

Politechnika Warszawska

W Y D Z I A Ł E L E K T R Y C Z N Y



Instytut Elektrotechniki Teoretycznej i Systemów Informacyjno-Pomiarowych

Praca dyplomowa inżynierska

na kierunku Informatyka Stosowana
w specjalności Inżynieria Oprogramowania

Aplikacja wspomagająca proces gotowania

Jakub Słabicki

Numer albumu 311471

promotor

dr inż. Zuzanna Krawczyk

WARSZAWA 2024

Streszczenie

Aplikacja wspomagająca proces gotowania

Praca inżynierska skupia się na opracowaniu aplikacji internetowej, która usprawnia zarządzanie domowymi zapasami żywności i pomaga w ich wykorzystywaniu. Aplikacja łączy funkcje menadżera lodówki, skanera produktów oraz rekomendacji połączeń smakowych na podstawie wybranego produktu.

Realizacja projektu obejmowała analizę istniejących rozwiązań, projektowanie systemu i interfejsu użytkownika oraz implementację. Rezultatem jest intuicyjna aplikacja, które nie tylko pomaga w organizacji kuchennych zapasów, ale również pozwala na zredukowanie wyrzucanej żywności poprzez propozycję wykorzystania posiadanych składników.

Słowa kluczowe:

aplikacja internetowa, zarządzanie zapasami żywności, menadżer lodówki, skaner produktów, rekomendacja połączeń smakowych, redukcja marnowania żywności

Application for cooking process improvement

The engineering thesis focuses on the development of a web application that helps with the management of home food supplies and suggests how to effectively use them. The application combines the functions of a fridge manager, product scanner and flavour pairing recommendations based on the selected product.

The project's execution included analysis of existing solutions, system design, user interface design and implementation. The result is an intuitive application that not only assists in the organisation of kitchen supplies but also allows for the reduction of food waste through the suggestion of using ingredients on hand.

Keywords:

Web application, food supply management, fridge manager, product scanner, flavour pairing recommendations, food waste reduction

Spis treści

1. Wstęp.....	6
1.1. Wprowadzenie	6
1.2. Sztuczna inteligencja	6
1.3. Chat-GPT.....	7
1.4. Cel.....	7
1.5. Motywacja	7
1.6. Podział pracy	8
2. Przegląd istniejących rozwiązań	9
2.1. Fridify.....	9
2.2. Fitatu	14
2.3. Blogi kulinarne.....	19
2.4. Porównanie aplikacji	21
3. Specyfikacja wymagań i projekt aplikacji	22
3.1. Wymagania biznesowa.....	22
3.2. Działy słownik pojęć.....	22
3.3. Wymagania funkcjonalne użytkownika.....	23
3.4. Wymagania нефункционалне użytkownika.....	23
3.5. Wymagania нефункционалне systemowe.....	23
3.6. Reguły biznesowe	23
3.7. Diagram przypadków użycia i scenariusze	24
3.8. Projekt Frontend	25
4. Użyte technologie.....	27
4.1. Frontend	27
4.1.1. React.....	27
4.1.2. TypeScript.....	27
4.1.3. React Router	27
4.1.4. Bootstrap i React-Bootstrap.....	27
4.2. Backend	27
4.2.1. Flask.....	27
4.2.2. Flask-JWT-Extended	27
4.2.3. Axios	28
4.2.4. Flask-Migrate.....	28
4.3. Baza Danych	28
4.3.1. SQLite	28

4.3.2.	SQLAlchemy.....	28
4.4.	Dodatkowe narzędzia.....	28
4.4.1.	OpenAI API:	28
5.	Główne funkcjonalności.....	30
5.1.	Ekran główny	30
5.2.	Rejestracja i logowanie.....	30
5.3.	Panel główny użytkownika	31
5.4.	Lodówka użytkownika	33
5.5.	Dodawanie produktów z listy	35
5.6.	Skanowanie produktów ze zdjęcia	36
5.7.	Połączenia smakowe	38
5.8.	Lista przepisów	39
5.9.	Lista polubionych przepisów	40
5.10.	Ustawienia użytkownika.....	41
6.	Implementacja	42
6.1.	Środowisko programistyczne	42
6.2.	Modele danych.....	42
6.2.1.	User	42
6.2.2.	FoodItem	42
6.2.3.	Recipe	43
6.3.	Implementacja metod CRUD dla użytkownika	44
6.3.1.	Create	44
6.3.2.	Read.....	45
6.3.3.	Update.....	46
6.3.4.	Delete	47
6.4.	Połączenie z OpenAI API.....	47
7.	Możliwości rozwoju.....	49
7.1.	Liczenie kalorii	49
7.2.	Rozszerzanie bazy przepisów przez użytkowników.....	49
7.3.	Planer jadłospisu i zakupów	49
8.	Podsumowanie.....	50
9.	Bibliografia	51
10.	Spis ilustracji.....	52

1. Wstęp

1.1. Wprowadzenie

W dzisiejszych czasach, kiedy już od wielu lat panuje era cyfrowej transformacji i granice pomiędzy rzeczywistością fizyczną a wirtualną stają się coraz bardziej płynne, informatyka na dobre przeniknęła już do niemalże każdego aspektu życia człowieka. Współczesne społeczeństwo jest kompletnie zdominowane przez technologię, bez której życie byłoby niezmiernie ciężkie. Jednak oferuje ona niespotykane dotąd możliwości, które wpływają na sposób komunikowania, uczenia się oraz pracy, jak i również odpoczynku czy rozrywki. Szczególnie branża aplikacji mobilnych oraz webowych przeżywa swój rozkwit oferując rozwiązanie dostępne pod ręką dla każdego, gdyż niewiele jest już osób, które nie posiadają przy sobie jakiegoś urządzenia niemalże cały czas. Taki stały dostęp do tylu możliwości ułatwia codzienne zadania, pozwala na samodzielną edukację w dowolnie wybrany sposób. Dodatkowo umożliwia monitorowanie swojego zdrowia oraz zaspokojenie potrzeb rozrywki wpływając tym na dobre samopoczucie użytkowników. Znaczącą rolę odgrywają również aplikacje gastronomiczne, które służą nie tylko do zamawiania jedzenia z dostawą do domu czy rezerwacji stolika w restauracjach, ale również pozwalają na zarządzanie domowymi zapasami żywności, planowanie posiłków. Mogą być również codzienną inspiracją w kuchni, umożliwiając rozwój swoich umiejętności kulinarnych zarówno osobom bardziej zaawansowanym jak i osobom, dla których gotowanie jest zupełnie czymś obcym. Pozwalają na czerpanie z doświadczeń innych ludzi pozwalając tym samym na zwiększenie jakości własnego życia.

1.2. Sztuczna inteligencja

Obok dobrze już rozwiniętego i tak chętnie użytkowanego przez społeczeństwo rynku aplikacji mobilnych oraz webowych, w ostatnich latach znacznie rozwinęła się sztuczna inteligencja (AI), która ogromnie rozszerzyła dotychczasowe możliwości aplikacji, czy systemów informatycznych. Jej korzenie sięgają lat 50-tych XX wieku, kiedy to naukowcy tacy jak Alan Turing zaczęli badać możliwości maszyn do naśladowania ludzkiego procesu myślowego. Turing, poprzez swój słynny test, zaproponował metodę oceny zdolności maszyny do wykazywania inteligentnego zachowania, nie do odróżnienia od zachowania człowieka. Od tego czasu sztuczna inteligencja przeszła długą drogę, poprzez ograniczenia technologiczne, które niestety spowalniały jej postęp, aż do obecnych czasów, kiedy jest ona wszechobecna, od prostych aplikacji po złożone systemy zarządzające danymi na masową skalę. Potencjał AI jest ogromny. Obejmuje dziedziny takie jak medycyna, w której algorytmy uczenia maszynowego potrafią

przewidzieć choroby na podstawie danych pacjentów, transport, gdzie powstają coraz bardziej udoskonalane samochody autonomiczne, czy edukacja, w której spersonalizowane metody uczenia się pozwalają na indywidualne przystosowanie do potrzeb osób uczących się. AI ma również ogromne znaczenie przy analizie danych, przetwarzaniu języka naturalnego, rozpoznawaniu wzorców i obrazów oraz w robotyce, a całkiem niedawno dobrze rozwinęły się jej możliwości w generowaniu sztucznej twórczości: obrazów, krótkich form poetyckich, utworów muzycznych, a nawet wideo.

1.3. Chat-GPT

Szczególnym osiągnięciem w dziedzinie sztucznej inteligencji w ostatnich latach jest rozwój generatywnych modeli przetwarzania języka naturalnego, takich jak ChatGPT. Został on opracowany przez OpenAI i jest model wariantu GPT – Generative Pretrained Transformer, który wykorzystuje głębokie uczenie do generowania tekstów o zdanej tematyce. Jest on trenowany na szerokim zestawie danych tekstowych, co pozwala mu na naśladowanie ludzkich wzorców komunikacji. Najnowsza wersja – GPT-4, inaczej jak jego poprzednicy, jest modelem multimodalnym, co oznacza, że może przyjmować zarówno obrazy, jak i tekst jako dane wejściowe. Posiada on możliwości rozpoznawania przedmiotów na obrazie, czy rozczytywania tekstu z obrazu, co znacznie rozszerza jego możliwości.

1.4. Cel

Dzięki połączeniu aplikacji wraz ze sztuczną inteligencją, użytkownicy zyskują nowe możliwości, które mogą być spersonalizowane pod ich własne potrzeby. W tej pracy chciałbym przedstawić możliwości sztucznej inteligencji w analizowaniu obrazów jak i również przedstawić możliwość wykorzystania jej w kuchni, dzięki połączeniu z aplikacją webową. Tematem mojej pracy jest aplikacja, która umożliwi użytkownikom zarządzanie posiadaną żywnością, jak i również proponuje, na podstawie rozpoznanych ze zdjęcia składników oraz tych posiadanych, możliwe do przygotowania potrawy. Pomaga również w dobieraniu połączeń smakowych na podstawie wybranego produktu w oparciu o sztuczną inteligencję.

1.5. Motywacja

Motywacją do podjęcia się takiego tematu pracy inżynierskiej jest związana z moimi osobistymi doświadczeniami i napotkanym problemem kreatywności w kuchni. Często zdarza się, że brak pomysłu na posiłek prowadzi do konieczności improwizacji. Niekiedy jednak, nawet jeśli już nadejdzie inspiracja co do chęci przygotowania jakiegoś jedzenia, często napotykam się z problemem braku potrzebnych składników na tę

potrawę, to natomiast wiąże się z koniecznością wybrania się na zakupy. Powyższa aplikacja, będąca przedmiotem tej pracy, ma za zadanie wspomóc potencjalnych użytkowników w przekształceniu dostępnych składników w pełnowartościowy, kreatywny posiłek. Dodatkową motywacją do stworzenia tej aplikacji jest powszechny problem marnowania żywności. Statystyki dotyczące wyrzucanej żywności w Polsce są niepokojące. Wskazują, że rocznie wyrzuca się około 4,8 miliona ton jedzenia, z czego za 60% odpowiadają gospodarstwa domowe. Przyczyną marnowania tak ogromnej ilości jedzenia może być niewykorzystywanie resztek składników, które pozostają po przygotowaniu danej potrawy. Często trafiają one do lodówki, gdzie z braku pomysłu na ich wykorzystanie zapomina się o nich. Ostatecznie kończą one swoją drogę w śmietniku, gdyż upływa ich termin przydatności. W przedstawionej pracy inżynierskiej rozważany jest sposób, w jaki technologia może przyczynić się do zmniejszenia marnotrawstwa poprzez efektywniejsze zarządzanie zasobami żywności w domach, co może w znacznym stopniu wpłynąć pozytywnie na środowisko naturalne oraz ekonomię domowego budżetu.

1.6. Podział pracy

Praca rozpoczyna się od wstępu, wprowadzającego w temat problemu, którego dotyczy. Dalsza część pracy skupia się na szczegółowym omówieniu koncepcji, przeglądu istniejących rozwiązań, stanowiących bazę porównawczą oraz inspirację do tworzenia nowych funkcjonalności. Ponadto zawiera ona specyfikację wymagań, szczegółowy opis użytych technologii, prezentacji procesu projektowania oraz implementacji systemu. Zakończenie pracy stanowi podsumowanie osiągniętych wyników, refleksja nad tworzeniem aplikacji oraz perspektywy jej rozwoju.

2. Przegląd istniejących rozwiązań

W ramach analizy istniejących już na rynku rozwiązań w zadanej tematyce został przeprowadzony przegląd aplikacji oraz stron internetowych związanych z zarządzaniem żywnością i gotowaniem. Rynek aplikacji mobilnych oferuje szereg narzędzi, które w różnym stopniu odpowiadają na potrzeby użytkowników w zakresie organizacji zapasów żywności czy czerpania inspiracji kulinarnych. Niemniej jednak, większość z nich koncentruje się na pojedynczych aspektach tej tematyki, nie oferując kompleksowego podejścia, które połączyłoby te funkcjonalności w jedną całość.

2.1. Fridify

Pierwsza aplikacja, która została wzięta pod lupę to aplikacja Fridify. Służy ona do kontroli i przypominania o końcu ważności produktów, które użytkownik może dodać do swojej wirtualnej lodówki, kosmetyczki czy apteczki. Dodatkowo aplikacja umożliwia użytkownikowi ustawienie przypomnień o otwartych produktach, dzięki czemu nie musi on pamiętać kiedy otworzył dany produkt. Poniżej znajdują się ekrany przedstawiające główne funkcjonalności aplikacji.

11:30 VoD LTE+ LTE1

← Fridify
Dodawanie schowka

Nazwa schowka

Lodówka

Wybierz logo

Refrigerator, First Aid Kit, Pill

DODAJ

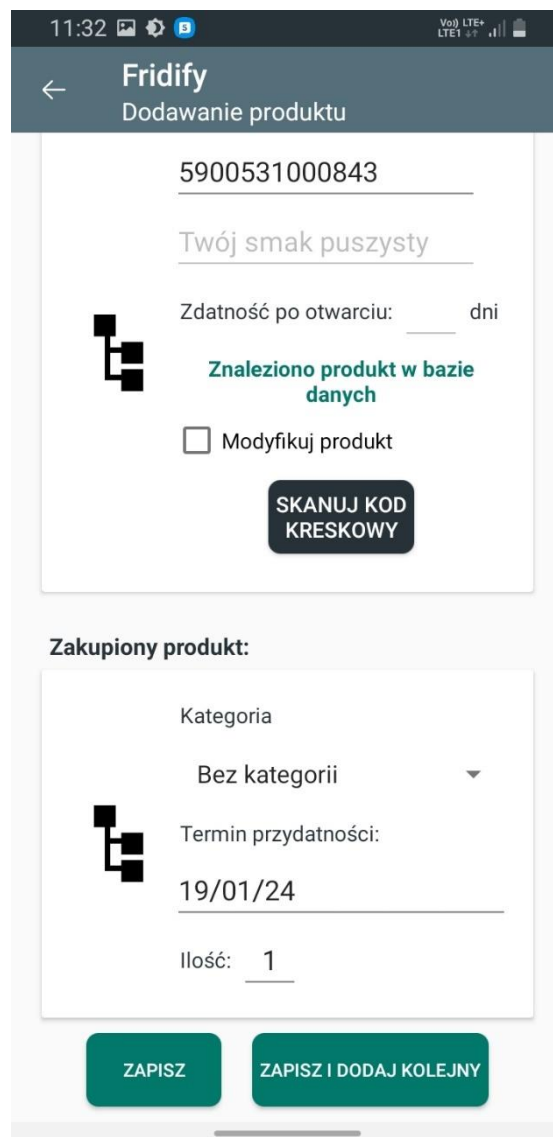
Rysunek 1 Ekran dodawania schowka Fridify

Na tym ekranie przedstawiony jest interfejs dodawania nowego schowka w aplikacji. Użytkownik może nadać własną nazwę swojemu schowkowi oraz wybrać jedno z trzech logo: lodówkę, kosmetyczkę lub apteczkę.



