

Opis zadania projektowego

1. Temat i cel projektu

Temat: "Bazodanowy system obsługi przychodni medycznej"

Cel projektu: implementacja systemu baz danych oraz interfejsu użytkownika do obsługi przychodni przychodni z poziomu strony internetowej.

2. Opis działania i funkcje systemu

System umożliwiać będzie umówienie wizyty u konkretnego lekarza pracującego w przychodni w oparciu o relacyjną bazę danych. Klient za pomocą strony www będzie mógł: zarejestrować się w systemie, wyszukać lekarza, zapisać się na wizytę, wystawić opinię lekarzowi. Pracownik za pomocą systemu będzie mógł: dodać nowego pracownika, usunąć opinię o lekarzu, odwołać wizytę.

3. Założenia architektoniczne przyjęte podczas realizacji systemu

W ramach projektu zostanie zaimplementowany 3-warstwowy model komunikacji klient/serwer, który umożliwi efektywną obsługę różnych aspektów systemu. W zastosowanym modelu, przetwarzanie danych oraz funkcje biznesowe będą podzielone między aplikację użytkownika a serwer bazy danych, co zapewni optymalną wydajność i skalowalność.

Warstwa Aplikacyjna (Serwer Internetowy): W warstwie tej, obsługa funkcji biznesowych zostanie zrealizowana, a dostęp do tych funkcji będzie możliwy za pośrednictwem serwera www. Wykorzystamy aplikację internetową opartą na technologii PHP do realizacji tych funkcji biznesowych. Klient będzie mógł interaktywnie korzystać z aplikacji, wywoływać różnego rodzaju operacje i przekazywać żądania do serwera internetowego.

Warstwa Bazy Danych (Serwer Bazodanowy): W tej warstwie będzie zachodzić zarządzanie danymi oraz częściowe ich przetwarzanie. Operacje bazodanowe, takie jak odczyt, zapis i aktualizacja danych, będą realizowane po stronie serwera bazy danych. Baza danych PostgreSQL będzie odpowiedzialna za efektywne przechowywanie danych medycznych i umożliwi wykonywanie operacji na tych danych.

Warstwa Klienta (Przeglądarka Internetowa): Po stronie klienta, głównie zajmować się będziemy prezentacją danych. Przeglądarka internetowa będzie używana do wyświetlania informacji oraz interakcji z użytkownikiem. Dzięki temu, klient będzie mógł przeglądać profile lekarzy, umawiać się na wizyty, zarządzać swoimi danymi i wyświetlać historię wizyt, korzystając z przyjaznego interfejsu graficznego.

4. Wykorzystywane technologie, narzędzia projektowania oraz implementacji systemu

4.1. Technologie Programistyczne:

- Java: Język programowania do implementacji backendu systemu.
- Spring Boot: Framework do tworzenia aplikacji Java z wydajnym zarządzaniem kontenera i obsługą RESTful API.
- PostgreSQL: Baza danych do przechowywania danych medycznych pacjentów, wizyt, lekarzy itp.
- HTML, CSS, JavaScript: Technologie front-endowe do tworzenia interfejsu użytkownika.

4.2. Narzędzia Rozwojowe:

- IntelliJ IDEA.
- Maven: Narzędzia do zarządzania zależnościami i budowy projektu.
- Postman: Narzędzia do testowania i debugowania API.

4.3. Biblioteki i Frameworki:

- Spring Security: Do obsługi uwierzytelniania i autoryzacji użytkowników.
- Thymeleaf Do tworzenia dynamicznego interfejsu użytkownika.
- Hibernate: Biblioteka do mapowania obiektowo-relacyjnego (ORM) dla dostępu do bazy danych.

4.4. Baza Danych:

- PostgreSQL: System zarządzania relacyjną bazą danych, z możliwością definiowania struktury danych medycznych.

4.5. Serwer Aplikacyjny:

- Apache Tomcat lub inny serwer aplikacyjny obsługujący aplikacje Spring Boot.

4.6. Kontrola Wersji i Kolaboracja:

- Git: System kontroli wersji dla zarządzania kodem źródłowym.
- GitHub Platformy internetowe do przechowywania i kolaboracji nad kodem projektu

5. Schemat komunikacji, struktura systemu



