Призначення: Діагнози, через які пацієнти записуються до лікаря.

Поля:

- Id: GUID – унікальний ідентифікатор діагноза.
- Name:VARCHAR(50) – назва діагноза.

Таблиця: Appointments

Призначення: Прийом пацієнта до лікаря з усією клінічною інформацією.

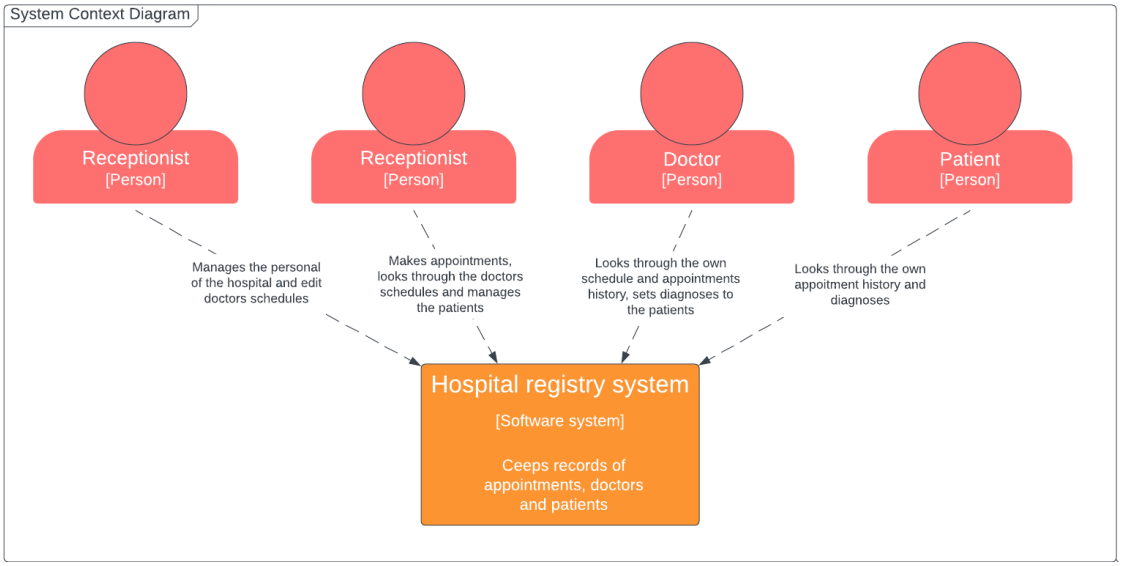
Поля:

- Id: GUID – унікальний ідентифікатор запису.
- DateAndTime:DateTime – дата та час запису.
- DoctorId:GUID - унікальний ідентифікатор доктора.
- PatientId:GUID - унікальний ідентифікатор пацієнта.
- DiagnosisId:GUID - унікальний ідентифікатор діагнозу.

- ExtraClinicalData:VARCHAR(70) – додаткові клінічні данні пацієнта перед прийомом.
- Status:VARCHAR(15) – статус прийому(“Scheduled”, “Active”, “Completed”, “Canceled”).
- Conclusion:VARCHAR(150) – висновок лікаря стосовно прийому.

