

Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **„LOKALizator” – WebAplikacja do ogłoszeń lokalnych**
Autorzy: **Przemysław Niemiec, Patryk Surmacz**
Grupa: **I1-210A**
Kierunek: **informatyka**
Rok akademicki: **...**
Poziom i semestr: **I/4**
Tryb studiów: **stacjonarne/niestacjonarne**

Należy pozostawić wszelkie nagłówki tego dokumentu, a umieszczać treść w odpowiednich miejscach zamiast obecnych objaśnień.

Stronę tytułową można sformatować w dowolny sposób, ale należy pozostawić zawartość informacyjną w układzie pokazanym powyżej.

Praca powinna zostać złożona wyłącznie w formacie pdf. Przed wygenerowaniem ostatecznej wersji należy zaktualizować spis treści – wyświetlane dwa poziomy.

Niniejszą informację należy również usunąć z wersji końcowej.

1 Spis treści

2	Odnośniki do innych źródeł.....	4
3	Słownik pojęć	5
4	Wprowadzenie	6
4.1	Cel dokumentacji.....	6
4.2	Przeznaczenie dokumentacji	6
4.3	Opis organizacji lub analiza rynku.....	6
4.4	Analiza SWOT organizacji	6
5	Specyfikacja wymagań	7
5.1	Charakterystyka ogólna.....	7
5.2	Wymagania funkcjonalne.....	8
5.3	Wymagania нефunkcjonalne.....	13
6	Zarządzanie projektem	15
6.1	Zasoby ludzkie	15
6.2	Harmonogram prac.....	15
6.3	Etapy/kamienie milowe projektu	15
7	Zarządzanie ryzykiem.....	17
7.1	Lista czynników ryzyka	17
7.2	Ocena ryzyka.....	17
7.3	Plan reakcji na ryzyko	17
8	Zarządzanie jakością.....	18
8.1	Scenariusze i przypadki testowe	18
9	Projekt techniczny	19
9.1	Opis architektury systemu.....	19
9.2	Technologie implementacji systemu.....	19
9.3	Diagramy UML	19
9.4	Charakterystyka zastosowanych wzorców projektowych.....	24
9.5	Projekt bazy danych	24
9.6	Projekt interfejsu użytkownika.....	24
9.7	Procedura wdrożenia	25
10	Dokumentacja dla użytkownika.....	26
11	Podsumowanie	27
11.1	Szczegółowe nakłady projektowe członków zespołu	27
12	Inne informacje	28

Odnosniki do innych źródeł

tj. do wykorzystywanych narzędzi / projektów w tych narzędziach

- Zarządzania projektem – Jira, Trello, itp.
- Wersjonowanie kodu – sugerowany Git (hosting np. na Bitbucket lub Github), ew. SVN
- System obsługi defektów – np. Bitbucket, Github, Bugzilla.

2 Słownik pojęć

Tabela lub lista z pojęciami, które wymagają wyjaśnienia, wraz z tymi wyjaśnieniami – w szczególności synonimy różnych pojęć używanych w dokumentacji.

3 Wprowadzenie

3.1 Cel dokumentacji

po co ją robimy i co zawiera (poziom szczegółowości)

Przeznaczenie dokumentacji

dla kogo ona jest

Opis organizacji lub analiza rynku

Jedna z dwóch opcji:

1. Jeśli dla konkretnej organizacji: Czym jest organizacja, dla której realizowany będzie system; jak działa (lub będzie działała), kiedy system będzie wdrożony – tutaj nie odwołujemy się do samego systemu, tylko opisujemy samo działanie organizacji i role. W szczególności – jak wyglądają główne procesy biznesowe.
2. Jeśli na masowy rynek: Pobieżna analiza rynku. Dla kogo będzie przydatny taki system. Ile jest organizacji, które będą mogły z niego skorzystać, użytkowników w poszczególnych organizacjach. Czy te organizacje stanowią jednorodną grupę czy są różne rodzaje. Co one mają ze sobą wspólnego. Jak ta liczba będzie się zmieniała w najbliższej przyszłości.

Analiza SWOT organizacji

- jeśli system dla konkretnej organizacji:
 - wystarczy sama tabela 2x2 (silne-słabe-szanse-zagrożenia)
- jeśli system na masowy rynek:
 - szanse i zagrożeniaC

4 Specyfikacja wymagań

4.1 Charakterystyka ogólna

4.1.1 Definicja produktu

"Lokalizator" – system internetowy do publikacji i przeglądania lokalnych ogłoszeń.

4.1.2 Podstawowe założenia

Serwis "Lokalizator" został zaprojektowany, aby służyć jako centralne miejsce dla mieszkańców danej lokalizacji do publikowania i przeglądania ogłoszeń lokalnych. Serwis wzorowany jest na facebookowych stronach typu „Spotted”. Będzie więc zawierał zarówno „luźne” ogłoszenia, pozdrowienia czy pytania, jak również oferty kupna, sprzedaży czy usług. Platforma ma również na celu integrację lokalnej społeczności i promocję lokalnej gospodarki poprzez umożliwienie bezpośredniego kontaktu między mieszkańcami. Ogłoszenia będzie można publikować zarówno anonimowo (na wzór „Spotted”) jak i pod własnym nazwiskiem.

4.1.3 Cel biznesowy

Celem biznesowym "Lokalizatora" jest stworzenie stabilnej i dochodowej platformy ogłoszeniowej, która zdobędzie popularność i zaufanie w lokalnych społecznościach. System ma na celu przyciągnięcie dużej liczby użytkowników dzięki swojej użyteczności, prostocie i dostępności, co w konsekwencji ma przekładać się na zyski z reklam oraz potencjalnych usług premium.