Rysunek 2 Ekran Panel Głównego Fridify

Ten ekran ukazuje panel główny użytkownika aplikacji, gdzie może on zobaczyć wszystkie swoje schowki. Widoczny jest wcześniej dodany schowek „Lodówka” wraz z informacją o liczbie przechowywanych produktów.



11:32 VoD LTE+ LTE1

← **Fridify**
Dodawanie produktu

5900531000843

Twój smak puszysty

Zdatność po otwarciu: _____ dni

Znaleziono produkt w bazie danych

☐ Modyfikuj produkt

SKANUJ KOD KRESKOWY

Zakupiony produkt:

Kategoria

Bez kategorii ▼

Termin przydatności:

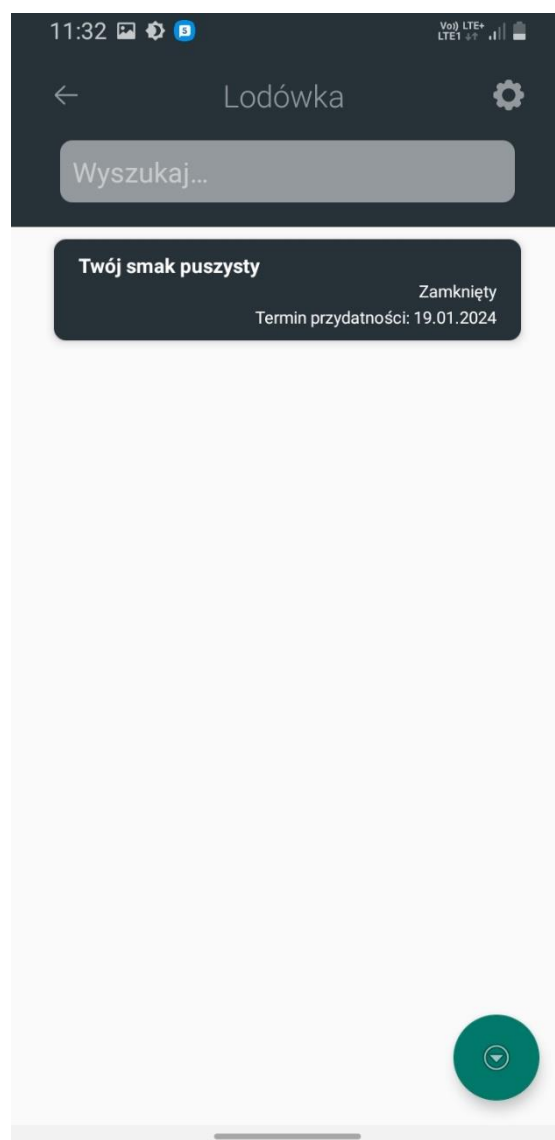
19/01/24

Ilość: 1

ZAPISZ **ZAPISZ I DODAJ KOLEJNY**

Rysunek 3 Ekran dodawania produktu Fridify

Ten widok aplikacji Fridify prezentuje proces dodawania produktu do schowka. Użytkownik może zeskanować kod kreskowy produktu, dzięki czemu automatycznie uzupełniane są jego dane lub może również wprowadzić je ręcznie. Możliwa jest również modyfikacja produktu, wybór kategorii, oraz określenie daty przydatności i ilości.



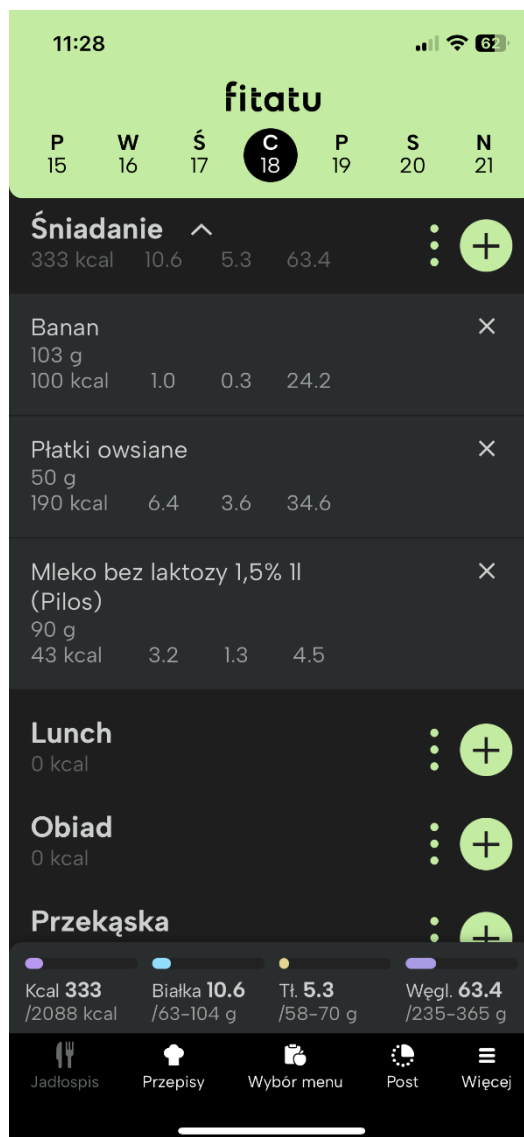
Rysunek 4 Ekran schowka "Lodówka" Fridify

Na tym ekranie został przedstawiony widok schowka „Lodówka”. Użytkownika ma możliwość przeszukiwania jego zawartości oraz dodawania do niego nowych produktów. Każdy produkt jest opatrzony nazwą oraz datą przydatności do spożycia co pomaga w monitorowaniu świeżości oraz unikaniu marnowania żywności.

Jak widać na powyżej przedstawionych ekranach aplikacja zdecydowanie może przydać się w organizacji w kuchni pozwalając na zarządzanie własnymi zapasami, jednak na tym kończą się jej funkcje. Nie zapewnia ona dostępu do bazy przepisów, ani nie sugeruje przepisów kulinarnych, które można by wykonać z posiadanych produktów. Ponadto nie zapewnia też możliwości sprawdzenia połączeń smakowych do produktów. Dodatkowo jej głównym ograniczeniem jest możliwość korzystania z niej jedynie na urządzeniach z systemem Android.

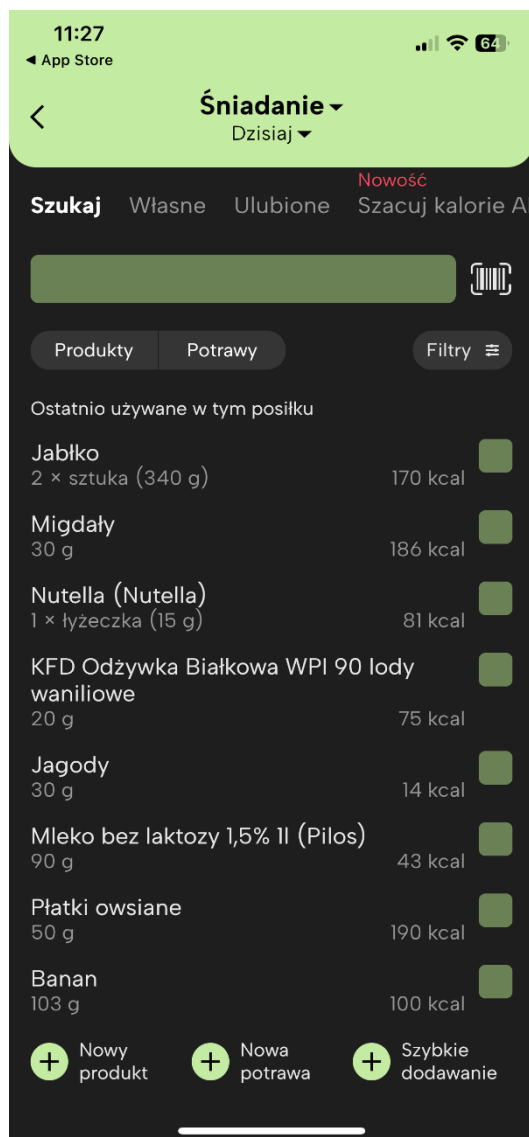
2.2. Fitatu

Kolejną aplikacją, która została poddana analizie jest aplikacja Fitatu. Jej główną funkcjonalnością jest umożliwienie użytkownikowi liczenia spożywanych w ciągu dnia kalorii. Pozwala ona wybierać produkty z bazy danych i dodawać je do swojego dziennego jadłospisu, sumując przy tym ilość spożytych kalorii oraz makroelementów. Jej baza składa się z wielu codziennie używanych produktów lub dań, dzięki czemu dodawanie poszczególnych składników lub całej potrawy nie zajmuje użytkownikowi wiele czasu. Jedyne co musi podać to wagę produktu lub porcji. Aplikacja dodatkowo jest wyposażona w bazę z przepisami wraz z ich wartościami odżywczymi, którą użytkownik może przeglądać poszukując inspiracji. Natomiast jeśli użytkownik nie znajdzie przepisu lub składnika w dostępnej bazie, może samemu wprowadzić dane na temat produktu lub przepisu i dodać go do bazy danych co pozwala na jej rozwój i możliwość dzielenia się swoimi inspiracjami, jak i również inspirowanie się pomysłami innych. Na poniższych zdjęciach zostały przedstawione poszczególne funkcjonalności aplikacji.



Rysunek 5 Ekran widoku głównego użytkownika Fitatu

Ekran przedstawia widok główny użytkownika, gdzie może on dodawać spożyte w ciągu dnia posiłki lub produkty. Pozwala to na śledzenie codziennego jadłospisu, gdyż aplikacja automatycznie kalkuluje sumę spożytych kalorii oraz makroskładników.



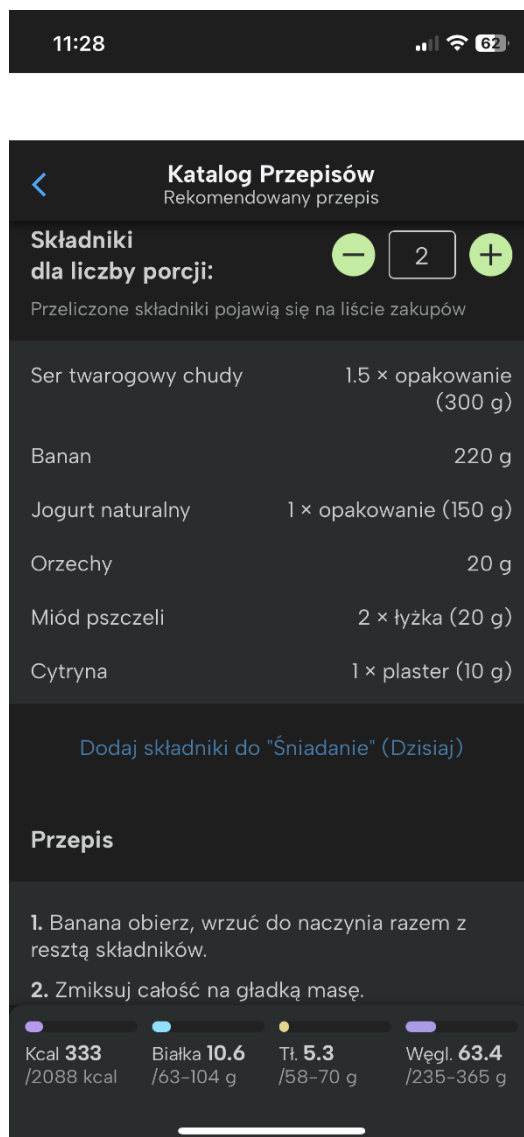
Rysunek 6 Ekran listy składników Fitatu

Na tym ekranie widać listę składników posortowanych od tych ostatnio używanych. Użytkownik może wyszukać produkt, który spożył, a następnie po podaniu ilości dodać go do swojego jadłospisu. Ponadto, jeśli danego produktu lub potrawy nie ma na liście, użytkownik może go dodać podając jego nazwę, ilość kalorii oraz makroskładników.



Rysunek 7 Ekran listy przepisów Fitatu

W tym widoku użytkownik może przeglądać potrawy w poszukiwaniu inspiracji. Każdy przepis jest wyraźnie opisany nazwą, kalorycznością oraz czasem potrzebnym do jego przygotowania. Ponadto, podana jest również ocena użytkowników dotycząca danego przepisu, co pozwala użytkownikowi oszacować jego jakość.



Rysunek 8 Ekran szczegółów przepisu Fitatu

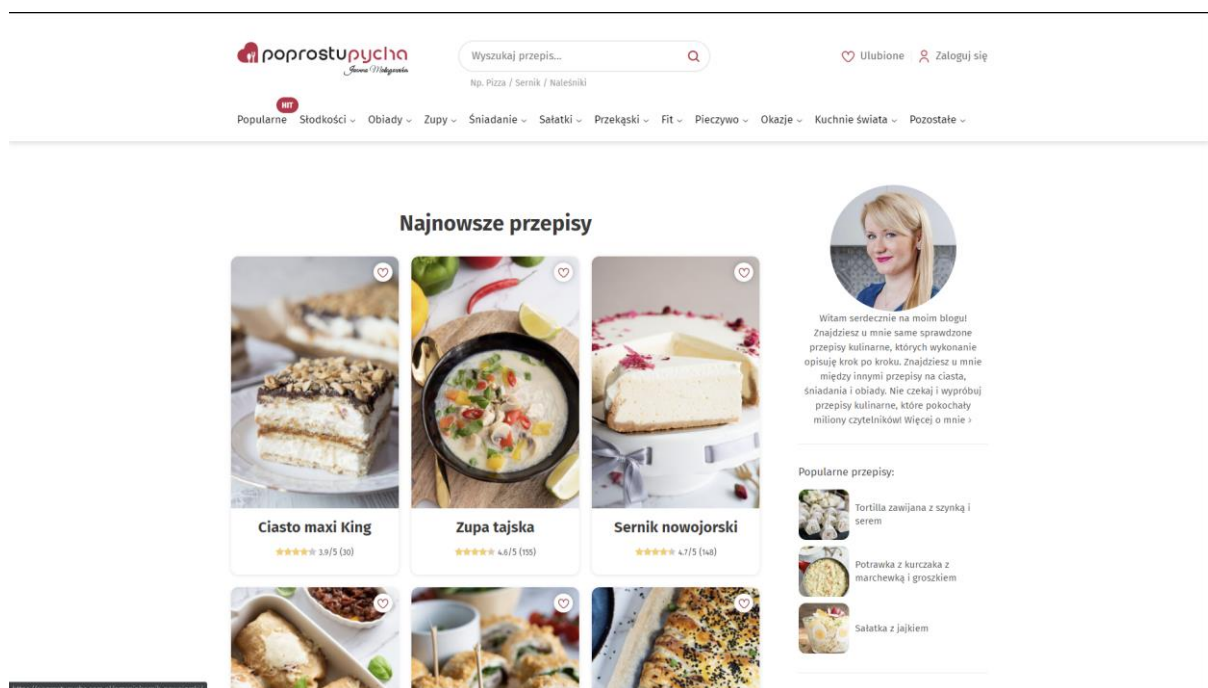
Po wybraniu przepisu użytkownik otrzymuje listę składników z przeliczeniem na wybraną liczbę porcji, co ułatwia planowanie i przygotowywanie posiłków. Dodatkowo zawarte są tam szczegółowe instrukcje dotyczące przepisu.

Aplikacja Fitatu jest bardzo rozbudowana i umożliwia użytkownikowi na sprawne monitorowanie spożywanych kalorii oraz makroskładników dostarczając przy tym różnych inspiracji kulinarnych pochodzących od innych użytkowników. Jej ograniczeniem jest brak możliwości zarządzania posiadanymi składnikami, a co za tym idzie brak sugerowania przepisów, która można by wykonać z posiadanymi składnikami. Nie zapewnia ona również informacji o połączeniach smakowych produktów.

2.3. Blogi kulinarne

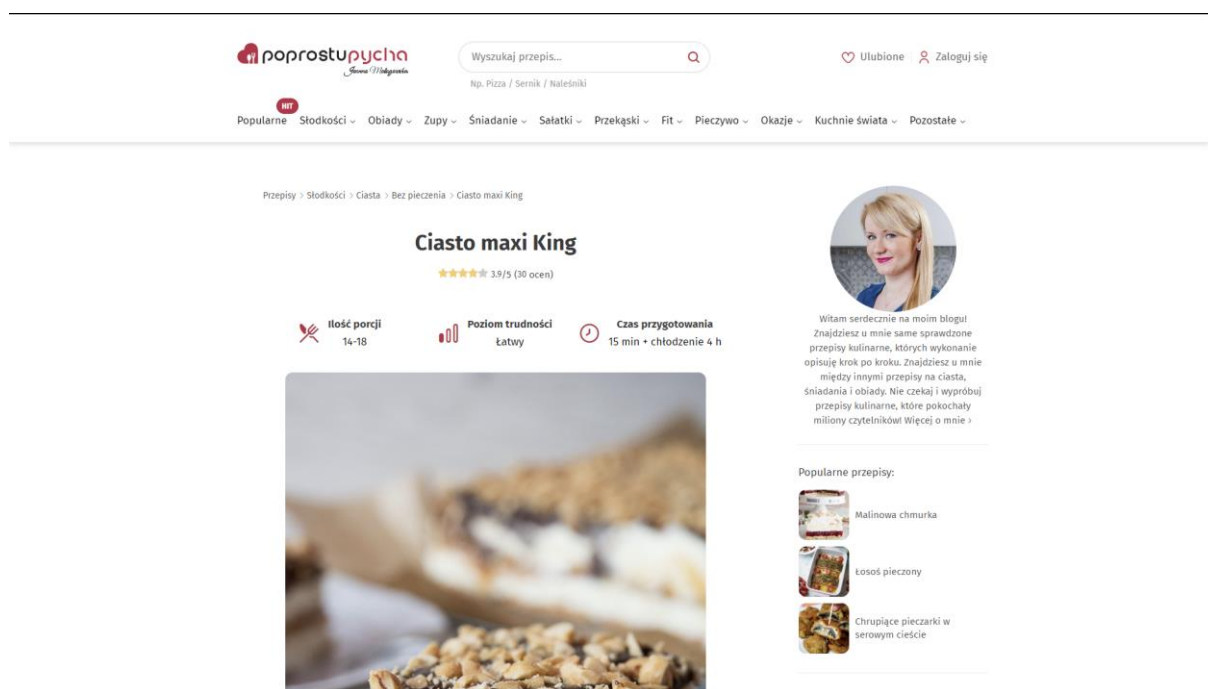
Istnieje wiele blogów oraz aplikacji, które oferują przepisy kulinarne tworzone przez ich twórców lub użytkowników. Ich głównym celem jest inspiracja oraz dostarczanie pomysłów kulinarnych dla osób, które tego potrzebują zapewniając przy tym szczegółowy opis instrukcji oraz potrzebnych składników przy przygotowywaniu danego posiłku.

Jako przykład bloga kulinarnego został opisany blog PoProstuPycha, który posiada ogromną bazę przepisów i jest prowadzony przez Joannę Małagowską. Użytkownicy mają możliwość założenia własnego konta na blogu, które między innymi pozwala na dodawanie przepisów do swojej listy ulubionych. Użytkownik ma możliwość wyszukania przepisu wpisując jego nazwę lub może skorzystać z zakładek przygotowanych na stronie głównej wybierając interesującą go kategorię. Poniżej został przedstawiony wygląd bloga.



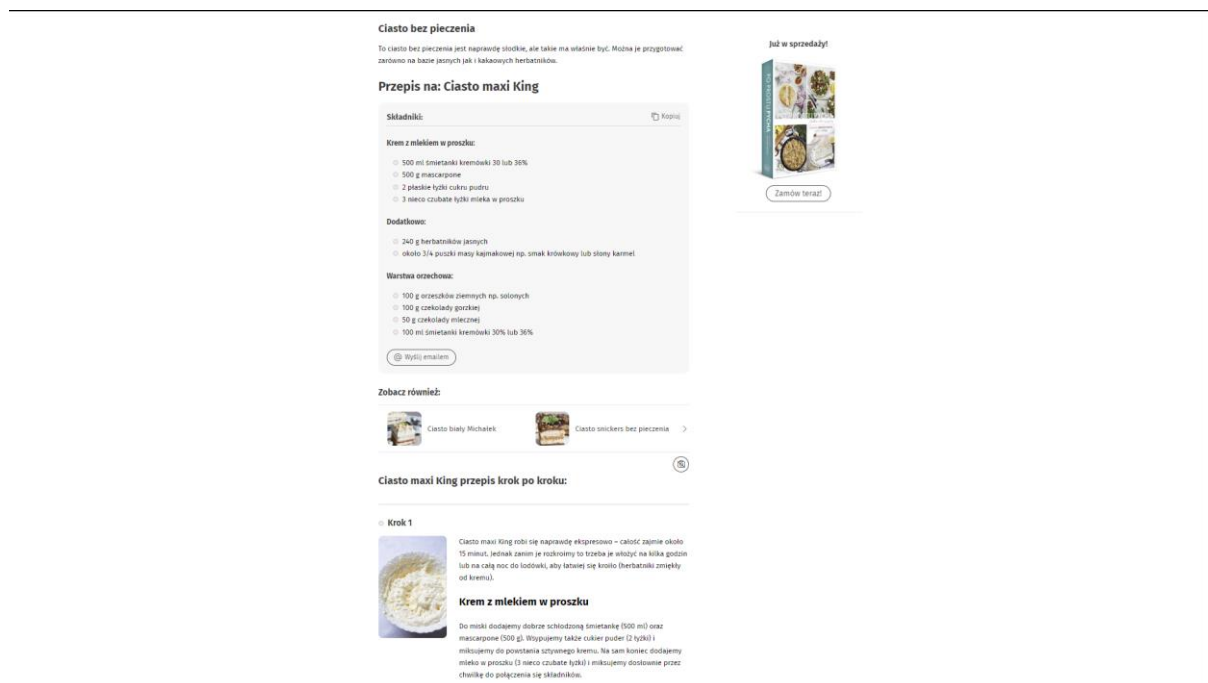
Rysunek 9 Strona główna bloga kulinarnego

Na zdjęciu widzimy stronę główną. Użytkownik poszukujący inspiracji może skorzystać z zakładki popularne lub przeszukać stworzoną bazę przepisów. Po wybraniu przepisu ukazuje się jego strona, zawierająca informacje na jego temat.



Rysunek 10 Strona przepisu bloga kuliarnego

Na stronie przepisu użytkownik jest informowany o ilości porcji, poziomie trudności i średnim czasie przygotowanie. Dodatkowo znajdują się tu również potrzebne składniki oraz dokładne instrukcje postępowania.



Rysunek 11 Instrukcje przepisu bloga kulinarnego

Blogi kulinarne mogą być dobrą inspiracją podczas przygody w kuchni, niemniej jednak jest to jedynie zbiór przepisów i nie oferują one personalizacji pod kątem aktualnie posiadanych składników.

2.4. Porównanie aplikacji

Podsumowując przeanalizowane aplikacje oraz blog można zauważyć, że każde z rozwiązań posiada swoje unikalne funkcjonalności i skupia się na określonych aspektach zarządzania żywnością lub gotowania. Jednak nie istnieje aplikacja, które by te najważniejsze funkcjonalności zintegrowała w jedną całość co pozwoliłoby na znacznie przyjemniejsze dla użytkowników korzystanie z tych rozwiązań. Zatem moja aplikacja będąca tematem tej pracy ma na celu zintegrowanie istniejących funkcjonalności zarządzania zapasami żywności oraz czerpania inspiracji z istniejących przepisów w oparciu o posiadane produkty. Ponadto jej celem jest również dostarczenie dodatkowych funkcjonalności takich jak sugerowanie połączeń smakowych oraz rozpoznawanie produktów na zdjęciach, które nie zostały jeszcze przedstawione na rynku aplikacji mobilnych czy internetowych.

3. Specyfikacja wymagań i projekt aplikacji

W tym rozdziale przedstawione zostały wymagania dotyczące aplikacji służącej do zarządzania zawartością lodówki i proponowania połączeń smakowych oraz przepisów na podstawie zdjęcia lub listy posiadanych produktów. Wymagania zostały podzielone na biznesowe, funkcjonalne użytkownika, niefunkcjonalne użytkownika oraz systemowe.

3.1. Wymagania biznesowa

Aplikacja ma za zadanie pomóc użytkownikom w efektywny sposób zarządzać zawartością ich lodówek, co bezpośrednio ma wpływ na zmniejszenie marnotrawienia żywności. Ponadto ma ona proponować, na podstawie zeskanowanego zdjęcia oraz składników znajdujących się w lodówce użytkownika, możliwe do wykonania z tych składników przepisy. Aplikacja powinna umożliwiać użytkownikowi dostęp do bazy danych z przepisami oraz proponować dobrane połączenia smakowe na podstawie wybranego z listy produktu. Celem projektu jest stworzenie aplikacji webowej z intuicyjnym interfejsem, która umożliwi użytkownikowi łatwy dostęp do jej funkcjonalności, ponadto umożliwi dodanie rozpoznanych ze zdjęcia przedmiotów do wirtualnej lodówki użytkownika oraz dodawanie wybranych przepisów do ulubionych. Dodatkowo powinien mieć również możliwość założenia własnego konta, edycji danych konta oraz usunięcia go. Językiem aplikacji jest język angielski.

3.2. Dziedzinowy słownik pojęć

- **Użytkownik** – osoba korzystająca z aplikacji
- **Produkt** – pojedynczy składnik spożywczy, który może być dodany do lodówki użytkownika lub na podstawie którego proponowane są pasujące do niego połączenia smakowe, posiada nazwę oraz średnią liczbę dni do zepsucia
- **Lista produktów** – lista wszystkich produktów z bazy danych
- **Przepis** – zestawienie produktów oraz instrukcji do przygotowania potrawy, posiada nazwę, zdjęcie, czas przygotowywania oraz trudność.
- **Lodówka użytkownika** – wirtualna reprezentacja rzeczywistego zbioru składników użytkownika
- **Połączenia smakowe** – zbiór produktów, które dobrze współgrają smakowo z innym produktem
- **Lista przepisów** – zbiór wszystkich przepisów z bazy danych
- **Lista ulubionych przepisów** – zbiór przepisów zapisanych przez użytkownika

3.3. Wymagania funkcjonalne użytkownika

- Użytkownik ma możliwość rejestracji za pomocą adresu e-mail, nazwy użytkownika i hasła oraz logowania się za pomocą adresu e-mail i hasła.
- Użytkownika może dodawać oraz usuwać produkty z listy produktów do swojej lodówki.
- Użytkownik może wyszukiwać za pomocą nazwy produkty w swojej lodówce oraz sortować je ze względu na nazwę, datę dodania, ilość dni do zepsucia oraz ilość posiadanych produktów w lodówce.
- Użytkownik może wyszukiwać produkty z listy produktów za pomocą nazwy oraz sortować je ze względu na nazwę, ilość dni do zepsucia oraz ilość posiadanych produktów w lodówce.
- Aplikacja umożliwia użytkownikowi przeglądanie listy przepisów, wyszukiwanie po nazwie oraz sortowanie po nazwie, trudności, ilości porcji, oraz czasie przygotowania.
- Użytkownik ma możliwość przesłania zdjęcia z produktami, na którego podstawie aplikacja wyświetla listę rozpoznanych produktów.
- Użytkownik ma możliwość wyświetlenia listy możliwych do wykonania przepisów ze względu na posiadane i rozpoznane ze zdjęcia składniki.
- Aplikacja wyświetla, na podstawie wybranego produktu, połączenia smakowe.
- Zalogowany użytkownik może zmienić swoje hasło, e-maila oraz nazwę użytkownika lub usunąć swoje konto.