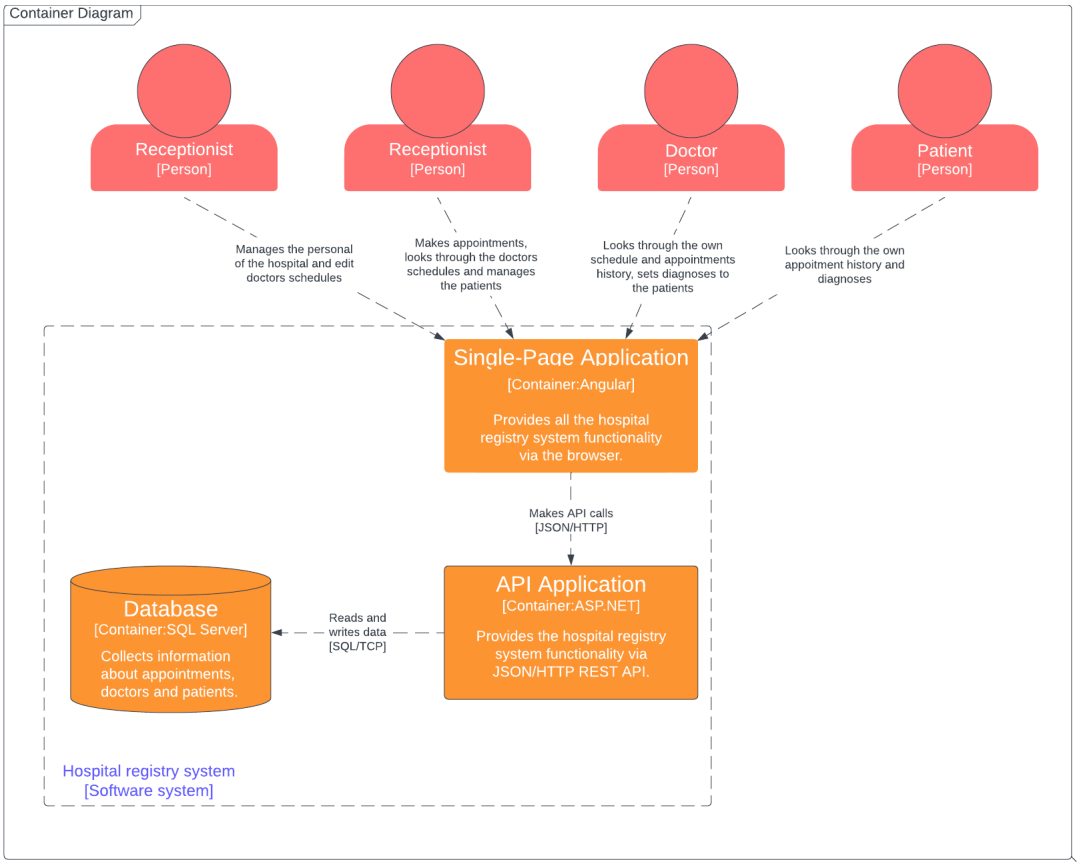
C4-діаграма

System Context Diagram



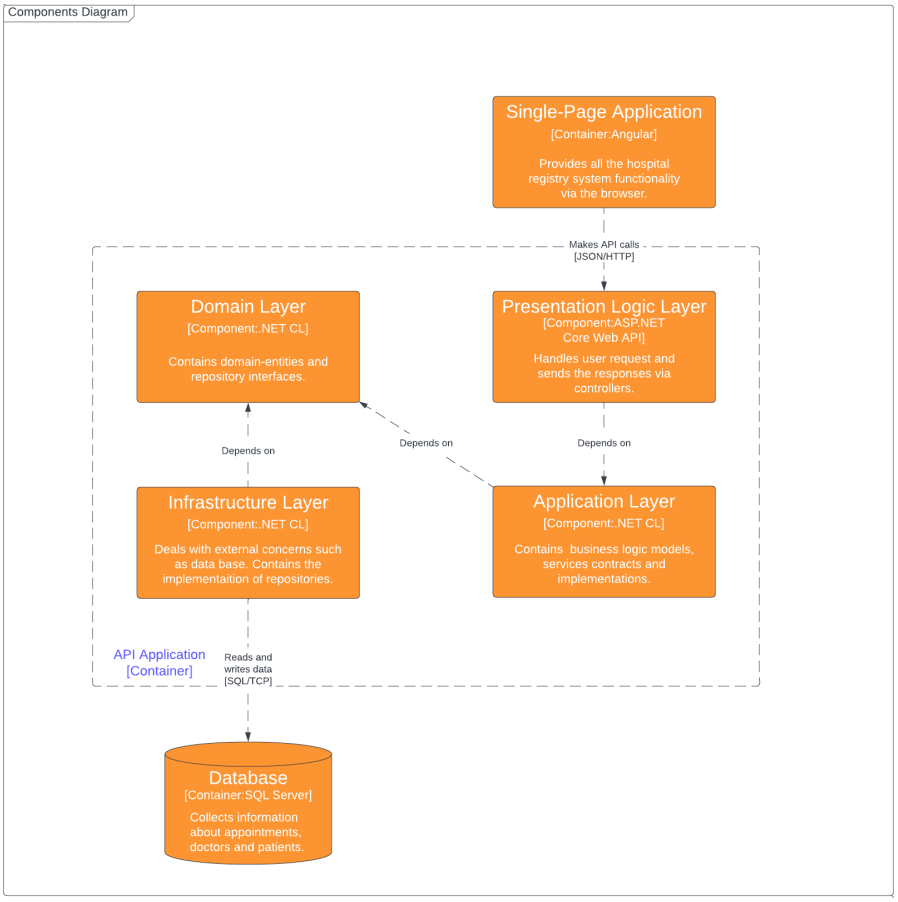
У системи “Реєстратура лікарні” будуть чотири типи користувачів: Адмін – керує даними про персонал лікарні(докторів), також налаштовує додаткові данні, наприклад, діагнози; Реєстратор – записує пацієнтів на прийом до лікаря, може продивлюватись графік роботи лікаря, веде картотеку пацієнтів; Доктор – продивляється свій робочий графік, історію прийомів, ставити діагноз пацієнту та висновок прийому; Пацієнт – продивлятись історію прийомів в лікарні.

