

Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat:	System do zgłaszania awarii oraz zliczania przepracowanych godzin
Autorzy:	Piotr Zawalski, Filip Ślęzak
Grupa:	21B
Kierunek:	informatyka
Rok akademicki:	2020/2021
Poziom i semestr:	I/4
Tryb studiów:	niestacjonarne

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	9
5.3	Wymagania niefunkcjonalne	20
6	Zarządzanie projektem	21
6.1	Zasoby ludzkie	21
6.2	Harmonogram prac	21
6.3	Etapy/kamienie milowe projektu	21
7	Zarządzanie ryzykiem	22
7.1	Lista czynników ryzyka	22
7.2	Ocena ryzyka	22
7.3	Plan reakcji na ryzyko	22
8	Zarządzanie jakością	23
8.1	Scenariusze i przypadki testowe	23
9	Projekt techniczny	28
9.1	Opis architektury systemu	28
9.2	Technologie implementacji systemu	30
9.3	Diagramy UML	31
9.4	Charakterystyka zastosowanych wzorców projektowych	41
9.5	Projekt bazy danych	42
10	Schemat	41
10.1	Projekty szczegółowe tabel	41
10.2	Projekt interfejsu użytkownika	44
10.3	Procedura wdrożenia	45
11	Podsumowanie	46
11.1	Szczegółowe nakłady członków zespołu	46
12	Inne informacje	46

1 Odnośniki do innych źródeł

- Zarządzania projektem – [Jira](#)
- Wersjonowanie kodu – Środowisko [Gitlab](#) (self-hosted)
- System obsługi defektów – [Jira](#)

2 Słownik pojęć

- CI - Ciągła implementacja (Continuous integration).
- CD - Ciągła integracja (Continuous deployment).
- Zleceniodawca - firma Amazon Polska
- Zleceniobiorca - firma Zawalski & Ślęzak
- Aktorzy:
 - Pracownik działu Robotics - jest to serwisant, zajmujący się obsługą zgłoszeń o wszelkich awariach robotów magazynowych.
 - Pracownik magazynowy - jest to osoba pracująca na magazynie, na stanowisku STOW lub PICK, mająca możliwość zgłoszenia jakiegokolwiek technicznej usterki do Managera, który przekaze sprawę dalej.
 - Team manager - osoba zlecająca naprawy w razie awarii oraz kontrolująca należyte wykonanie pracy przez pracowników Robotics.

3 Wprowadzenie

3.1 Cel dokumentacji

Dokumentacja ma na celu przedstawienie projektu oraz uzgodnienie kwestii jego realizacji pomiędzy zleceniodawcą, a zleceniobiorcą. Zawiera szczegółowy plan wykonania projektu w konkretnych etapach, dzięki czemu daje możliwość kontrolowania postępu prac oraz dokładnego omówienia potrzeb zleceniodawcy. Celem wprowadzanego systemu jest usprawnienie funkcjonowania działalności całego magazynu poprzez wdrożenie oprogramowania, dzięki któremu serwis robotów magazynowych będzie znacznie wydajniejszy, co przełoży się na większą ilość obsłużonych przez firmę Amazon zleceń każdego dnia.

3.2 Przeznaczenie dokumentacji

Stworzona jest dla firm Amazon oraz Zawalski & Ślęzak. Ma ona na celu ustalenie opisu projektu oraz plan jego realizacji.

3.3 Opis organizacji lub analiza rynku

Firma Amazon to amerykańskie przedsiębiorstwo handlowe założone w 1994 roku w Seattle. Prowadzi największy na świecie sklep internetowy. Mają 110 magazynów w samych Stanach Zjednoczonych oraz 185 rozproszonych po całym świecie, z czego 9 znajduje się w Polsce, a centrum w Kolbaskowie jest jednym z najnowocześniejszych w Europie. Firma wdrażając system będzie w stanie znacznie usprawnić działanie magazynu, gdyż roboty są w nim kluczowe (przywożą one pracownikom towary, które oni wybierają, a dalej są wysyłane w świat. Każda awaria robota, bądź kolizja, powoduje straty pieniężne dla firmy, gdyż obsługują one wówczas mniej zleceń klientów.

3.4 Analiza SWOT organizacji

Silne: Firma ta jest zdecydowanym liderem wśród sklepów internetowych. Z powodu ogólnoświatowej pandemii ilość zakupów online znacznie się zwiększyła, dzięki czemu jest to obecnie jedna z największych firm świata.	Słabe: Mimo, że firma ma miliardy przychodu to ich realny zysk jest stosunkowo mały, gdyż mają stosunkowo niskie prowizje na produktach. Istnieje również dużo artykułów na temat złego traktowania pracowników przez Amazona, co szkodzi ich wizerunkowi.
--	---

<p>Szanse: Firma cały czas ma szanse na rozwój, gdyż poza samym sklepem internetowym, oferują również usługi w Amazon Web Services, co znacznie zwiększa dochód firmy.</p>	<p>Zagrożenia: Największym zagrożeniem obecnie dla tej firmy jest możliwy atak cyberprzestępców, dzięki któremu wyciekły by prywatne dane użytkowników, co spowodowałoby, że firma straciłaby część klientów.</p>
---	--

4 Specyfikacja wymagań

5.1 Charakterystyka ogólna

4.1.1 Definicja produktu

System informatyczny do zarządzania zgłoszeniami działu Robotics oraz zliczania przepracowanych godzin w firmie Amazon.

4.1.2 Podstawowe założenia

System będzie służył do zarządzania zgłoszeniami w firmie Amazon. Będą to zgłoszenia dla pracowników działu Robotics, wówczas gdy nastąpi np. kolizja robotów lub będzie je trzeba po prostu serwisować. Poprzez wdrożenie takiego systemu, dzięki czemu cały proces zgłaszania i przyjmowania zleceń zostanie zautomatyzowany, firma będzie mogła obsługiwać więcej zleceń, ponieważ naprawa robotów będzie przebiegała sprawniej. System ma również funkcje, dzięki której możliwe jest zliczanie godzin przepracowanych przez pracowników oraz to kto i ile zleceń wykonał np. w skali miesiąca. Dzięki tej funkcji dział księgowości również usprawni swoją pracę, poprzez częściową automatyzację tego zadania.

4.1.3 Cel biznesowy

- Usprawnienie funkcjonowania magazynu(większa ilość wysyłanych paczek)
- Usprawnienie działania księgowości.
- Uproszczenie procesu zgłaszania problemów z robotami w magazynie.
- Śledzenie czasu spędzonego na poszczególnych zadaniach działu Robotics.

4.1.4 Użytkownicy

- Serwisanci z działu Robotics
- Team Managerowie
- Pracownicy księgowości

4.1.5 Korzyści z systemu

1. Pracownicy działu Robotics:
 - obniżenie czasu obsługi zleceń (ID 11)
 - usprawniona komunikacja z między serwisantami a team managerami (ID 12)
 - dostęp do historii wcześniejszych napraw (ID 13)
2. Team Manager:
 - ułatwienie zarządzania i śledzenia zgłoszeń w systemie (ID 21)
 - dostęp do pełnej dokumentacji wykonanych zgłoszeń (ID 22)
3. Dział księgowości:
 - generowanie miesięcznych raportów o wykonanych zleceniach (ID 31)
 - zautomatyzowanie procesu zliczania godzin spędzonych w pracy (ID 32)

4.1.6 Ograniczenia projektowe i wdrożeniowe

przepisy prawne, specyficzne technologie, narzędzia, b.d., protokoły komunikacyjne, aspekty zabezpieczeń, zgodność ze standardami, powiązania z innymi aplikacjami, platforma sprzętowa, system operacyjny, inne komponenty niezbędne do współpracy – wszystko wraz z uzasadnieniem!

Specyficzne Technologie:

- Docker
- Django
- Angular
- Postgresql
- Kubernetes
- Helm (Kubernetes)
- Ingress (Kubernetes)
- [Fauria](#)
- Node.js

- Webpack
- Istio
- Haproxy

Platforma Sprzętowa:

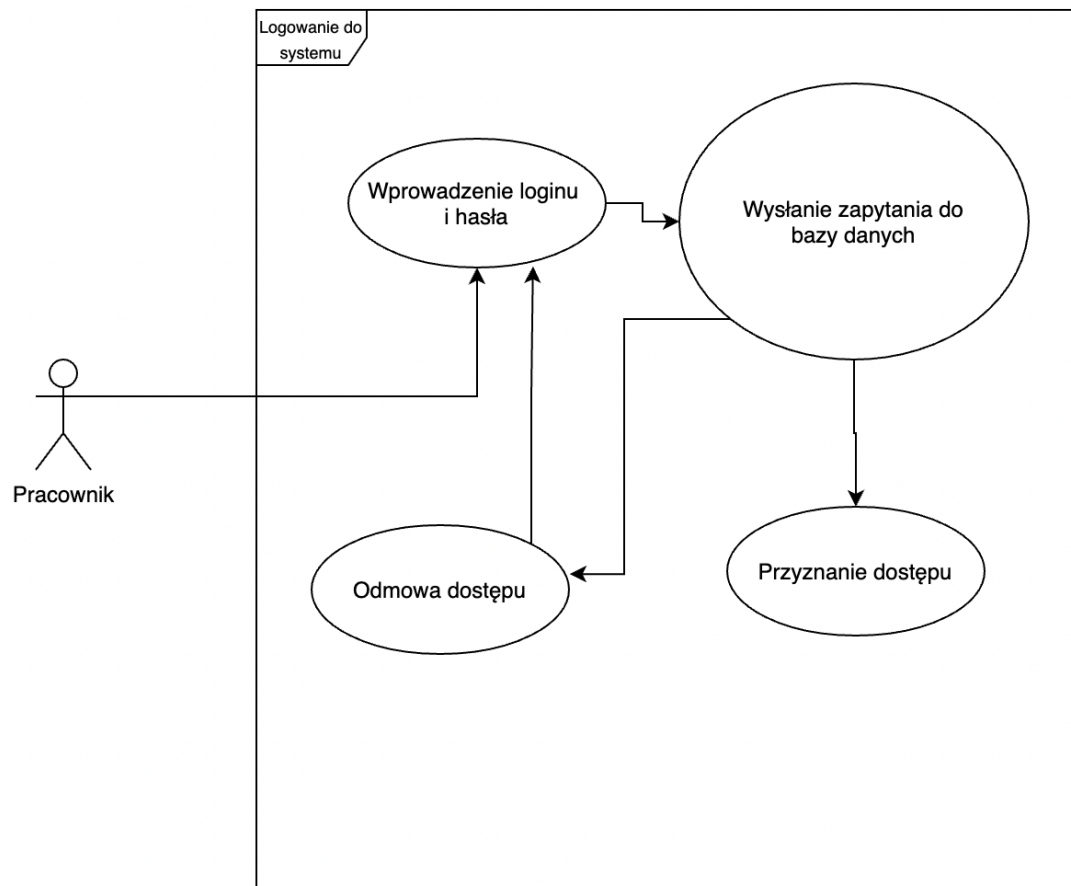
Przeglądarka Internetowa (mobile, desktop)