3.4. Wymagania niefunkcjonalne użytkownika

- Aplikacja ma być responsywna i dostępna na urządzeniach mobilnych i desktopowych
- Powinna posiadać intuicyjny i prosty interfejs użytkownika
- Przechowywane hasła użytkowników powinny być zabezpieczone

3.5. Wymagania niefunkcjonalne systemowe

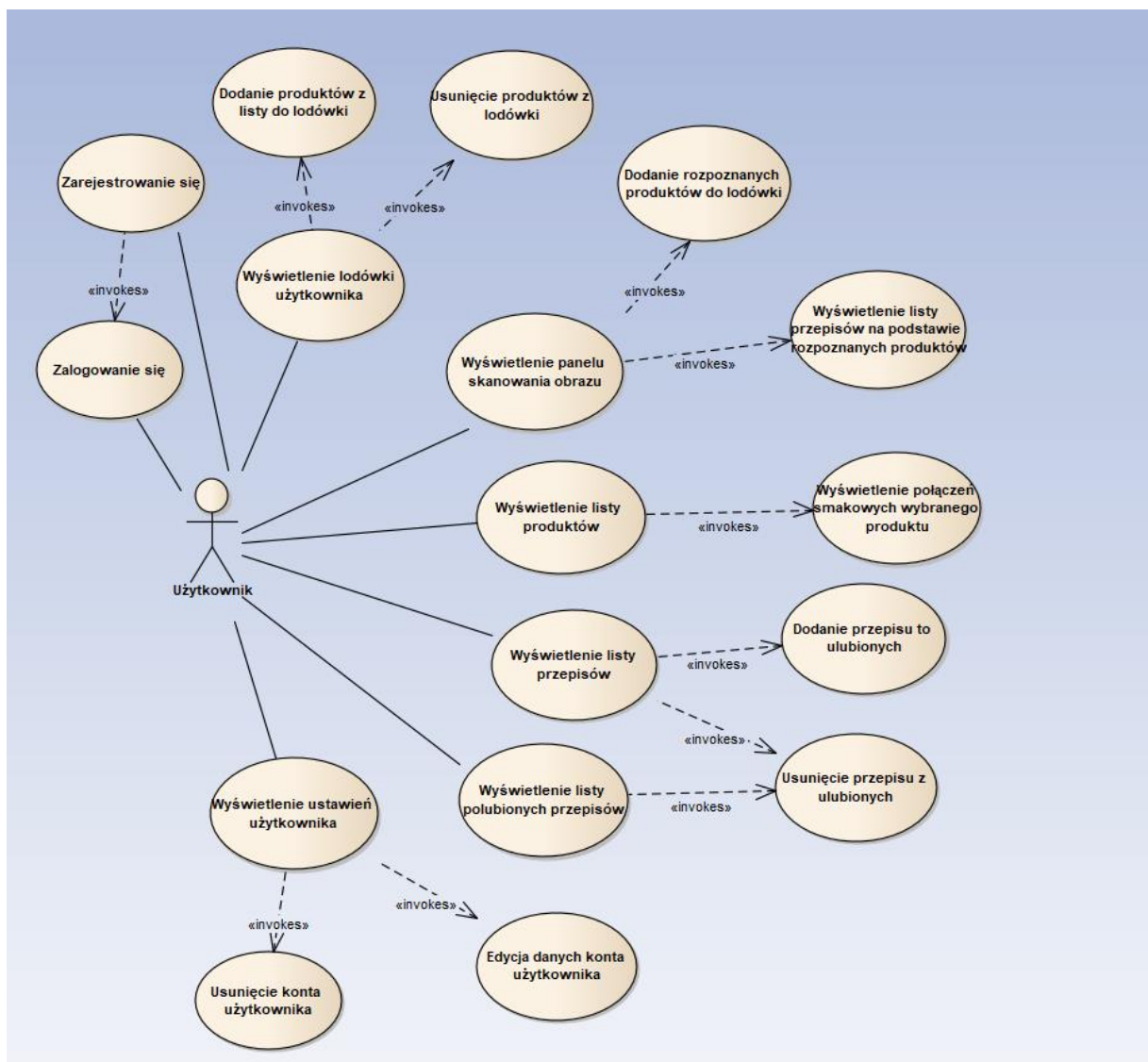
- Aplikacja powinna mieć możliwość integracji z zewnętrznymi API do uzyskiwania połączeń smakowych do produktów

3.6. Reguły biznesowe

- Użytkownik niezalogowany nie może wywołać funkcji aplikacji zwracających danych z systemu

- Osoba niezalogowana w chwili próby dostania się do widoku aplikacji innego niż logowanie, rejestracja lub ekran główny powinna zostać przekierowana do strony głównej.
- Każdy zalogowany użytkownik otrzymuje dostęp do funkcjonalności aplikacji oraz listy przepisów
- Przy usuwaniu konta lodówka użytkownika oraz lista polubionych przepisów również są usuwane

3.7. Diagram przypadków użycia i scenariusze



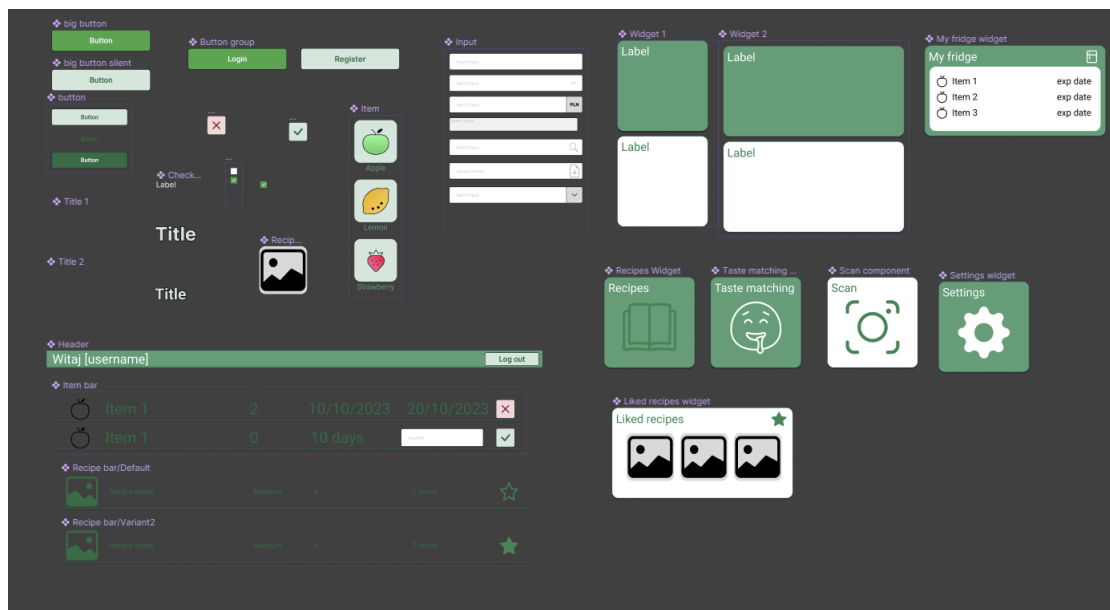
Rysunek 12 Diagram przypadków użycia

Diagram przypadków użycia przedstawia główne funkcjonalności projektowanej aplikacji. Poszczególne przypadki odpowiadają pojedynczym funkcjonalnościom. Głównym aktorem aplikacji jest użytkownik, który może zarządzać swoją lodówką oraz polubionymi przepisami. Może wchodzić w interakcje z aplikacją dodając lub usuwając

produkty do swojej lodówki. Ponadto może wyświetlić listę połączeń smakowych do wybranego produktu oraz zarządzać rozpoznanymi na obrazie produktami.

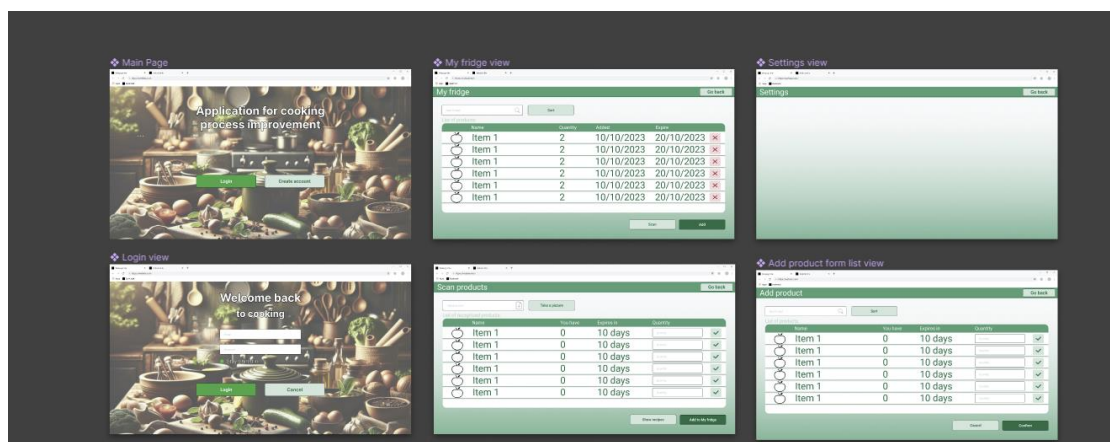
3.8. Projekt Frontend

Do stworzenia projektu interfejsu użytkownika została wykorzystana Figma. Jest to współczesne narzędzie wykorzystywane właśnie do takich celów. Oferuje szeroki zakres funkcji takich jak prototypowanie, dynamiczne komponenty oraz obsługę bibliotek UI. W moim projekcie została wykorzystana biblioteka Bootstrap.

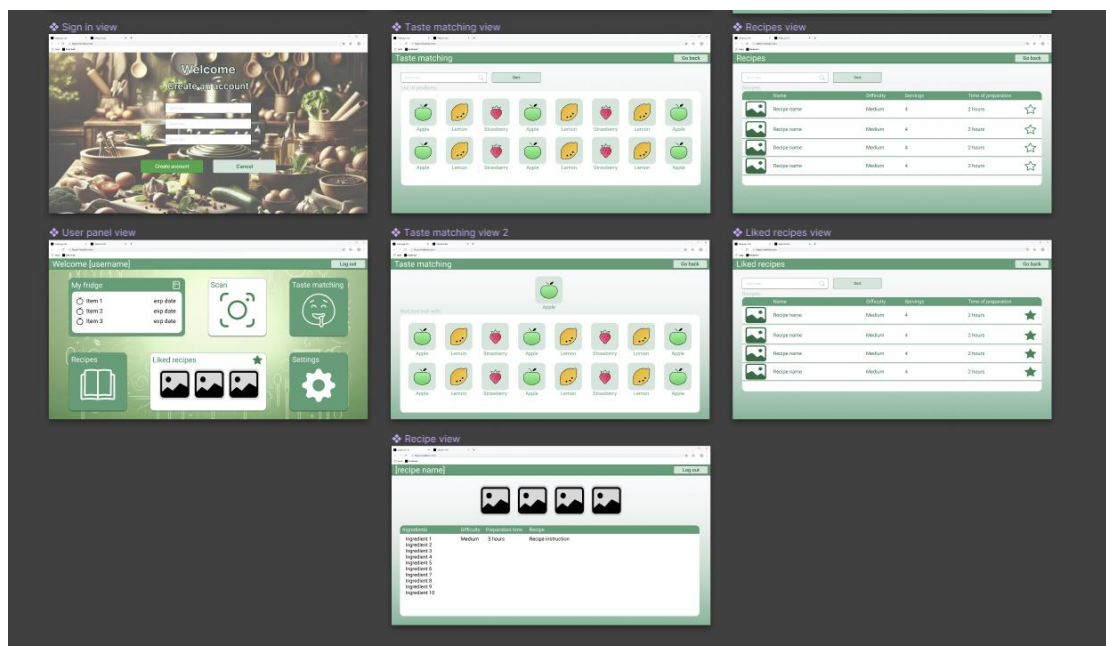


Rysunek 13 Projekt komponentów w Figmie

Na rysunku widać komponenty, które udało mi się stworzyć podczas prototypowania interfejsu użytkownika. Dzięki temu mogłem wykorzystać te same komponenty w różnych miejscach interfejsu, co zapewnia spójność w jego wyglądzie.



Rysunek 14 Projekt ekranów w Figmie (1)



Rysunek 15 Projekt ekranów w Figmie (2)

Tworząc projekt interfejsu komponowane były kolejne widoki przyszłej aplikacji. Znacznie uprościło to późniejszą implementację, gdyż można było wykorzystać zaprojektowane już wcześniej komponenty, aby móc je wykorzystywać następnie w całej aplikacji. Pozwoliło to również na zobrazowanie jej funkcjonalności co także było pomocne w późniejszej implementacji.

4. Użyte technologie

4.1. Frontend

4.1.1. React

Jest to biblioteka JavaScript stworzona przez Facebooka, dzięki której można tworzyć interaktywne interfejsy użytkownika w sposób zorganizowany i wydajny. Używając konceptu komponentów, Ract pozwala na budowę dużych aplikacji, mających możliwość zmiany danych bez przeładowania strony. Wykorzystanie komponentów pozwala na tworzenie izolowanych i ponownie wykorzystywanych bloków kodu. Moja aplikacja wykorzystuje bibliotekę React do stworzenia dynamicznego interfejsu, zapewniającego płynne i intuicyjne doświadczenia użytkownikowi.

4.1.2. TypeScript

Jest to nadzbiór JavaScriptu dodający typowanie statyczne do języka. Użycie TypeScriptu w projekcie aplikacji pozwoliło na wczesne wykrywanie błędów i zapewnienie większej spójności kodu, co w przypadku bardziej złożonych aplikacji jest kluczowe.

4.1.3. React Router

Jest to standardowa biblioteka Reacta, służy ona do routing w jego ekosystemie. Pozwala to na implementację nawigacji między widokami aplikacji SPA – Single Page Application. W tworzonej aplikacji React Router odpowiada za przejścia między różnymi ekranami, jak panel użytkownika, lista przepisów, czy ustawienia.

4.1.4. Bootstrap i React-Bootstrap

Jest to zestaw narzędzi służących do stylizacji interfejsów użytkownika. Pozwala on na szybkie i spójne projektowanie responsywnych interfejsów. W opisywanej aplikacji został wykorzystany w celu stworzenia estetycznego i funkcjonalnego layoutu, który może dostosować się do różnych urządzeń i rozdzielczości ekranu.

4.2. Backend

4.2.1. Flask

To lekki elastyczny Framework webowy napisany w języku Python. Zapewnia on prostotę i szybkość tworzenia aplikacji internetowych. W mojej aplikacji służy on jako podstawa dla REST API – obsługuje zapytania od klienta, zarządza danymi użytkownika oraz komunikuje się z bazą danych oraz zewnętrznymi API.

4.2.2. Flask-JWT-Extended

Jest to rozszerzenie Flask umożliwiające dość prostą integrację zabezpieczeń opartych na tokenach JWT – JSON Web Tokens. Są one kluczowe dla uwierzytelniania oraz autoryzacji w aplikacjach internetowych. Aplikacja wykorzystuje go do zapewnienia bezpieczeństwa i ochrony danych użytkowników oraz do kontroli dostępu do jej zasobów.

4.2.3. Axios

Axios jest popularną biblioteką JavaScriptu, która używana jest do wykonywania żądań HTTP. Zapewnia ona nieskomplikowany i czysty interfejs do komunikacji z API. W opisywanej aplikacji wykorzystywany jest do asynchronicznego łączenia się z backendem, umożliwiając przy tym obsługę żądań i odpowiedzi oraz zarządzanie stanem aplikacji.

4.2.4. Flask-Migrate

Jest to narzędzie pomagające w wersjonowaniu i migracji bazy danych. Szczególnie przydaje się podczas cyklu tworzenia aplikacji, kiedy model danych często ulega zmianie. Podczas tworzenia aplikacji było wykorzystywane właśnie do celów rozbudowy bazy danych o kolejne elementy.

4.3. Baza Danych

4.3.1. SQLite

Jest to lekki i samowystarczalny system zarządzania bazy danych SQL. Nie wymaga on dedykowanego serwera i idealnie nadaje się do prototypowania mniejszych aplikacji. W projekcie został wykorzystany jako prosta i niezawodna baza danych.

4.3.2. SQLAlchemy

SQLAlchemy jest biblioteką ORM – Object-Relational Mapping dla języka Python. Ułatwiła ona znacznie pracę z bazami danych SQL poprzez mapowanie klas Pythona na tabelę bazy danych. W aplikacji została wykorzystana do modelowania i interakcji z danymi co znacznie przyspieszyło jej rozwój.

4.4. Dodatkowe narzędzia

4.4.1. OpenAI API:

Jest to interfejs programistyczny, który umożliwił wykorzystanie zaawansowanych modeli sztucznej inteligencji w aplikacji takich jak GPT-3 oraz GPT-4. Głównym zadaniem w aplikacji jest komunikacja z tymi modelami w celu uzyskania sugestii dotyczącej najbardziej odpowiednich połączeń smakowych do wybranego przez użytkownika produktu. Ponadto wykorzystywana jest również umiejętność modelu GPT-4 do

rozpoznawania produktów znajdujących się na zdjęciu co pozwoliło znacznie usprawnić tę funkcjonalność.

5. Główne funkcjonalności

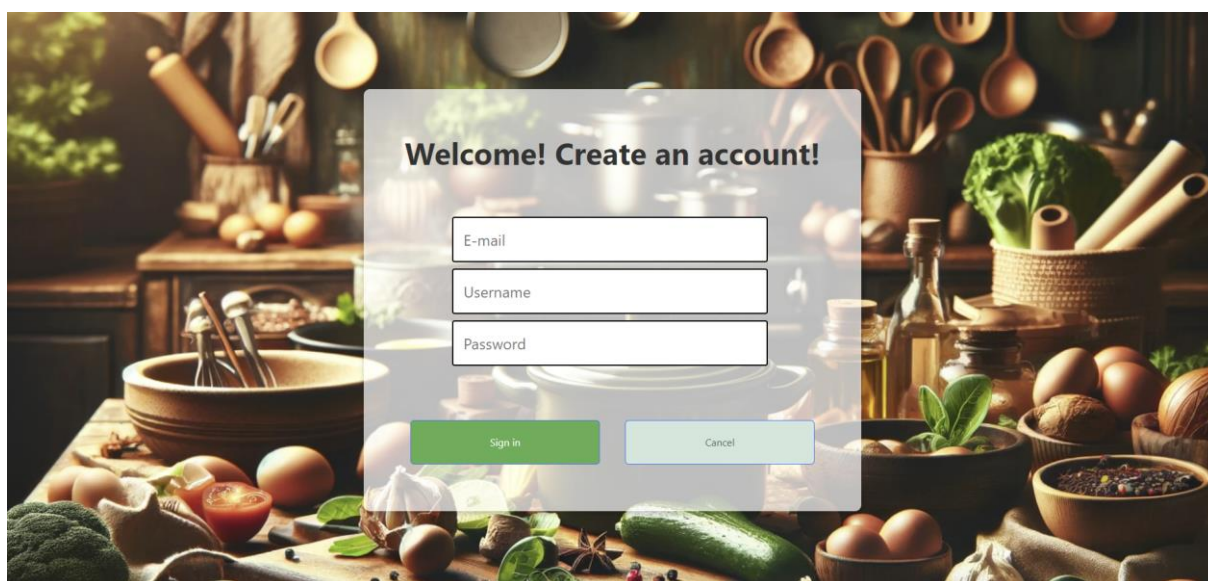
5.1. Ekran główny



Rysunek 16 Ekran główny aplikacji

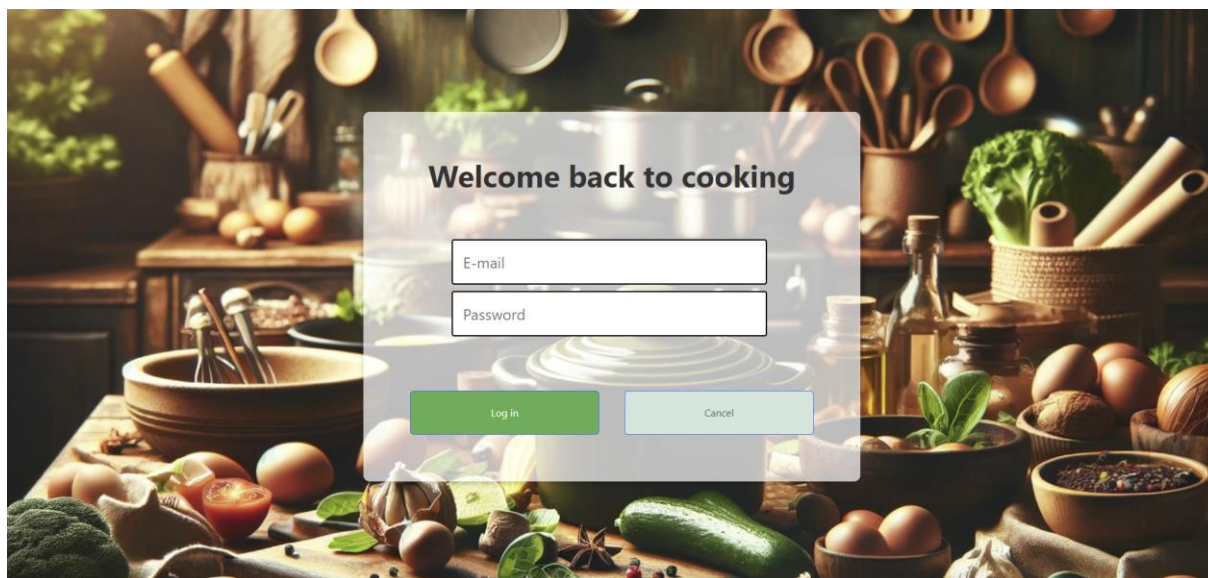
Pierwszy ekran aplikacji jest jej wizytówką, daje użytkownikowi możliwość zalogowania się, jeśli już posiada konto lub stworzenia nowego, dla tych którzy go nie mają. Widok ten jest prosty i intuicyjny, co zachęca użytkownika do interakcji z nim. Pozwala na nawigację do pozostałych funkcji aplikacji.

5.2. Rejestracja i logowanie



Rysunek 17 Ekran tworzenia konta

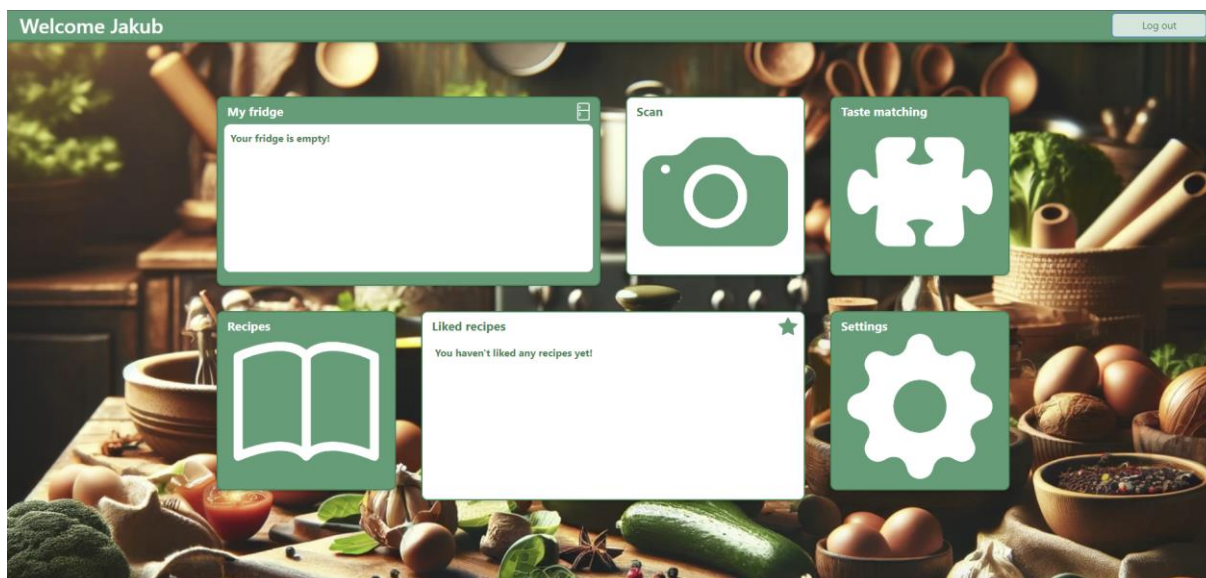
Ekran rejestracji zawiera prosty formularz, zachęcający użytkownika do stworzenia nowego konta. Do tej czynności wymagane jest podanie podstawowych informacji takich jak e-mail, nazwa użytkownika oraz hasło. Gdy użytkownika nie wprowadzi danych lub jeśli te podane nie spełniają założonych wymogów zostanie on poinformowany czerwonym komunikatem. Po wciśnięciu przycisku „Sign in” tworzone jest konto użytkownika, po czym zostaje on przekierowany do ekranu logowania w celu dalszej możliwości korzystania z aplikacji.



Rysunek 18 Ekran logowania

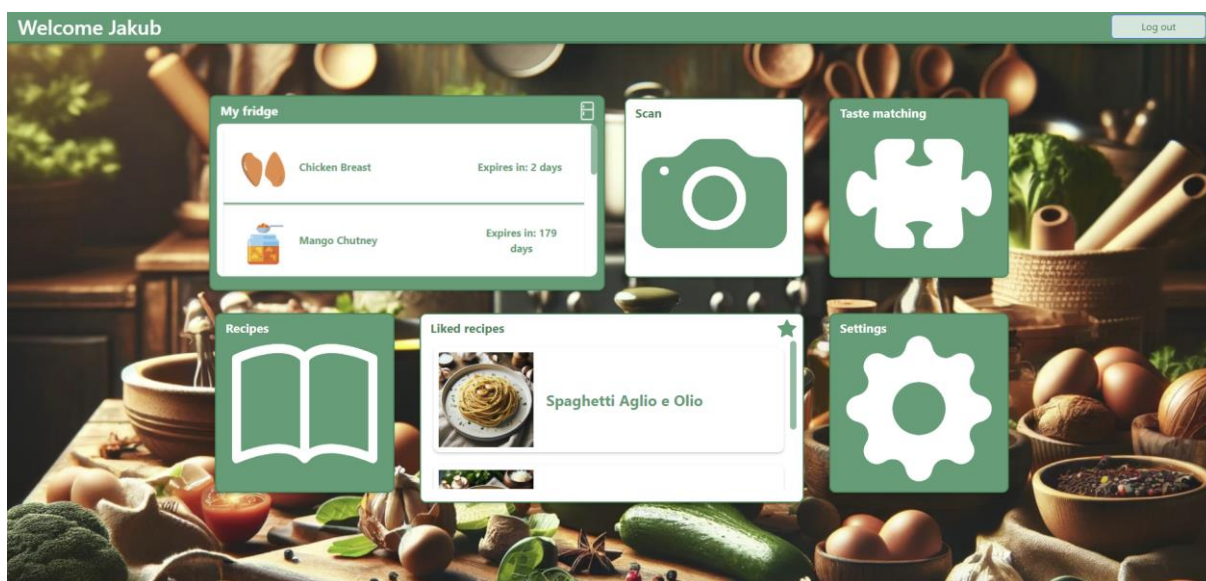
Ekran logowania wyglądem nawiązuje do ekranu rejestracji. Jest on prosty i intuicyjny. Użytkownik, aby się zalogować, jest proszony o podanie adresu e-mail oraz hasła, które podał podczas tworzenia konta. Jeśli wprowadzone dane będą niepoprawne zostanie wyświetlony czerwony komunikat informujący o tym. Po wciśnięciu przycisku „Log in” użytkownik jest autoryzowany i przekierowywany do swojego panelu głównego.

5.3. Panel główny użytkownika



Rysunek 19 Panel główny użytkownika, niezapełniony

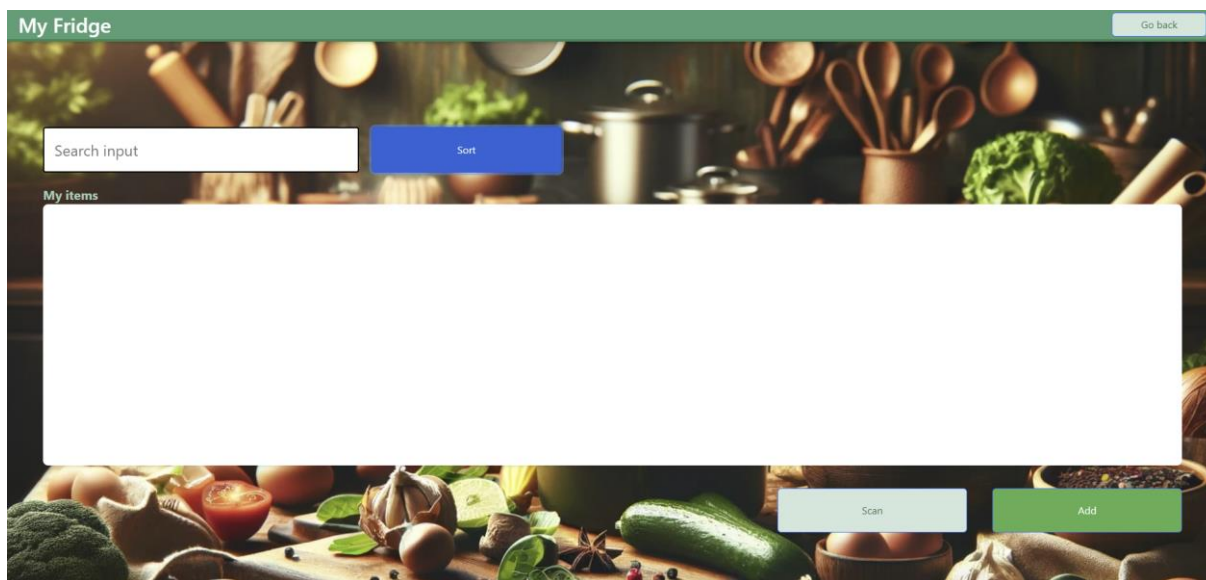
Panel główny użytkownika wita go wprowadzoną przez niego nazwą podczas rejestracji. Dla nowego użytkownika, który jeszcze nie dodał żadnych produktów do swojej lodówki, sekcja „My fridge” jest pusta. Kolejne widgety odpowiadają głównym funkcjom aplikacji. „Scan” odpowiada za skanowanie zdjęcia w poszukiwaniu produktów, „Taste matching” to funkcja odpowiedzialna za dobieranie połączeń smakowych do wybranego produktu. Widget „Recipes” daje użytkownikowi możliwość przejścia do listy wszystkich przepisów, gdzie może on je przeglądać oraz dodawać do ulubionych. Zostaną one wtedy wyświetlone w sekcji „Liked recipes”. Ikona „Settings” umożliwia użytkownikowi zarządzanie swoim kontem.



Rysunek 20 Panel główny użytkownika zapełniony

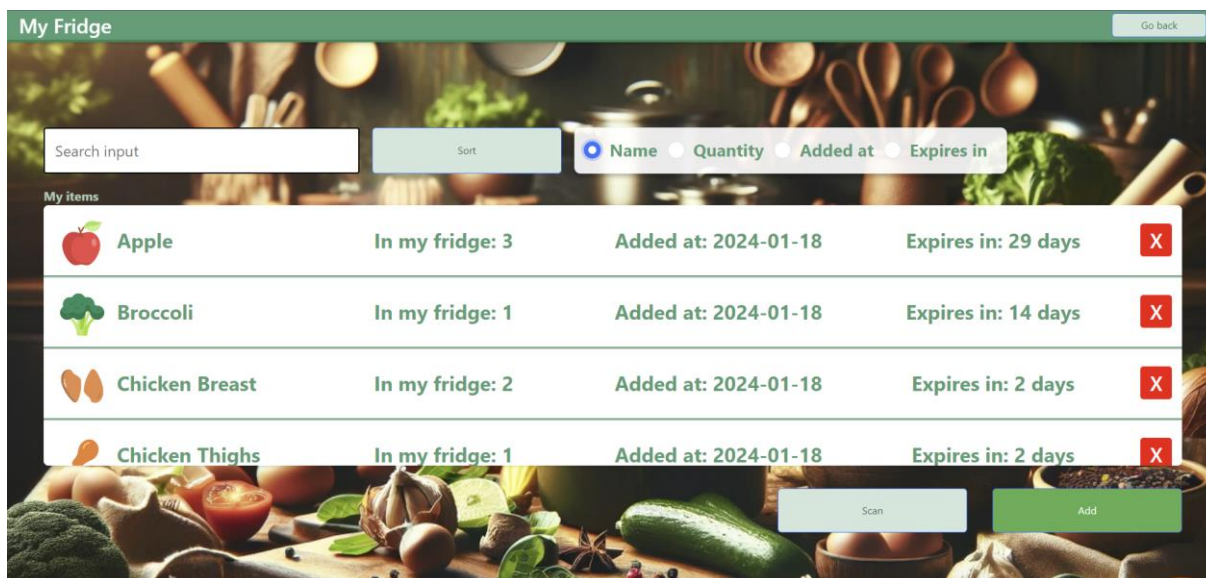
Gdy użytkownik zaczynie korzystać z aplikacji, jego panel główny odzwierciedla jego interakcje z aplikacją. W sekcji „My fridge” zostają wyświetlone produkty, które użytkownik dodał do swojej lodówki, wraz z liczbą dni po których ich termin ważności upłynie. Dodatkowo w sekcji „Liked Recipes” zostają wyświetlone polubione przez użytkownika przepisy, aby ułatwić mu do nich dostęp.

5.4. Lodówka użytkownika



Rysunek 21 Ekran pustej lodówki użytkownika

Ekran „My Fridge” przedstawia stan lodówki użytkownika. W miarę dodawania nowych produktów jest ona zapełniana oraz umożliwia użytkownikowi wyszukiwanie ich po nazwie lub posortowania, ze względu na kryteria takie jak: nazwa, ilość w lodówce, datę dodania oraz liczbę dni, po których produkt straci przydatność. Dwa przyciski w dolnej części pozwalają użytkownikowi na nawigację do kolejnych funkcji aplikacji, kolejno: „Scan” do skanowania produktów na podstawie zdjęcia oraz „Add” do dodawania produktów z listy składników.



Rysunek 22 Ekran zapelnionej lodówki użytkownika

Gdy użytkownik zacznie zapelnia lodówkę dodając produkty poprzez listę czy też skanowanie, zostaną one wyświetlone w panelu „My fridge” wraz z informacją o ilości danego produktu w lodówce, datą dodania oraz ilością dni, po których minie jego termin przydatności. Użytkownik w celu przeszukiwania swoich produktów może je sortować lub szukać bezpośredni po nazwie produktu. Dodatkowo może on usunąć wybrany produkt ze swojej lodówki.

5.5. Dodawanie produktów z listy

The screenshot shows the 'Add product' interface. At the top, there's a 'Go back' button. Below it is a search input field with the placeholder 'Search input' and a 'Sort' button. The main area displays a list of products under the heading 'Nazwa'. The products are Apple, Avocado, and Banana. Each product row includes an icon, the product name, 'In my fridge' count (all 0), 'Expires in' duration (30 days for Apple, 7 days for others), a 'Quantity' input field, and a green checkmark confirmation button. At the bottom, there are 'Cancel' and 'Confirm' buttons.

Nazwa	In my fridge	Expires in	Quantity	Confirm
Apple	0	30 days	<input type="text"/>	✓
Avocado	0	7 days	<input type="text"/>	✓
Banana	0	7 days	<input type="text"/>	✓

Rysunek 23 Ekran dodawania składników z listy

Kolejny ekran pozwala użytkownikowi przeglądać listę produktów oraz dodania dowolnej ich ilości do swojej lodówki. Aby to zrobić musi on podać wartość „Quantity” a następnie zatwierdzić zielonym przyciskiem obok.

This screenshot shows the 'Add product' interface with search and sorting capabilities. The search input field contains 'Chicken'. Below it, there are radio buttons for sorting: 'Name', 'Quantity', and 'Expires in' (which is selected). The product list under 'Nazwa' includes Chicken, Chicken Breast, and Chicken Thighs. Each row shows the product name, 'In my fridge' count (0, 2, and 1 respectively), 'Expires in' duration (2, 3, and 3 days), a 'Quantity' input field, and a green checkmark confirmation button. 'Cancel' and 'Confirm' buttons are at the bottom.

Nazwa	In my fridge	Expires in	Quantity	Confirm
Chicken	0	2 days	<input type="text"/>	✓
Chicken Breast	2	3 days	<input type="text"/>	✓
Chicken Thighs	1	3 days	<input type="text"/>	✓

Rysunek 24 Ekran dodawania składników z listy z sortowaniem i wyszukiwaniem

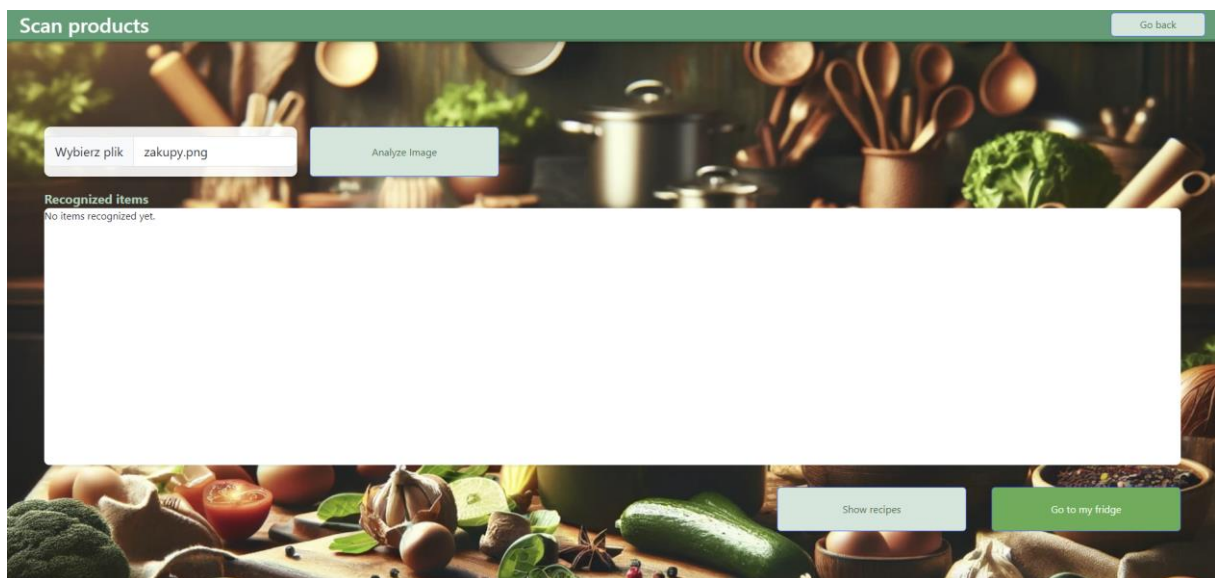
W celu przeszukiwania listy produktów, użytkownik może wprowadzić nazwę poszukiwanego produktu, co pozwala w łatwy i szybki sposób dodawać produkty do swojej lodówki. Dodatkowo użytkownik może posortować listę produktów ze względu na nazwę, ilość posiadanych produktów w lodówce oraz liczbę dni do minięcia terminu ważności.

5.6. Skanowanie produktów ze zdjęcia



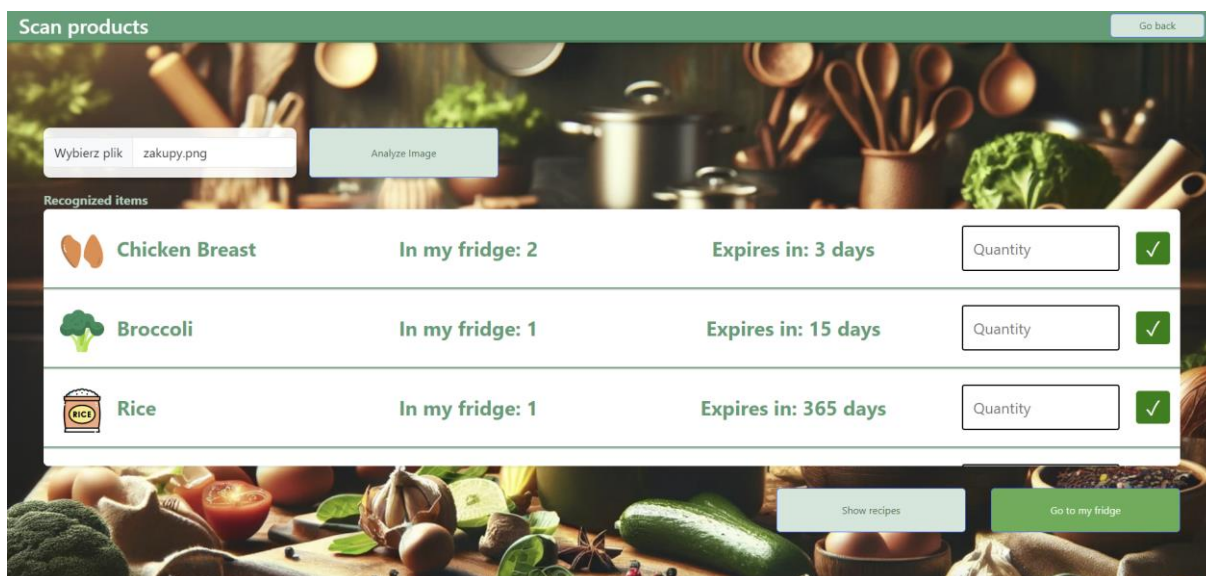
Rysunek 25 Przykładowy obraz produktów do skanowania

Na rysunku przedstawiono przykładowe zdjęcie zawierające produkty spożywcze, które następnie zostało przesłane do aplikacji, aby ta mogła rozpoznać co się na nich znajduje.



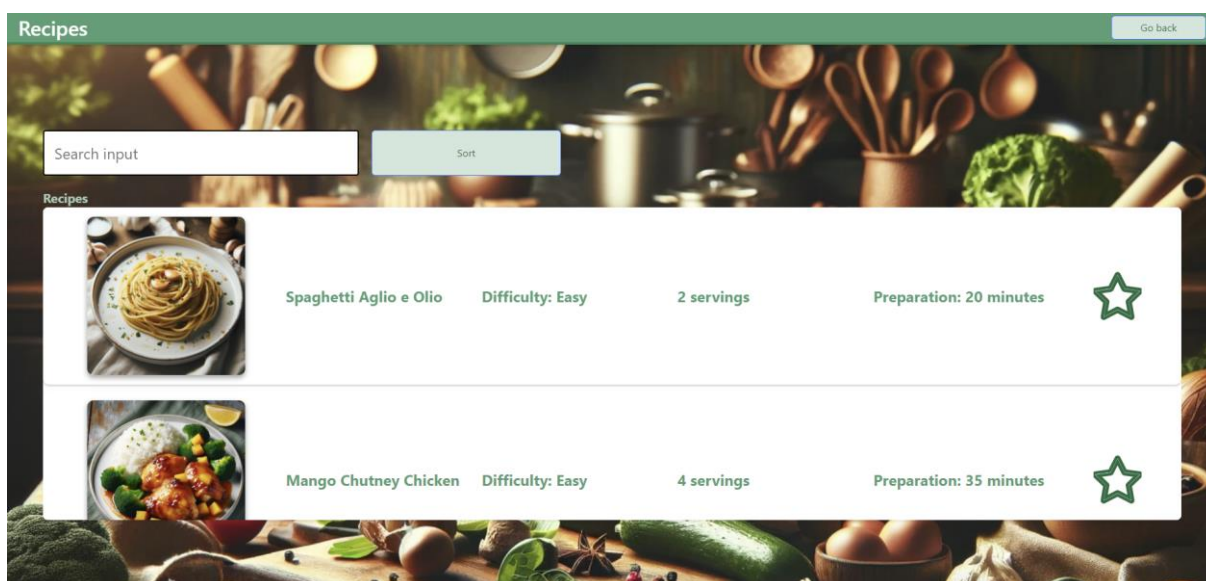
Rysunek 26 Ekran skanowania produktów przed rozpoznaniem

Panel „Scan products” umożliwia użytkownikowi przesłanie obrazu, który ma zostać poddany analizie w celu znalezienia na nim produktów spożywczych. Po załadowaniu obrazu i wciśnięciu przycisku „Analyze” aplikacja zabiera się za jego skanowanie i po chwili wyświetla rozpoznane produkty.



Rysunek 27 Ekran skanowania po rozpoznaniu produktów na obrazie

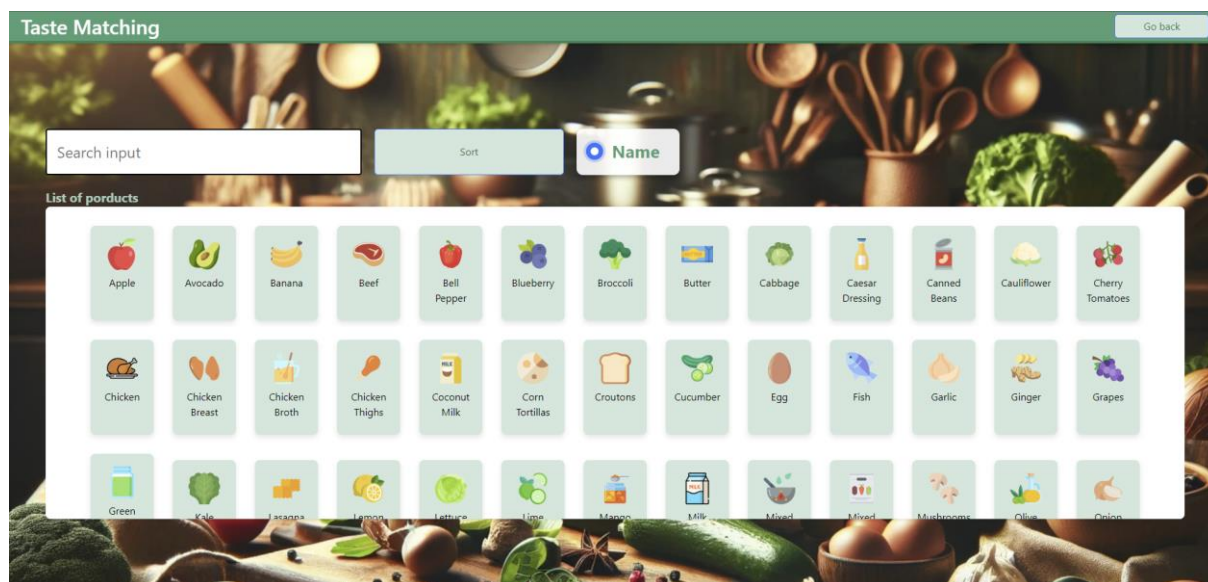
Aplikacja poprawnie rozpoznała produkty na zdjęciu i wyświetliła ich listę. Użytkownik ma teraz możliwość dodania ich do swojej lodówki. Na dole ekranu znajdują się dwa przyciski. „Go to my Fridge”, który prowadzi do lodówki użytkownika, natomiast po wciśnięciu „Show recipes” aplikacja analizuje listę zeskanowanych składników oraz listę składników znajdujących się w lodówce użytkownika a następnie wyświetla listę przepisów, które użytkownik jest w stanie przygotować ze względu na zeskanowane i posiadane składniki.



Rysunek 28 Ekran przepisów posortowanych ze względu na rozpoznane i posiadane składniki

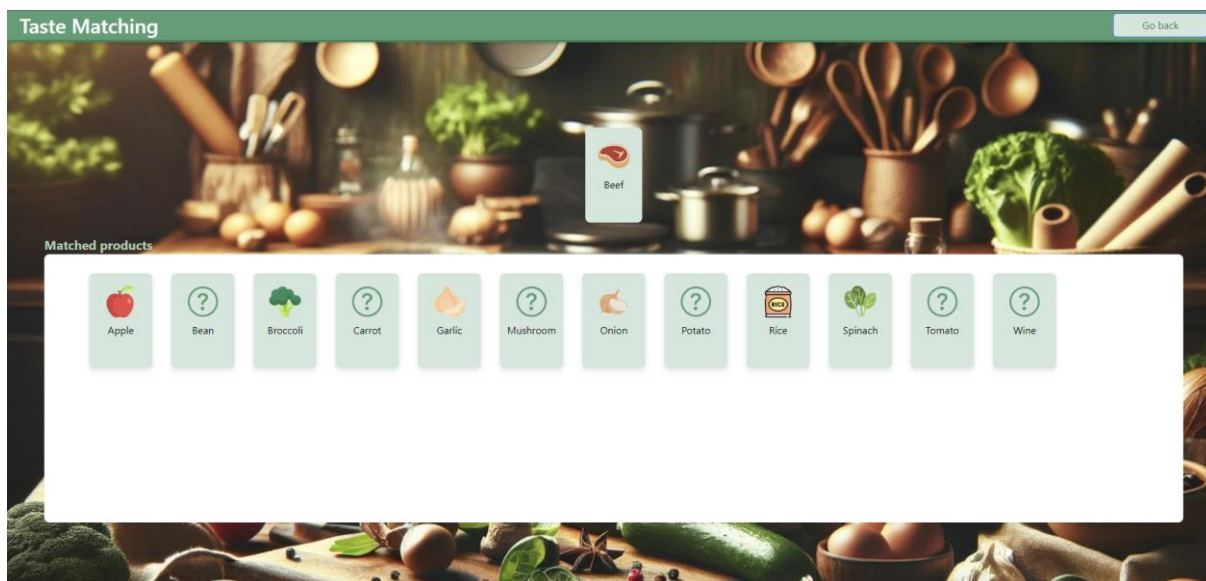
W tym widoku wyświetlone są przepisy, które użytkownik może wykonać wraz z informacją o ich trudności, ilości porcji oraz czasie przygotowania. Użytkownik może kliknąć w wybrany przepis w celu wyświetlenia szczegółowych informacji oraz kliknąć w gwiazdkę znajdującą się na panelu przepisu w celu dodania go do ulubionych.

5.7. Połączenia smakowe



Rysunek 29 Ekran połączeń smakowych

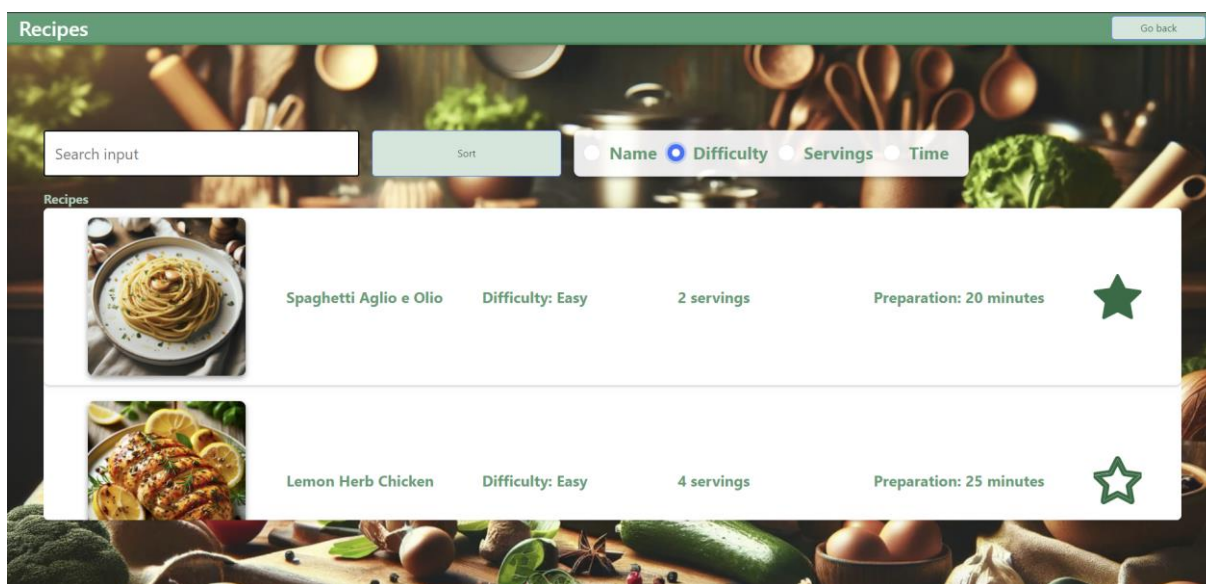
W panelu „Taste Matching” zostaje użytkownikowi wyświetlona siatka produktów wraz z ich ikonami w celu prostszej identyfikacji produktu. Użytkownik może wyszukiwać żądany produkt po nazwie lub posortować je alfabetycznie. Po kliknięciu wybrany produkt użytkownik zostaje przekierowany do panelu, gdzie wyświetlane są pasujące smakowo produkty do wybranego.



Rysunek 30 Ekran zaproponowanych przez aplikację połączeń smakowych

Ekran wybranego produktu prezentuje wybrany przez użytkownika produkt oraz siatkę innych produktów, które idealnie komponują się z tym wybranym.

5.8. Lista przepisów



Rysunek 31 Ekran listy przepisów

Po przejściu do panelu „Recipes” użytkownikowi zostaje wyświetlona lista wszystkich przepisów wraz z podstawowymi informacjami na ich temat. Może on również wyszukać przepis po nazwie lub posortować je ze względu na nazwę, trudność przygotowania, liczbę porcji oraz czas przygotowania. Dodatkowo klikając w gwiazdkę na panelu przepisu użytkownik może dodać go do ulubionych, jeśli wcześniej tego nie zrobił lub usunąć go z ulubionych, jeśli już taki znajduje się na jego liście. Gdy użytkownik

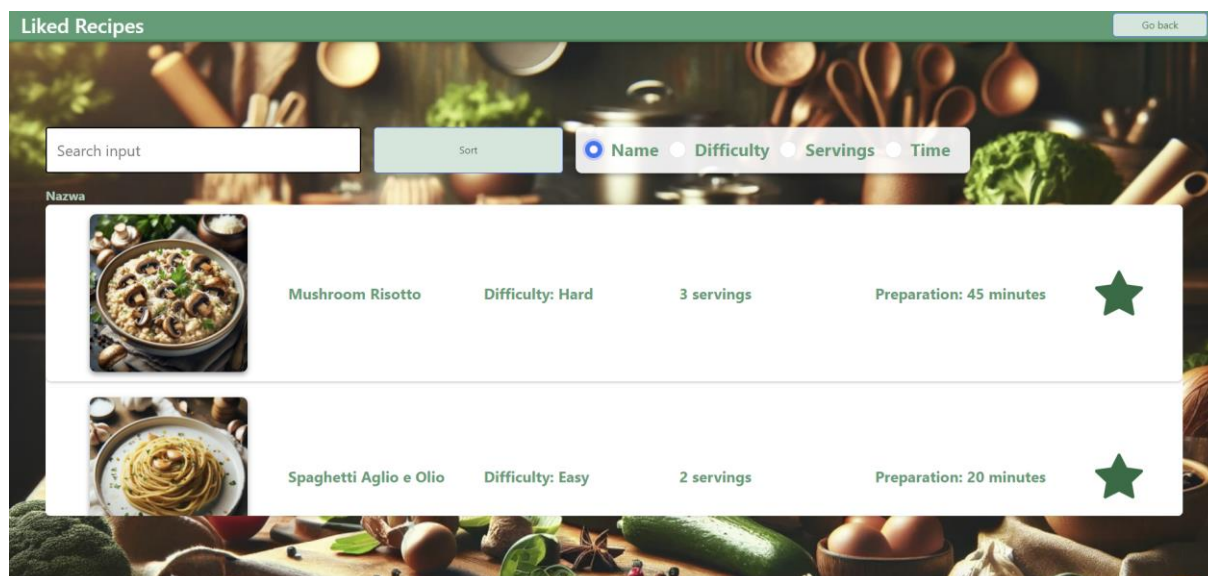
kliknie w wybrany przepis, zostanie przekierowany do jego strony ze szczegółowymi informacjami na jego temat.



Rysunek 32 Ekran szczegółów przepisu

Strona przepisu zawiera wszystkie jego informacje, takie jak, trudność przygotowania, czas przygotowania, liczbę porcji, potrzebne składniki oraz instrukcje jakie należy wykonać przy przygotowywaniu posiłku.

5.9. Lista polubionych przepisów

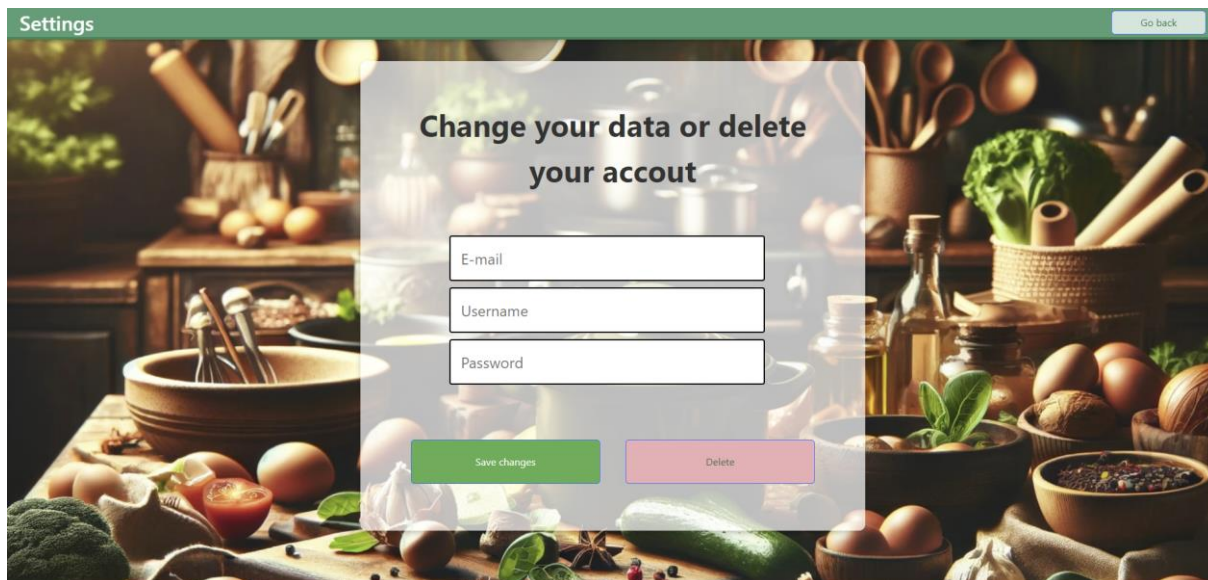


Rysunek 33 Ekran polubionych przepisów

Panel „Liked Recipes” wyświetla listę wszystkich polubionych przez użytkownika przepisów. Jak w panelu „Recipes” użytkownik może przeszukać swoje polubione przepisy wprowadzając nazwę poszukiwanego lub posortować je ze względu na nazwę,

trudność przygotowania, liczbę porcji oraz czas przygotowania. Może on również usunąć dany przepis z ulubionych klikając na gwiazdkę na jego panelu.

5.10. Ustawienia użytkownika



Rysunek 34 Ekran ustawień użytkownika

Widok ustawień użytkownika zezwala mu na zarządzanie swoimi danymi. Może on wprowadzając dane zmienić swój adres e-mail, nazwę użytkownika oraz hasło lub zupełnie usunąć konto. Podczas usuwania użytkownik pytany jest o potwierdzenie usunięcia konta, a następnie aplikacja usuwa dane użytkownika oraz przekierowuje go do ekranu głównego aplikacji.

6. Implementacja

6.1. Środowisko programistyczne

Aplikacja została zaimplementowana z wykorzystaniem Visual Studio Code. Jest to wszechstronny edytor kodu stworzony przez Microsoft. Ma rozbudowane wsparcie dla wielu języków programowania oraz bogaty zestaw rozszerzeń. Zintegrowany z edytorem klient Git pozwolił na zarządzanie wersjami aplikacji.

6.2. Modele danych

W aplikacji zostały wykorzystane obiekty modeli danych, które odpowiadają strukturom przechowywanym w bazie danych. Są one zdefiniowane przy użyciu ORM (Object-Relational Mapping) z SQLAlchemy, co umożliwia efektywne zarządzanie danymi ich relacjami. Poniżej zostały przedstawione przykładowe modele danych.

6.2.1. User

Model „User” reprezentuje użytkownika aplikacji. Każdy z nich posiada unikalny identyfikator – „id”, adres e-mail – „email”, nazwę użytkownika – „username” oraz zahashowane hasło – „password”. Adres e-mail oraz nazwa użytkownika są unikalne dla każdego rekordu, co zapobiega duplikacji kont. Hashowanie haseł zapewnia bezpieczeństwo przechowywanych danych.

```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(120), unique=True, nullable=False)
    username = db.Column(db.String(20), unique=True, nullable=False)
    password = db.Column(db.String(60), nullable=False)
```

6.2.2. FoodItem

Model ten zawiera informacje o artykułach spożywczych. Posiada pola takie jak unikalny identyfikator – „id”, nazwę produktu – „name”, liczbę dni do zepsucia – „spoilage_days” oraz URL do ikony reprezentującej produkt – „icon_url”. Relacja „recipe_ingredient” definiuje związek z modelem „RecipeIngredient” umożliwiając odwzorowanie składników w przepisach. Metoda „serialize” służy do konwersji danych na format JSON, co ułatwia wymianę danych z frontendem.

```

class FoodItem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    spoilage_days = db.Column(db.Integer, nullable=False)
    icon_url = db.Column(db.String(255))

    recipe_ingredients = db.relationship('RecipeIngredient',
back_populates='food_item')

    def serialize(self):
        return {
            "id": self.id,
            "name": self.name,
            "spoilage_days": self.spoilage_days,
            "icon_url": self.icon_url
        }

```

6.2.3. Recipe

Model „Recipe” opisuje przepisy kulinarne. Zawiera informacje takie jak identyfikator – „id”, nazwa – „name”, poziom trudności – „difficulty”, czas przygotowania – „preparation_time”, instrukcje przygotowania – „instructions” oraz URL do obrazu – „image_url”. Posiada również związki z „FoodItem” poprzez „recipe_ingredient” oraz model „RecipeIngredient”, co pozwala na zdefiniowanie listy składników danego przepisu. Metoda „serialize” pozwala na łatwe przekształcenie danych przepisu wraz ze składnikami do formatu JSON.

```

class Recipe(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    difficulty = db.Column(db.String(50), nullable=False)
    preparation_time = db.Column(db.Integer, nullable=False)
    servings = db.Column(db.Integer, nullable=False)
    instructions = db.Column(db.Text, nullable=False)
    image_url = db.Column(db.String(255))

    ingredients = db.relationship('FoodItem', secondary='recipe_ingredient',
overlaps="recipe_ingredients")
    recipe_ingredients = db.relationship('RecipeIngredient',
back_populates='recipe')

    def serialize(self):
        return {
            "id": self.id,
            "name": self.name,

```

```

        "difficulty": self.difficulty,
        "preparation_time": self.preparation_time,
        "servings": self.servings,
        "instructions": self.instructions,
        "image_url": self.image_url,
        "ingredients": [ingredient.serialize() for ingredient in
self.ingredients],
        "recipe_ingredients": [
            {
                "food_item_id": ri.food_item_id,
                "quantity": ri.quantity
            } for ri in self.recipe_ingredients
        ]
    }

```

6.3. Implementacja metod CRUD dla użytkownika

W aplikacji zostały zaimplementowane podstawowe operacje CRUD (Create, Read, Update i Delete) dla zarządzania kontami użytkowników. Wykorzystany do tego framework Flask dostarczył prosty, ale również mocny zestaw narzędzi do tworzenia aplikacji webowych. Dzięki wykorzystaniu Flask-JWT-Extended możliwe było też zaimplementowanie uwierzytelniania i autoryzacji opartej na tokenach JWT, co zapewnia bezpieczny dostęp do API aplikacji.

6.3.1. Create

Rejestracja nowego użytkownika odbywa się poprzez endpoint '/register', który przyjmuje dane – email, username oraz password, weryfikuje je, a następnie tworzy nowy rekord w bazie danych.

```

@auth_blueprint.route('/register', methods=['POST'])
def register():
    data = request.json

    is_valid, message = validate_field(data.get('email'), validate_email)
    if not is_valid:
        return {'message': message}, 400

    is_valid, message = validate_field(data.get('username'),
validate_username)
    if not is_valid:
        return {'message': message}, 400

    is_valid, message = validate_field(data.get('password'),
validate_password)
    if not is_valid:

```

```

        return {'message': message}, 400
    if User.query.filter_by(email=data['email']).first():
        return {'message': 'Email already in use'}, 400

    if User.query.filter_by(username=data['username']).first():
        return {'message': 'Username already in use'}, 400

    hashed_password =
bcrypt.generate_password_hash(data['password']).decode('utf-8')
    user = User(username=data['username'], email=data['email'],
password=hashed_password)
    db.session.add(user)
    db.session.commit()
    return {'message': 'User created successfully'}, 201

```

6.3.2. Read

Logowanie użytkownika oraz odczytywanie danych profilu realizowane są przez endpoint '/login', który po pomyślnym uwierzytelnieniu zwraca token JWT.

```

@auth_blueprint.route('/login', methods=['POST'])
def login():
    data = request.json
    is_valid, message = is_field_empty(data.get('email'), 'E-mail')
    if not is_valid:
        return {'message': message}, 400

    is_valid, message = is_field_empty(data.get('password'), 'Password')
    if not is_valid:
        return {'message': message}, 400

    user = User.query.filter_by(email=data['email']).first()
    if user and bcrypt.check_password_hash(user.password, data['password']):
        access_token = create_access_token(identity=user.id)
        return {'access_token': access_token, 'username': user.username,
'user_id': user.id, 'message': 'Login successful'}, 200
    else:
        return {'message': 'Incorret e-mail or password'}, 401

```

6.3.3. Update

Zmiana danych użytkownika takich jak adres e-mail, hasło czy nazwę użytkownika jest możliwa jedynie przy uwierzytelnieniu. Po walidacji wprowadzonych danych zostają zaktualizowane dane użytkownika dzięki użycia endpointu '/update-user/<user_id>'.

```
@auth_blueprint.route('/update-user/<int:user_id>', methods=['PUT'])
@jwt_required()
@cross_origin()
def update_user(user_id):
    current_user_id = get_jwt_identity()
    if current_user_id != user_id:
        return {'message': 'Unauthorized access'}, 401

    data = request.json
    user = User.query.get(user_id)
    if not user:
        return {'message': 'User not found'}, 404

    is_valid, message = validate_field(data.get('email'), validate_email)
    if not is_valid:
        return {'message': message}, 400

    is_valid, message = validate_field(data.get('username'),
validate_username)
    if not is_valid:
        return {'message': message}, 400

    is_valid, message = validate_field(data.get('password'),
validate_password)
    if not is_valid:
        return {'message': message}, 400

    if User.query.filter_by(email=data['email']).first():
        return {'message': 'Email already in use'}, 400

    if User.query.filter_by(username=data['username']).first():
        return {'message': 'Username already in use'}, 400

    user.password =
bcrypt.generate_password_hash(data['password']).decode('utf-8')
    user.username = data['username']
    user.email = data['email']
    db.session.commit()
    return {'message': 'User updated successfully'}, 200
```

6.3.4. Delete

Usunięcie konta użytkownika implementowane jest przez endpoint `/delete-user/<user_id>`. Wymaga ono uwierzytelnienia.

```
@auth_blueprint.route('/delete-user/<int:user_id>', methods=['DELETE'])
@cross_origin()
@jwt_required()
def delete_user(user_id):
    current_user_id = get_jwt_identity()
    if current_user_id != user_id:
        return {'message': 'Unauthorized access'}, 401

    user = User.query.get(user_id)
    if not user:
        return {'message': 'User not found'}, 404

    FridgeItem.query.filter_by(user_id=user_id).delete()
    LikedRecipe.query.filter_by(user_id=user_id).delete()

    db.session.delete(user)
    db.session.commit()
    return {'message': 'User deleted successfully'}, 200
```

6.4. Połączenie z OpenAI API

Aby rozszerzyć funkcjonalność aplikacji o zaawansowane możliwości sztucznej inteligencji została ona zintegrowana z OpenAI API. Umożliwia to na wykorzystanie modeli językowych GPT-3 oraz GPT-4. Integracja ta pozwala na generowanie sugestii dotyczących połączeń smakowych do wybranego produktu oraz na analizowanie przesłanych zdjęć pod kątem znajdujących się na nich produktach. Wykorzystanie API odbywa się poprzez endpointy zdefiniowane w aplikacji, które komunikują się z serwerami OpenAI wysyłając zapytania i odbierając odpowiedzi. Poniżej został ukazany endpoint służący do komunikacji z modelem językowym OpenAI GPT-3.5-turbo.

```
@resources_blueprint.route('/ask-gpt', methods=['POST'])
@jwt_required()
def ask_gpt():
    data = request.json
    input_label = data.get('label') # Assuming you receive a 'label' field

    if not input_label:
        return jsonify({'error': 'No label provided'}), 400
```

```

    prompt = f"List between 10 and 25 foods that pair well with {input_label},
just the names in singular form. Each word of the name should start with
capital letter. Separate each item with '\n' and don't numerate them and
don't use any other characters"

    try:
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=[{"role": "system", "content": "You are a helpful
assistant."},
                    {"role": "user", "content": prompt}]
        )
        # Process the response to extract only item names if necessary
        print("Response from GPT:", response)
        items_text = response.choices[0].message['content']
        items = process_response(items_text)

        items_with_icons = []
        for item_name in items:
            item = FoodItem.query.filter_by(name=item_name.strip()).first()
            if item:
                items_with_icons.append({'name': item_name, 'icon_url':
item.icon_url})
            else:
                items_with_icons.append({'name': item_name, 'icon_url': None})

        return jsonify({'items': items_with_icons})
    except Exception as e:
        print(f"RequestException occurred: {e}")
        return jsonify({'error': str(e)}), 500

```


7. Możliwości rozwoju

Aplikacja wspomagająca proces gotowania posiada już szeroki zakres funkcjonalności, jednak może być ciągle rozwijana i ulepszana, aby jeszcze lepiej odpowiadać na potrzeby użytkowników. Oto kilka kierunków, w jakich aplikacja może się rozwijać.

7.1. Liczenie kalorii

Podobnie jak w aplikacji Fitatu, jedną z atrakcyjnych możliwości rozwoju aplikacji jest dodanie funkcji monitorowania swojego jadłospisu i liczenia kalorii. Pozwoliłoby to użytkownikom nie tylko na planowanie posiłków z dostępnych produktów, ale również świadome zarządzanie dietą oraz utrzymywanie zdrowego stylu życia. Funkcja ta mogłaby obejmować:

- Śledzenie kalorii spożywanych w poszczególnych posiłkach,
- Monitorowanie codziennego jadłospisu,
- Analiza makroskładników w spożywanych posiłkach.

7.2. Rozszerzanie bazy przepisów przez użytkowników

Umożliwienie użytkownikom dodawania własnych przepisów do bazy danych pozwoliłoby na jej dynamiczny rozwój. Zapewniłoby do innym użytkownikom zdecydowanie większą różnorodność co do wyboru posiłku oraz umożliwiłoby na czerpanie inspiracji od innych. Dodatkowo możliwość dzielenia się opiniami na temat przepisów pozwoliłoby na selekcję tylko tych, które spełniają oczekiwania użytkowników.

7.3. Planer jadłospisu i zakupów

Wprowadzenia planera jadłospisu na cały tydzień to kolejny pomysł na rozwój aplikacji. Mogłaby ona jeszcze znacznie poprawić sposób zarządzania żywnością użytkowników eliminując przy tym marnowanie się żywności do minimum. Dodatkowo ograniczyło by to również liczbę potrzebnych wyjść do sklepu, gdyż aplikacja mogłaby tworzyć listy zakupów zależne od tego, co w danym tygodniu zostało zaplanowane do spożycia przez użytkownika.

8. Podsumowanie

Podsumowując, przedstawiony projekt aplikacji wspomagającej proces gotowania stanowi w pewnym stopniu rozwiązanie problemu dotyczącego nie tylko planowania posiłków i zarządzaniem zapasami żywności ale również dzięki swoim funkcjom jest w stanie przyczynić się do ograniczenia marnotrawstwa żywności w gospodarstwach domowych. Aplikacja ta umożliwia użytkownikom przechowywanie danych na temat posiadanego jedzenia, ale też dzięki wykorzystaniu sztucznej inteligencji pozwala na skanowanie obrazów w poszukiwaniu produktów spożywczych oraz proponuje na podstawie wybranego produktu pasujące do niego inne produkty. Może być również inspiracją do tworzenia zróżnicowanych posiłków z dostępnych składników.

W trakcie realizacji największy nacisk został nałożony na intuicyjność interfejsu użytkownika, co było możliwe do osiągnięcia dzięki zastosowaniu frameworka React w połączeniu z Bootstrapem. Zapewniło to atrakcyjność wizualną aplikacji. Backend stworzony z użyciem Flaska zapewnia solidną strukturę oraz bezpieczeństwo kluczowe dla ochrony danych użytkowników oraz płynnego działania systemu.

Integracja z OpenAI API, wykorzystując modele GPT-3 oraz GPT-4 otworzyła nowe możliwości w zakresie skanowania obrazów oraz generowaniu sugestii dotyczących połączeń smakowych. Dodatkowo możliwości dalszego rozwoju aplikacji, takie jak implementacja planera jadłospisu, czy funkcji liczenia kalorii, do których również może zostać wykorzystana sztuczna inteligencja, pokazuje na szerokie perspektywy aplikacji.

Podczas tworzenia aplikacji nie obeszło się bez przeszkód i wyzwań. Jedną z nich było dostarczenie do aplikacji ikon produktów. Wykorzystano do tego ikony ze strony flaticon.com. Jednak podczas skanowania zdjęć lub otrzymywania sugerowanych połączeń smakowych zdarzały się sytuacje, że rozpoznane lub sugerowane produkty nie istniały w bazie danych produktów, co skutkowało potrzebą wyświetlenia uniwersalnej ikony. Aby rozwiązać ten problem należałoby rozbudować istniejącą bazę danych produktów, aby zawierała ich ogromną ilość.

Realizacja projektu była również okazją do pogłębienia wiedzy zakresu programowania oraz zastosowania teorii nabytej podczas nauki na studiach w praktyce. To doświadczenie pozwoliło na rozwój umiejętności technicznych oraz na zrozumienie znaczenia myślenia analitycznego i kreatywnego w procesie tworzenia oprogramowania.

9. Bibliografia

- https://pl.wikipedia.org/wiki/Sztuczna_inteligencja
- <https://en.wikipedia.org/wiki/GPT-4>
- <https://openai.com/research/gpt-4>
- <https://edukacja.rp.pl/trendy/art37668171-marnowanie-zywnosci-przestaje-byc-w-polsce-problemem-to-wciaz-ponura-statystyka>
- <https://www.fitatu.com/>
- <https://www.droptica.pl/blog/co-jest-react-i-jak-go-uzywac-w-projektach-kompleksowy-przewodnik/>
- <https://platform.openai.com/docs/guides/vision>
- <https://flask.palletsprojects.com/en/3.0.x/>
- <https://vitejs.dev/guide/>
- <https://platform.openai.com/docs/api-reference/authentication?lang=python>
- <https://www.sqlite.org/docs.html>
- <https://docs.sqlalchemy.org/en/20/>

10. Spis ilustracji

Rysunek 1 Ekran dodawania schowka Fridify.....	10
Rysunek 2 Ekran Panel Głównego Fridify.....	11
Rysunek 3 Ekran dodawania produktu Fridify.....	12
Rysunek 4 Ekran schowka "Lodówka" Fridify	13
Rysunek 5 Ekran widoku głównego użytkownika Fitatu	15
Rysunek 6 Ekran listy składników Fitatu	16
Rysunek 7 Ekran listy przepisów Fitatu	17
Rysunek 8 Ekran szczegółów przepisu Fitatu.....	18
Rysunek 9 Strona główna bloga kulinarnego.....	19
Rysunek 10 Strona przepisu bloga kulinarnego.....	20
Rysunek 11 Instrukcje przepisu bloga kulinarnego	20
Rysunek 12 Diagram przypadków użycia	24
Rysunek 13 Projekt komponentów w Figmie	25
Rysunek 14 Projekt ekranów w Figmie (1).....	25
Rysunek 15 Projekt ekranów w Figmie (2).....	26
Rysunek 16 Ekran główny aplikacji.....	30
Rysunek 17 Ekran tworzenia konta	30
Rysunek 18 Ekran logowania	31
Rysunek 19 Panel główny użytkownika, niezapełniony.....	32
Rysunek 20 Panel główny użytkownika zapełniony	32
Rysunek 21 Ekran pustej lodówki użytkownika.....	33
Rysunek 22 Ekran zapełnionej lodówki użytkownika	34
Rysunek 23 Ekran dodawania składników z listy	35
Rysunek 24 Ekran dodawania składników z listy z sortowaniem i wyszukiwaniem	35
Rysunek 25 Przykładowy obraz produktów do skanowania	36
Rysunek 26 Ekran skanowania produktów przed rozpoznaniem	36
Rysunek 27 Ekran skanowania po rozpoznaniu produktów na obrazie.....	37
Rysunek 28 Ekran przepisów posortowanych ze względu na rozpoznane i posiadane składniki	37
Rysunek 29 Ekran połączeń smakowych	38
Rysunek 30 Ekran zaproponowanych przez aplikację połączeń smakowych.....	39
Rysunek 31 Ekran listy przepisów	39
Rysunek 32 Ekran szczegółów przepisu	40
Rysunek 33 Ekran polubionych przepisów	40
Rysunek 34 Ekran ustawień użytkownika	41