4.1.4 Użytkownicy

- Użytkownicy anonimowi
- Indywidualni Sprzedawcy: Osoby prywatne chcące sprzedać lub wynająć swoje rzeczy lub usługi.
- Kupujący: Osoby szukające produktów lub usług do zakupu lub wynajmu.
- Firmy lokalne: Przedsiębiorstwa zainteresowane reklamowaniem swoich produktów i usług lokalnie

4.1.5 Korzyści z systemu

- Anonimowi użytkownicy (ID-00): Łatwość i szybkość dodawania ogłoszeń
- Indywidualni sprzedawcy (ID-01): Dostęp do klientów znajdujących się w tym samym mieście, co zwiększa szanse na szybką sprzedaż.
- Kupujący (ID-02): Dostęp do szerokiej gamy lokalnych ofert w jednym miejscu, co ułatwia porównywanie i wybieranie najlepszych opcji.
- Firmy lokalne (ID-03): Możliwość skutecznej reklamy w lokalnym środowisku, co zwiększa rozpoznawalność marki i przyciąga nowych klientów.

4.1.6 Ograniczenia projektowe i wdrożeniowe

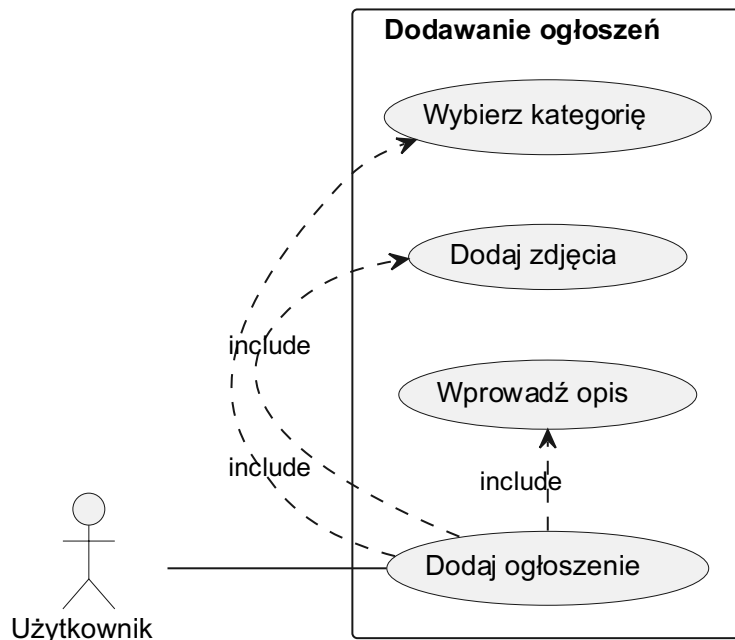
Projekt musi być zgodny z RODO w zakresie ochrony danych osobowych, co wymaga implementacji odpowiednich mechanizmów bezpieczeństwa i zasad prywatności. System musi być również dostosowany do obsługi na różnych platformach (desktop, mobile), co wymaga responsywnego projektowania interfejsów. Wymagane jest także zapewnienie wysokiej dostępności i odporności na awarie, co będzie wymagać zastosowania odpowiednich rozwiązań w infrastrukturze serwerowej.

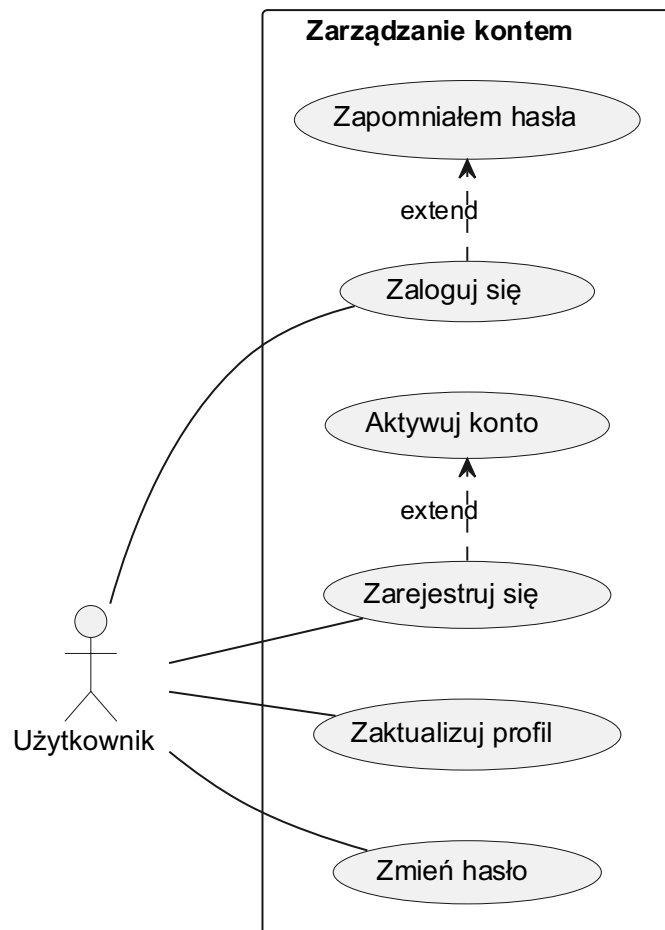
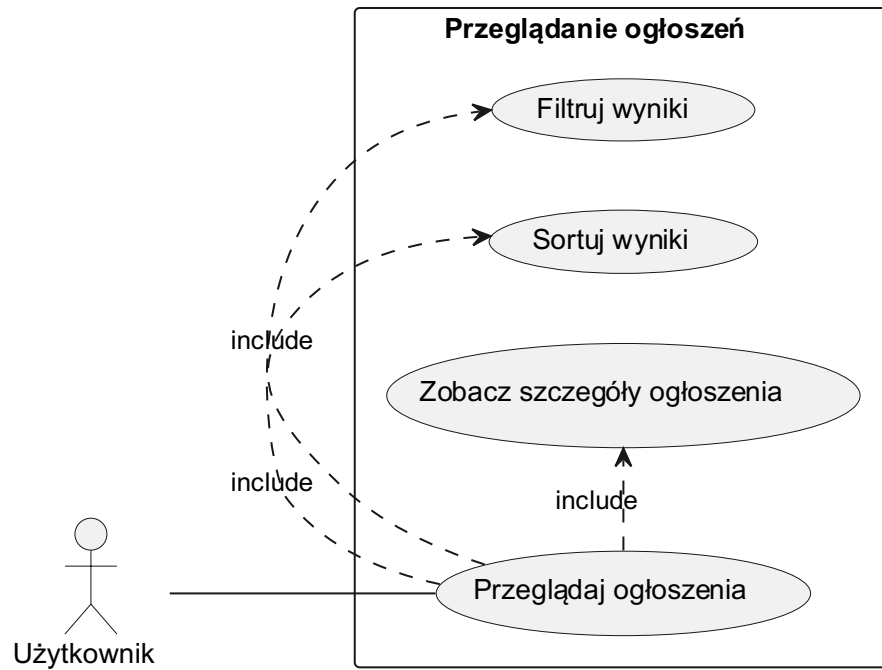
4.2 Wymagania funkcjonalne