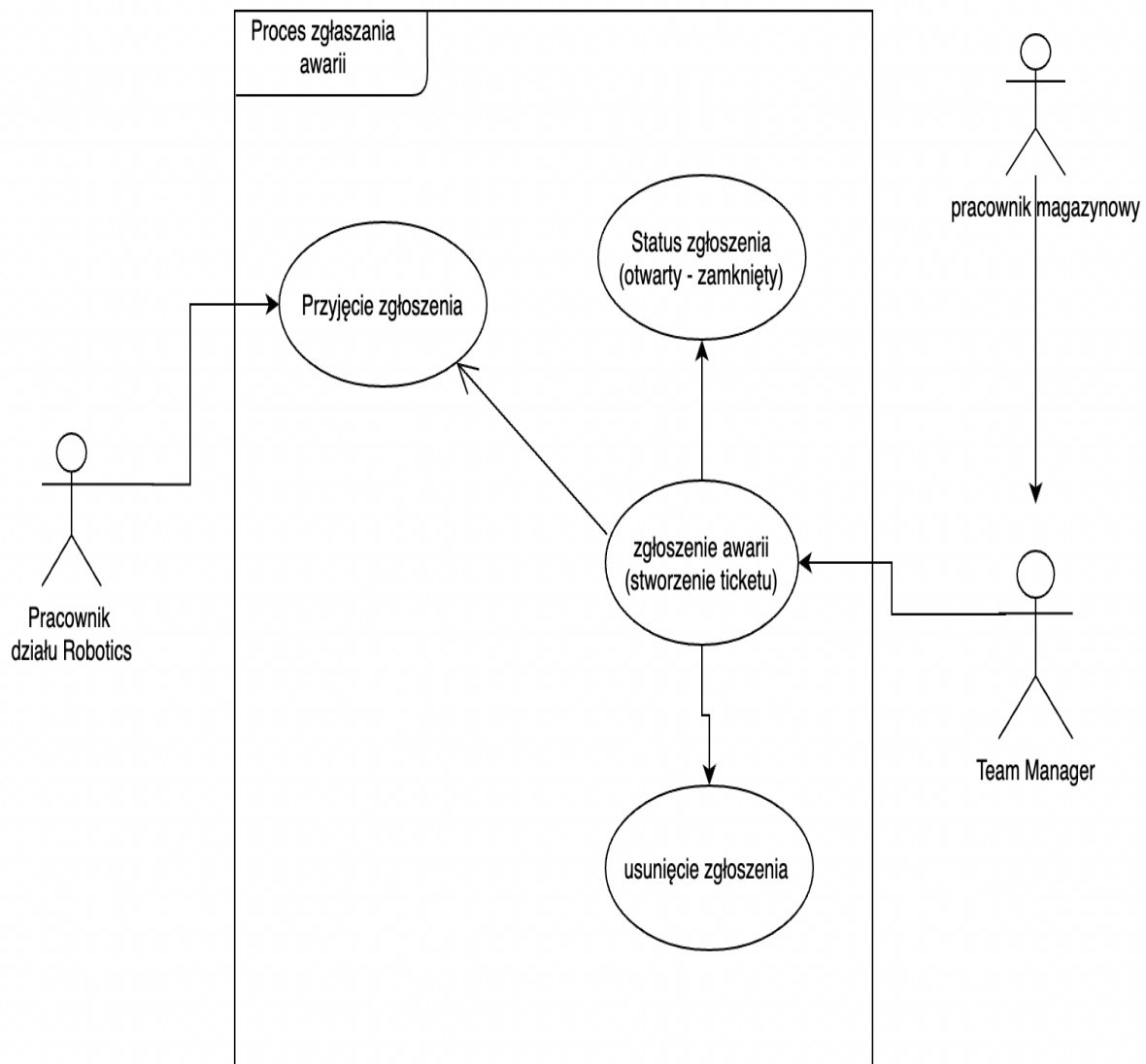
4.2 Wymagania funkcjonalne

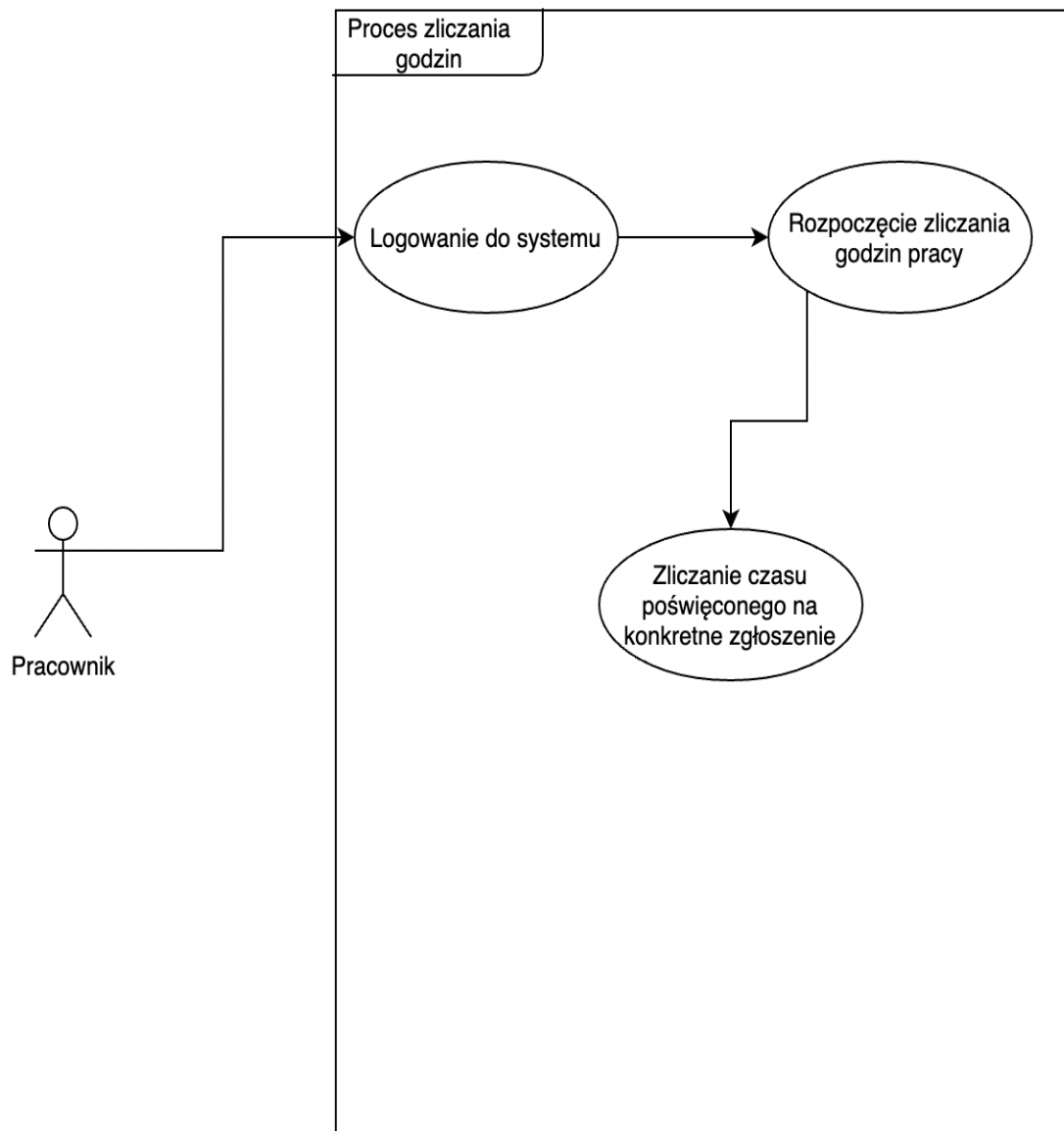
4.2.1 Lista wymagań

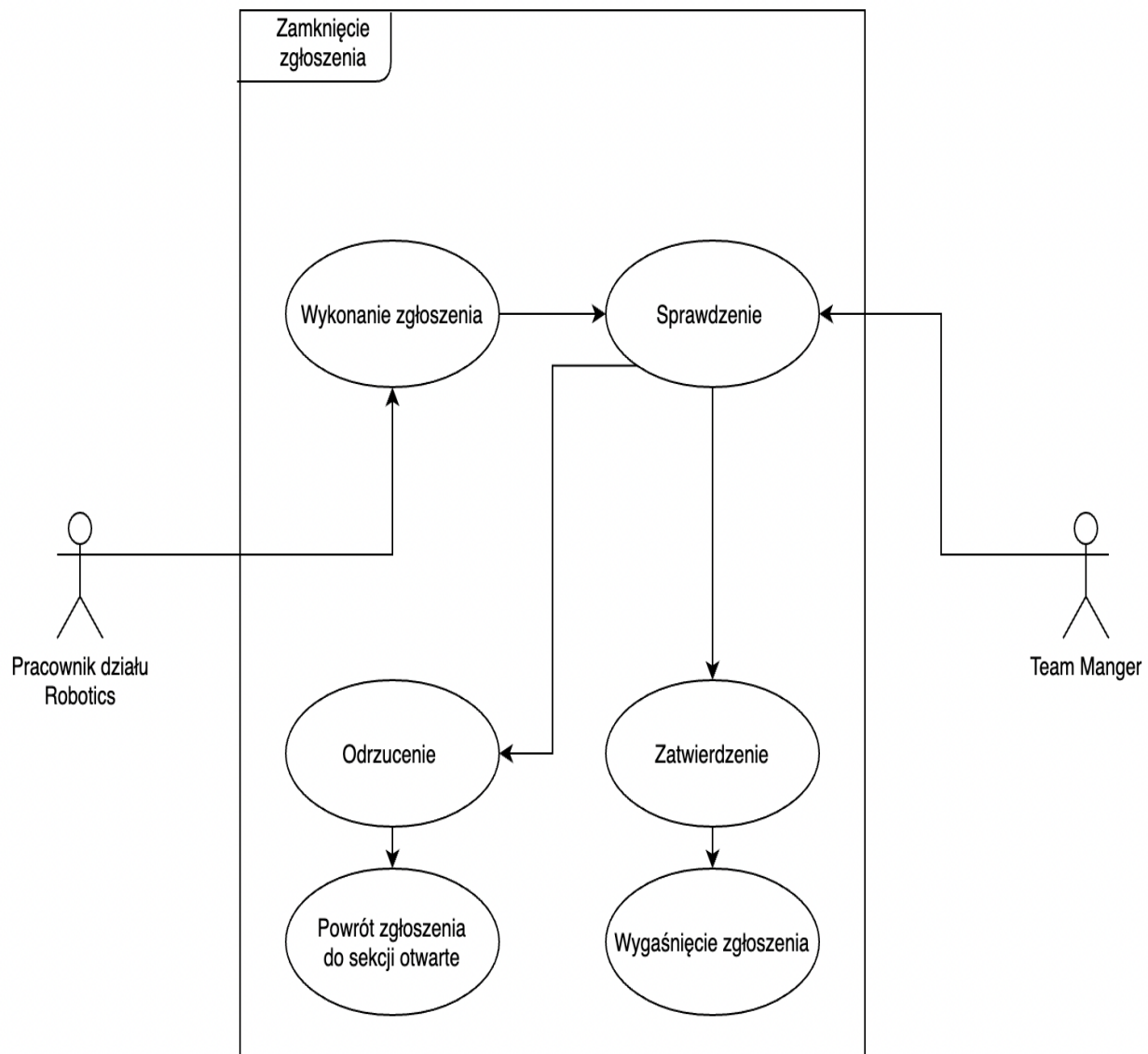
1. Logowanie do systemu.
Pracownik może zalogować się do systemu tylko w lokalnej sieci prywatnej firmy. Jest to możliwe w dowolnej przeglądarce internetowej - na komputerach oraz na telefonach.
2. Proces zgłaszania awarii (ticket)
Podczas każdej awarii w systemie tworzone jest zgłoszenie, które następnie pracownik działu Robotics może przypisać do swojego konta i zacząć nad nim pracować.
3. Proces zliczania godzin przepracowanych.
Podczas logowania do systemu rozpoczyna się zliczanie godzin, które uwzględnia czas przez który pracownik był zalogowany w systemie oraz czas poświęcony na konkretne zgłoszenia.
4. Zamknięcie zgłoszenia po wcześniejszym sprawdzeniu przez managera.
Po wykonaniu zgłoszenia manager sprawdza czy awaria została należycie usunięta. Jeśli została naprawiona zatwierdza ją w systemie, a zgłoszenie wygasa. Natomiast jeśli coś jest zrobione nie tak jak należy manager ma prawo odmówić zatwierdzenia, przy czym musi wskazać konkretne przyczyny i zapisać co wymaga poprawy, wówczas zgłoszenie dalej jest w systemie i czeka na przypisanie go przez pracownika z działu Robotics.
5. Zarządzanie zgłoszeniami w systemie.
Pracownik działu robotics może przyjąć wybrane zgłoszenie i przypisać je do swojego konta. Po zrealizowaniu wysłane zostaje zapytanie do Team Managera w celu weryfikacji rozwiązane problemu.

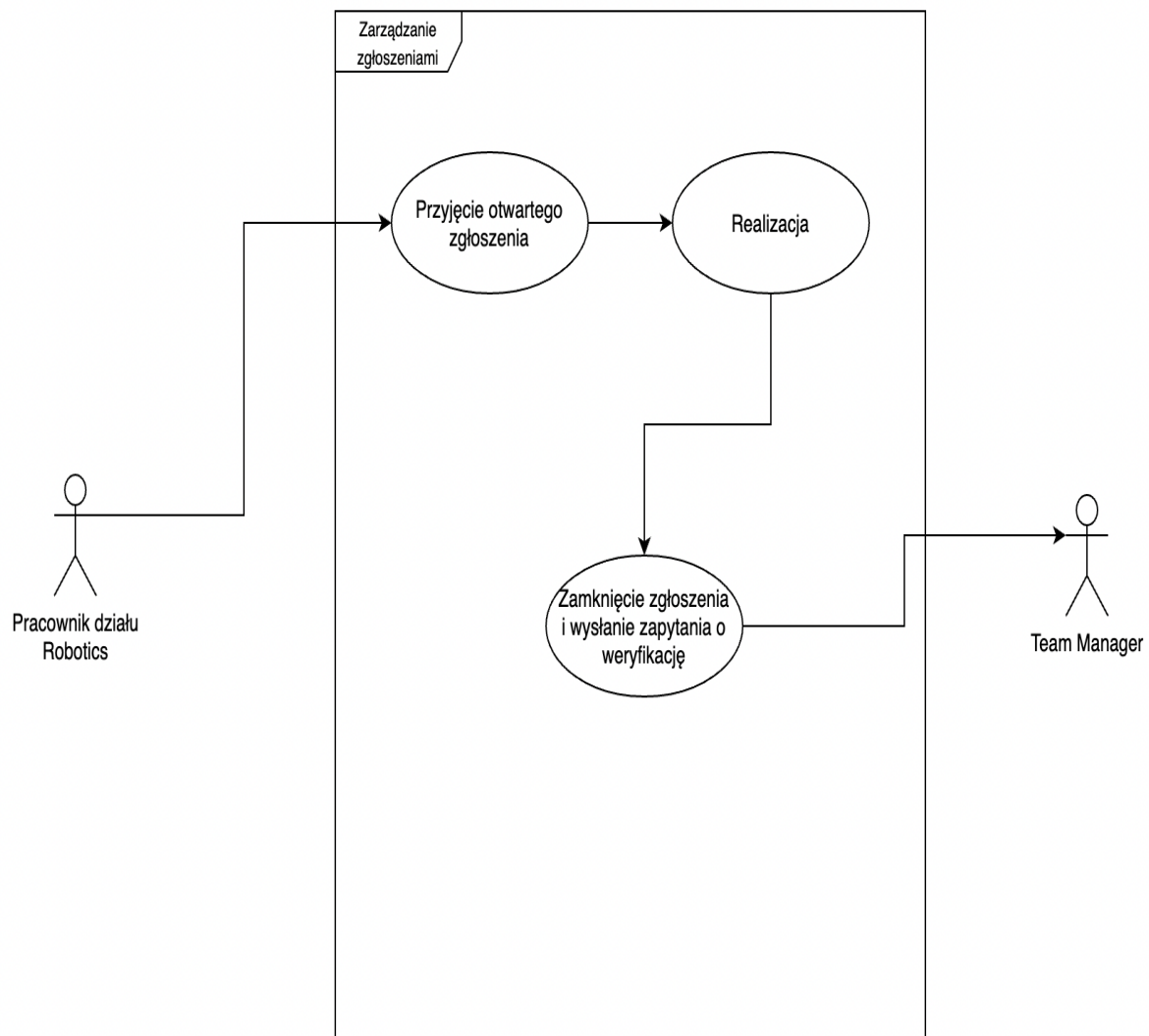
4.2.2 Diagramy przypadków użycia











4.2.3 Szczegółowy opis wymagań

ID: 1

Nazwa: Logowanie do systemu

Korzyści biznesowe: ID11, ID12

Użytkownik: pracownik działu Robotics

Warunki początkowe: Pracownik musi być zarejestrowany w systemie

Efekty: Pracownik uzyskuje dostęp do panelu aplikacji

Istotność: 5

Przebieg działań:

1. Użytkownik wprowadza dane w panelu logowania (login i hasło).
2. Zostaje wysłane zapytanie do bazy danych.
3. Serwer przetwarza informacje i porównuje zgodność dostarczonych informacji z bazą danych.
 - 3.1. W przypadku nieprawidłowych danych:
 - 3.1.1. Dane dostarczone w formularzu logowania zostają wymazane.
 - 3.1.2. Pracownik jest proszony o ponowne wpisanie danych w formularzu.
 - 3.2. W przypadku podania prawidłowych danych:
 - 3.2.1. Użytkownik zostaje przekierowany na stronę główną aplikacji.

ID: 2

Nazwa: Proces zgłaszania awarii

Korzyści biznesowe: ID11, ID12, ID21, ID22

Użytkownik: pracownik działu Robotics, Team Manager, Pracownik magazynowy

Istotność: 5

Warunki początkowe: Wystąpiła awaria uniemożliwiająca lub utrudniająca pracę magazynierów.

Efekty: Serwisanci z działu Robotics posiadają wiedzę na temat usterki i mogą przystąpić do naprawy.

Przebieg działań:

1. Pracownik magazynowy powiadamia Team Managera.
2. Team Manager sprawdza usterkę i sporządza wstępny raport.
3. Następnie Manager wprowadza te dane do systemu i tworzy zgłoszenie.
4. Serwisant dostaje powiadomienie o awarii.
5. Serwisant przyjmuje zgłoszenie i rozpoczyna nad nim pracę.

ID: 3

Nazwa: Proces zliczania godzin

Korzyści biznesowe: ID32

Użytkownik: pracownik działu Robotics, pracownik księgowości

Istotność: 4

Warunki początkowe: Pracownik musi być zarejestrowany w systemie i mieć do niego dostęp

Efekty: Automatyczne wysłanie raportu o czasie pracy.

Przebieg działań:

1. Pracownik loguje się do systemu.
2. System automatycznie rozpoczyna naliczanie czasu pracy.
3. Przy wylogowaniu następuje koniec liczenia..
4. Na serwer zostaje wysłany raport z danego dnia.
5. Pracownik księgowości może przeglądać raporty.

ID: 4

Nazwa: Zamknięcie zgłoszenia

Korzyści biznesowe: ID12, ID13, ID22

Użytkownik: pracownik działu Robotics, Team Manager

Istotność: 5

Warunki początkowe: Zgłoszenie musi być widoczne w systemie i przypisane do konkretnego serwisanta.

Efekty: Zamknięcie otwartego zgłoszenia i przeniesienie go do archiwum.

Przebieg działań:

1. Pracownik oznacza zgłoszenie jako wykonane.
2. Zostaje wysłane powiadomienie do Team Managera o wykonaniu zgłoszenia.
3. Manager weryfikuje czy naprawa została przeprowadzona należycie.
 - 3.1. Naprawa zostaje zatwierdzona przez Managera
 - 3.1.1. Zgłoszenie znika z systemu i zostaje zarchiwizowane.
 - 3.2. Naprawa została odrzucona.
 - 3.2.1. Zgłoszenie powraca na listę główną do ponownej naprawy.

ID: 5

Nazwa: Zarządzanie zgłoszeniami

Korzyści biznesowe: ID11, ID12, ID13, ID21, ID22

Użytkownik: pracownik działu Robotics, Team Manager

Istotność: 5

Warunki początkowe: Użytkownicy muszą być zalogowani do systemu.

Efekty: Szybkie zarządzanie zgłoszeniami w systemie.