Container Diagram



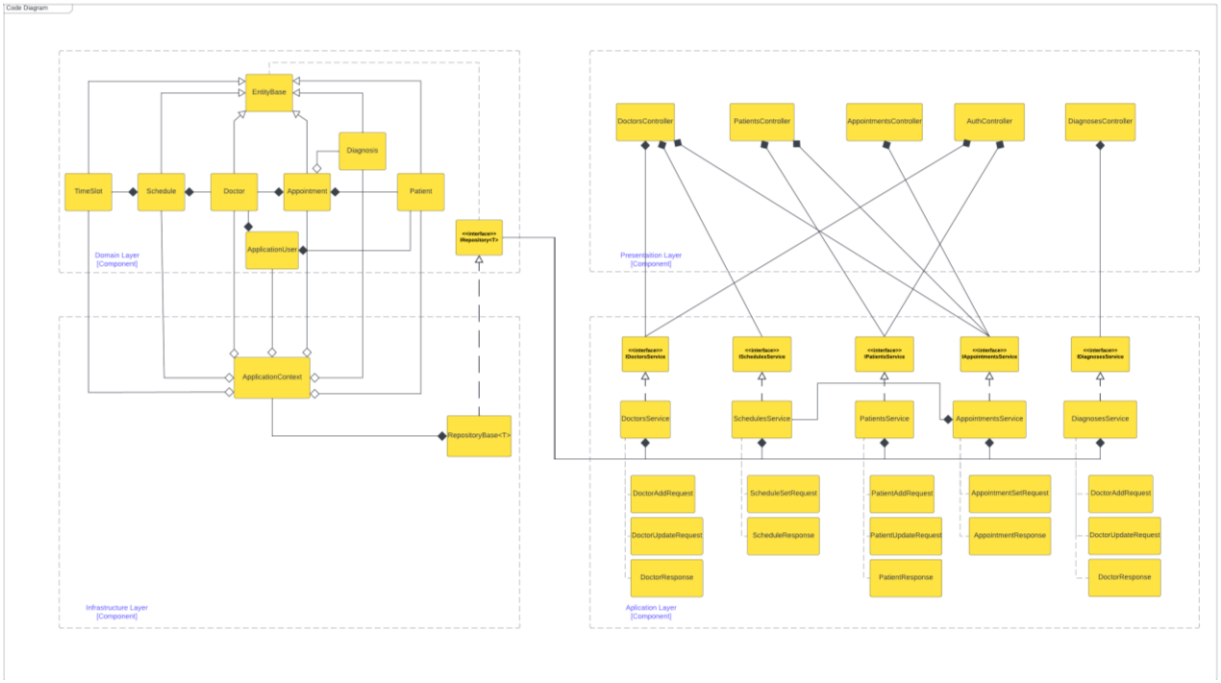
Система “Реєстратура лікарні” розділяється на три основні компоненти: веб-застосунок, який надає функціонал реєстратури через веб-браузер, серверний застосунок, який реалізує функціонал через API, та база даних – яка зберігає усю інформацію про лікарів, пацієнтів та прийоми.

Component Diagram



Для розробки бекенд-застосунку буде застосовано архітектурний паттерн “Clean Architecture”, який розділяє застосунку на наступні шари: Domain – ядро застосунку, містить сутності доменної області та контракти репозиторіїв; Application – шар бізнес логіки, містить контракти та реалізації сервісів, DTO класи та інші компоненти бізнес логіки; Infrastructure – компонент, який комунікує зі сторонніми сервісами або джерелами даних(таких як база даних), містить контекст бази даних та реалізацію репозиторіїв; Presentation – шар, який відповідає за комунікацію з користувачами та обробку запитів, містить контролери.

Code Diagram



Всі шари застосунку будуть реалізовані в якості бібліотеки класів, окрім Presentation(це проект ASP.NET Core Web API).

Доменний шар буде реалізовано з сутностями бази даних, які успадотковуються від абстрактного класу EntityBase. Це допоможе в дженеріковому інтерфейсі репозиторія прописати обмеження типу аргумента. Також шар містить клас ApplicationUser, який буде використовуватись для Identity.

Шар інфраструктури містить реалізацію дженерікового рипозиторію RepositoryBase та контекст бази даних ApplicationContext, який в свою чергу містить датасети всіх сутностей з домену. Для комуннікації з базою даних використовується EntityFramework.

Шар бізнес-логіки містить інтерфейси та реалізацію сервісів: IDoctorsService, ISchedulesService, IPatientsService, IAppointmentsService, IDiagnosesService. Також присутні відповідні DTO об’єкти: DoctorAddRequest, DoctorUpdateRequest, DoctorResponse, PatientAddRequest, PatientUpdateRequest, PatientResponse, ScheduleSetRequest, ScheduleResponse, AppointmentSetRequest, AppointmentResponse, DiagnosisAddRequest, DiagnosisUpdateRequest, DiagosisResponse.
Шар логіки інтерфейсу містить наступні контроллери: DoctorsController, PatientsControllet, AppointmentsController, AuthController, DiagnosesController.

Посилання:
https://lucid.app/lucidchart/fb9efa17-618d-41bf-a919-257b6424584c/edit?viewport_loc=-9191%2C-4159%2C32079%2C15288%2C0_0&invitationId=inv_6675198b-ced4-4c9b-b2ad-a63e059a9bd2