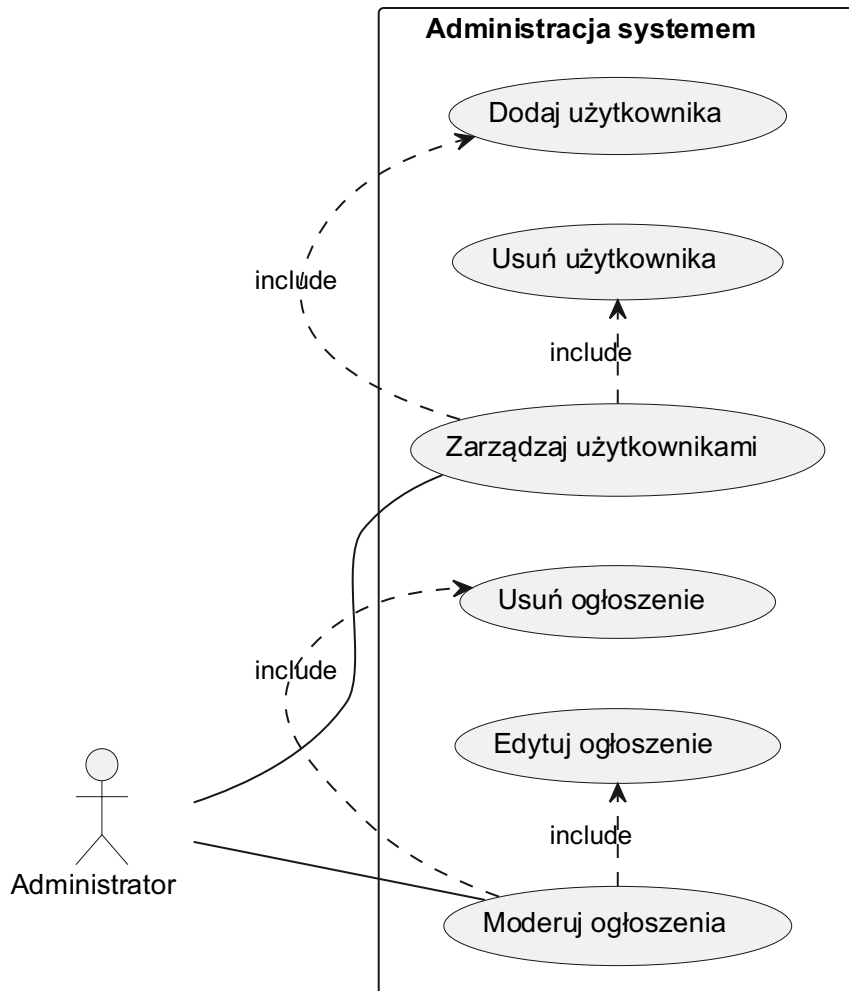
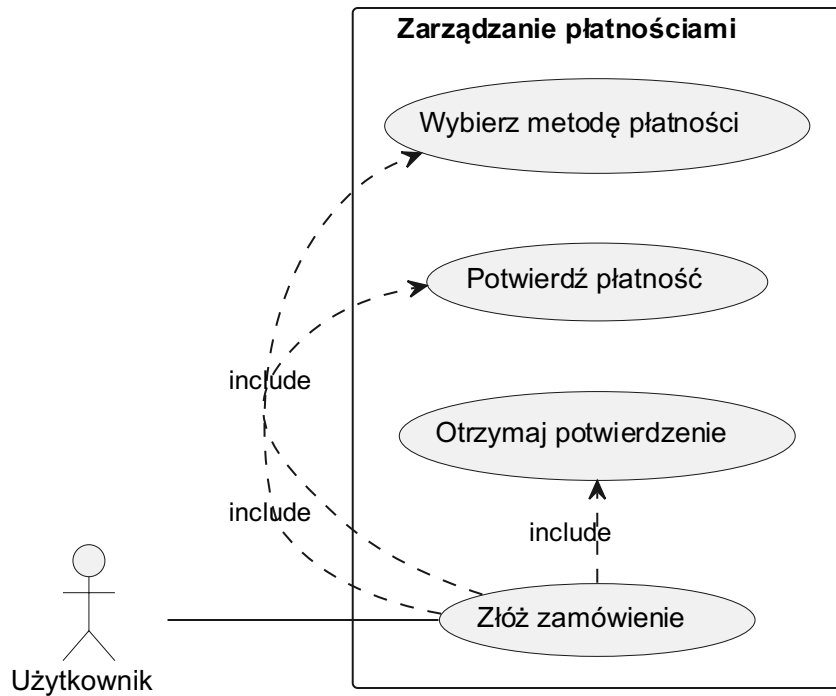
4.2.1 Lista wymagań

1. Użytkownik może tworzyć konto i zarządzać nim.
2. Użytkownik może dodawać, edytować i usuwać ogłoszenia.
3. Użytkownik może przeglądać ogłoszenia według kategorii.
4. Użytkownik może wyszukiwać ogłoszenia używając filtrów (np. lokalizacja, rodzaj ogłoszenia, cena).
5. Użytkownik może dodawać ogłoszenia do ulubionych.
6. System oferuje opcje kontaktu między kupującymi a sprzedającymi.
7. System umożliwia zarządzanie płatnościami w przypadku usług premium.

4.2.2 Diagramy przypadków użycia







4.2.3 Szczegółowy opis wymagań

Przypadek użycia 1: Dodawanie Ogłoszenia

ID: UC-01

Nazwa: Dodawanie Ogłoszenia

Uzasadnienie biznesowe: (ID-01) Pozwala sprzedawcom na szybkie i łatwe publikowanie ofert, co jest główną domeną serwisu.