Przebieg działań:

1. Pracownik działu Robotics przegląda aktualne zlecenia i może dla siebie wybrać dowolne.
2. Następuje przypisania numeru zgłoszenia do konta serwisanta.
3. Jeśli naprawa wymaga sprawdzenia historii awarii danego urządzenia to pracownik ma taką możliwość z poziomu systemu.
4. Realizacja naprawy.
5. Team Manager może kontrolować proces wykonania i dopytać serwisanta o szczegóły w przypadku niejasności.
6. Serwisant zaznacza wykonanie zadania
7. Team Manager weryfikuje poprawność rozwiązania.

4.3 Wymagania niefunkcjonalne

Wydajność

- Replikacja
- Wersjonowanie
- Automatyczne testowanie i deployment.
- Skalowalność (kubernetes)
- Proces identyfikacji podczas logowania nie powinien trwać dłużej niż 20ms
- Zgłoszenie awarii powinno zostać zatwierdzone w systemie nie dłużej niż po 2 sekundach.

Bezpieczeństwo

- Baza danych utrzymywana przez zewnętrzne rozwiązanie AWS.
- Dostęp do systemu możliwy tylko w sieci magazynu.
- Cotygodniowe tworzenie kopii zapasowych danych.

Zabezpieczenia

- Dostępność tylko w obrębie sieci magazynu, zabezpieczone przez haproxy.
- Dostęp do panelu administracyjnego tylko poprzez zweryfikowane urządzenia (MAC), za pomocą poprawnie skonfigurowanego ingressa i haproxy.

Inne cechy

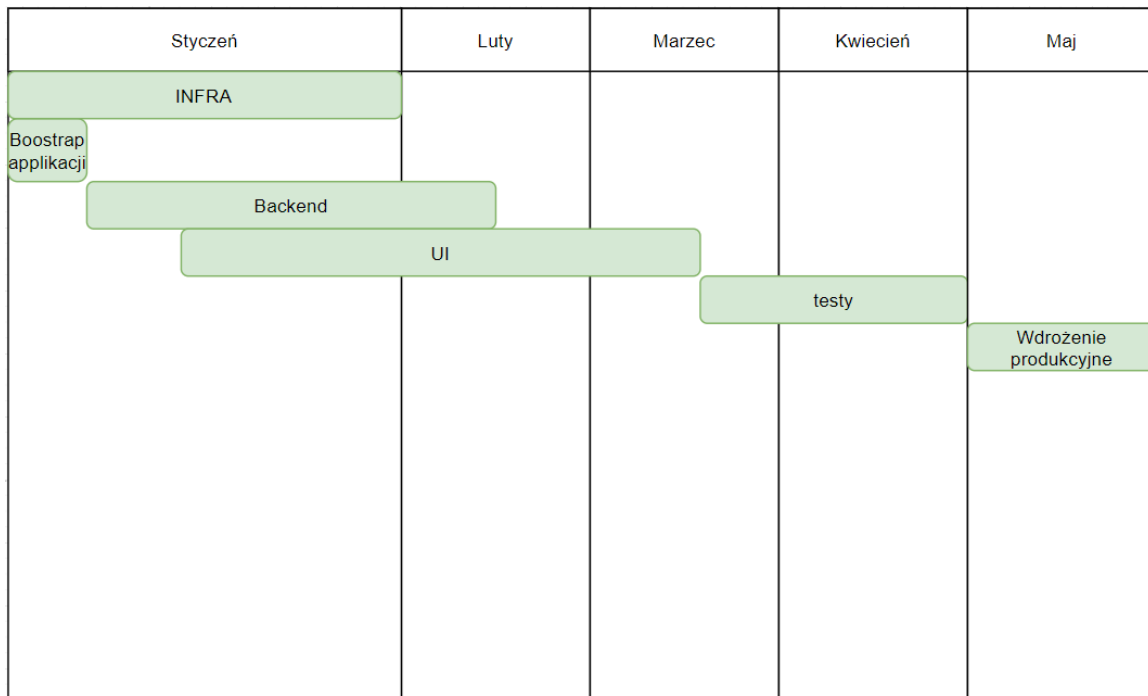
- Responsywność
- Skalowalność
- Użyteczność

5 Zarządzanie projektem

5.1 Zasoby ludzkie

- Administratorzy systemowi (3 osoby)
- Testerzy (testy jednostkowe, integracyjne, obciążeniowe i end-to-end) (5 osób)
- Front-end Developerzy (4 osoby)
- Back-end Developerzy (5 osób)

5.2 Harmonogram prac



5.3 Etapy/kamienie milowe projektu

- Przygotowanie infrastruktury
- Bootstrap projektów (repozytoria, pliki projektów itd) .
- API/Backend
- UI/Frontend
- Testy (jednostkowe, obciążeniowe, end-to-end)
- Wdrożenie instancji produkcyjnej.
- Oddanie aplikacji klientowi.

6 Zarządzanie ryzykiem

6.1 Lista czynników ryzyka

- możliwość ingerencji w system osoby nieupoważnionej
- błąd systemowy (niewykryty podczas testowania)
- problem z bazą danych / utrata danych

6.2 Ocena ryzyka

Największym ryzykiem jest dostęp do systemu osób niepowołanych, jednakże aby można było dokonać faktycznych szkód potrzeba dostęp do konta administratora (root). Poprzez to że, dostęp do konta ma tylko dwóch administratorów systemowych w firmie szansa ta jest stosunkowo mała.

Utrata bazy danych byłaby dużym zagrożeniem jednakże dzięki regularnemu tworzeniu kopii zapasowych możliwe jest przywrócenie systemu do stanu używalności w ciągu kilku godzin (3-4).

Błędy systemowe, które użytkownik mógłby napotkać nie powinny być krytyczne w skutkach (choć takie nie są oczywiście niemożliwe). Powinny być to raczej drobne błędy nie wyrządzają większych szkód dla integralności całego systemu.

6.3 Plan reakcji na ryzyko

- tworzenie kopii zapasowych bazy danych
- używanie wewnętrznej sieci lokalnej w firmie, aby uniemożliwić osobom postronnym dostęp do systemu
- firma powinna mieć zatrudnionych programistów, którzy natychmiast zareagowaliby w sytuacji nieoczekiwanego błędu systemu, który mógł zostać niewykryty w testach

8 Zarządzanie jakością

8.1 Scenariusze i przypadki testowe

Test nr 1

Testowanie wewnętrzne – weryfikacja wykonanej pracy, wyszukiwanie błędów

Cel: sprawdzenie działania programu w przypadku logowania się do systemu

Grupa testerów: Jan Nowak, Piotr Wiśniewski, Tomasz Kowalski

Data przeprowadzenia: 04.06.2021

Narzędzia: Jira software

Lista poszczególnych kroków:

- 1. Wyświetlenie ekranu logowania – wprowadzenie danych przez użytkownika*
- 2. Łączenie z bazą danych – weryfikacja połączenia, sprawdzenie czy jest szyfrowane*
- 3. Weryfikacja poprawności danych logowania – przetworzenie danych funkcją haszującą sha256 i porównanie wartości z bazą danych*
- 4. Niepoprawne dane logowania - resetuje wprowadzone dane i prosi o ponowne wpisanie*
- 5. Wczytanie panelu użytkownika na stronie – po zwróceniu wartości TRUE przez serwer użytkownik przekierowywany jest na stronę główną*

Zestaw danych testowych: dane logowania

Krok 1 login (string)

TRUE/FALSE

Krok 2 hasło (string)

TRUE/FALSE

Krok 3 numer telefonu(string)

TRUE/FALSE

Krok 4 id pracownika (int)

TRUE/FALSE

Test nr 2

Testowanie wewnętrzne – weryfikacja wykonanej pracy, wyszukiwanie błędów

Cel: sprawdzenie działania programu w przypadku zgłoszenia awarii

Grupa testerów: Jan Nowak, Piotr Wiśniewski, Tomasz Kowalski

Data przeprowadzenia: 05.06.2021

Narzędzia: Jira software

Lista poszczególnych kroków:

- 1. Tabela w menu „Zgłoś awarię” – otwiera się nowa strona gdzie należy wprowadzić wymagane dane*
- 2. Wypełnienie pól danymi – zostają one wysłane do serwera*
- 3. Niepoprawne dane – resetuje wprowadzone dane i prosi o ponowne wpisanie*
- 4. Wpisanie awarii na liście aktywnych zgłoszeń - jest ona dostępna od teraz dla serwisantów*

Zestaw danych testowych: dane identyfikacyjne zgłoszenia

Krok 1 id zgłoszenia(int)

TRUE/FALSE

Krok 2 id managera(int)

TRUE/FALSE

Krok 3 data zgłoszenia (date)

TRUE/FALSE

Krok 4 id pracownika (int)

TRUE/FALSE

Test nr 3

Testowanie wewnętrzne – weryfikacja wykonanej pracy, wyszukiwanie błędów

Cel: sprawdzenie działania programu w przypadku zliczania przepracowanych godzin

Grupa testerów: Jan Nowak, Piotr Wiśniewski, Tomasz Kowalski

Data przeprowadzenia: 06.06.2021

Narzędzia: Jira software

Lista poszczególnych kroków:

- 1. Zalogowanie użytkownika do systemu – rozpoczęcie naliczania godzin*
- 2. Przydzielenie do konta konkretnego zgłoszenia – rozpoczęcie zliczania czasu naprawy*
- 3. Zamknięcie zgłoszenia – koniec zliczania czasu naprawy*
- 4. Wylogowanie z systemu – koniec zliczania godzin i zapisanie danych na serwerze*

Zestaw danych testowych: liczba godzin

Krok 1 id użytkownika(int)

TRUE/FALSE

Krok 2 start(date)

TRUE/FALSE

Krok 3 koniec(date)

TRUE/FALSE

Krok 4 start naprawy(date)

TRUE/FALSE

Krok 5 koniec naprawy(date)

TRUE/FALSE

Test nr 4

Testowanie wewnętrzne – weryfikacja wykonanej pracy, wyszukiwanie błędów

Cel: sprawdzenie działania programu w przypadku procesu zamknięcia zgłoszenia

Grupa testerów: Jan Nowak, Piotr Wiśniewski, Tomasz Kowalski

Data przeprowadzenia: 08.06.2021

Narzędzia: Jira software

Lista poszczególnych kroków:

- 1. Zakończenie zgłoszenia – pracownik oznacza w systemie zgłoszenie jako wykonane*
- 2. Prośba o weryfikację – automatycznie jest wysyłana do managera*
- 3. Zamknięcie zgłoszenia – status zgłoszenia zwraca wówczas wartość TRUE*
- 4. Nie zamknięcie zgłoszenia – próba poprawy danego przypadku*

Zestaw danych testowych: zamknięcie zgłoszenia

Krok 1 id użytkownika (int)

TRUE/FALSE

Krok 2 id managera (int)

TRUE/FALSE

Krok 3 id zgłoszenia (int)

TRUE/FALSE

Krok 4 status zgłoszenia (bool)

TRUE/FALSE

Krok 5 czas trwania naprawy (int)

TRUE/FALSE

Test nr 5

Testowanie wewnętrzne – weryfikacja wykonanej pracy, wyszukiwanie błędów

Cel: sprawdzenie działania programu w przypadku procesu zarządzania zgłoszeniami

Grupa testerów: Jan Nowak, Piotr Wiśniewski, Tomasz Kowalski

Data przeprowadzenia: 09.06.2021

Narzędzia: Jira software

Lista poszczególnych kroków:

- 1. Otwarte zgłoszenie – możliwość przypisania przez serwisanta*
- 2. Usterka w trakcie naprawy – zgłoszenie zostaje podjęte, wówczas status zgłoszenia zwraca wartość FALSE*
- 3. Zamknięcie zgłoszenia – status zgłoszenia zwraca wówczas wartość TRUE*
- 4. Niezamknięcie zgłoszenia – próba poprawy danego przypadku*

Zestaw danych testowych: zarządzanie zgłoszeniami

Krok 1 id użytkownika (int)

TRUE/FALSE

Krok 2 id zgłoszenia (int)

TRUE/FALSE

Krok 3 status zgłoszenia (bool)

TRUE/FALSE

9 Projekt techniczny

9.1 Opis architektury systemu

Interfejs Użytkownika



Aplikacja webowa oparta na frameworku [Angular](#). Pozwala na interakcję użytkowników z systemem.

