

Laboratorium analityczne to specjalistyczne środowisko, w którym przeprowadza się różnego rodzaju testy i analizy na próbek. Stosuje się w nim często systemy informatyczne do zarządzania danymi, gromadzenia wyników testów oraz obsługi zamówień i użytkowników systemu. W opisie klas laboratorium analitycznego wyróżniamy główne moduły, takie jak **BusinessLogic** i **ClassLibrary** oraz **Views**.

1. **BusinessLogic**- zawierający klasy reprezentujące obiektowo podstawowe tabele bazy danych takie jak:

- a. **DatabaseManagement**- Klasa odpowiedzialna za zarządzanie bazą danych SQL w laboratorium analitycznym. Podstawowe metody tej klasy to:
  - **LoadClients()**: Metoda pobiera listę klientów z bazy danych.
  - **LoadTechnicians()**: Metoda pobiera listę techników z bazy danych.
  - **InsertNewClient(Client c)**: Metoda dodaje nowego klienta do bazy danych.
  - **InsertNewTechnician(User t)**: Metoda dodaje nowego technika do bazy danych.
  - **ChangeUserPassword(int id, string newPassword)**: Metoda zmienia hasło użytkownika o podanym identyfikatorze.
  - **ChangeUserEmail(int id, string newEmail)**: Metoda zmienia adres e-mail użytkownika o podanym identyfikatorze.
  - **LoadTestTypes()**: Metoda pobiera listę rodzajów testów z bazy danych.
  - **InsertNewTestType(TestType t)**: Metoda dodaje nowy rodzaj testu do bazy danych.
  - **LoadTests()**: Metoda pobiera listę testów z bazy danych.
- b. **OrderManagement**- Klasa obsługująca proces zamówień klientów. Podstawowe metody tej klasy to:
  - **Initialize()**: Metoda inicjalizuje listy zamówień na podstawie danych pobranych z bazy danych.
  - **getFreeOrderID()**: Metoda zwraca wolny identyfikator dla nowego zamówienia.
- c. **TestManagement**- Klasa zarządzająca procesem przeprowadzania testów analitycznych. Podstawowe metody tej klasy to:
  - **Initialize(List<FlowLayoutPanel> flp)**: Metoda inicjalizuje listę testów na podstawie danych pobranych z bazy danych oraz przypisuje przekazane panele do odpowiednich zmiennych.
  - **CreateTest(string name, string type, string description, decimal minVal, decimal maxVal, int unit, decimal price)**: Metoda tworzy nowy test i dodaje go do listy testów.
  - **RemoveTest(int id)**: Metoda usuwa test o podanym identyfikatorze z listy testów.
- d. **TestTypeManagement**- Klasa obsługująca różne typy testów dostępne w laboratorium analitycznym. Podstawowe metody tej klasy to:
  - **Initialize()**: Metoda inicjalizuje listę rodzajów testów na podstawie danych pobranych z bazy danych.

- CreateType(string name): Metoda tworzy nowy rodzaj testu i dodaje go do listy.
  - RemoveType(string name): Metoda usuwa rodzaj testu o podanej nazwie z listy.
  - GetNameList(): Metoda zwraca listę nazw rodzajów testów.
  - GetTypeID(string name): Metoda zwraca identyfikator rodzaju testu na podstawie podanej nazwy.
  - GetTypeName(int id): Metoda zwraca nazwę rodzaju testu na podstawie podanego identyfikatora.
- e. **UserManagement:** Klasa zarządzająca danymi użytkowników laboratorium.
- Podstawowe metody tej klasy to:
- Initialize(): Metoda inicjalizuje listy kont użytkowników na podstawie danych pobranych z bazy danych.
  - RegisterClient(string login, string password, string name, string surname, DateTime birthdate, string email, string pesel, string residence, string phoneNum): Metoda rejestruje nowego klienta i dodaje go do listy kont klientów.
  - RegisterTechnician(string login, string password, string name, string surname, DateTime birthdate, string pesel, string residence, string phoneNum): Metoda rejestruje nowego technika i dodaje go do listy kont techników.
  - ChangeAccountStatus(int id, int type): Metoda zmienia status konta użytkownika o podanym identyfikatorze i typie.
  - GetTechnicianList(): Metoda zwraca listę informacji o technikach.
  - GetTechnicianByID(int id): Metoda zwraca obiekt technika na podstawie podanego identyfikatora.
  - IsValidLogin(string login): Metoda sprawdza poprawność formatu loginu.
  - IsValidPassword(string password): Metoda sprawdza poprawność formatu hasła.
  - IsValidEmail(string email): Metoda sprawdza poprawność formatu adresu e-mail.
  - IsValidPesel(string pesel): Metoda sprawdza poprawność formatu numeru PESEL.
  - IsValidBirthdate(DateTime date): Metoda sprawdza poprawność podanej daty urodzenia.
  - IsEmailAlreadyUsed(string email): Metoda sprawdza, czy podany adres e-mail jest już używany.
  - IsLoginAlreadyUsed(string login, int userType): Metoda sprawdza, czy podany login jest już używany dla określonego typu użytkownika.
  - IsPeselAlreadyUsed(string pesel, int userType): Metoda sprawdza, czy podany numer PESEL jest już używany dla określonego typu użytkownika.
  - LogInUser(int userType, string login, string password): Metoda loguje użytkownika o podanym typie, loginie i hasle.
  - LogOutUser(): Metoda wylogowuje obecnie zalogowanego użytkownika.

- `ChangeAccountPassword(int id, string newPassword)`: Metoda zmienia hasło konta o podanym identyfikatorze.
- `ChangeAccountEmail(int id, string newEmail)`: Metoda zmienia adres e-mail konta o podanym identyfikatorze

## 2. ClassLibrary

**3. Views-** implementuje graficzny interfejs użytkownika. Zawarte w nim klasy umożliwiają użytkownikowi aplikacji modyfikowanie bazy danych, wyświetlanie jej zawartości zgodnie z oczekiwaniami oraz szeroko pojętą nawigację wewnątrz programu. Zawarte w module klasy do działania wykorzystują m.in. okna, pola tekstowe, przyciski, pola wyboru, listy, zakładki. Wyświetlane informacje z bazy danych przechowywane są w odpowiednich obiektach modeli. Oto klasy w module Views:

- `AccountDeleteConfirmation` - Klasa reprezentuje widok potwierdzenia usunięcia konta. Zapewnia interfejs, który umożliwia użytkownikowi potwierdzenie decyzji o usunięciu swojego konta.
- `AdminView` - Klasa reprezentuje widok panelu administracyjnego. Ten widok jest dostępny tylko dla administratorów systemu. Zapewnia interfejs, który umożliwia administratorowi zarządzanie użytkownikami, testami i innymi funkcjonalnościami systemu.
- `ClientView` - Klasa reprezentuje widok panelu klienta. To jest widok dostępny tylko dla klientów. Zapewnia interfejs, który umożliwia klientowi przeglądanie dostępnych testów, składanie zamówień, sprawdzanie statusu zamówień itp.
- `LoginRegisterView` - Klasa reprezentuje widok logowania i rejestracji. Zapewnia interfejs, który umożliwia użytkownikowi logowanie się do systemu lub rejestrację nowego konta.
- `TechnicianView` - Klasa reprezentuje widok panelu technika. Ten widok jest dostępny tylko dla techników. Zapewnia interfejs, który umożliwia technikowi przeglądanie przypisanych mu zamówień, przeprowadzanie testów, wprowadzanie wyników i innych działań związanych z pracą technika.