Użytkownicy: Indywidualni Sprzedawcy, Firmy Lokalne

Scenariusze:

SCENARIUSZ 1

Nazwa scenariusza: Dodawanie Standardowego Ogłoszenia

Warunki początkowe: Użytkownik jest zalogowany.

Przebieg działań:

1. Użytkownik wybiera opcję „Dodaj Ogłoszenie”.
2. Wypełnia formularz ogłoszenia (tytuł, opis, cena, zdjęcia).
3. Wybiera kategorie ogłoszenia.
4. Przesyła formularz.

Efekty: Ogłoszenie jest dodane do systemu i widoczne dla innych użytkowników.

Wymagania niefunkcjonalne: System powinien obsłużyć żądanie dodania pojedynczego ogłoszenia w czasie poniżej 3 sek.

Częstotliwość: 3 (średnio często)

Istotność: 5 (krytyczne)

SCENARIUSZ 2

Nazwa scenariusza: Dodawanie Anonimowego Ogłoszenia

Warunki początkowe: Użytkownik nie musi być zalogowany.

Przebieg działań:

1. Użytkownik wybiera opcję „Dodaj Ogłoszenie Anonimowo”.
2. Wypełnia formularz ogłoszenia bez podawania swoich danych.
3. Wybiera kategorie ogłoszenia.
4. Przesyła formularz.

Efekty: Ogłoszenie jest dodane anonimowo do systemu i widoczne dla innych użytkowników.

Wymagania niefunkcjonalne: System musi zapewnić anonimowość użytkownika.

Częstotliwość: 3 (średnio często)

Istotność: 4 (ważne)

Przypadek Użycia 2: Wyszukiwanie Ogłoszeń

ID: UC-02

Nazwa: Wyszukiwanie Ogłoszeń

Uzasadnienie biznesowe: (ID-02) Umożliwia użytkownikom efektywne odnajdywanie ofert spełniających ich kryteria, co poprawia doświadczenia użytkownika i zwiększa zaangażowanie.

Użytkownicy: Kupujący, Przeglądający

Scenariusze:

SCENARIUSZ 1

Nazwa scenariusza: Wyszukiwanie zaawansowane

Warunki początkowe: Użytkownik jest na stronie wyszukiwania.

Przebieg działań:

1. Użytkownik wpisuje słowa kluczowe w pole wyszukiwania.
2. Ustawia filtry (cena, lokalizacja, kategoria).
3. Uruchamia wyszukiwanie.
4. Przegląda wyniki.

Efekty: Wyświetlane są ogłoszenia spełniające kryteria wyszukiwania.

Wymagania niefunkcjonalne: System musi obsługiwać wyszukiwanie w czasie rzeczywistym. Czas odpowiedzi wyszukiwarki powinien wynosić poniżej 2 sek.

Częstotliwość: 5 (bardzo często)

Istotność: 5 (krytyczne)

SCENARIUSZ 2

Nazwa scenariusza: Przeglądanie ogłoszeń na stronie głównej

Warunki początkowe: Użytkownik jest na stronie głównej.

Przebieg działań:

1. Użytkownik przegląda ogłoszenia na stronie głównej
2. Użytkownik może zmienić stronę (paginacja)
3. Użytkownik może kliknąć w dowolne ogłoszenie i przeczytać o nim więcej.

Efekty: Wyświetlane są wszystkie ogłoszenia posortowane po dacie dodania malejąco.

Wymagania niefunkcjonalne: System musi wyświetlić najnowsze ogłoszenia w czasie poniżej 0.5 sek od wejścia na stronę.

Częstotliwość: 5 (bardzo często)

Istotność: 5 (krytyczne)

Przypadek Użycia 3: Zarządzanie Kontem Użytkownika

ID: UC-03

Nazwa: Zarządzanie Kontem Użytkownika

Uzasadnienie biznesowe: (ID-01, ID-03) Umożliwia użytkownikom personalizację i kontrolę nad ich kontem i ogłoszeniami.

Użytkownicy: Indywidualni Sprzedawcy, Kupujący

Scenariusze:

SCENARIUSZ 1

Nazwa scenariusza: Edycja Danych Konta

Warunki początkowe: Użytkownik jest zalogowany i jest na podstronie swojego profilu.

Przebieg działań:

1. Użytkownik wybiera „Edytuj Profil”.
2. Aktualizuje dane (np. nazwisko, adres e-mail).
3. Zatwierdza zmiany.

Efekty: Dane konta są aktualizowane.

Wymagania niefunkcjonalne: Zapewnienie bezpieczeństwa danych.

Częstotliwość: 2 (rzadko)

Istotność: 4 (ważne)

4.3 Wymagania niefunkcjonalne

1. Wydajność

Wydajność systemu "Lokalizator" musi być dostosowana do obsługi wielu użytkowników jednocześnie, szczególnie w godzinach szczytu, co jest krytyczne dla użytkowników przeglądających i dodających ogłoszenia.

Specyfikacja:

- System powinien obsługiwać do 10,000 równoczesnych użytkowników bez znaczącego spadku wydajności.
- Czas odpowiedzi systemu na żądania użytkowników nie powinien przekraczać 2 sekund.

2. Bezpieczeństwo

Bezpieczeństwo jest kluczowym aspektem, mając na uwadze ochronę danych osobowych użytkowników oraz anonimowość w przypadku dodawania ogłoszeń.

Specyfikacja:

- Wdrożenie zabezpieczeń przed atakami typu SQL injection, XSS, i CSRF.
- Stosowanie szyfrowania danych wrażliwych, w tym danych osobowych i komunikacji między klientem a serwerem.
- Regularne kopie zapasowe danych oraz strategie szybkiego przywracania systemu po ewentualnych awariach.

3. Zabezpieczenia

Zabezpieczenia obejmują zarówno środki techniczne, jak i odpowiednie procedury.

Specyfikacja:

- Implementacja certyfikatów SSL/TLS dla całego ruchu sieciowego.
- Mechanizmy autentykacji dwuetapowej dla użytkowników.
- Regularne audyty bezpieczeństwa przeprowadzane przez zewnętrzne firmy.

