

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Paweł Giżka

Nr albumu: 385522

Kamil Ćwintal

Nr albumu: 385369

Szymon Gajda

Nr albumu: 385477

Tomasz Kanas

Nr albumu: 385674

Aplikacja mobilna "Podziel się"

Praca licencjacka
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
dra Janusza Jabłonowskiego
Instytut Informatyki

Czerwiec 2019

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpisy autorów pracy

Streszczenie

“Podziel się” to aplikacja mobilna przeznaczona do wzajemnego dzielenia się żywnością dla użytkowników systemów operacyjnych Android oraz iOS. Celem projektu było stworzenie platformy, która umożliwi łatwe i bezpłatne dzielenie się jedzeniem z innymi mieszkańcami w okolicy: osoby posiadające nadwyżkę artykułów żywnościowych lub gotowych potraw mogą przekazać je użytkownikom, którzy zadeklarują chęć ich odbioru. W pracy przedstawiono szczegółowy opis aplikacji oraz całościowej architektury projektu, a także przybliżono proces jego powstawania.

Słowa kluczowe

informatyka, aplikacja mobilna, żywność, foodsharing, React Native

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.3 Informatyka

Klasyfikacja tematyczna

Software and its engineering

Applied computing → Law, social and behavioral sciences → Sociology

Spis treści

Wprowadzenie	5
1. Przegląd konkurencyjnych rozwiązań	9
1.1. Sytuacja na rynku	9
1.2. Olio	9
1.3. Too Good To Go	9
1.4. Grupa “Freeganism, dumpster diving, food sharing — Warszawa”	10
2. Implementacja	11
2.1. Architektura systemu	11
2.1.1. Aplikacja mobilna	11
2.1.2. Część Serwerowa	12
2.1.3. Komunikacja Klient - Serwer	12
2.2. Postgis	12
2.3. Przechowywanie zdjęć	12
2.4. Powiadomienia	13
2.5. Logowanie za pomocą Facebooka	14
2.6. Działanie aplikacji mobilnej na dwóch systemach operacyjnych	14
2.7. Infrastruktura serwera	18
3. Przypadki użycia	19
3.1. Rejestracja	19
3.1.1. Krótki opis	19
3.1.2. Cele	19
3.1.3. Warunki wstępne	19
3.1.4. Czynności	19
3.2. Dodawanie nowego ogłoszenia	20
3.2.1. Krótki opis	20
3.2.2. Cele	20
3.2.3. Warunki wstępne	20
3.2.4. Czynności	20
3.2.5. Możliwe rozwinięcia	22
3.3. Konwersacja (czat) z innym użytkownikiem	22
3.3.1. Krótki opis	22
3.3.2. Cele	22
3.3.3. Warunki wstępne	22
3.3.4. Czynności	22
3.3.5. Możliwe rozwinięcia	23

3.4. Ocena innego użytkownika	23
3.4.1. Krótki opis	23
3.4.2. Cele	23
3.4.3. Warunki wstępne	23
3.4.4. Czynności	24
3.4.5. Możliwe rozwinięcia	25
3.5. Odbiór przedmiotów oferowanych w liście ogłoszeń	25
3.5.1. Krótki opis	25
3.5.2. Cele	25
3.5.3. Warunki wstępne	25
3.5.4. Czynności	25
3.5.5. Możliwe rozwinięcia	26
4. Zarządzanie projektem	27
4.1. Zbieranie wymagań	27
4.2. Podział obowiązków	27
5. Instrukcja budowania aplikacji i zawartość płytki	29
6. Podsumowanie	31
Bibliografia	33

Wprowadzenie

Kontekst społeczny

Według współczesnych szacunków ([FAO]), około 1/3 wytworzonej na świecie żywności nigdy nie zostanie zjedzona. Skala problemu jest szczególnie widoczna w danych Eurostatu ([Eurostat]), z których wynika, że w Polsce rocznie marnuje się około 9 milionów ton żywności. Jej wartość sięga około 50 złotych miesięcznie w przeliczeniu na jednego mieszkańca Polski. Badanie [Kantar Millward Brown] wykazało, że do wyrzucania jedzenia przyznaje się 42% Polaków, podczas gdy wyrzucane produkty mogłyby przyczynić się do zaspokojenia podstawowych potrzeb żywieniowych niemal 3 mln osób żyjących w skrajnym ubóstwie. Nasz kraj jest piątym wśród państw Unii Europejskiej, w których marnuje się najwięcej żywności.

W krajach rozwiniętych około 50% wyrzucanego jedzenia pochodzi z gospodarstw domowych, w których na skutek nierozważnego planowania zakupów jedzenie ulega zepsuciu i trafia do śmietnika ([Polityka]). Wierzimy, że możliwe jest przeciwdziałanie temu zjawisku na poziomie lokalnym.

Istnieją ruchy społeczne (dumpster diving, freeganizm, zero waste) promujące możliwe racjonalny obrót żywnością oraz uwrażliwiające społeczeństwo na problem masowego wyrzucania jedzenia.

Problem do rozwiązania

Obecnie w Polsce istnieją liczne grupy entuzjastów wymiany żywności funkcjonujące na portalach społecznościowych (przeważnie na Facebooku), jednak duża liczebność tych grup uniemożliwia skuteczną organizację wymiany jedzenia w satysfakcjonujący sposób. Nasza aplikacja „*Podziel się*” powstała, aby wspierać dalsze funkcjonowanie tej inicjatywy oraz zapewnić zwolennikom idei food sharingu platformę do efektywnej wymiany żywności, także w znacznie większej skali.

Zamawiający

Aplikacja „*Podziel się*” powstała na zamówienie pani Marii Skołożyńskiej, działaczki społecznej oraz pomysłodawczyni akcji *Podziel się Posiłkiem z Bezdomnymi* ([podzielmysie.pl]), podczas której mieszkańcy całej Polski mogą przekazać pozostałe po świętach (Bożego Narodzenia, Wielkanocy) posiłki do wolontariuszy, którzy zajmują się transportem jedzenia do osób potrzebujących. Pani Maria jest aktywistką działającą na rzecz przeciwdziałania marnowaniu jedzenia oraz główną koordynatorką akcji dystrybucji żywności. Działalność pani Marii Skołożyńskiej ma zasięg ogólnopolski; akcje gromadzenia żywności dla bezdomnych i ubogich działają obecnie w kilkudziesięciu miastach, m. in. w Warszawie, Krakowie, Wrocławiu, Trójmieście, Szczecinie, Toruniu, Bydgoszczy, Katowicach, Łodzi. W 2018 r. akcja *Podziel się Po-*

siłkiem z Bezdomnymi, w uznaniu dla skuteczności działania oraz promocji pozytywnych postaw społecznych, otrzymała Europejską Nagrodę Reduce Food Waste ([reducefoodwaste.eu]).

