

Wyższa Szkoła Bankowa w Poznaniu

Wydział Finansów i Bankowości

Kierunek: Informatyka

Studia niestacjonarne I stopnia

Komis meblowy

Twój Meblownik

Przedmiot: Programowanie zaawansowane

Grupa: nlinz_3_K07_Inf

Prowadzący: dr Paweł Płaczek

Autorzy:

- 1. Hubert Baluszyński**
- 2. Szymon Rajczyk**

Spis treści

1. Krótki opis działania projektu.....	3
2. Krótki opis funkcji jakie będą znajdować się na stronie	3
3. Specyfikacja wykorzystanych technologii	3
4. Instrukcje pierwszego uruchomienia projektu	3
5. Opis struktury projektu	3
6. Wylistowane wszystkie modele	4
7. Wylistowane kontrolery wraz z metodami.....	5
8. Opis systemu użytkowników	6
9. Krótka charakterystyka najciekawszych funkcjonalności	6

1. Krótki opis działania projektu

Projekt ma na celu stworzenie strony internetowej, na której użytkownicy będą mogli wystawiać używane bądź nowe meble, które zalegają im w mieszkaniu lub potrzebują nowego właściciela. Nie będzie konieczności zalogowania się, aby przeglądać wystawione meble, jednakże osoby zalogowane będą miały możliwość dodawania (na stronę publiczną) i zarządzania swoimi meblami.

2. Krótki opis funkcji jakie będą znajdować się na stronie

Użytkownicy będą mieli możliwość zarejestrowania się na stronie oraz zalogowania się na utworzone konto. Po zalogowaniu użytkownik będzie miał dostęp do głównej tablicy, na której będzie mógł przeglądać wystawione tam meble.

Zalogowany użytkownik będzie miał również dostęp do zakładki ze swoją bazą mebli, które wystawił na stronie. Osoba niezalogowana będzie miała jedynie możliwość przeglądania głównej tablicy oraz wysłania maila do sprzedającego.

Po wejściu w mebel użytkownik będzie w stanie sprawdzić podobne meble z tej samej kategorii w sekcji "Podobne meble".

3. Specyfikacja wykorzystanych technologii

W ramach realizacji projektu zastosowano technologię C# ASP.NET 6.0, baza danych: MS SQL Server.

4. Instrukcje pierwszego uruchomienia projektu

W celu pierwszego uruchomienia projektu na komputerze lokalnym zalecamy wykonać następujące czynności:

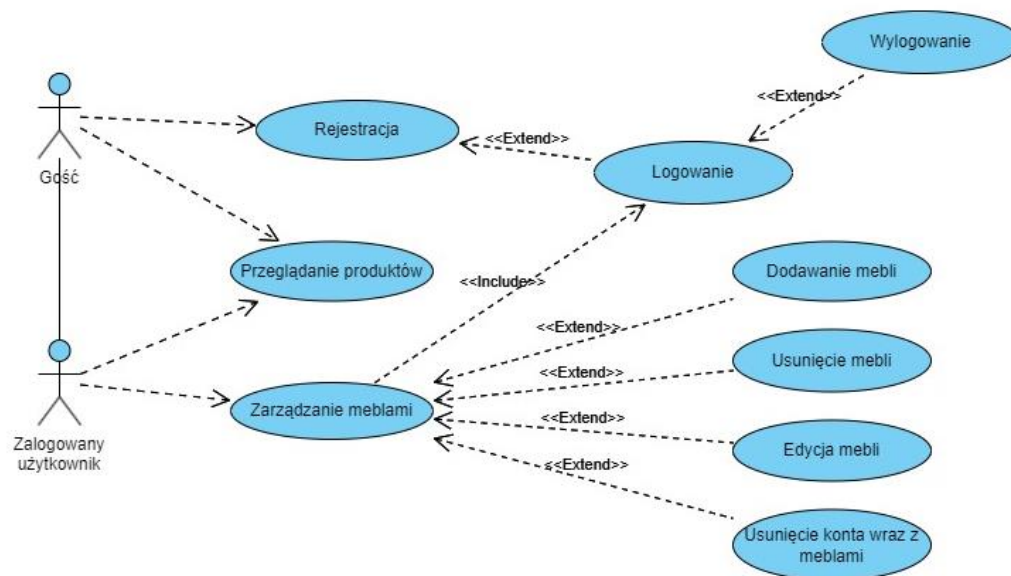
- zmiana nazwy serwera (aby sprawdzić nazwę serwera należy w CMD użyć komendy „hostname”) w appsettings.json na lokalny serwer w polu "Data Source"
- uruchomienie Konsoli Menedżera Pakietów i uruchomienie komendy "Update-Database"