4. Inne cechy jakości

- **Adaptowalność:** System musi być łatwy do dostosowania i rozbudowy w odpowiedzi na zmieniające się wymagania rynku lub regulacje prawne. Każda nowa funkcja powinna być w stanie zostać wdrożona w ciągu maksymalnie 1 miesiąca.
- **Dostępność:** System powinien być dostępny 24/7, z maksymalnym czasem przestoju nie przekraczającym 99,9% dostępności rocznie.
- **Poprawność:** Wszystkie funkcje systemu muszą działać zgodnie ze swoimi specyfikacjami, bez błędów krytycznych.
- **Elastyczność:** System powinien być zdolny do skalowania w zależności od wzrostu liczby użytkowników i obciążenia.
- **Łatwość konserwacji:** Kod źródłowy powinien być napisany zgodnie ze standardami branżowymi, co ułatwia jego utrzymanie i aktualizacje.
- **Przenośność:** System powinien być kompatybilny z różnymi systemami operacyjnymi i urządzeniami (desktop, mobile).
- **Awaryjność:** System powinien posiadać mechanizmy szybkiego wykrywania błędów i ich naprawy, z minimalnym wpływem na działanie systemu.
- **Testowalność:** Architektura systemu powinna wspierać łatwe tworzenie i wykonanie testów automatycznych.
- **Użyteczność:** Interfejs użytkownika powinien być intuicyjny i prosty w obsłudze, z minimalnym czasem nauki dla nowych użytkowników.

5 Zarządzanie projektem

5.1 Zasoby ludzkie

Zespół odpowiedzialny za realizację projektu składa się z następujących ról:

- Project Manager: odpowiedzialny za nadzorowanie wszystkich etapów projektu, zarządzanie zespołem i komunikację z interesariuszami. (1 osoba)
- Backend Developer(s): odpowiedzialni za projektowanie i implementację architektury serwerowej oraz baz danych, głównie w technologii Laravel 11. (2 osoby)
- Frontend Developer(s): odpowiedzialni za projektowanie i implementację interfejsu użytkownika oraz interakcji z API, korzystając z biblioteki React. (3 osoby)
- DevOps Engineer: odpowiedzialny za konfigurację środowisk, zarządzanie wersjami oraz wdrożenie ostatecznego produktu. (1 osoba)
- Quality Assurance Specialist: odpowiedzialny za testowanie całego systemu, zapewnienie jakości i spełnienie wymagań funkcjonalnych i нефункциональных. (1 osoba)
- UX/UI Designer: odpowiedzialny za projektowanie estetyczne i funkcjonalne interfejsu użytkownika, zapewniając przyjazne doświadczenie dla użytkownika końcowego. (1 osoba)

5.2 Harmonogram prac

- Faza 1: Planowanie i Projektowanie (1 miesiąc)
 - Analiza wymagań
 - Projekt architektury systemu i bazy danych
 - Projekt interfejsu użytkownika
- Faza 2: Implementacja (2 miesiące)
 - Budowa backendu
 - Budowa frontendu
 - Integracja frontendu z backendem
- Faza 3: Testowanie (1 miesiąc)
 - Testy jednostkowe i integracyjne
 - Testy użytkowe
 - Rozwiązywanie wykrytych błędów
- Faza 4: Wdrożenie (2 tygodnie)
 - Konfiguracja środowiska produkcyjnego
 - Wdrożenie ostateczne
 - Szkolenie użytkowników

5.3 Etapy/kamienie milowe projektu

Główne kamienie milowe projektu to:

- Zakończenie analizy wymagań i zatwierdzenie projektu (koniec Fazy 1)

- Ukończenie głównych modułów backendu i frontendu (koniec Fazy 2)
- Zakończenie wszystkich testów i potwierdzenie jakości (koniec Fazy 3)
- Pomyślne wdrożenie i akceptacja systemu przez użytkowników (koniec Fazy 4)

Każdy z tych kamieni milowych będzie monitorowany przez Project Managera, aby zapewnić, że projekt przebiega zgodnie z planem i osiąga wszystkie wyznaczone cele.

6 Zarządzanie ryzykiem

6.1 *Lista czynników ryzyka*

Wypełniona lista kontrolna

6.2 *Ocena ryzyka*

prawdopodobieństwo i wpływ

6.3 *Plan reakcji na ryzyko*

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

Rozdział obowiązkowy w zespołach co najmniej 3-osobowych, w mniejszych – do uzgodnienia z prowadzącym zajęcia.

7 Zarządzanie jakością

7.1 Scenariusze i przypadki testowe

głównie testowanie funkcjonalności, ale może być też testowanie wymagań niefunkcjonalnych/zgodności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

- numer – jako ID
- nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
- kategoria – poziom/kategoria testów
- opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
- tester - konkretna osoba lub klient/pracownik,
- termin – kiedy testowanie ma być przeprowadzane,
- narzędzia wspomagające – jeśli jakieś są używane przy danym scenariuszu
- przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu
- założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów
- zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań
- *przebieg lub zestaw danych testowych musi zawierać jawną informację o warunku zaliczenia testu*

8 Projekt techniczny

8.1 Opis architektury systemu

"LOKALizator" jest oparty na architekturze opartej na modelu klient-serwer. Serwer API stworzony jest w Laravel 11, zapewniając wydajne zarządzanie danymi i logiką biznesową, natomiast klient (frontend) zbudowany jest z użyciem ReactJS.

- **Frontend:** Aplikacja webowa stworzona w React.js, zapewniająca dynamiczne i responsywne interfejsy użytkownika, które komunikują się z backendem za pośrednictwem RESTful API.
- **Backend:** Aplikacja serwerowa w Laravel 11, która obsługuje wszystkie żądania z frontendu, zarządzanie danymi, autentykację użytkowników i inne procesy biznesowe.
- **Baza danych:** PostgreSQL, wybrany dla swojej niezawodności i wsparcia dla zaawansowanych funkcji SQL, który jest odpowiedni do zarządzania relacyjnymi danymi w aplikacjach o dużym obciążeniu.