Osoby zaangażowane w projekt

Aplikacja *“Podziel się”* została opracowana oraz wdrożona przez zespół studentów informatyki Uniwersytetu Warszawskiego: Kamila Ćwintala, Szymona Gajdę, Pawła Giżkę oraz Tomasza Kanasa. Opiekę nad projektem sprawował dr Janusz Jabłonowski. Wymagania funkcjonalne zostały dostarczone zespołowi przez pomysłodawczynię projektu: p. Marię Skołożyńską oraz p. Martynę Dakowską.

Grupa docelowa

Aplikacja *“Podziel się”* jest przeznaczona dla użytkowników smartfonów, którym nie jest objętny problem marnowania żywności. *“Podziel się”* może stanowić interesującą propozycję dla osób, które nie mają czasu na przygotowanie posiłku albo gotują zbyt dużo (i jednocześnie nie chcą wyrzucać nadwyżki jedzenia). Aplikacja *“Podziel się”* będzie docelowo zrzeszać społeczność osób zainteresowanych tematyką food sharingu i zero waste, które dbają o środowisko naturalne oraz racjonalną gospodarkę żywnością.

Opis funkcjonalności

Aplikacja *“Podziel się”* umożliwia:

- bezpłatną rejestrację nowego konta,
- uzupełnienie własnego profilu o podstawowe dane personalne,
- dodawanie nowego ogłoszenia (opisu produktu wraz z preferowaną lokalizacją miejsca odbioru),
- zamieszczenie w ogłoszeniu zdjęć produktu, opakowania, etykiet itp.,
- przeglądanie dostępnych do odebrania posiłków w okolicy aktualnej lokalizacji użytkownika,
- sortowanie dostępnych ogłoszeń w oparciu o ustalone kryteria (odległość, data wystawienia produktu, data przydatności itp.),
- funkcję czatu między odbiorcą a darczyńcą w celu ustalenia szczegółów odbioru, zadawania dodatkowych pytań,
- przechowywanie historii konwersacji z innymi użytkownikami,
- wskazanie użytkownika (ze zbiorczej listy osób zainteresowanych ogłoszeniem), która otrzyma wystawiony produkt/artkuł żywnościowy,
- pośrednictwo w procesie przekazania żywności,
- usuwanie ogłoszeń zrealizowanych pomyślnie lub ogłoszeń, dla których minęła data przydatności do spożycia,
- system recenzji/ocen użytkowników,
- oznaczanie wybranych użytkowników jako “obserwowanych”, dzięki czemu ich nowe ogłoszenia będą dodatkowo eksponowane na liście dostępnych ogłoszeń,
- zgłaszanie ogłoszeń o nieodpowiedniej zawartości, co skutkuje powiadomieniem moderatora o konieczności weryfikacji zamieszczonej treści.

Aplikację wyróżnia nowoczesny interfejs graficzny, zaprojektowany specjalnie na potrzeby projektu.

Zawartość pracy

Rozdział 1 stanowi przegląd oraz analizę konkurencyjnych rozwiązań. W rozdziale 2 przedstawiono całościową architekturę systemu, zadania poszczególnych modułów oraz wykorzystane technologie. Rozdział 3 szczegółowo opisuje najważniejsze przypadki użycia aplikacji. Rozdział 4 poświęcony jest metodyce tworzenia projektu. Końcowe rozdziały zawierają: instrukcję zbudowania aplikacji, spis zawartości płyty dołączonej do pracy, podział zadań w zespole oraz krótkie podsumowanie projektu.

Rozdział 1

Przegląd konkurencyjnych rozwiązań

1.1. Sytuacja na rynku

Od paru lat można zaobserwować ciągły wzrost zainteresowania tematyką marnowania żywności wśród społeczeństwa. Przejawia się to na przykład w rosnącym zainteresowaniu polityków i władz, w tym Komisji Europejskiej (patrz [IDC]), bardzo licznym publikacjom na ten temat, a także w rosnącej popularności ruchów społecznych takich “freeganizm”, czy “zero waste”. Zainteresowanie tym tematem przejawia się też w licznych aplikacjach pomagających wymieniać się jedzeniem które inaczej by się zmarnowało. Co ciekawe w różnych państwach popularne są różne aplikacje. Na przykład w Wielkiej Brytanii najpopularniejsze jest *Olio*, w Zachodniej Europie *Too Good To Go*, a w Stanach Zjednoczonych “Unsung”. W Polsce nie ma jeszcze żadnej popularnej aplikacji, chociaż funkcjonują grupy na portalach społecznościowych spełniające podobną funkcję. W dalszej części tego rozdziału zostaną dokładniej omówione wybrane z konkurencyjnych rozwiązań.

1.2. Olio

Z wymienionych w tym rozdziale aplikacji *Olio* jest najbardziej podobne do *Podziel Się*. Posiada bardzo prosty interfejs, pozwalający już kilka chwil po zainstalowaniu dodać swoje pierwsze ogłoszenie, czy znaleźć z listy ogłoszeń (która może być sortowana po odległości, lub czasie dodania ogłoszenia) produkt na który mamy ochotę. *Olio* tak samo jak *Podziel Się* jest zintegrowane z Mapami Google i lokalizacją telefonu (choć można ją wyłączyć). *Olio* posiada też proste forum, które uznaliśmy za zbędne w takiej aplikacji, nie posiada natomiast podziału ogłoszeń na kategorie (np. wegańskie, bezglutenowe itp.), w *Olio* specyfikuje się tylko czy produkt jest żywnością czy nie.

Olio jest bardzo popularną aplikacją. Na chwilę obecną posiada ponad milion użytkowników i ponad półtora miliona ogłoszeń ([Olioex.com]). Jest najbardziej popularna w Wielkiej Brytanii, w Polsce praktycznie nie funkcjonuje.

1.3. Too Good To Go

Ta aplikacja reprezentuje zupełnie inne podejście niż *Podziel Się*. Służy do umożliwienia restauracjom i jadalniom do odsprzedaży niesprzedanych posiłków po znacznie niższych cenach niedługo przed zamknięciem. Oznacza, to że oferty są zwykle płatne, w przeciwieństwie do *Podziel Się*, gdzie wszystkie oferty są darmowe. *Too Good To Go* ma trochę bardziej skomplikowany interfejs niż *Olio*, czy *Podziel się*. Tak jak poprzednie aplikacje jest zintegrowana

z Mapami Google i lokalizacją telefonu, umożliwia też przeglądanie mapy z zaznaczonymi wszystkimi dostępnymi jadłodajniami, oraz dodawania jadłodajni do listy swoich ulubionych.