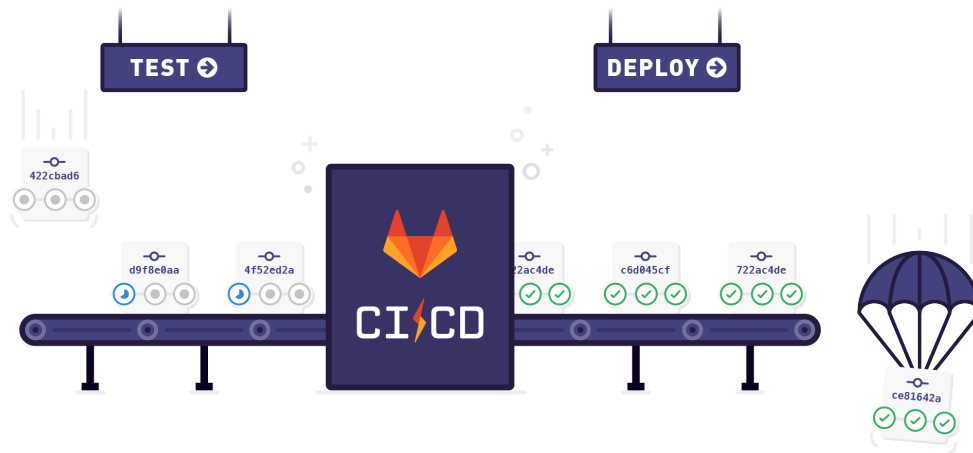
Interfejs programowania aplikacji (API)



Pośrednik pomiędzy interfejsem użytkownika a bazą danych. Wystawia odpowiednie ścieżki komunikacyjne i ma bezpośredni dostęp do bazy danych. Napisany w języku python z wykorzystaniem frameworka [Django](#). Zawiera także interfejs administratora / zarządzania zadaniami.

CI/CD

Zaimplementowany za pośrednictwem środowiska Gitlab. Automatyzuje proces testowania i integracji nowych wersji poszczególnych elementów systemu (API / Front).



Kubernetes



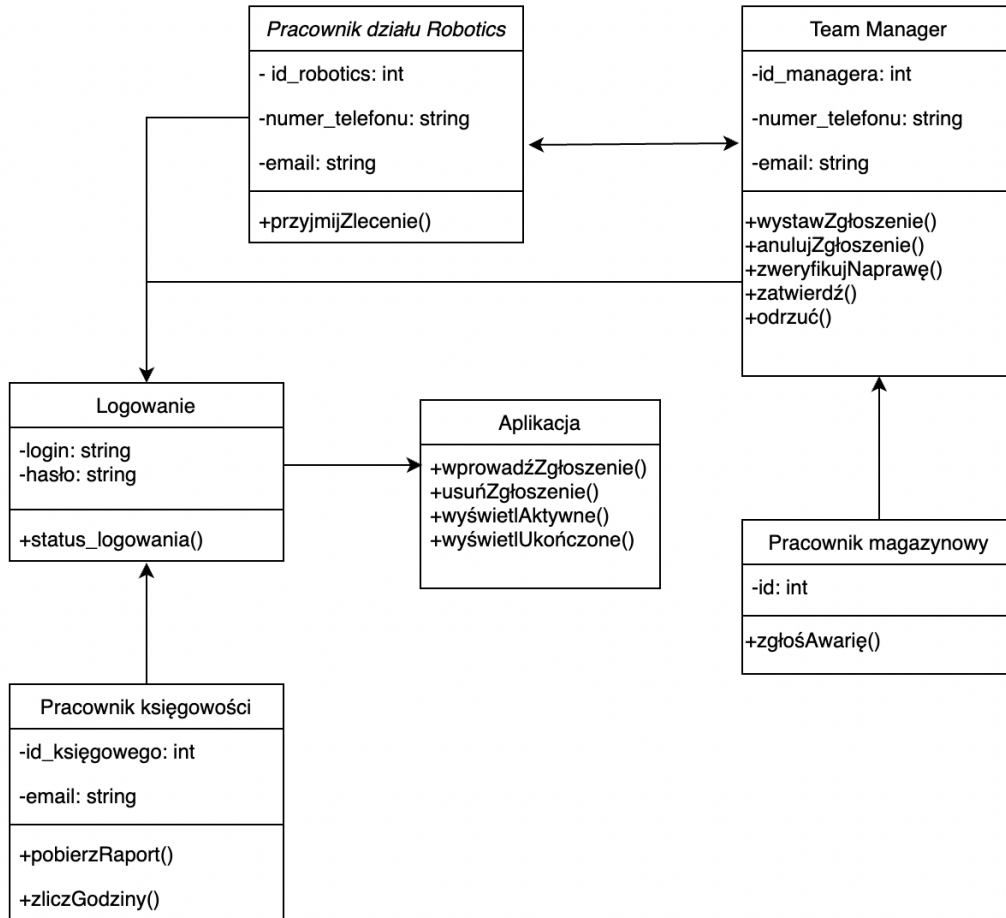
Środowisko do kontroli i zarządzania integracją oraz skalowaniem poszczególnych serwisów systemu.

9.2 Technologie implementacji systemu

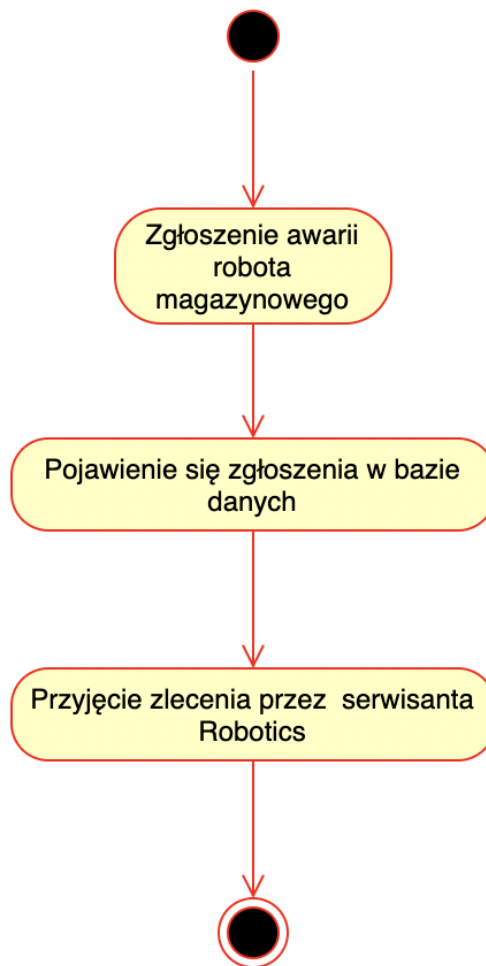
Technologia	Uzasadnienie
Docker	Rozwiązanie kontenerowe dla stabilnego środowiska developerskiego poszczególnych serwisów systemu.
Kubernetes	Platforma służąca do zarządzania zadaniami i serwisami uruchamianymi w kontenerach (Docker).
Django	Pozwala na szybkie zaimplementowanie systemu autoryzacji, wystawienia przestrzeni administracyjnej oraz implementacji logiki komunikacji pomiędzy interfejsem użytkownika a bazą danych. Z uwagi na możliwości frameworka zawarty również będzie w nim zamknięty interfejs administracyjny systemu.
Angular	Framework do zbudowania interfejsu użytkownika.
Postgresql	Implementacja relacyjnej bazy danych.
Gitlab	Kontrola wersji. CI/CD

9.3 Diagramy UML

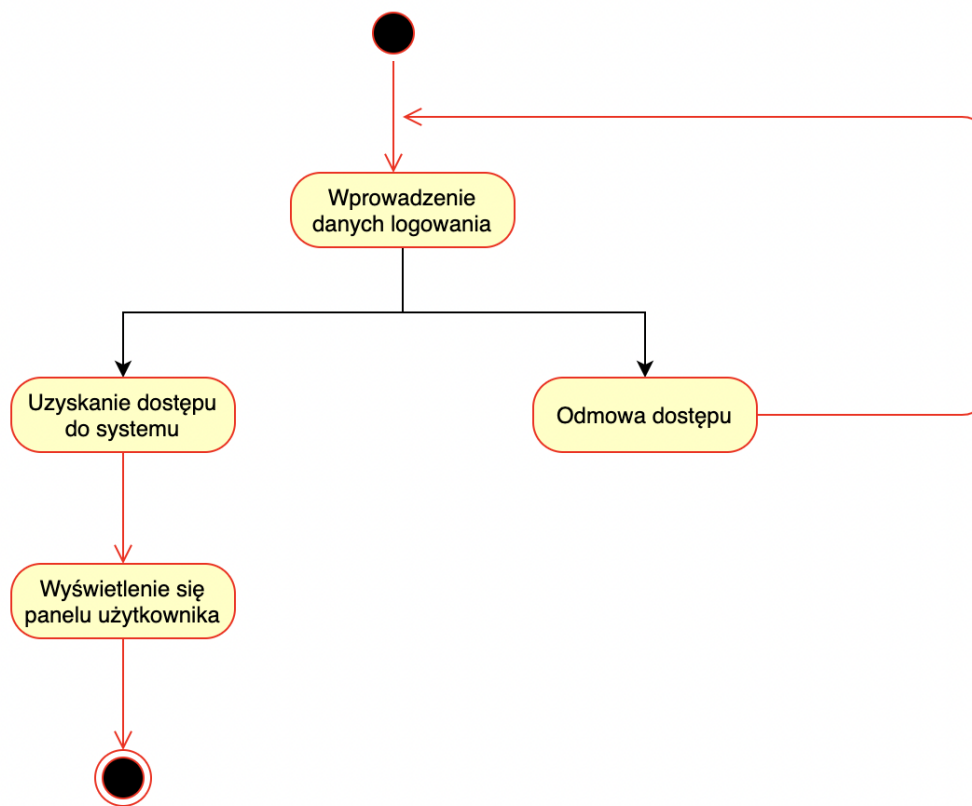
9.3.1 Diagram klas



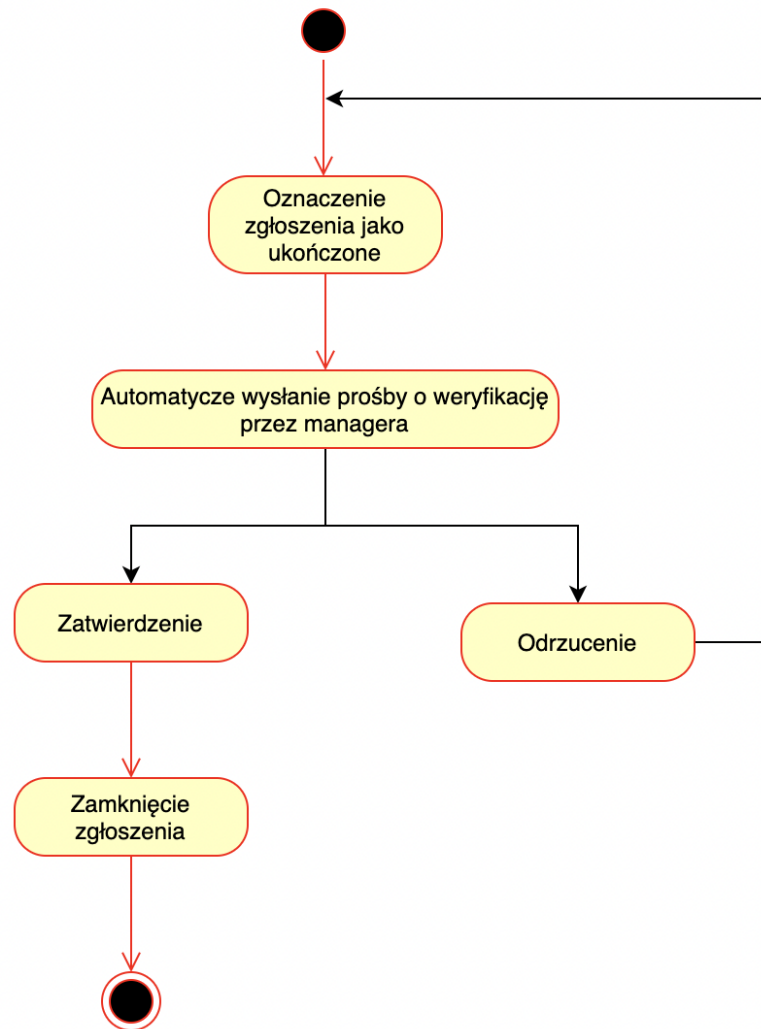
9.3.2 Diagram(-y) czynności



Naprawa awarii i wznowienia działania robota na hali magazynowej



Zalogowanie się do systemu



Proces zamykania zgłoszeń

9.3.3 Diagramy sekwencji

Diagram sekwencji dla procedury zgłaszania usterki

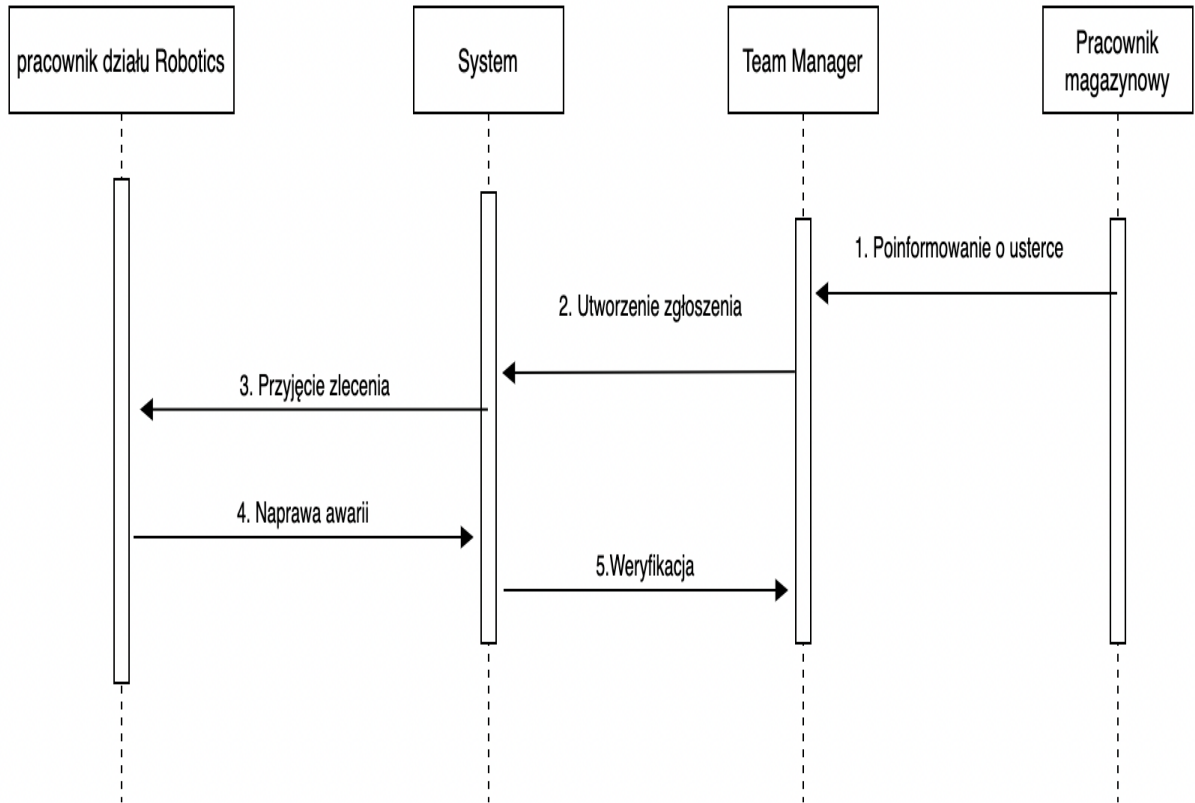


Diagram sekwencji dla procedury logowania do systemu

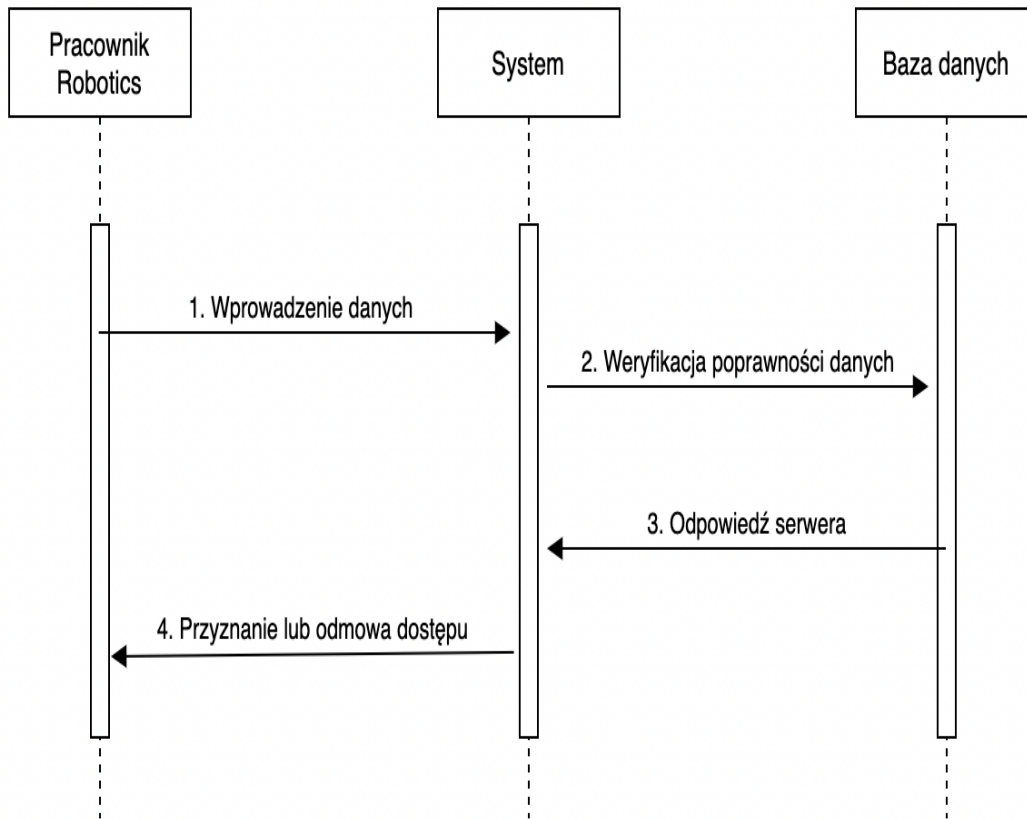


Diagram sekwencji dla procedury tworzenia raportów o liczbie przepracowanych godzin

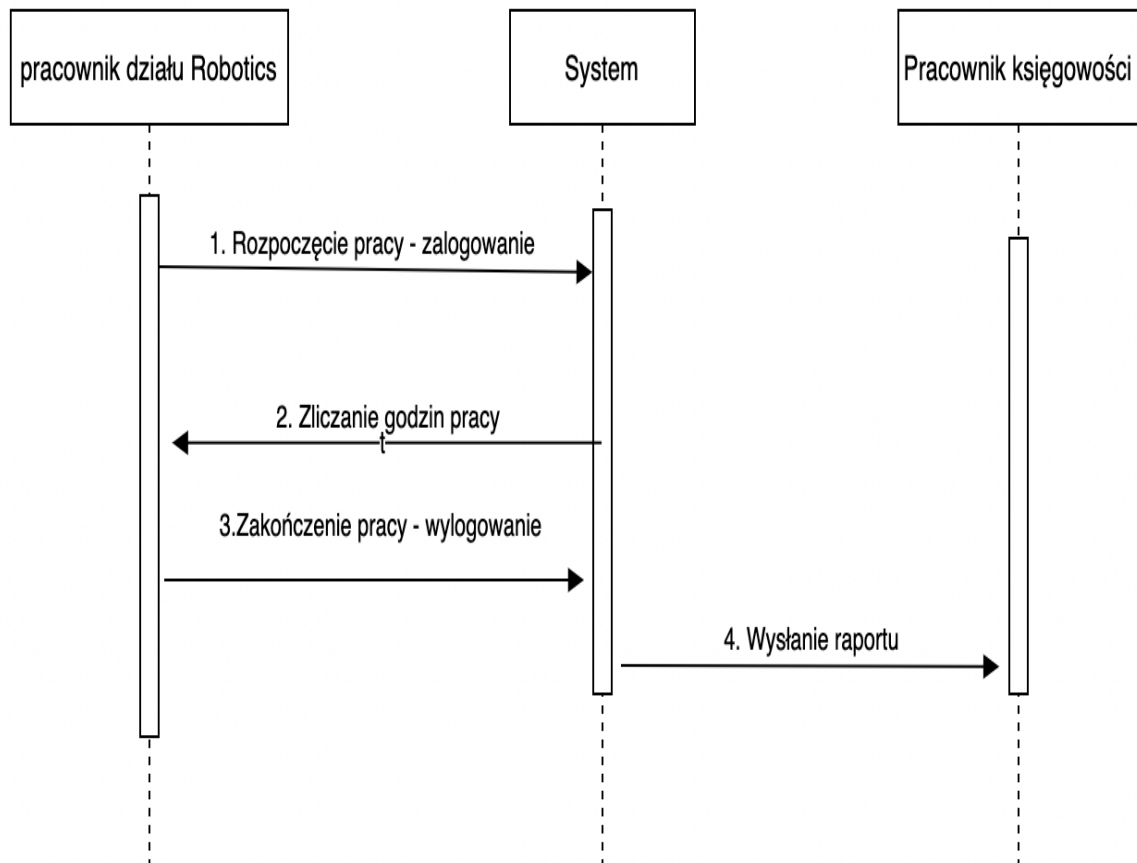


Diagram sekwencji dla procedury weryfikacji zakończonych zgłoszeń

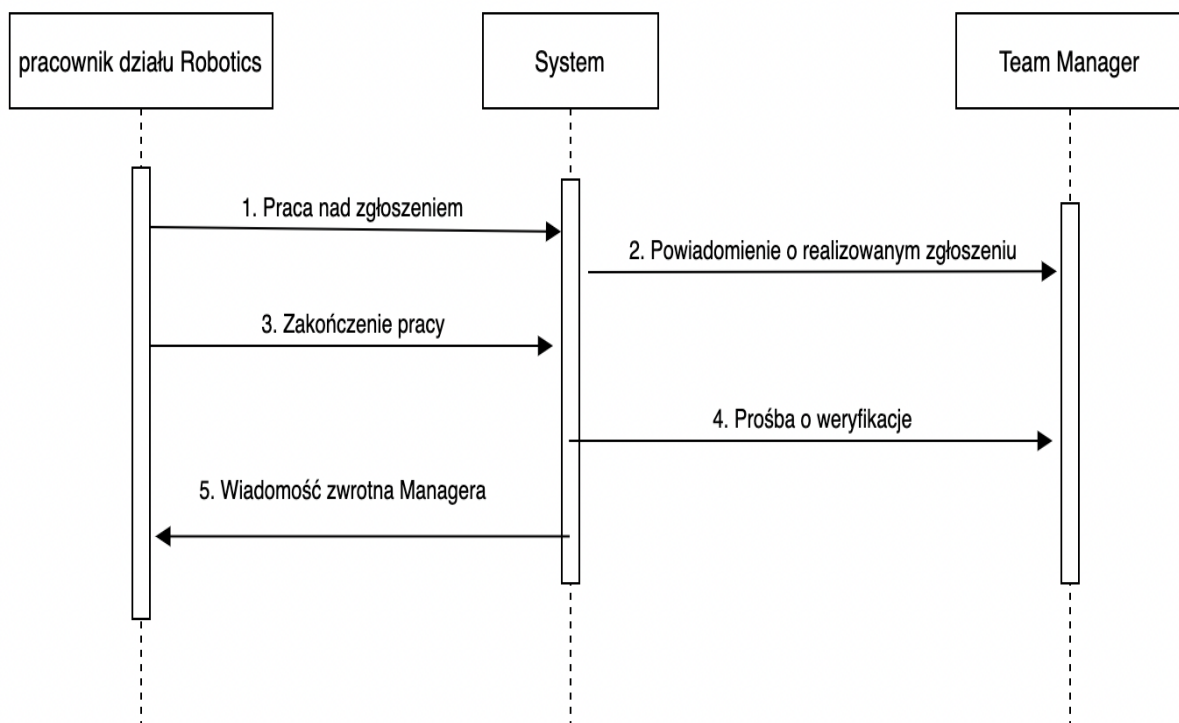
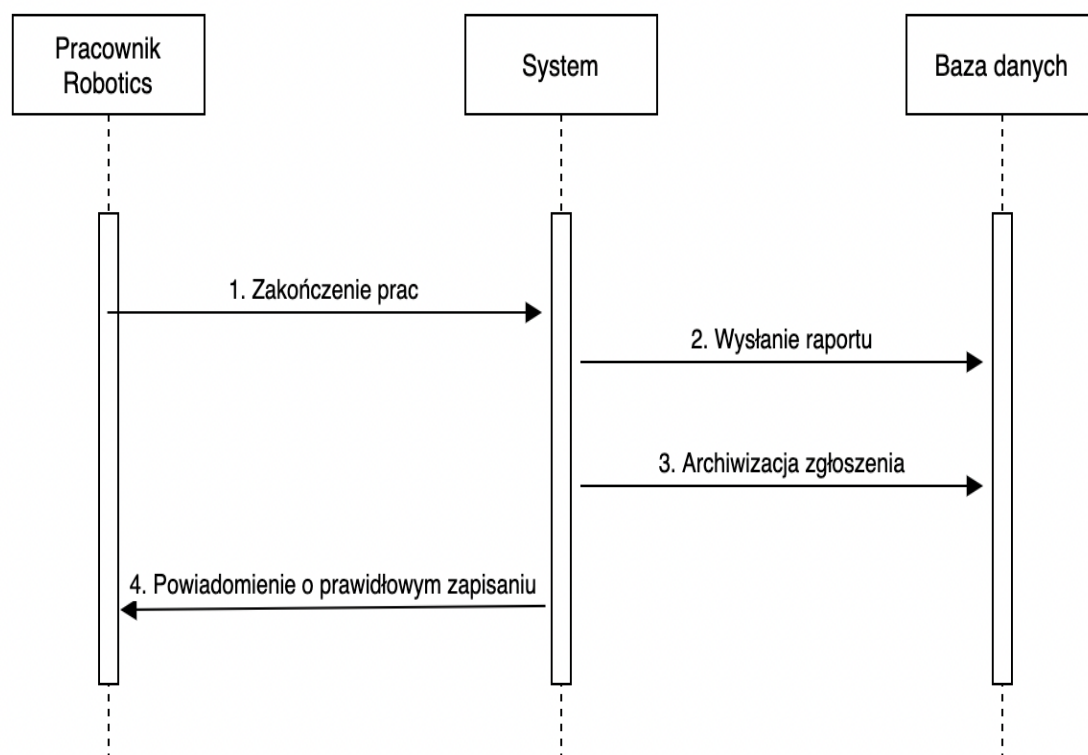


Diagram sekwencji dla procedury zapisu zgłoszenia w bazie danych



9.3.4 Inne diagramy

Diagram maszyny stanowej

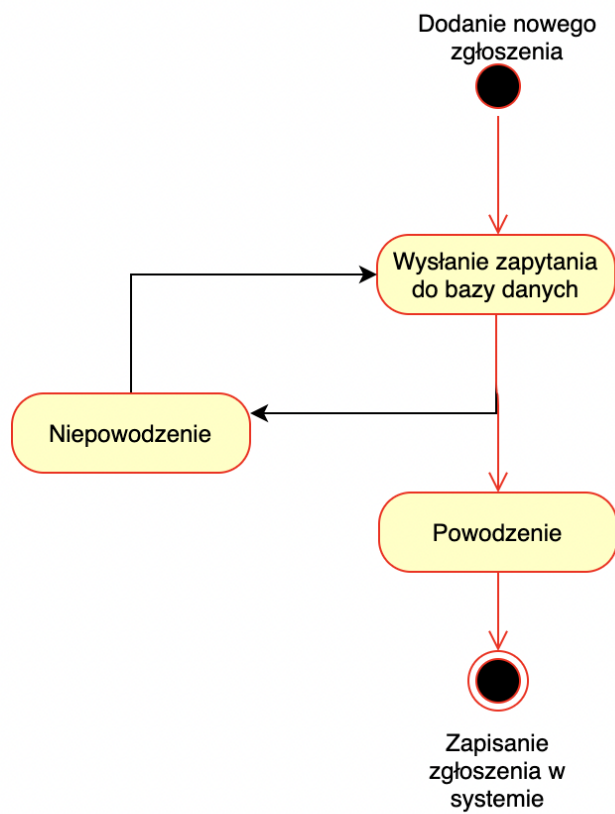


Diagram komponentów

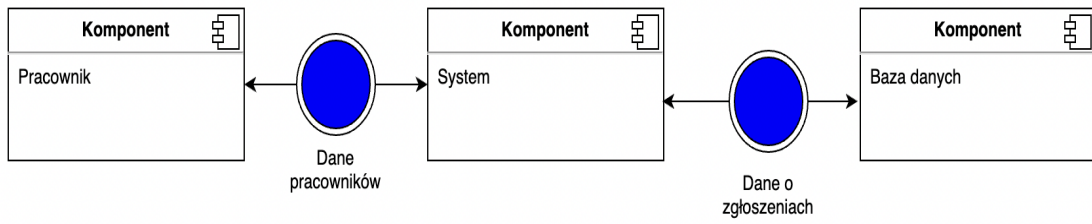
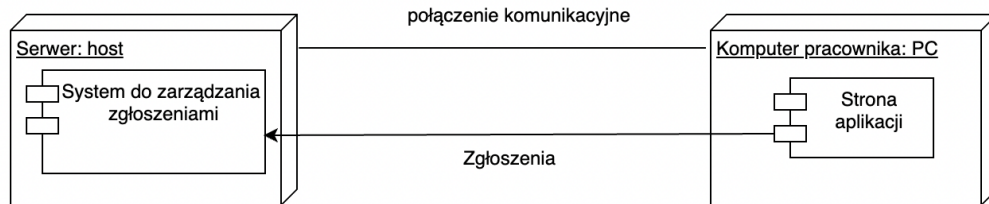


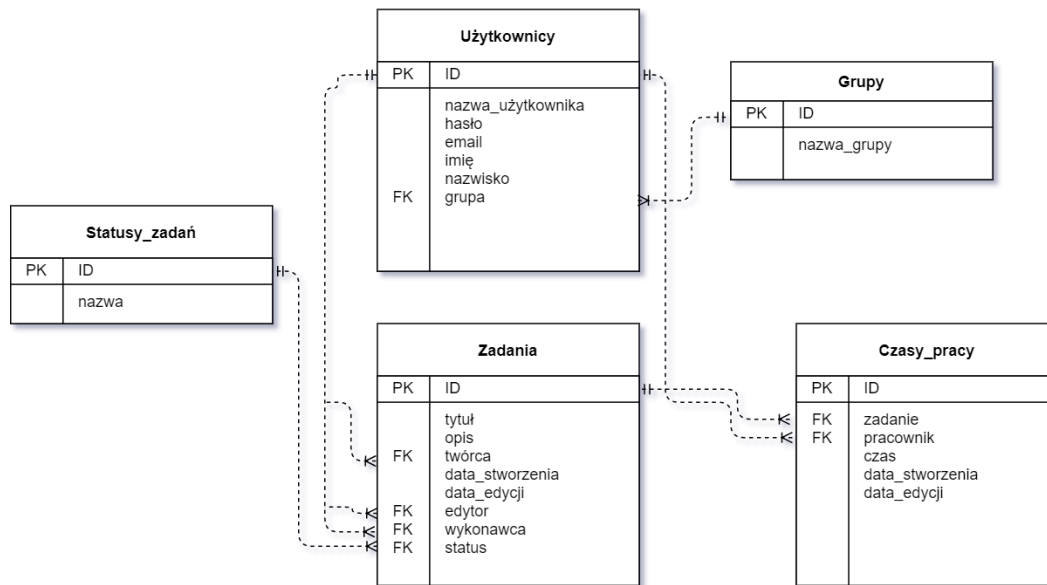
Diagram wdrożeniowy



9.4 Charakterystyka zastosowanych wzorców projektowych

Wzorzec kreacyjny -> Metoda wytwórcza

9.5 Projekt bazy danych



10 Schemat

10.1.1 Projekty szczegółowe tabel

Użytkownicy

Pole	Typ pola	Typ klucza
ID	int(64)	PK
nazwa_użytkownika	varchar(32)	-
hasło	varchar(64)	-

email	varchar(32)	-
imię	varchar(32)	-
nazwisko	varchar(32)	-
grupa	int(64)	FK

Zadania

Pole	Typ pola	Typ klucza
ID	int(64)	PK
tytuł	varchar(32)	-
opis	varchar(32)	-
twórca	int(64)	FK
data_stworzenia	dateTime	-
data_edycji	dateTime	-
edytor	int(64)	FK
wykonawca	int(64)	FK
status	int(64)	FK

Czas pracy

Pole	Typ pola	Typ klucza
ID	int(64)	PK
zadanie	int(64)	FK
pracownik	int(64)	FK

czas	int(64)	-
data_stworzenia	dateTime	-
data_edycji	dateTime	-

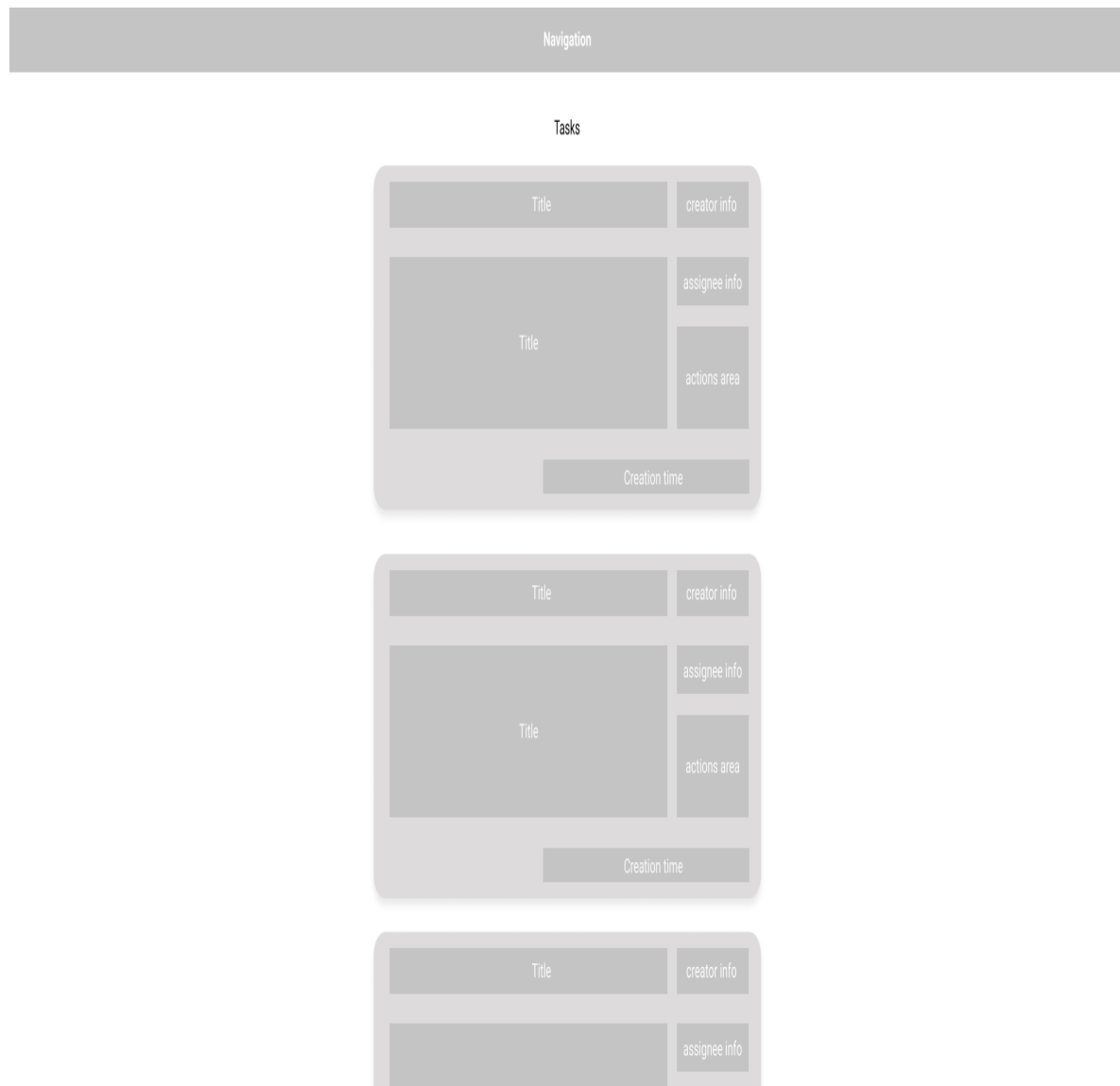
Grupy

Pole	Typ pola	Typ klucza
ID	int(64)	PK
nazwa_grupy	varchar(32)	-

Statusy zadań

Pole	Typ pola	Typ klucza
ID	int(64)	PK
nazwa	varchar(32)	-

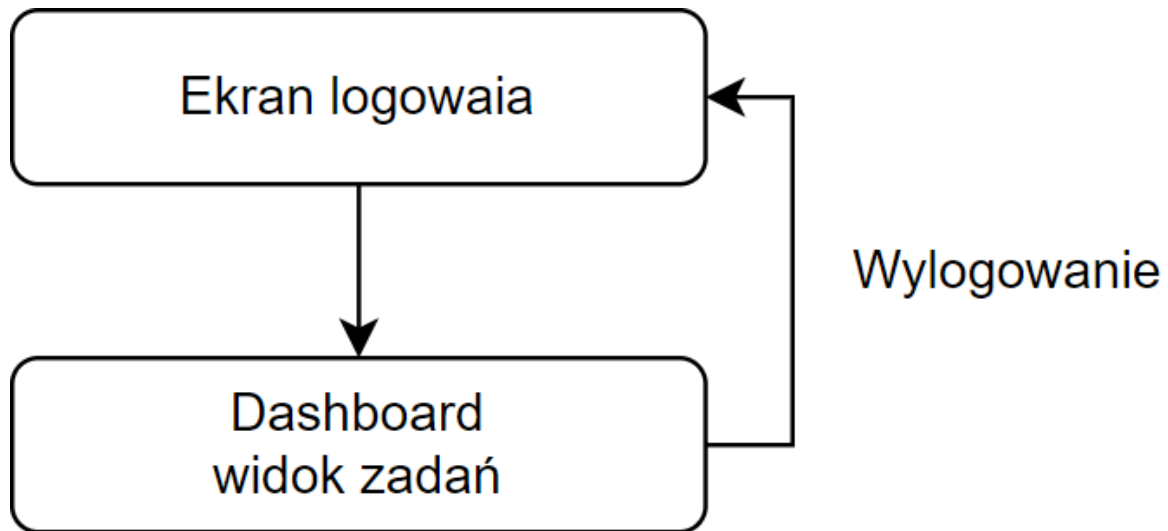
10.2 Projekt interfejsu użytkownika



10.2.1 Lista głównych elementów interfejsu

1. Nawigacja
2. Lista zadań

10.2.2 Przejścia między głównymi elementami



10.2.3 Projekty szczegółowe poszczególnych elementów

Nazwa	Ekran logowania
Wykorzystane dane	nazwa_użytkownika, hasło
Opis działania	Uwierzytelnienie użytkownika

Nazwa	Dashboard
Wykorzystane dane	Dane zadań
Opis działania	Wyświetla zadania oraz informacje o nich.

10.3 Procedura wdrożenia

W pierwszej kolejności należy przygotować i wdrożyć kluster kubernetes. Dla serwisów obsługujących frontend i backend należy przygotować konfigurację konteneryzacji oraz runner i konfigurację CI/CD dla ich repozytoriów w gitlab. Dla gotowego klastra należy skonfigurować usługę istio oraz dodatkowo zabezpieczyć dostęp poprzez haproxy. Wersja produkcyjna powinna mieć osobny namespace lub klaster.

11 Podsumowanie

11.1 Szczegółowe nakłady projektowe członków zespołu

Piotr Zawalski	Filip Ślęzak
2, 3, 4, 5, 7, 8, 9.3	2, 3, 6, 9.1, 9.2, 9.4, 9.5, 9.6, 9.7, 10 implementacja / demo dashboard

12 Inne informacje