W przypadku problemów z uruchomieniem projektu (projekt jest działający) prosimy o kontakt poprzez Facebook - Szymon Rajczyk lub Hubert Bałuszyński, lub poprzez Discord - brekker755 (Hubert Bałuszyński) lub szefraju (Szymon Rajczyk).

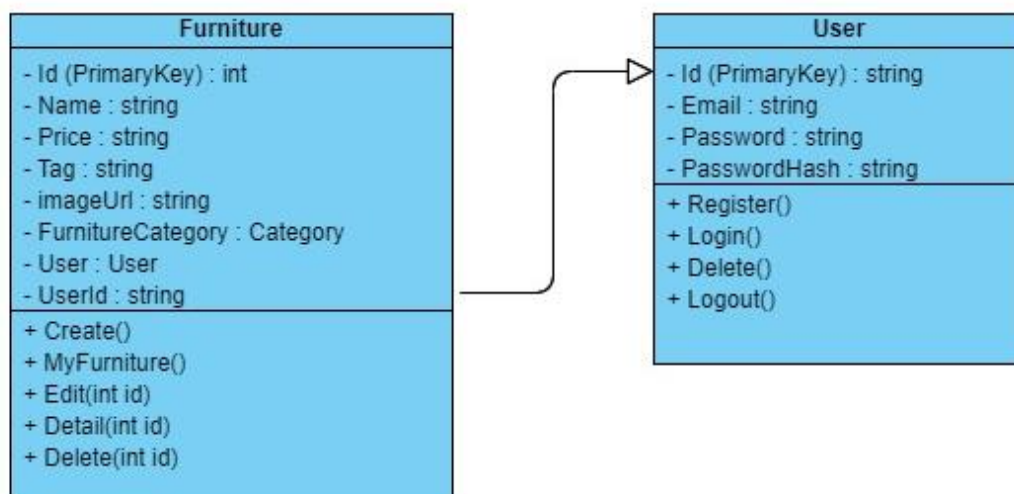
5. Opis struktury projektu

Projekt został stworzony w architekturze ASP.NET MVC (Model-View-Controller), który dzieli aplikację na trzy główne komponenty: Model, Widok i Kontroler. W projekcie posiadamy Modele: ErrorViewModel, Furniture oraz User (dokładniej opisane w dalszej części dokumentacji technicznej). Widoki: Account (Login.cshtml, Register.cshtml, Welcome.cshtml), Furniture (Create.cshtml, Delete.cshtml, Detail.cshtml, Edit.cshtml, Index.cshtml, MyFurniture.cshtml), Home (Privacy.cshtml) oraz Shared. Kontrolery: AccountController,

FurnitureController oraz HomeController (dokładniej opisane w dalszej części dokumentacji technicznej).



Obraz 1 - Diagram przypadków użycia



Obraz 2 - Diagram klas

6. Wylistowane wszystkie modele

Model User.cs w aplikacji ASP.NET Core MVC opartej na Microsoft Identity Framework jest klasą reprezentującą użytkownika systemu. Klasa User dziedziczy z klasy IdentityUser, która

jest częścią biblioteki Microsoft.AspNetCore.Identity. Klasa IdentityUser dostarcza zestaw właściwości oraz funkcji używanych do zarządzania użytkownikami w aplikacjach ASP.NET Core. Model User jest używany do przechowywania informacji o użytkownikach, takich jak nazwa użytkownika, email, hasło itp. W naszym projekcie wykorzystaliśmy pola Id - typ: string, jest to unikalny identyfikator użytkownika, Email - Typ: string, jest to adres email użytkownika, ograniczenia: maksymalna długość to 256 znaków oraz PasswordHash - typ: string, hasło posiada ograniczenia - musi mieć co najmniej 8 znaków, dużą i małą literę, cyfrę oraz znak specjalny, jest to hash hasła podanego przez użytkownika.

Model Furniture.cs reprezentuje mebel w aplikacji ASP.NET Core. Służy do przechowywania informacji o poszczególnych meblach, takich jak nazwa, cena, obraz, tag, użytkownik, który dodał mebel, oraz kategoria mebla. W naszym projekcie stworzyliśmy pola Id, typ: int, posiada atrybut: [Key], jest to unikalny identyfikator mebla, jest to klucz główny, Name - typ: string, jest to nazwa dodanego mebla, Tag - typ: string, jest to pole, które ma przechowywać opis mebla, Price - typ: string, przechowuje cenę mebla, ImageUrl - typ: string, przechowuje adres URL do obrazu przedstawiającego mebel, UserID - typ: string, posiada atrybut [ForeignKey("User")], przechowuje identyfikator użytkownika, który dodał mebel, User - typ: string, obiekt reprezentujący użytkownika, który dodał mebel, posiada relacje jeden do wielu z modelem User, FurnitureCategory - typ: string, zawiera kategorię do której należy mebel.

7. Wylistowane kontrolery wraz z metodami

AccountController - metody HTTP: GET i POST, *GET /Account/Login* wyświetla stronę logowania, brak parametrów, zwraca widok z pustym modelem LoginViewModel. *POST /Account/Login* obsługuje proces logowania użytkownika, posiada parametry LoginViewModel (parametry przesyłane w formularzu), EmailAddress (string) - adres email użytkownika, Password (string) - hasło użytkownika. Podczas działania sprawdza poprawność modelu, wyszukuje użytkownika na podstawie adresu email, jeżeli użytkownik istnieje wyświetla komunikat o błędzie, jeśli użytkownik nie istnieje, tworzy nowego użytkownika, rejestruje go i przekierowuje na stronę główną. *GET /Account/Logout* obsługuje wylogowanie użytkownika, brak parametrów, przekierowuje na stronę główną. *POST /Account/Delete* usuwa konto użytkownika oraz wszystkie powiązane z nim meble, brak parametrów, jest wykonywane na podstawie aktualnie zalogowanego użytkownika, usuwa konto wraz z wszystkimi powiązanymi z nim meblami, wylogowuje użytkownika oraz przekierowuje na stronę główną. Kontroler posiada modele *LoginViewModel* z polami EmailAddress oraz Password, oraz *RegisterViewModel* z takimi samymi polami.

FurnitureController - metody HTTP: GET i POST, *GET /Furniture/Index* wyświetla listę wszystkich mebli, brak parametrów, zwraca widok z listą obiektów Furniture. *GET /Furniture/MyFurniture* wyświetla listę mebli aktualnie zalogowanego użytkownika, brak parametrów, zwraca widok z listą obiektów Furniture należących do zalogowanego użytkownika. *GET /Furniture/Detail/{id}* wyświetla szczegóły mebla o podanym id, posiada parametr id (identyfikator mebla), pobiera szczegóły mebla z bazy danych, pobiera listę powiązanych mebli z tej samej kategorii, zwraca widok z modelem DetailFurnitureViewModel, zawierającym szczegóły mebla oraz listę powiązanych mebli. *GET /Furniture/Create* wyświetla formularz tworzenia nowego mebla, brak parametrów, zwraca widok z pustym modelem CreateFurnitureViewModel. *POST /Furniture/Create* obsługuje proces tworzenia nowego mebla. Posiada parametry przesyłane w formularzu: Name (string) - nazwa mebla, Tag (string) - opis mebla, Price (decimal) - cena mebla, ImageUrl (string) - URL obrazka

mebla, UserId (string) - identyfikator użytkownika tworzącego mebel, Category (string) - kategoria mebla, sprawdza poprawność modelu, a następnie tworzy nowy obiekt Furniture i zapisuje go w bazie danych w przypadku sukcesu, następnie przekierowuje na stronę główną. *GET /Furniture/Edit/{id}* wyświetla formularz edycji mebla o podanym id, posiada parametr id (identyfikator mebla), pobiera szczegóły mebla z bazy danych, sprawdza czy mebel należy do aktualnie zalogowanego użytkownika, zwraca widok z modelem EditFurnitureViewModel w przypadku sukcesu lub widok błędu w przypadku braku uprawnień. *POST /Furniture/Edit* obsługuje proces edycji mebla, posiada parametry przesyłane w formularzu: Id (int) - identyfikator mebla, Name (string) - nazwa mebla, Tag (string) - opis mebla, Price (decimal) - cena mebla, ImageUrl (string) - URL obrazka mebla, Category (string) - kategoria mebla, sprawdza poprawność modelu, a następnie aktualizuje obiekt Furniture w bazie danych, następnie przekierowuje na stronę główną. *GET /Furniture/Delete/{id}* wyświetla stronę potwierdzenia usunięcia mebla o podanym id, posiada parametr id (identyfikator mebla), pobiera szczegóły mebla z bazy danych, sprawdza, czy mebel należy do aktualnie zalogowanego użytkownika. *POST /Furniture/Delete/{id}* obsługuje proces usuwania mebla, posiada parametr id, pobiera szczegóły mebla z bazy danych, sprawdza, czy mebel należy do aktualnie zalogowanego użytkownika, usuwa model z bazy danych, następnie przekierowuje do strony głównej. Kontroler posiada modele Furniture z polami Id (int) - identyfikator mebla, Name (string) - nazwa mebla, Tag (string) - opis mebla, Price (decimal) - cena mebla, ImageUrl (string) - URL obrazka mebla, UserId (string) - identyfikator użytkownika, Category (string) - kategoria mebla. DetailFurnitureViewModel z polami Furniture (Furniture) - szczegóły mebla, RelatedFurnitures (List<Furniture>) - lista powiązanych mebli. CreateFurnitureViewModel - z polami identycznymi jak Furniture bez pola Id, EditFurnitureViewModel z polami identycznymi jak Furniture.

HomeController - metody HTTP: GET, *GET /Home/Index* wyświetla stronę główną, brak parametrów, zwraca widok strony głównej. *GET /Home/Privacy* wyświetla stronę polityki prywatności, brak parametrów, zwraca widok strony polityki prywatności. *GET /Home/Error* wyświetla stronę błędu, brak, zwraca widok z modelem ErrorViewModel zawierającym RequestId. Kontroler posiada model ErrorViewModel z polem RequestId (string) - identyfikator żądania.

8. Opis systemu użytkowników

Gość (globalny) - jest to każdy niezalogowany użytkownik, może przeglądać meble, a następnie skontaktować się mailowo z wystawiającym użytkownikiem, ma on również możliwość rejestracji, a następnie zalogowania.

Zalogowany użytkownik - posiada możliwość dodawania mebli na stronę, może on również edytować i usuwać swoje meble, posiada możliwość wylogowania oraz usunięcia konta wraz ze swoimi wszystkimi meblami.

9. Krótka charakterystyka najciekawszych funkcjonalności

Uważamy, że najciekawszymi funkcjonalnościami są: możliwość podejrzenia podobnych mebli z tej samej kategorii po wejściu w mebel w sekcji "Podobne meble", pozwala to użytkownikowi na łatwiejsze znalezienie mebla, którego potrzebuje. Możliwość usunięcia

konta wraz ze wszystkimi aktualnie dodanymi meblami, eliminuje to sytuacje w której, użytkownik usuwa konto, a jego meble są cały czas dostępne na stronie.