Too Good To Go jest popularna w praktycznie całej Zachodniej Europie, posiadając obecnie prawie 25.000 współpracujących jadłodajni ([toogoodtogo.com]). W Warszawie jest tylko 20 jadłodajni.

1.4. Grupa “Freeganism, dumpster diving, food sharing — Warszawa”

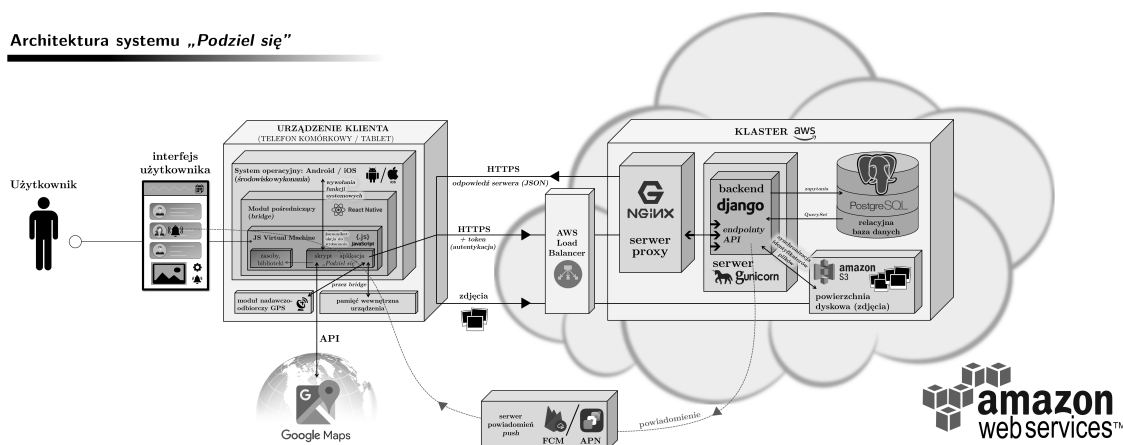
Jest to grupa funkcjonująca na portalu społecznościowym Facebook. Nie jest to co prawda aplikacja, jednak służy do tego samego celu i jest największą lokalną konkurencją. Grup tego rodzaju jest wiele zarówno w Polsce jak i w samej Warszawie, ta została wybrana jako przykładowa głównie ze względu na największą popularność, ponieważ w chwili obecnej liczy prawie 9 tysięcy członków i publikowanych na niej jest około 700 postów miesięcznie. Główny problem takich grup wynika z niedostosowania platformy — portalu Facebook. Nie posiada on narzędzi umożliwiających wygodne przeglądanie listy aktualnych ofert, nie pozwala filtrować, ani sortować ogłoszeń po odległości, ani nawet łatwo odfiltrować już wygasłe ogłoszenia. Bardzo ogranicza to rozrost takich grup. Z ciekawych rozwiązań z których ta grupa korzysta jest concept jadłodzielni. Są to zwykle ogólnodostępne lodówki i szafki na żywność w których można umieszczać produkty których się nie spożyje. Udostępniane są one zwykle przez różne jednostki jak: urząd miasta, uniwersytet, czy centrum kultury. Jest to szczególnie wygodne w odbiorze, ponieważ nie trzeba nawet wiedzieć, że ktoś ma coś do oddania, wystarczy jak się jest w pobliżu sprawdzić, czy nie ma w niej akurat czegoś na co ma się ochotę.

Rozdział 2

Implementacja

2.1. Architektora systemu

Architektora aplikacji *“Podziel się”* składa się z dwóch głównych komponentów: aplikacji mobilnej oraz części serwerowej. Wykorzystywane są również zewnętrzne usługi takie jak Firebase, Facebook, GoogleMaps, AmazonS3.



2.1.1. Aplikacja mobilna

Aplikacja *“Podziel się”* przeznaczona jest dla użytkowników dwóch najpopularniejszych mobilnych systemów operacyjnych: Androida oraz iOS. Interfejs aplikacji powstał z wykorzystaniem nowoczesnego frameworka React Native, pozwalającego opisać graficzne rozmieszczenie elementów na ekranie w składni zbliżonej do arkusza stylów CSS, zaś logikę działania aplikacji w kodzie JavaScriptu. W React Native, przy użyciu specjalnego języka znaczników JSX, zostały zdefiniowane tzw. komponenty (lista ogłoszeń, przycisk wysyłania wiadomości, menu szufladkowe, okienko z komunikatem itp.), na bazie których React Native buduje komponenty natywne dla Androida oraz iOS. Pozwala to na ujednolicenie wyglądu interfejsu graficznego dla użytkowników różnych systemów i architektur — aplikacja nie odbiega wyglądem od rozwiązań natywnych, projektowanych specjalnie z myślą o konkretnej platformie. Filozofia React Native polega na stosowaniu schematów tworzenia aplikacji webowych przy projektowaniu aplikacji działających w środowisku mobilnym, w szczególności możliwe jest korzystanie z rozmaitych pomocniczych bibliotek JavaScriptu.

Ważnym elementem React Native jest moduł pośredniczący (ang. *bridge*), który tłumaczy

wywołania pochodzące z wątków JavaScriptu na polecenia skierowane do systemu operacyjnego. Komunikaty opisujące akcje do wykonania są przesyłane w formacie JSON do modułu pośredniczącego, a po ich otrzymaniu *bridge* wywołuje odpowiednie funkcje systemowe. Wymiana komunikatów jest asynchroniczna oraz nieblokująca, zatem React Native nie ingeruje nadmiernie w pracę systemu operacyjnego oraz umożliwia sprawne renderowanie widoków na ekranie. Co więcej, aplikacja ma możliwość swobodnej komunikacji ze środowiskiem wykonania; w razie potrzeby może uzyskać dostęp do modułu GPS, aparatu, pamięci wewnętrznej urządzenia itp.

2.1.2. Część Serwerowa

Część serwerowa została stworzona w języku Python, w oparciu o framework Django. Pozwala on na szybkie tworzenie webowych interfejsów programistycznych aplikacji (po angielsku Web API) oraz łatwy dostęp do relacyjnych baz danych dzięki rozbudowanemu mechanizmowi mapowania obiektowo-relacyjnego (ORM). Polega ono na odwzorowaniu logicznych modeli encji, reprezentowanych jako klasy w języku Python, na rzeczywistą treść skryptów tworzących tabele i więzy spójności w bazie danych. Podobnie, wyniki zapytań są konwertowane do obiektu typu QuerySet, który może podlegać dalszemu przetwarzaniu przez serwer.

Jednym z głównych wymogów zamawiającej było stworzenie panelu administracyjnego. Wybrany framework umożliwia automatyczne generowanie panelu administratora. Wygenerowany panel umożliwia wykonywanie podstawowych operacji: pobierz, dodaj, usuń, modyfikuj na danych znajdujących się w bazie.

Do przechowywania danych wykorzystana została relacyjna baza danych PostgreSQL z wtyczką PostGIS, która pozwala na wydajne wykonywanie zapytań zapytań przestrzennych. Zdjęcia zapisywane są w usłudze AmazonS3.

2.1.3. Komunikacja Klient - Serwer

Wymiana danych pomiędzy aplikacją mobilną a serwerem odbywa się za pomocą bezstanowego protokołu HTTP. Dane przesyłane są w formacie JSON, jest on formatem tekstowym i bazuje na podzbiorze języka JavaScript. Każde zapytanie do serwera uwierzytelniane jest unikalnym tokenem przyznawanym w momencie logowania bądź rejestracji, w związku z tym hasło użytkownika nie jest przechowywane w pamięci telefonu. Przy każdym ponownym logowaniu generowany jest nowy token.

Dane z serwera do urządzenia mobilnego (powiadomienia) wysyłane są natomiast za pośrednictwem usługi Firebase.

2.2. PostGIS

Jedną z funkcjonalności naszej aplikacji jest prezentowanie użytkownikowi ofert z adresem odbioru w pobliżu jego aktualnej lokalizacji. Żeby to osiągnąć musieliśmy po stronie serwera być w stanie wyznaczyć wszystkie ogłoszenia z bazy danych, które są oddalone od zadanej lokalizacji nie więcej niż pewna wartość otrzymana w zapytaniu użytkownika. Pierwszym podejściem do rozwiązania tego problemu było wyznaczanie dla każdego zapytania wszystkich odległości pomiędzy adresem odbioru ogłoszeń a otrzymaną lokalizacją i wybranie tych, które są bliżej niż maksymalna odległość, którą wybrał użytkownik. Byłoby to jednak rozwiązanie bardzo nieefektywne, ponieważ dla każdego zapytania wymaga przejrzania wszystkich ofert z bazy co przy większej ilości użytkowników i ofert może powodować problemy z wydajnością całego systemu.

Rozwiązaniem, na które ostatecznie się zdecydowaliśmy jest skorzystanie z rozszerzenia bazy danych PostgreSQL o nazwie PostGIS. Pozwala ono na wprowadzanie danych geograficznych do bazy i ich odpowiednią obsługę. Dla każdego ogłoszenia w bazie danych zapisujemy współrzędne geograficzne miejsca odbioru, dzięki czemu odpowiednia funkcja rozszerzenia PostGIS pozwala na znalezienie wszystkich ofert w ustalonej odległości od lokalizacji użytkownika. Operacja ta jest efektywna dlatego, że wyżej wspomniane rozszerzenie zmienia sposób przechowywania danych geograficznych w porównaniu do standardowej bazy danych. Zamiast B-drzewa używa R-drzewa, które jest strukturą danych wspomagającą wyszukiwanie obiektów w przestrzeni wielowymiarowej. Dla rzeczywistych danych to rozwiązanie jest dużo szybsze od podejścia naiwnego przedstawionego powyżej.

2.3. Przechowywanie zdjęć

W trakcie tworzenia systemu, musieliśmy wybrać sposób przechowywania zdjęć. Pierwszym pomysłem było zapisywanie zdjęć na lokalnym dysku serwera. Szybko zdaliśmy sobie sprawę, że takie rozwiązanie posiada szereg wad. Pierwszą z nich jest brak elastyczności, w przypadku wyczerpania miejsca bylibyśmy zmuszeni zapłacić za droższy serwer, nawet jeżeli nie potrzebowalibyśmy szybszego procesora czy większej pamięci ram. Kolejną wadą jest duże obciążenia serwera, który musiałby obsługiwać zapytania HTTP związane z transferem tych zdjęć na urządzenia mobilne.

Zdecydowaliśmy skorzystać z serwisu Amazon Simple Storage Service (Amazon S3). Usługa pozwala przechowywać dane w sposób trwały i bezpieczny. Zaletą Amazon S3 jest nieograniczona pojemność dyskowa oraz brak narzuconych limitów na transfer wychodzący. Co więcej, koszt usługi jest proporcjonalny do sumarycznego rozmiaru przechowywanych danych, mamy więc pełną elastyczność jeśli chodzi o koszty. Transfer plików jest szyfrowany protokołem SSL, ponadto same dane podlegają 256-bitowemu szyfrowaniu po stronie serwera.

Obsługa zdjęć zrealizowana jest w następujący sposób: Część mobilna kompresuje fotografie przed wysłaniem, następnie wysyła je w jednym zapytaniu wraz z informacjami o ogłoszeniu do serwera aplikacji *“Podziel się”*. Serwer zapisuje dane ogłoszenia i metadane zdjęć w bazie PostgreSQL, a następnie wysyła obrazy do usługi Amazon S3. Przy pobieraniu informacji o dostępnych ogłoszeniach lub szczegółach danego ogłoszenia serwer zwraca odnośniki do zdjęć, następnie urządzenie mobilne pozyskuje dane bezpośrednio z serwisu Amazon S3. W ten sposób w znacznym stopniu zredukowane jest obciążenie serwera aplikacji *“Podziel się”*.

W dalszej perspektywie, w miarę rozwoju projektu, Amazon S3 oferuje rozszerzone plany wykorzystania oraz opcję archiwizacji danych z możliwością łatwego odtworzenia w przypadku nadpisania, usunięcia lub utracenia plików.

2.4. Powiadomienia

Aplikacja *“Podziel się”* wspiera użytkowników w zarządzaniu licznymi ogłoszeniami wykorzystując mechanizm powiadomień. Wspomniana funkcjonalność pozwala na wyświetlanie użytkownikom krótkich komunikatów o zajściu zdarzeń wymagających uwagi.

Powiadomienia wyświetlane są w przypadku zająścia jednego z następujących zdarzeń:

- otrzymanie wiadomości od innego użytkownika,
- dodanie nowego ogłoszenia w okolicy,
- dodanie nowego ogłoszenia przez obserwowaną osobę
- polubienie naszego ogłoszenia,

Najprostszym sposobem implementacji tej funkcjonalności jest regularne odpytywanie serwera czy są dostępne nowe powiadomienia. Jednakże ze względu na ograniczenia systemu Android jak i iOS, obsługa procesów w tle (to znaczy wtedy kiedy użytkownik nie korzysta z aplikacji) jest bardzo mocno ograniczona. Procesy w tle nie mogą być uruchamiane częściej niż co kilkadziesiąt minut. Związane jest to z oszczędzaniem energii.

Z tego względu zdecydowaliśmy się na użycie zewnętrznych usług takich jak: Firabase Cloud Messaging (FCM), dla systemu Androida oraz Apple Push Notification (APN) w przypadku iOS. Umożliwiają one niemalże natychmiastowe dostarczanie wiadomości z serwera do telefonu użytkownika.

Schemat działania systemu powiadomień wygląda następująco: urządzenie mobilne przy pierwszym uruchomieniu pobiera unikalny identyfikator (token) z serwisu FCM bądź APN, jest on wysyłany na serwer aplikacji *“Podziel się”* w trakcie logowania lub rejestracji, a następnie zapisywany w bazie danych. W przypadku potrzeby dostarczenia powiadomienia na dane urządzenie, serwer aplikacji *“Podziel się”* wyśle do serwera FCM wiadomość zawierającą token danego telefonu oraz zawartość powiadomienia. Usługa FCM przekieruje notyfikację do konkretnego użytkownika na podstawie danego identyfikatora urządzenia. Zachowanie telefonu lub tabletu po odebraniu nowego powiadomienia zależy od preferencji użytkownika, może to być okienko z wiadomością, sygnał dźwiękowy lub ikonka na ekranie.

Udało nam się zaimplementować w pełni funkcjonalne rozwiązanie na platformie Android. Jeśli chodzi o system iOS to wymogiem usługi APN było posiadanie konta w programie *Apple Developer Program*. Niestety ze względów czasowych i formalnych nie byliśmy w stanie przystąpić do tego programu. Jednakże zarówno aplikacja mobilna jak i część serwerowa są już przystosowane do obsługi APN, wystarczy jedynie odpowiednio skonfigurować platformę Firebase (wgrać odpowiednie certyfikaty wygenerowane na koncie dewelopera Apple), wtedy powiadomienia do użytkowników Apple będą przekierowywane do APN.

2.5. Logowanie za pomocą Facebooka

2.6. Działanie aplikacji mobilnej na dwóch systemach operacyjnych

System operacyjny Android jest niezwykle popularny w naszym kraju. Według IDC 91,5% sprzedanych smartfonów w 2017 roku stanowiły urządzenia działające pod kontrolą systemu firmy Google.

Z tego względu postanowiliśmy, że w pierwszej kolejności tworzymy aplikację na platformę Android, a dopiero później zajmiemy się systemem iOS. W trakcie pracy nad wersją androidową używaliśmy tylko i wyłącznie bibliotek, których twórcy wyraźnie informowali o kompatybilności z obydwoma systemami. Pozwoliło nam to dosyć bezboleśnie uruchomić aplikację na symulatorze iPhone.

Nie udało nam się jedynie zaimplementować dwóch funkcjonalności:

Pierwszą z nich jest logowanie z Facebookiem. Pomimo tego, że twórcy biblioteki z której skorzystaliśmy, zapewniali o działaniu na platformie iOS, to nie udało nam się odpowiednio jej skonfigurować do działania z naszą aplikacją.

Drugą są powiadomienia. Tak jak pisaliśmy we wcześniejszym podrozdziale, wymogiem było posiadanie konta w programie *Apple Developer Program*.

Cała pozostała funkcjonalność działa dokładnie tak jak na systemie firmy Google. W szczególności, obie wersje mają taki sam interfejs użytkownika. Różni on się jedynie szczegółami, typowymi dla danej platformy. Na przykład komponentami wyboru daty lub zdjęć z

galerii. Poniżej zrzuty tych samych ekranów aplikacji, wykonane odpowiednio na emulatorze Android i symulatorze iPhone:



(a) Android.



(b) iPhone.

Rysunek 2.1: Lista ogłoszeń.

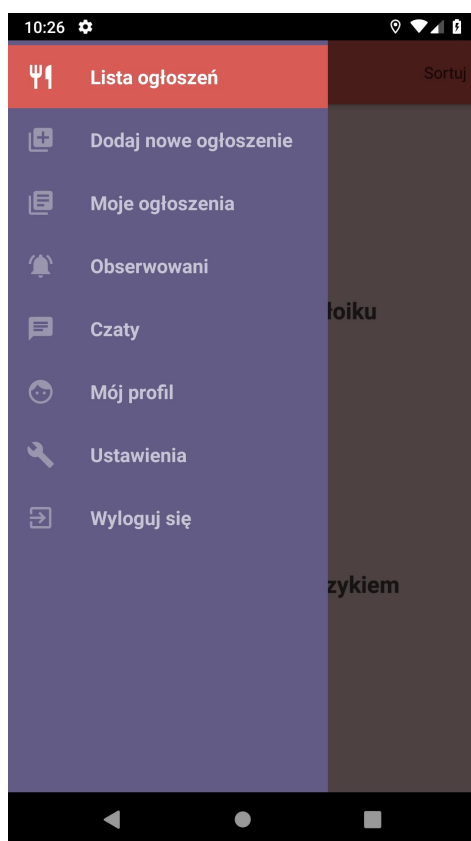


(a) Android.

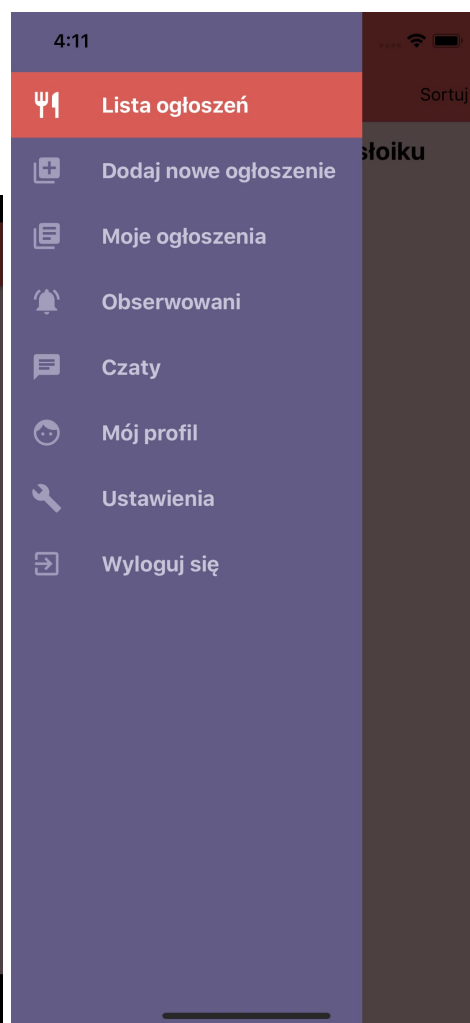


(b) iPhone.

Rysunek 2.2: Szczegóły ogłoszenia.



(a) Android.



(b) iPhone.

Rysunek 2.3: Główne menu.

2.7. Infrastruktura serwera

Serwer projektu został wdrożony na platformie AWS (Amazon Web Services). Decyzja o wybraniu chmury AWS została podjęta z uwagi na jej wydajność, skalowalność, stosunkowo niską awaryjność oraz obecność rozbudowanych mechanizmów zapewniających bezpieczeństwo danych. AWS udostępnia widoczny globalnie w Internecie klastery komputerów w chmurze obliczeniowej. Wspomniane klastry emulują zachowanie prawdziwego komputera, począwszy od wirtualizacji sprzętu (procesora, pamięci RAM, dysku twardego), poprzez infrastrukturę sieciową i system operacyjny, skończywszy na wirtualnej konsoli umożliwiającej połączenie się oraz sterowanie klastrem z poziomu przeglądarki. Wirtualne maszyny AWS udostępniają gotowe rozwiązania serwerowe i oprogramowanie do specjalistycznych zastosowań.

Część serwerowa aplikacji działa pod kontrolą serwera HTTP Gunicorn. Gunicorn odpowiada za wykonanie odpowiedniego kodu języka Python oraz współpracę z serwerem pośredniczącym nginx. Nginx odbiera połączenia od klientów oraz obsługuje statyczne zapytania HTTP, zaś dynamiczne przekierowuje do Gunicorna i Django z wykorzystaniem standardu WSGI (Web Server Gateway Interface). Po otrzymaniu odpowiedzi nginx odsyła klientowi wynik zapytania. Wprowadzenie serwera proxy pozwala na łatwiejszą regulację obciążenia oraz zapobieganie opóźnieniom wynikającym z konieczności obsługi klientów dysponującym powolnym połączeniem internetowym.

Rozdział 3

Przypadki użycia

Zaprezentowane są tu przykładowe przypadki użycia.

3.1. Rejestracja

3.1.1. Krótki opis

Utworzenie konta użytkownika w aplikacji.

3.1.2. Cele

Umożliwienie aplikacji identyfikacji użytkowników w celu dostarczenia im określonych funkcjonalności.

3.1.3. Warunki wstępne

1. Aplikacja jest zainstalowana na urządzeniu użytkownika.
2. Użytkownik dysponuje stałym połączeniem internetowym.

3.1.4. Czynności

Czynności podstawowe

1. Użytkownik naciska przycisk “Załącz konto”.
2. Aplikacja wyświetla użytkownikowi formularz rejestracyjny.
3. Użytkownik wprowadza swoje imię, nazwisko (lub nazwę firmy), numer telefonu, adres e-mail, hasło.
4. W polu “Potwierdź hasło” użytkownik wpisuje drugi raz to samo hasło.
5. Użytkownik akceptuje regulamin.
6. Użytkownik naciska przycisk “Zarejestruj”.
7. Aplikacja potwierdza poprawną rejestrację.
8. Użytkownik jest domyślnie zalogowany do aplikacji na danym urządzeniu.
9. Użytkownik może nacisnąć przycisk “Wyloguj”, aby się wylogować.

Czynności alternatywne

1. Użytkownik:
 - wprowadza drugi raz inne hasło.
 - nie wypełnia któregoś z pól.
 - wprowadza niepoprawny składniowo adres e-mail.
 - nie akceptuje regulaminu.
2. Użytkownik naciska przycisk “Zarejestruj”.
3. Aplikacja powiadamia użytkownika o popełnionym błędzie i odsyła go z powrotem do edycji formularza.

3.2. Dodawanie nowego ogłoszenia

3.2.1. Krótki opis

Wprowadzenie przez użytkownika informacji potrzebnych do stworzenia nowego ogłoszenia oraz opublikowanie go na liście aktualnie dostępnych ogłoszeń.

3.2.2. Cele

Zapewnienie możliwości przeglądania nowo utworzonego ogłoszenia innym użytkownikom aplikacji.

3.2.3. Warunki wstępne

- Aplikacja jest zainstalowana na urządzeniu użytkownika.
- Użytkownik posiada poprawnie założone konto.
- Użytkownik jest zalogowany w aplikacji.
- Użytkownik dysponuje stabilnym połączeniem internetowym.

3.2.4. Czynności

Czynności podstawowe

1. Użytkownik naciska oznaczony przycisk, który rozwija szufladkowe menu.
2. Użytkownik wskazuje w menu pozycję “Dodaj nowe ogłoszenie”.
3. Na ekranie pojawia się formularz dodawania ogłoszenia z podpisanymi polami.
4. Użytkownik wprowadza (w dowolnej kolejności) wymagane informacje:
 - Kategorie ogłoszenia (np. pieczywo, mięso, przetwory, wyroby cukiernicze. . .)
 - Użytkownik rozwija listę dostępnych kategorii.
 - Użytkownik wskazuje kategorie, które wydają się najbardziej odpowiednie do zawartości ogłoszenia.
 - Tytuł ogłoszenia, który będzie widoczny na liście ogłoszeń

- Właściwa treść ogłoszenia
 - Sugerowana data przydatności do spożycia
 - Użytkownik otwiera okienko kalendarza z graficznym rozkładem dni bieżącego miesiąca.
 - W razie potrzeby użytkownik przechodzi za pomocą przycisków-strzałek do dalszych miesięcy.
 - Użytkownik wskazuje konkretny dzień w kalendarzu.
 - Preferowane miejsce odbioru (z rozwijanej listy uprzednio wgranych miejsc odbioru)
5. Użytkownik przesuwając widok ekranu na koniec formularza i naciskając przycisk “Wyślij”.
 6. Aplikacja komunikuje się z serwerem, wysyłając prośbę o dodanie wprowadzonego ogłoszenia do bazy danych.
 7. Aplikacja otrzymuje odpowiedź od serwera — ogłoszenie zostało dodane pomyślnie.
 8. Użytkownik dostaje informację zwrotną: aplikacja wyświetla na ekranie okienko o pomyślnym dodaniu nowego ogłoszenia.

Czynności alternatywne

Użytkownik może opcjonalnie dodać zdjęcia artykułu żywnościowego, opakowania, etykiet.

1. Użytkownik wskazuje przycisk “Dodaj nowe zdjęcie”.
2. Aplikacja wyświetla okienko z prośbą o wskazanie źródła zdjęcia, do wyboru: załączenie zdjęcia z pamięci urządzenia (galerii) lub utworzenie aparatem nowego zdjęcia.
 - (a) Jeżeli użytkownik wyrazi chęć dodania istniejącego zdjęcia, aplikacja uruchamia klasyczny interfejs przeglądania galerii systemu operacyjnego.
 - (b) Wybór opcji zrobienia nowego zdjęcia przekazuje sterowanie wbudowanej w system operacyjny aplikacji obsługującej aparat.
3. Jeżeli zdjęcie ma zbyt duży rozmiar, następuje jego kompresja (po stronie aplikacji).
4. Na ekranie formularza pojawia się miniaturka dodanego zdjęcia.

Czynności niepoprawne

We wszystkich wymienionych poniżej przypadkach na ekranie wyświetla się (jaskrawym czerwonym kolorem) stosowny komunikat. Użytkownik zostaje przekierowany z powrotem do ekranu edycji formularza, aby mógł wprowadzić zmiany.

1. Użytkownik nie wprowadził kompletu wymaganych informacji.
2. Wartości niektórych pól formularza nie spełniają kryteriów akceptacji (np. zbyt długa treść ogłoszenia)
3. Nastąpił błąd komunikacji z serwerem — prośba o dodanie ogłoszenia nie została rozpatrzona.
4. Serwer zasygnalizował błąd podczas operacji dodawania ogłoszenia do bazy danych.

3.2.5. Możliwe rozwinięcia

Automatyczna klasyfikacja ogłoszeń

Wyświetlenie podpowiedzi co do wyboru odpowiedniej kategorii na podstawie analizy słownictwa użytego w treści ogłoszenia.

3.3. Konwersacja (czat) z innym użytkownikiem

3.3.1. Krótki opis

Wykorzystanie funkcji czatu do komunikacji z innym użytkownikiem aplikacji.

3.3.2. Cele

- Możliwość zadawania wystawcy ogłoszenia dodatkowych pytań.
- Udostępnienie prywatnego kanału do ustalania szczegółów odbioru produktu.
- Zachęcenie użytkowników aplikacji do współtworzenia aktywnej społeczności.

3.3.3. Warunki wstępne

- Aplikacja jest zainstalowana na urządzeniu użytkownika.
- Użytkownik posiada poprawnie założone konto.
- Użytkownik jest zalogowany w aplikacji.
- Użytkownik dysponuje stabilnym połączeniem internetowym.

3.3.4. Czynności

Czynności podstawowe

1. Użytkownik A otwiera ekran czatu z użytkownikiem B. Może to zrobić na kilka sposobów:
 - Bezpośrednio ze strony profilowej użytkownika B.
 - Naciskając przycisk “Zadaj pytanie” widocznego na ekranie ogłoszenia wystawionego przez B.
 - Wybierając pseudonim użytkownika B z zakładki “Moje konwersacje” w menu szufladkowym.
2. W dolnej części ekranu pojawia się pole tekstowe do wpisania wiadomości oraz przycisk służący do jej wysłania. Jeżeli użytkownicy A i B rozmawiali ze sobą wcześniej, na ekranie będzie także widoczna przewijana lista poprzednich wiadomości, uporządkowana chronologicznie.
3. Użytkownik A wpisuje treść wiadomości i wysyła ją.
4. Nowa wiadomość zostaje dopisana do prowadzonej historii konwersacji.
5. Użytkownik A oczekuje na odpowiedź użytkownika B:

- Jeżeli osoba B udzieli odpowiedzi od razu, jej treść pojawi się na ekranie użytkownika A w czasie rzeczywistym.
- W przypadku, gdy użytkownik A nie korzysta w danej chwili z aplikacji “Podziel się”, oraz korzysta z systemu Android, otrzyma powiadomienie (notyfikację) systemu operacyjnego o nowej wiadomości od użytkownika B.

Czynności niepoprawne

We wszystkich wymienionych poniżej przypadkach na ekranie wyświetla się okienko ze stosownym komunikatem.

1. Osoba A znajduje się na “czarnej liście” użytkownika B (tzn. B zablokował możliwość interakcji z użytkownikiem A)
2. Nastąpił błąd komunikacji z serwerem — nie udało się wysłać wiadomości.
3. Serwer zasygnalizował błąd podczas dopisywania wiadomości do historii konwersacji.

3.3.5. Możliwe rozwinięcia

Przesyłanie obrazów

Umożliwienie przesyłania wiadomości w formie zdjęć. W tym celu użytkownik wysyłający wiadomość wybiera piktogram “prześlij obraz”. Dalszy scenariusz jest analogiczny do opisanego w punkcie 5.4.2.

3.4. Ocena innego użytkownika

3.4.1. Krótki opis

Wystawienie innemu użytkownikowi oceny w skali od 1 do 5 oraz możliwość napisania krótkiego komentarza dotyczącego współpracy z nim.

3.4.2. Cele

Informowanie innych osób o jakości współpracy z daną osobą, w szczególności ostrzeganie innych przed nierzetelnymi użytkownikami.

3.4.3. Warunki wstępne

- Aplikacja jest zainstalowana na urządzeniu użytkownika.
- Użytkownik posiada poprawnie założone konto.
- Użytkownik jest zalogowany w aplikacji.
- Użytkownik dysponuje stabilnym połączeniem internetowym.

3.4.4. Czynności

Czynności podstawowe

1. Użytkownik A otwiera ekran, na którym znajduje się publiczny profil użytkownika B. Może to zrobić na kilka sposobów:
 - W oknie konwersacji z użytkownikiem B, poprzez kliknięcie na jego imię lub zdjęcie w lewym górnym rogu ekranu.
 - Naciskając przycisk “Pokaż profil użytkownika” na ekranie ogłoszenia, które użytkownik B wystawił.
 - Wpisując: imię, nazwisko lub nazwę użytkownika B w polu tekstowym znajdującym się na samej górze ekranu prostej wyszukiwarki.
2. Wyświetla się ekran z profilem użytkownika B, a w nim następujące elementy:
 - Imię, nazwisko oraz nazwa użytkownika.
 - Zdjęcie profilowe.
 - Zbiorcza ocena innych użytkowników w skali od 1 do 5.
 - Przycisk “Rozpocznij chat”.
 - Przycisk “Dodaj ocenę”.
 - Lista z ocenami i komentarzami innych osób.
3. Użytkownik A naciska przycisk “Dodaj ocenę”.
4. Wyświetla mu się ekran z formularzem dodawania oceny.
5. Użytkownik A ocenia użytkownika B w skali od 1 do 5.
6. Użytkownik A wpisuje krótki komentarz odnośnie użytkownika B.
7. Użytkownik A naciska przycisk “Dodaj ocenę”.

Czynności alternatywne

Użytkownika A może co najwyżej raz ocenić użytkownika B. Jeśli miało już to miejsce, użytkownik A może edytować swoją opinię:

1. Zamiast przycisku “Dodaj ocenę” na ekranie z profilem użytkownika B będzie widoczny przycisk “Edytuj ocenę”.
2. Formularz dodawania oceny będzie uzupełniony o poprzednią ocenę i komentarz.
3. Użytkownik po edycji formularza klika przycisk “Zapisz ocenę”.

Czynności niepoprawne

We wszystkich wymienionych poniżej przypadkach na ekranie wyświetla się (jaskrawym czerwonym kolorem) stosowny komunikat. Użytkownik zostaje przekierowany z powrotem do ekranu edycji formularza, aby mógł wprowadzić zmiany.

1. Nastąpił błąd komunikacji z serwerem — prośba o dodanie / edycję opinii nie została rozpatrzona.
2. Serwer zasygnalizował błąd podczas operacji dodawania / edycji opinii do bazy danych.

3.4.5. Możliwe rozwinięcia

Blokada możliwości oceny danego użytkownika do momentu, w którym odbierzemy od niego (lub on od nas) pożywienie z przynajmniej jednego ogłoszenia.

3.5. Odbiór przedmiotów oferowanych w liście ogłoszeń

3.5.1. Krótki opis

Umożliwienie użytkownikowi, który wystawił ogłoszenie (A) i użytkownikowi, który jest zainteresowany odbiorem przedmiotu z ogłoszenia (B) ustalenia szczegółów odbioru.

3.5.2. Cele

- Umożliwienie użytkownikom ustalenia szczegółów odbioru przedmiotu (np. czas odbioru, miejsce odbioru).
- Umożliwienie użytkownikowi zarządzania statusem swoich ogłoszeń.

3.5.3. Warunki wstępne

- Aplikacja jest zainstalowana na urządzeniach użytkowników A i B.
- Użytkownicy posiadają poprawnie założone konta.
- Użytkownicy są zalogowani w aplikacji.
- Użytkownik B dodał ogłoszenie.
- Użytkownicy dysponują stabilnym połączeniem internetowym.

3.5.4. Czynności

Czynności podstawowe

Użytkownik A

1. Użytkownik naciska oznaczony przycisk “Szczegóły ogłoszenia” przy ogłoszeniu na liście ogłoszeń.
2. Na ekranie ze szczegółami ogłoszenia użytkownik naciska przycisk “Zadaj pytanie”.
3. Użytkownikowi otwiera się ekran czatu z autorem ogłoszenia.
4. Użytkownik wysyła wiadomość, w której wyraża chęć odbioru przedmiotu z ogłoszenia.
5. Jeżeli użytkownik otrzyma odpowiedź pozytywną, użytkownicy ustalają między sobą szczegóły odbioru.

Użytkownik B

1. Użytkownik otrzymuje nową wiadomość na czacie z chęcią odbioru oferowanego przedmiotu.
2. Użytkownik decyduje, czy chce oddać oferowany produkt użytkownikowi A
 - (a) Jeżeli użytkownik nie chce oddać produktu użytkownikowi A, wysyła wiadomość odmowną i interakcja się kończy.
 - (b) Jeżeli użytkownik chce oddać produkt, wysyła wiadomość pozytywną, interakcja trwa dalej.
3. Na ekranie szczegółów swojego ogłoszenia użytkownik naciska przycisk “Oznacz jako zarezerwowane”
4. Użytkownicy ustalają między sobą szczegóły odbioru.
5. Po oddaniu oferowanego przedmiotu, użytkownik naciska przycisk “Oznacz jako oddane” na ekranie szczegółów swojego ogłoszenia, przez co ogłoszenie znika z listy ogłoszeń.

Czynności alternatywne

Jeżeli po oznaczeniu ogłoszenia jako zarezerwowane, użytkownik A albo B zmieni zdanie co do chęci odbioru/oddania produktu, użytkownik B może odznaczyć ogłoszenie jako zarezerwowane poprzez ponowne kliknięcie w przycisk “oznacz jako zarezerwowane” na ekranie szczegółów swojego ogłoszenia.

Czynności niepoprawne

We wszystkich wymienionych poniżej przypadkach na ekranie wyświetla się (jaskrawym czerwonym kolorem) stosowny komunikat.

1. Czynności niepoprawne z punktu 6.4.2 dotyczące funkcjonalności czatu
2. Nastąpił błąd komunikacji z serwerem — prośba o oznaczenie jako zarezerwowane / oddane nie została rozpatrzona.
3. Serwer zasygnalizował błąd podczas operacji związanych z oznaczeniem ogłoszenia jako zarezerwowane / oddane.

3.5.5. Możliwe rozwinięcia

Podczas oznaczania ogłoszenia jako zarezerwowane użytkownik może oznaczyć, dla którego użytkownika dokonywana jest rezerwacja.

Rozdział 4

Zarządzanie projektem

4.1. Zbieranie wymagań

4.2. Podział obowiązków

Rozdział 5

Instrukcja budowania aplikacji i zawartość płytki

Rozdział 6

Podsumowanie

Bibliografia

- [FAO] raport Organizacji Narodów Zjednoczonych do spraw Wyżywienia i Rolnictwa
“*Global food losses and food waste*”, fao.org/3/mb060e/mb060e.pdf
- [Eurostat] raport techniczny Komisji Europejskiej z 2010 roku
“*Preparatory Study on Food Waste Across EU 27*”,
ec.europa.eu/environment/eussd/pdf/bio_foodwaste_report.pdf
- [Kantar Millward Brown] raport Federacji Polskich Banków Żywności “*Nie marnuję jedzenia 2018*” z wynikami badań zrealizowanych przez Kantar Millward Brown,
bzsos.pl/2018/raportu-nie-marnuj-jedzenia-2018
- [Polityka] opracowanie “*Zgubione kalorie. Jak skutecznie walczyć z marnotrawieniem żywności*” dla magazynu Polityka Insight,
politykainsight.pl/gospodarka/ryzykaitrendy/_resource/multimediu/20145149
- [podzielmysie.pl] oficjalna strona akcji społecznej “*Podziel się Posiłkiem z Bezdomnymi*”,
podzielmysie.pl
- [reducefoodwaste.eu] oficjalna strona internetowa projektu Reduce Food Waste,
reducefoodwaste.eu
- [Olioex.com] Statystyki dotyczące wykorzystania żywności na świecie z oficjalnej strony aplikacji *Olio*, olioex.com
- [toogoodtogo.com] Oficjalna strona aplikacji *Too Good To Go*, toogoodtogo.com.
- [ec.europa.eu] Działania KE ws. marnowania żywności
https://ec.europa.eu/food/safety/food_waste_en
- [IDC] Popularność systemu Android w Polsce
<https://www.telepolis.pl/wiadomosci/prawo-finanse-statystyki/w-2017-roku-w-polsce-sprzedano-8-8-mln-smartfonow>
- [ReactNative] facebook.github.io/react-native
- [GoogleMaps] developers.google.com/maps/documentation
- [Firebase] firebase.google.com/docs/cloud-messaging
- [Notifications] developer.apple.com/notifications
- [AWS] aws.amazon.com
- [Django] djangoproject.com

[Gunicorn] gunicorn.org

[NginX] nginx.com

[PostgreSQL] postgresql.org