Akademia Górniczo - Hutnicza im. Stanisława Staszica w Krakowie

Wydział Informatyki, Elektroniki i Telekomunikacji

Katedra Informatyki



***Mniam Mniam!***

***Aplikacja webowa umożliwiająca przechowywanie, ocenianie, oraz udostępnianie innym użytkownikom przepisów kulinarnych***

**Dokumentacja deweloperska**

**Autorzy**: Krzysztof Wrona, Krzysztof Suruło, Piotr Puszkarski, Piotr Maciejowski

## **Spis treści.**

[Opis struktury.](#_hmgtmofbydvw)

[Opis zaimplementowanych modułów.](#_iu7k95ultn9m)

[Możliwe rozszerzenia.](#_tox28zv3pzr5)

## **Opis struktury.**

**Folder MniamMniam**

Główny folder, w którym zawierają się wszystkie foldery, oraz dwie ważne klasy. Struktura folderów przedstawia wzorzec strukturalny MVC.

**Klasa Program**

Klasa zawierająca Main(), która odpowiada za stworzenie hosta i odpalenie go. Wykorzystuje klasę Startup.

**Klasa Startup**

Klasa odpowiadająca za ustawienie konfiguracji jak np. użycie zewnętrznej autoryzacji Google, ustawienie secretów, lub też ustawienie kontekstu bazy danych.

**Folder MniamMniam/Controllers**

Zawiera wszystkie kontrolery do obsługi widoków odpowiadających im widoków.

**Klasa AccountController**

Jest to klasa rozszerzająca klasę Controller. Obsługuje odpowiednie zapytania GET i POST odpowiednich widoków.

AccountController zawiera metody:

GET/POST Login - metoda odpowiadająca za obsługę żądań GET lub POST w widoku login. GET Login odpowiada za wyświetlenie widoku Login, a POST Login za zareagowanie na wprowadzone dane do widoku. Podobnie jest z resztą metod kontrolerów.

GET/POST Register - analogicznie do Login, odpowiada za interakcję z widokiem register.

POST Logout - obsługą żądania wylogowania użytkownika.

POST ExternalLogin - metoda przekierowująca do zewnętrznego dostawcy logowania, w tym przypadku logowania za pomocą konta Google.

GET ExternalLoginCallback - odpowiada za to czy zalogować użytkownika jeśli jego login istnieje, czy też poprosić go o zarejestrowanie się jeśli nie ma konta.

GET ExternalLoginConfirmation - potwierdzenie utworzenia konta przez użytkownika przy użyciu zewnętrznego dostawcy.

GET/POST ForgotPassword - obsługa widoku forgotpassword.

GET/POST ResetPassword - obsługa widoku resetpassword.

GET/POST SendCode - obsługa widoku sendCode odpowiadającego za dwustopniową weryfikację.

GET/POST VerifyCode - obsługa widoku verifyCode.

GET AccessDenied - obsługa widoku accessDenied, który odpowiada za wyświetlenie informacji o nieautoryzowanym wstępie.

**Klasa HomeController**

Klasa rozszerzająca Controller, która ma trzy metody odpowiadająca za stronę domową.  
  
Metoda Index zwraca widok przedstawiający tekst na stronie głównej.

Metoda About, Contact zwraca widoki, odpowiednio About i Contact, wyświetlające dane o aplikacji i informacje o kontakcie z twórcami.

**Klasa IngredientsController**

Klasa rozszerzająca Controller, odpowiada za obsługę składników. Mająca metody Create, Edit, Details i Delete.

GET/POST Create - obsługa widoku Create. GET zwraca widok tworzenia składników, natomiast metoda z adnotacją POST obsługuje wprowadzenie danych i stworzenie nowego składniku.

Pozostałe metody są analogiczne do Create.

**Klasa RecipesController**

Klasa analogiczna do IngredientsController. Zawiera metody Create, Edit, Details, Delete. Metody odpowiadające za tworzenie, edytowanie, wyświetlanie i usuwanie przepisów. Możliwe tylko na własnych przepisach.

**Klasa TagsController**

Klasa podobna do IngredientsController i RecipesController. Ma podstawowe operacje - Create, Edit, Details, Delete. Tworzenie, wyświetlanie i modyfikacja tagów.

**Folder MniamMniam/Data**

W folderze Data zawarte są migracje oraz kontekst bazy danych.

**Klasa ApplicationDbContext**

Rozszerza klasę IdentityDbContext<ApplicationUser>. W tej klasie zapisane są związki między klasami modeli, a bazą danych. Na podstawie właściwości tej klasy powstaną relacje w bazie danych.

**Folder MniamMniam/Models**

Models zawiera trzy foldery: AccountViewModels, CookBookModels, ManageViewModels.

Zawiera też klasę ApplicationUser.

**Klasa ApplicationUser**

Klasa ta rozszerza klasę IdentityUser. Odpowiada za model użytkownika -

przechowywanie jego przepisów - AvailableRecipes,

ulubionych przepisów - FavouriteRecipes,

czy też ulubionych przepisów, czy też opinii o przepisie - Review

**Folder MniamMniam/Models/AccountViewModels**

Folder zawiera następujące kalsy odpowiadające za konto użytkownika:

ExternalLoginConfimationViewModel - model przechowujący Email

ForgotPasswordViewModel - przechowuje Email

LoginViewModel - przechowuje Login, Password i RememberMe - czy zapamiętać użytkownika

RegisterViewModel - przechowuje Email, Password, oraz potwierdzenia hasła czyli ConfirmPassword

ResetPasswordViewModel - stare hasło, nowe hasło z odpowiednimi wymaganiami oraz potwierdzenie tego hasła

SendCodeViewModel - wybór providera, czy zapamiętać oraz adres powrotu - ReturnUrl

VerifyCodeViewModel

**Folder MniamMniam/Models/CookBookModels**

Zawiera modele odpowiadające tagom, przepisom, składnikom, opiniom

FavouriteRecipe - model ulubionego przepisu, zawiera informacje o twórcy i przepisie

Ingredient - model składniku, zawiera takie dane jak nazwa przepisu i jednostka w jakiej składnik będzie liczony w ilościach.

Recipe - model składniku. Oczywiście zawiera informacje o składniku.

RecipeAvailability

RecipeIngredient - model relacji składniku z przepisem. Zawiera dane o składniku i przepisie.

RecipeTag - model relacji tagu z przepisem. Zawiera dane o tagui przepisie.

Review - model opinii. Zawiera użytkownika wstawiającego opinie, przepis, komentarz i ocenę.

Tag - model tagu. Zawiera dane o tagu, jak nazwa tagu i jego powiązaniu z przepisami.

**Folder MniamMniam/Models/ManageViewModels**

Modele odpowiadające dodawaniu telefonu, modyfikacji hasła czy zarządzaniu loginami.

**Folder MniamMniam/Views**

Istotny model, który zawiera wszystkie widoki. Widoki są podzielone na foldery:

Account - zawiera widoki związane z kontem, jak ForgotPassword, ResetPassword czy też AccessDenied.

Home - zawiera Index, About i Contact

Ingredients - widoki operacji na składnikach, Create, Delete, Edit, Details

Manage - widoki do zarządania.

Recipes - widoki operacji na przepisach, Create, Delete, Edit, Details

Reviews - widoki operacji na opiniach, Create, Delete, Edit, Details

Shared - widoki dzielone przez wszystkie widoki. Jak np. pasek nawigacyjny etc.

Tags - widoki operacji natagach, Create, Delete, Edit, Details

## **Opis zaimplementowanych modułów.**

**W obszarze klienta mobilnego zostały wykonane następujące moduły:**

* Modułkomunikacji z zewnętrznym serwerem
  + Dane przesyłane są poprzez protokół HTTP

Został wykonany jak zostało zaplanowane w dokumentacji Koncepcyjnej

* Moduł odpowiedzialny za poprawną interakcję użytkownika z interfejsem
  + Wyświetla pola wprowadzania danych, przyciski, generalny interfejs.
  + Zarządza odpowiednią kolejnością prezentowanych widoków.

Został wykonany jak zostało zaplanowane w dokumentacji Koncepcyjnej

**W obszarze serwera zostały wykonane następujące moduły:**

* Moduł odpowiedzialny za komunikację z bazą danych
  + Do komunikacji z bazą wykorzystany zostanie entity framework
  + Pozwala na pobieranie i zapisywanie danych do bazy

Został wykonany jak zostało zaplanowane w dokumentacji Specyfikacyjnej. Diagram przedstawiający tabele widoczne w bazie znajduje się w Dokumentacji Specyfikacyjnej,

* Moduł odpowiedzialny za autentykację użytkowników
  + Pozwala rozpoznać użytkownika za pomocą autoryzacji Google.

Został wykonany jak zostało zaplanowane w Koncepcyjnej

## **Możliwe rozszerzenia.**

Aplikacja może być w łatwy sposób poszerzana o nowe funkcje, przez dodanie widoku, modelu i kontrolera.

Proponowane przez nas rozszerzenia:

1. Implementacja porównywania przepisów
2. Umożliwienie zapisu przepisu na dysk w formie pdf
3. Zaimplementowanie modułu który pozwoliłby na zaznaczenie przepisów które chce się wykonać i obliczył liczbę potrzebnych składników.
4. Możliwe dodanie lekkiego tła do widoków
5. Wystylizowanie zdjęć
6. Zdefiniowanie jednostek miary składników
7. Zmiana nazwy pola “Text” w tworzeniu/edycji przepisu na Description