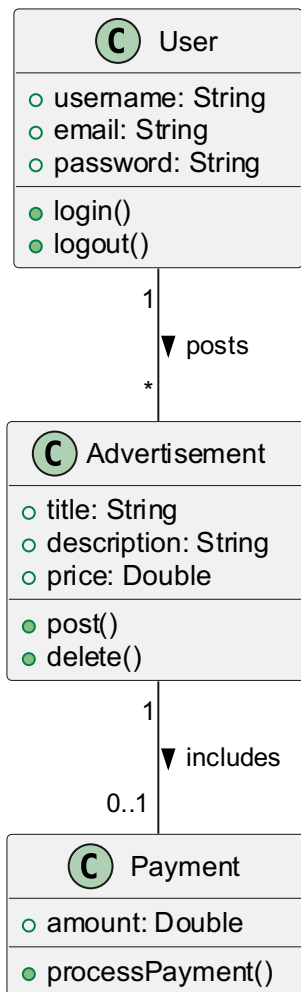
8.2 Technologie implementacji systemu

Technologia	Uzasadnienie wyboru
Laravel 11	Laravel to popularny framework PHP, który zapewnia bogaty zestaw funkcji dla szybkiego rozwoju aplikacji webowych. Jego wsparcie dla wzorców projektowych, jak MVC, oraz wbudowane mechanizmy autentykacji i ochrony sesji, sprawiają, że jest idealny do budowy bezpiecznego API backendowego.
React.js	Biblioteka JavaScript od Facebooka, znana z efektywnego renderowania UI i obsługi stanu aplikacji. Wybrana ze względu na modułowość, szeroką społeczność oraz łatwość integracji z różnymi backendami. Idealna do tworzenia dynamicznych, responsywnych interfejsów użytkownika.
PostgreSQL	Zaawansowany system zarządzania relacyjnymi bazami danych, który oferuje wsparcie dla dużych obciążeń i skomplikowanych zapytań. Wybrany dla stabilności, wydajności oraz doskonałego zarządzania transakcjami, co jest kluczowe w aplikacjach wymagających niezawodnego przechowywania danych.
Docker	Platforma do konteneryzacji aplikacji, która umożliwia łatwe pakowanie, dystrybucję i zarządzanie aplikacjami w izolowanych środowiskach. Docker jest wykorzystywany do usprawnienia procesów wdrażania oraz zapewnienia spójności środowisk od rozwoju po produkcję.

8.3 Diagramy UML

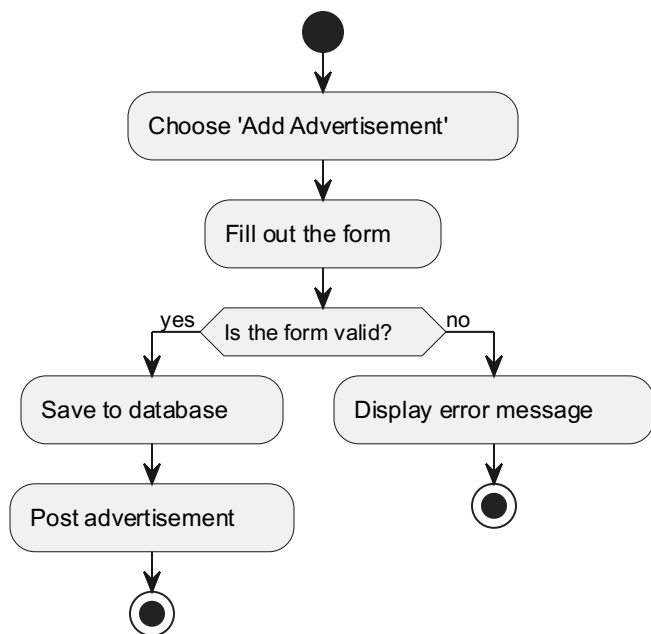
każdy diagram ma mieć tytuł oraz ma być na osobnej stronie
diagramy przypadków użycia umieszczone w punkcie 4.2.2, a nie tutaj.

8.3.1 Diagram(-y) klas



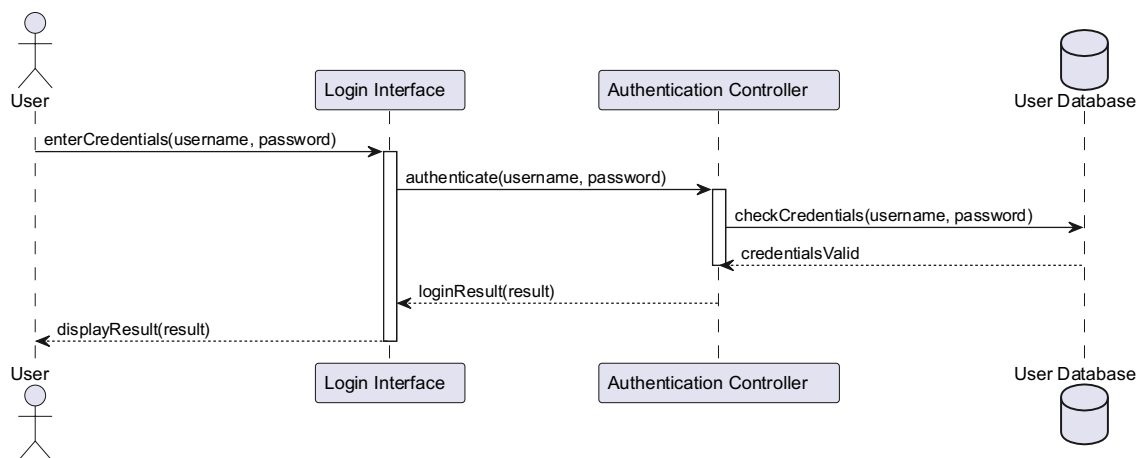
8.3.2 Diagram(-y) czynności

Diagram dodawania:



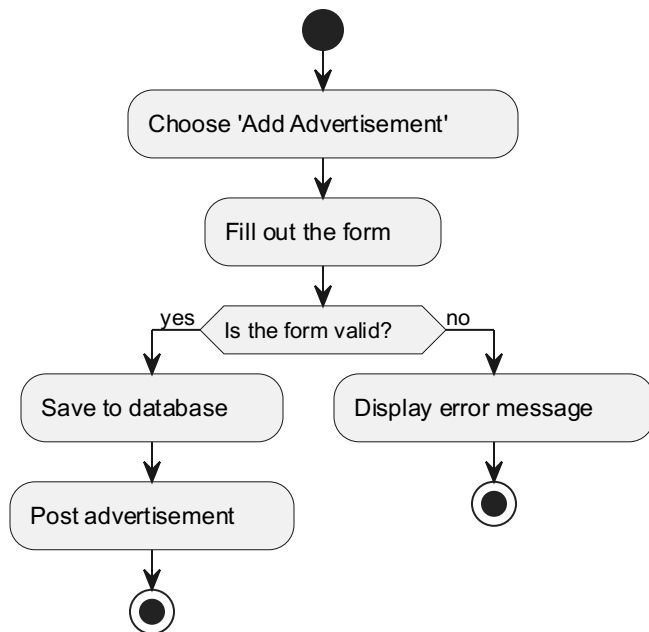
8.3.3 Diagramy sekwencji

Logowanie:

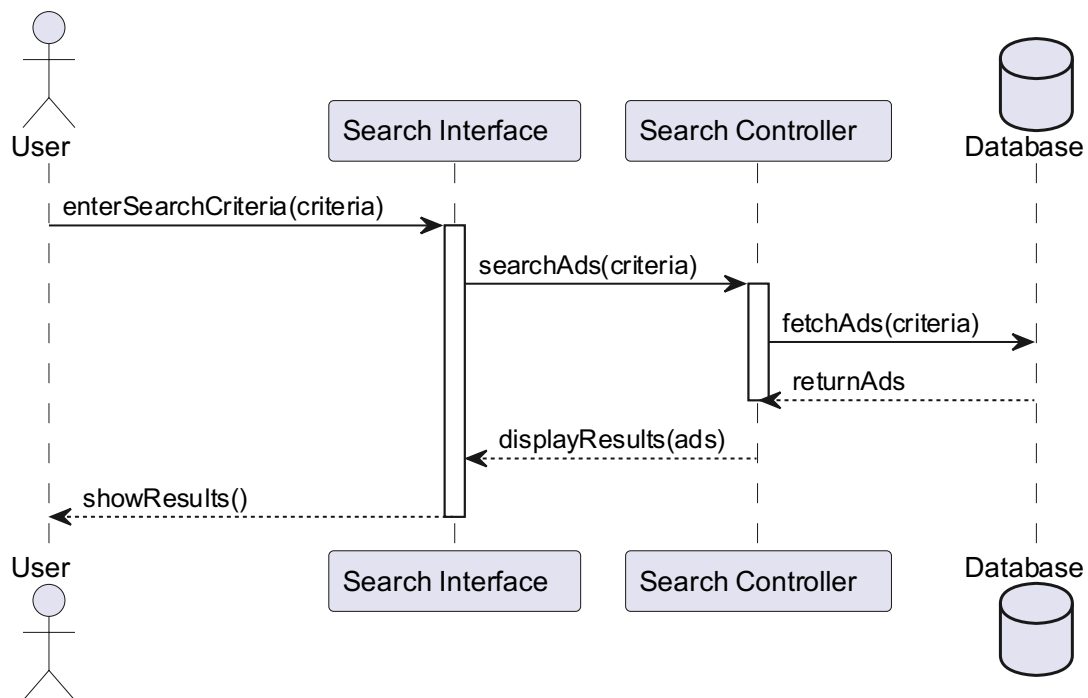


CRUD (Create, Read, Update, Delete) ogłoszeń:

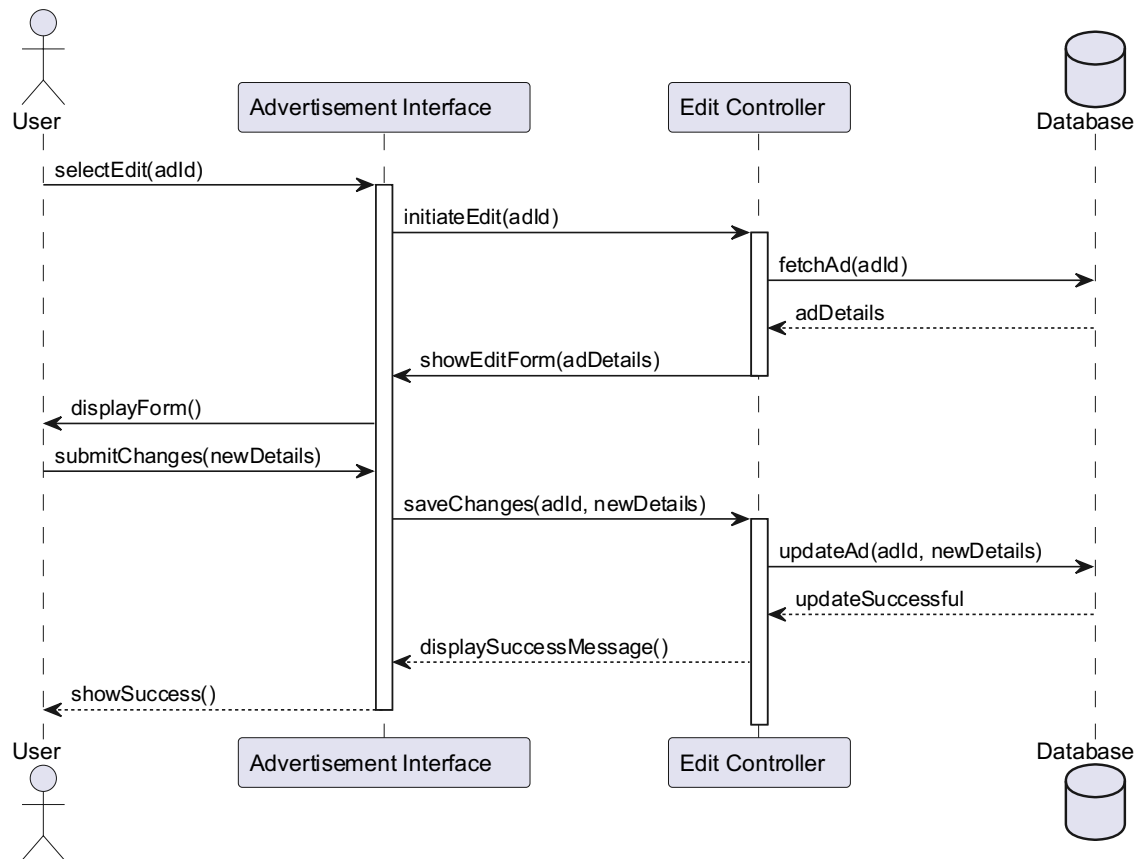
1. Dodawanie:



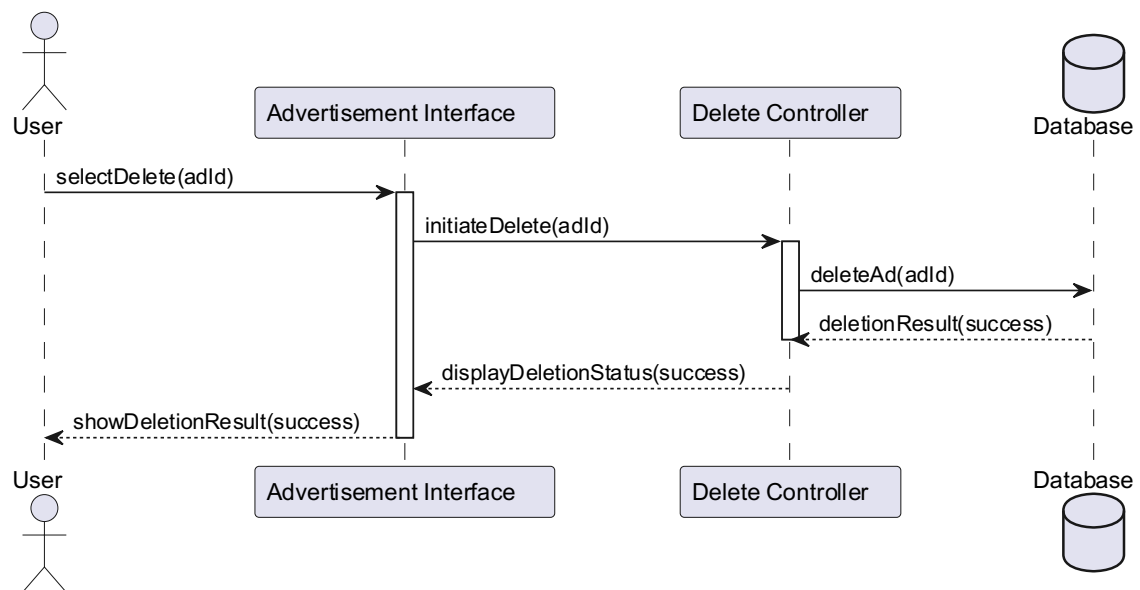
2. Odczyt (wyszukiwanie):



3. Edycja:



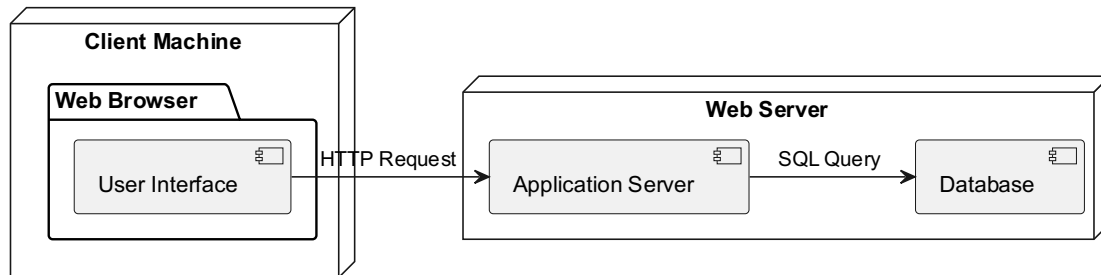
4. Usuwanie:



8.3.4 Inne diagramy

co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.

Diagram rozmieszczania:



8.4 Charakterystyka zastosowanych wzorców projektowych

informacja opisowa wspomagana diagramami (odsyłaczami do diagramów UML); jeśli wykorzystano wzorce projektowe, to należy wykazać dwa z nich
uwaga – wzorce projektowe nie są omawiane na wykładach!

8.5 Projekt bazy danych

8.5.1 Schemat

w trzeciej formie normalnej; jeśli w innej to umieć uzasadnić wybór

8.5.2 Projekty szczegółowe tabel

w zależności, czy następujące elementy są widoczne na schemacie b.d.: nazwa tabeli, nazwy pól, typ danych, wartości NULL, klucz główny, klucz obcy –

- jeśli TAK: i nie ma potrzeby pokazania dodatkowych elementów b.d., to ten punkt może być pusty,

- jeśli NIE: to podać te elementy, których nie widać na schemacie.

dodatkowymi elementami mogą być np. triggery, procedury, funkcje, indeksy, użytkownicy, role.

8.6 Projekt interfejsu użytkownika

co najmniej dla głównej funkcjonalności programu – w razie wątpliwości, uzgodnić z prowadzącym zajęcia

8.6.1 Lista głównych elementów interfejsu

okien, stron, aktywności (Android)

8.6.2 Przejścia między głównymi elementami

np. storyboard, schemat blokowy lub inna notacja

8.6.3 Projekty szczegółowe poszczególnych elementów

dla 5-7 głównych elementów (w zespołach 2-osobowych)

każdy element od nowej strony z następującą minimalną zawartością:

- numer – ID elementu
- nazwa – np. formularz danych produktu
- projekt graficzny – wystarczy schemat w narzędziu graficznym lub zrzut ekranu – z przykładowymi danymi (nie pusty!!!)
- opcjonalnie:
- opis – dodatkowe opcjonalne informacje o przeznaczeniu, obsłudze – jeśli nazwa nie będzie wystarczająco czytelna
- wykorzystane dane – jakie dane z bazy danych są wykorzystywane
- opis działania – tabela pokazująca m.in. co się dzieje po kliknięciu przycisku, wybraniu opcji z menu itp.

8.7 Procedura wdrożenia

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

9 Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

- pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
- może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

10 Podsumowanie

10.1 Szczegółowe nakłady projektowe członków zespołu

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – procentowy udział każdej osoby w skali całego projektu

11 Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach